NOVATEUR PUBLICATIONS
INTERNATIONAL JOURNAL OF INNOVATIONS IN ENGINEERING RESEARCH AND TECHNOLOGY [IJIERT]
ISSN: 2394-3696 Website: ijiert.org
VOLUME 7, ISSUE 8, Aug.-2020

LEARNING BASED APPROACH FOR HINDI TEXT SENTIMENT ANALYSIS USING NAIVE BAYES CLASSIFIER

V. B. PARTHIV DUPAKUNTLA,
Maturi Venkata Subba Rao Engineering College, Hyderabad

HEMISH VEERABOINA,
Maturi Venkata Subba Rao Engineering College, Hyderabad

M. VAMSI KRISHNA REDDY,
Maturi Venkata Subba Rao Engineering College, Hyderabad

M. MOHANA SATYANARAYANA,
Maturi Venkata Subba Rao Engineering College, Hyderabad

Y. SAI SAMEER
Maturi Venkata Subba Rao Engineering College, Hyderabad

ABSTRACT
Sentiment analysis can be briefly described as the process of analyzing the emotion and opinion a particular sentence carries using natural language processing techniques. With the increase in the amount of information being communicated via regional languages like Hindi, 4th commonly spoken language in the world and its high potential for knowledge discovery comes a promising opportunity to apply sentiment analysis on this information. Hindi, being morphologically rich and free order language when contrasted with English, adds intricacy while dealing with the user-generated content. Most of the work in this domain has been done in the English language. This paper attempts to classify the polarities of the reviews or opinions expressed in the Hindi language into positive or negative sentiments using a supervised machine learning algorithm called Naïve Bayes Classifier and evaluate the overall model's performance with respect to various parameters.

INDEX TERMS: Naïve Bayes Classifier, Natural Language Processing, Sentiment Analysis, Polarities, Hindi, Reviews.

INTRODUCTION
One of the most prominent domains in the field of Natural Language Processing (NLP) is that of Sentiment Analysis. It is a field of study that analyzes people's opinions, sentiments and emotions towards certain entities such as individuals, organizations, products or services. The term sentiment analysis perhaps first appeared in (Nasukawa and Yi, 2003) [1].

There are fundamentally two types of approaches to Sentiment Analysis. They are Learning-based approach and Lexicon-based approach. We implement an algorithm that falls under the purview of Learning-based approach and precisely comes under probabilistic classifiers. There are several probabilistic classifiers such as Naïve Bayes, Bayesian Network, Maximum Entropy, etc.,[2] and we have applied the Naïve Bayes Classifier to determine the polarity of a Hindi Language sentence.

Hindi is the national language of the Indian subcontinent. It is widely regarded as the common tongue of India and hence has a prolific significance in broadcasting one's opinion. Therefore, researchers have shown significant interests in Hindi Language Sentiment Analysis. Namita Mittal, Basant Agarwal, Garvit Chouhan, Prateek Pareek, and Nitin Bania (2013)[3] have studied on how by maintaining a balanced relation between negation and discourse may increase the performance of Hindi Review Sentiment Analysis.

The remainder of the paper is composed as follows: Section II depicts related work. Section III clarifies the proposed model for sentiment analysis. Trial results are talked about in Section IV. Segment V outlines the conclusion along with future work.

RELATED WORK

There have been numerous developments in the sentiment analysis domain with respect to various Indian languages such as Telugu, Bengali, Malayalam, etc. In Telugu, Pravarsha et al. [4] used a Rule-based Approach for Telugu sentiment analysis and Naidu et al. [5] proposed a two-stage sentiment analysis for Telugu news sentences with the help of Telugu Senti Word Net. This was further extended by Garapati et al. [6] wherein they have implemented Senti Phrase Net, which covers the drawbacks of Senti Word Net. Das and Bandyopadhyay [7] implemented a technique on English sentiment lexicons and developed a Bengali Senti Word Net using the English-Bengali bilingual dictionary. They have further extended their work by creating and validating Senti Word Net for Hindi and Telugu as well through a gaming strategy called "Dr. Sentiment" [8]. Here, they implemented Senti Mentality analysis on the data collected with the help of Internet users. They utilized this Senti Word Net to foresee the polarity of a given word and ordered the methodologies into four classifications in particular, the dictionary-based, Word Net-based, corpus-based and intelligent game (Dr Sentiment) to enlarge the extent of produced Senti Word Net. At last, an intuitive game is intended to recognize the polarity of a word dependent on four inquiries which must be replied by the users [9-11]. In Malayalam, a rule-based approached was proposed by Deepu S. Nair and Co. [12] to analyze the sentiment of text from film reviews given by users and to categorize them into positive, negative or neutral based on their writings.

To rouse more analysts towards the sentiment analysis in Indian dialects, Patra et al. [13] directed a mutual assignment called SAIL (Sentiment Analysis in Indian Languages). There, numerous analysts have introduced their techniques to examine the sentiment of Indian dialects, for example, Hindi, Bengali, Tamil and so forth. Kumar et al. [14] proposed regularized least square methodology with randomized feature learning on how to distinguish various sentiments in the Twitter dataset. Sarkar et al. [15] built up a sentiment analysis framework for Hindi and Bengali tweets utilizing multinomial Naïve Bayes classifier that utilizes unigrams, bigrams and trigrams for choosing features. Additionally, Prasad et al. [16] proposed decision tree-based analyzer for Hindi tweets.

PROPOSED SCHEME

This section deals with the various stages involved in implementing the Naïve Bayes Classifier to analyze Hindi text sentiment. It begins with data collection, followed by implementing the proposed classifier to train the collected data. Further, it gives a brief understanding of the algorithms applied. Finally, Fig.1 outlines the schematic of the entire process.

A. Data Collection:

We have utilized the collected data of 250 movie reviews available for research from IIT Bombay and the annotated dataset of 750 reviews from jagran.com by the user Shubam Goyal (GitHub Username: shubam721)[17] and also the annotated dataset obtained from Shivangi Arora (GitHub Username: nacre13)[18] to create a comprehensive collection of both polarities. We have used a 90/10 split to create training and testing datasets.

Positive Sentences	1,693
Negative Sentences	1,693
Positive Sentences (Train)	1,512
Negative Sentences (Train)	1,504
Positive Sentences (Test)	181
Negative Sentences (Test)	189

B. Naïve Bayes Classifier

A Naïve Bayes classifier is a probabilistic machine learning model that's used for classification task along with a strong independence assumption. Given a class (positive or negative), the words are conditionally independent of each other. Rennie et al. discuss the performance of Naïve Bayes on text classification tasks in their 2003 paper. [19] For a particular word, the maximum likelihood probability is given by:

$$P\left(\frac{x_i}{c}\right) = \frac{\text{Count of } x_i \text{ in documents of class c}}{\text{Total no of words in document of class c}} \quad (1)$$

Where,
x_i = i^{th}word in the sentence

The probability of a given document belonging to a class c_i according to bayes rule is given by:

$$P\left(\frac{c_i}{d}\right) = \frac{P\left(\frac{d}{c_i}\right) * P(c_i)}{P(d)} \quad (2)$$

Where,
c_i= i^{th} Class
d = document d

The model is termed as "naïve" due to the simplifying conditional independence assumption. Assuming the words to be conditionally independent of each other, the equation becomes as follows

The output of the classifier would be the class with maximum posterior probability [20].

$$P\left(\frac{c_i}{d}\right) = \frac{\left(\Pi P\left(\frac{x_i}{c_j}\right)\right) * P(c_j)}{P(d)} \quad (3)$$

Where,
c_i= i^{th} class
x_i= i^{th} word in the sentence
d = document d
c_j = j^{th} class

The classifier outputs the class with the maximum posterior probability [20].

Laplacian Smoothing
If a new word has been encountered from the training dataset, the probability of both the classes would become zero [20]. Laplacian smoothing is performed to avoid this problem:

$$P\left(\frac{x_i}{c_j}\right) = \frac{\text{Count}(x_i) + k}{(k+1) * (\text{No of words in class } c_j)} \quad (4)$$

Where,

x_i= i^{th} word in the sentence
c_j= j^{th} class
k = constant (usually taken as 1)

1) Algorithm:
1. The preprocessed Dataset which is divided into class Pos and class Neg is considered.
2. For both the classes Pos and Neg, prior probabilities are calculated as follows.
Class Pos=total number of sentences in class Pos / total number of sentences.
Class Neg=total number of sentences in class Pos / total number of sentences.

3. Word Frequencies for both the classes A &B i.e. n_i is calculated.
n_a= the total word frequency of class Pos.
n_b=the total word frequency of class Neg.
4. For the given class, conditional probability of keyword occurrences are calculated as follows
P(word1 / class Pos) = wordcount / n_i (Pos)
P(word1 / class Neg) = wordcount / n_i (Neg)
P(word2 / class Pos) = wordcount / n_i (Pos)
P(word2 / class Neg) = wordcount / n_i (Neg)
……
……
P(wordn / class Pos) = wordcount / n_i (Pos)
P(wordn / class Neg) = wordcount / n_i (Neg)
5. Laplacian Smoothing is done to avoid the problem of zero probability..
6. Given a new sentence M, the probability of the sentence being classified into class Pos and class Neg is calculated as follows
a) P(Pos / M) = P(Pos) * P(1st word/class Pos)* P(2nd word/ class Pos)……* P($n^{th}word$ / class Pos).
b) P(Neg / M) = P(Neg) *P(1st word /class Neg)*P(2nd word / class Neg)……* P($n^{th}word$/ class Neg).

For example, consider the following sentence

S = "उसनेभगवानकोमाननाशुरूकरदिया"

The probability for the polarity to be positive for a sentence is represented by:
P(Pos/S)=P(Pos)*P(उसने/class Pos)*P(भगवान/class Pos)*P(को/class Pos)*P(मानना/class Pos)*P(शुरू/class Pos)*P(कर/class Pos)*P(दिया/class Pos)

The probability for the polarity to be positive for a sentence is represented by:
P(Neg/S) = P(Neg)*P(उसने/class Neg)*P(भगवान/class Neg)*P(को/class Neg)*P(मानना/class Neg)*P(शुरू/class Neg)*P(कर/class Neg)*P(दिया/class Neg)

7. After the calculation of probabilities for both the classes, the one with higher probability is assigned to the sentence.

2. Training Phase Algorithm

Algorithm 1 is responsible for training the classifier with the given dataset. Initially, the count of the total number of sentences from D in class C is obtained. Then, log prior is calculates for that given class. To get the total count for the number of words in class C, looping is done over our vocabulary. At last, log-likelihoods are calculated considering Laplacian smoothing for each word in class C. Laplacian Smoothing is done to avoid the problem of zero probability.

Algorithm 1: Training Phase

for each: $class\ c \in C$
$Ns = Number\ of\ sentences\ in\ file\ D$
$Nc = Number\ of\ sentences\ from\ D\ in\ class\ C$
$\text{logprior}[c] \leftarrow \log\left(\frac{Nc}{Ns}\right)$
N ← vocabulary of D
dic[c] ← append(d) for d ∈ D with class C
for each: word w in V
count(w, c) ← #ofoccurences of w in dic[c]

$\text{loglikelihood}[w, c] \leftarrow \log((\text{count}(w, c) + 1)/\sum_{w' \text{ in } v} \text{count}(w', c))$
Return logprior, loglikelihood

3. Classification Phase Algorithm

Algorithm 2 handles the classification phase in which probabilities for each class are stored in a dictionary named sums. For each class c, the first term of our probability equation, i.e. log prior is added to the second term which is computed by adding the probability of the log-likelihood of each word over all the words in the sentence. Prediction is made based on the key of the maximum value of our dictionary.

Algorithm 2: Classification Phase

for each: class $c \in \mathcal{C}$
  $\text{sum}[c] \leftarrow \text{logprior}[c]$
for each: position i in sent
  $\text{word} \leftarrow \text{sent}[i]$
if $\text{word} \in \mathcal{V}$ **then** $\text{sum}[c] \leftarrow \text{sum}[c] + \text{loglikelihood}$
Return $\text{argmax}_c \text{sum}[c]$

FLOW CHART

The following figure depicts the flow of the proposed implementation scheme. The collected dataset is further preprocessed into two sets of polarities and by associating tags (positive/negative) to each sentence respectively. The classifier is then trained by feeding the input dataset.
Finally, the statistical parameters (accuracy, precision, recall, F1 score) are calculated based on the model's performance on the testing dataset.

Fig 1: Flow Chart for Sentiment Analysis

EXPERIMENTAL RESULTS & ANALYSIS

This section deals with the outcomes derived from the Naïve Bayes Classifier approach.
The total number of sentences in the dataset is 3396 out of which positive sentences are 1693 and negative sentences are 1693. The split used for training/testing the classifier is 90/10. The first phase in sentiment analysis has been performed using the training dataset. After training, the classifier is evaluated on the testing dataset which consists of 370 sentences out of which the positive/negative split is 181/189. The true positives, true negatives, false positives and false negatives obtained from the classifier are 152, 29, 57, and 132 respectively.

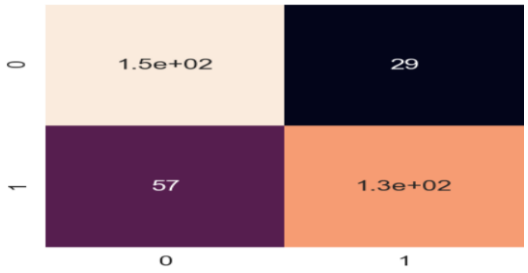

Fig 2: Confusion Matrix

$$\text{accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{FN} + \text{TN}} \quad (5)$$

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (6)$$

$$\text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (7)$$

$$\text{F Score} = \frac{2 * (\text{recall} * \text{precision})}{\text{recall} + \text{precision}} \quad (8)$$

Where,
TP= True Positives
TN= True Negatives
FP= False Positives
FN= False Negatives

With the help of the confusion matrix, four statistical parameters namely accuracy, precision, recall and F-score are obtained to evaluate the performance of the classifier using the Eqs. (5),(6),(7) and (8) respectively.

TABLE I
Results in terms of Accuracy, Precision, Recall, F-score.

	Accuracy (%)	Precision	Recall	F -score
NB	76.7	0.8415	0.7298	0.7817

The results thereby obtained from the proposed classifier are tabulated in TABLE 1 as show above.

CONCLUSION& FUTURE WORK

There are very few annotated datasets in Hindi on which sentiment analysis could be performed. This paper exploits two available datasets on which the Naïve Bayes Classifier was implemented to perform sentiment analysis. The authors would like to thank Shubham Goyal and Shivangi Arora for their datasets which were of great help to the work.

These datasets mainly include movie reviews. We have achieved an accuracy of 76.7% through the proposed model for sentiment analysis in the domain of user reviews.

During the research, we have found that learning-based approaches for text sentiment analysis have a vast scope in developing an ideal sentiment classifier. In addition, usage of ensemble methods like a combination of Naïve Bayes with neural networks or Support Vector Machines(SVM's) can improve the accuracy of the classifier further and have a high potential in extending this work.

REFERENCES

1) Liu and Bing, "Sentiment analysis and opinion mining," Synthesis Lectures on Human Language Technologies, 2012, pp. 1–167.
2) Walaa Medhata, Ahmed Hassan, Hoda Korashy, "Sentiment analysis algorithms and applications: A survey",Ain Shams Engineering Journal.
3) Namita Mittal and Basant Agarwal (2013), "Sentiment Analysis of Hindi Review based on Negation and Discourse Relation", International Joint Conference on Natural Language Processing, pages 45–50, Nagoya, Japan, 14-18 October.
4) Pravarsha Jonnalagadda, Krishna Pratheek Hari, Sandeep Batha, Hashresh Boyina "A rule based sentiment analysis in Telugu", International Journal of Advance Research, Ideas and Innovations in Technology.
5) Reddy Naidu, Santosh Kumar Bharti, Korra Sathya Babu and Ramesh Kumar Mohapatra,"Sentimental Analysis using SentiWordNet", IEEE WiSPNET 2017 Conference.
6) Abhinav Garapati, Naveen Bora, Hanisha Balla, Mohan Sai, "SentiPhraseNet: An extended SentiWordNet approach for Telugu sentiment analysis", International Journal of Advance Research, Ideas and Innovations in Technology.
7) Amitava Das and Sivaji Bandyopadhyay, "Sentiwordnet for bangla," Knowledge Sharing Event-4: Task 2, 2010.
8) Amitava Das and S. Bandyopadhay, "Dr sentiment creates SentiWordNet (s) for Indian languages involving internet population",Proceedings of Indo-wordnet workshop, 2010.
9) Amitava Das and Sivaji Bandyopadhyay, "SentiWordNet for Indian languages", Asian Federation for Natural Language Processing, China, 2010, pp. 56–63.
10) Das Amitava and Sivaji Bandyopadhyay, "Dr Sentiment knows everything!",Proceedings of the 49th annual meeting of the association for computational linguistics, human language technologies, systems demonstrations, Association for Computational Linguistics, 2011.
11) Das Amitava and Bjrn Gambck, "Sentimantics: conceptual spaces for lexical sentiment polarity representation with contextuality", Proceedings of the 3rd Workshop in Computational Approaches to Subjectivity and Sentiment Analysis, Association for Computational Linguistics, 2012
12) Deepu S. Nair, Jisha P. Jayan, Rajeev R.R, Elizabeth Sherly, "SentiMa - Sentiment Extraction for Malayalam", IIITM, Kerala
13) Braja Gopal Patra, Dipankar Das, Amitava Das, and Rajendra Prasath, "Shared Task on Sentiment Analysis in Indian Languages (SAIL) Tweets - An Overview", Springer International Publishing Switzerland 2015.
14) S. S. Kumar, B. Premjith, M. A. Kumar, and K. P. Soman, "AMRITA_CEN-NLP@ SAIL2015 Sentiment analysis in Indian Language using regularized least squares approach with randomized feature learning", International Conference on Mining Intelligence and Knowledge Exploration, Springer International Publishing, 2015, vol. 9468.
15) Kamal Sarkar and Saikat Chakraborty, "A sentiment analysis system for Indian language tweets," in International Conference on Mining Intelligence and Knowledge Exploration, Springer International Publishing, 2015, vol. 9468.

16) S. S. Prasad, J. Kumar, D. K. Prabhakar, and S. Pal, "Sentiment Classification: An Approach for Indian Language Tweets Using Decision Tree", in International Conference on Mining Intelligence and Knowledge Exploration, Springer International Publishing, 2015, vol. 9468.
17) Shubham Goyal, Sentiment-Analysis-On-Hindi-Reviews, GitHub. See: https://github.com/ shubham 721/Sentiment-Analysis-On-Hindi-Reviews
18) Shivangi Arora, Sentiment-Analysis-of-Hindi-Text-File, GitHub. See: https://github.com /nacre 13/ Sentiment-Analysis-of-Hindi-Text-File
19) Rennie, J.D., et al "Tackling the poor assumptions of naive bayes text classifiers.", Machine Learning-International Workshop then Conference, vol. 20(2) (2003)
20) Vivek Narayanan, Ishan Arora, Arjun Bhatia,"Fast and accurate sentiment classification using an enhanced Naive Bayes model", Intelligent Data Engineering and Automated Learning – IDEAL 2013: 14th International Conference, IDEAL 2013, Hefei, China, October 20-23, 2013.