A Novel Approach for Sentiment Analysis of Punjabi Text using SVM

Amandeep Kaur and Vishal Gupta
Department Computer Science and Engineering, Panjub University, India

Abstract: Opinion mining or sentiment analysis is to identify and classify the sentiments/opinion/emotions from text. Over the last decade, in addition to English language, many Indian languages include interest of research in this field. For this paper, we compared many approaches developed till now and also reviewed previous researches done in case of Indian languages like Telugu, Hindi and Bengali. We developed a hybrid system for Sentiment analysis of Punjabi text by integrating subjective lexicon, N-gram modelling and support vector machine. Our research includes generation of corpus data, algorithm for stemming, generation of Punjabi subjective lexicon, developing Feature set, Training and testing support vector machine. Our technique proves good in terms of accuracy on the testing data. We also reviewed the results provided by previous approaches to validate the accuracy of our system.

Keywords: Sentiment analysis, subjective lexicon, Punjabi language, n-gram modeling, support vector machine.

 Received June 17, 2014; accepted December 16, 2014

1. Introduction

Although Today, Punjabi is widely used and well spoken language in the various parts of world. This language has 100+ million speakers and wide coverage area across the web. But the thing which is scarce is the resources and tools to do successive research for this language. In this research, we used some tools of native and other languages such as Hindi subjective Lexicon developed by Piyush [5], Punjabi dictionary which contains 35031 words. We combined the features of Lexicon and N-gram modelling to train support vector machine and developed a classification model which will further used for classifying the text. In our work, we tried to provide following contributions as research work:

1. Punjabi subjective lexicon is developed using Punjabi dictionary and Hindi Subjective Lexicon provided by [22].
2. Developed corpus dataset for training and testing the system.
3. Generation of Algorithm for stemming using the rules given by Dr. Gupta [19].
4. Devised an Algorithm for the use of Subjective lexicon by combining the simple scoring method and unigram method which provides the better efficiency.
5. Generation of feature set for better identification of type of sentiment class.

7. Training and testing support vector machine to resolve the error.
8. Machine learning is implemented through SVM.

2. Related Work

Through the literature survey [27], we have come up with the following levels at which research of Sentiment analysis has been done:

- Document level [32, 43]
- Sentence level [21, 24, 28, 45, 47]
- the Word level [2, 46]

2.1. Non-Indian Languages

With the research of General Inquirer system [42] in 1966, IBM has begun the research under this area. Selection of features for this kind of research is explained by khan et al. [8]. In the area of research for English language, most of research has been done by Sebastiani and Esuli. In 2006, they come up with great lexical resource named Sentiwordnet [6, 18].

For languages having scarcity of resources, a bootstrapping method was given by Banea et al. [7]. Rao and Ravichandran [37] developed the semi supervised label propagation. Kim and Hovy [29] did research to analyse opinions or judgement. Kamps et al. [26] used adjectives given in wordnet to do research in the field of sentiment analysis.

2.2. Indian Languages

In case of Indian Languages, major issue is limitation of resources. For Bengali language, Das and
Bandhopaday did a lot of research and developed Sentiwordnet [13, 14]. Joshi et al. [25] worked for Hindi language. They used English Sentiwordnet and English-Hindi Word net Linking for developing H-SWN (Hindi-Sentiwordnet). Our research work is related to Punjabi Language.

3. Approach Used

In Natural Language Processing, English Language is taken as test language to apply different techniques. Following are the common approaches used in this field:

- N-gram Modeling,
- Subjective Lexicon,
- Machine Learning.

Using the above approaches, we have developed a Hybrid System by combining Subjective Lexicon, N-grams and Support Vector Machine. Implementation is done using Following Framework tools:

- Weka: It is java based tool kit developed at University of Waikato, New Zealand for machine learning [44].
- Naïve Bayes: This classifier uses Bayes Theorem which relates the probabilities of event given and event already occurred.
- Support Vector Machine: SVM classifier separates the dataset into classes by constructing N-dimensional hyper plane. WLSVM and LibSVM are the built-in libraries which we have used.

3.1. Stemming

To get the maximum coverage, we used the concept of stemming. Root words and stemmed words are given Table 1. We have formulated the following Algorithm for stemming:

Table 1. Stemming.

<table>
<thead>
<tr>
<th>Word</th>
<th>Root Word</th>
<th>Suffixes</th>
</tr>
</thead>
<tbody>
<tr>
<td>ਵੱਡੀ</td>
<td>ਵੱਡਾ</td>
<td>ਾਾ</td>
</tr>
<tr>
<td>਷ੁੁੰਦਰਤਾ</td>
<td>਷ੁੁੰਦਰ</td>
<td>ਤਾ</td>
</tr>
<tr>
<td>ਕੌੜੀ</td>
<td>ਕੌੜਾ</td>
<td>ਾੀ</td>
</tr>
</tbody>
</table>

Algorithm for Stemming:

1. Develop Array of Suffixes which we have to be identify and remove from the word to get a Root Word. List of suffixes is collected from research By Dr. Gupta [19]. Say this Array as SuffixArray.
2. Boolean = Check wordtobeStemmed Exists in the Dictionary.
3. If Boolean = true then no Stemming required. Else RootWord=getRootWord(WordtobeStemmed)

getRootWord (WordtobeStemmed)

Loop
Position < length
Suffix = Split the word at position
Flag = check if Suffix exits in the SuffixArray developed in Step 1.
If Flag = true
Output= ReplaceSuffix (WordtobeStemmed,Suffix)
Return Output
End if

If flag = false and position = length-1
Return WordtobeStemmed // no stemming possible
End if

End loop

ReplaceSuffix (WordtobeStemmed , Suffix)

Develop two arrays to determine replacement of suffix with kanna ( ਾਾ ) or BadiE ( ਾੀ ):
KannaArray = { "ਾ", "ਾ", "ਿਾਾ", "ਿਾ", "ਿਾਾ" } BadiEArray = {"ਿਾ", "ਿਾ"}
If Suffix lies in KannaArray Then Replace the suffix with Kanna ( ਾਾ ) Else if (Suffix lies in BigEArray) Then Replace the suffix with BadiE ( ਾੀ ) Else Remove the suffix from WordtobeStemmed End if
Return Root_word

3.2. Subjective Lexicon

Punjabi Language is very scarce because of the lack of limited resources developed till now. Basically, three popular methods are used for the generation of subjective lexicon:

- Use of bi-lingual dictionary [15].
- machine translation [15].
- use of word net [5].

In Our generation of Lexicon for Punjabi language, Translation process is applied at word level on the Hindi Subjective Lexicon developed by Arora in 2013. Arora [5] The resultant lexicon is refined by adopting various techniques of error reduction. We have also used Punjabi dictionary (contains 35031 words) to add the words which were not covered by the Hindi Subjective Lexicon. In the resultant Subjective lexicon, we have come with total of 7860 words. The developed lexicon is introduced with the concepts of synonyms and antonyms.

Every entry of the lexicon is tagged with one of the following four parts of speech [20]:

- Noun
- Verb
• Adjective
• Adverb

Structure and Statistics of developed lexicon is formulated in Table 2.

<table>
<thead>
<tr>
<th>Part of speech</th>
<th>Lexicon Statistics (Total = 7860)</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>noun</td>
<td>2448</td>
<td>2091</td>
</tr>
<tr>
<td>verb</td>
<td>1203</td>
<td>939</td>
</tr>
<tr>
<td>adverb</td>
<td>398</td>
<td>299</td>
</tr>
<tr>
<td></td>
<td>251</td>
<td>231</td>
</tr>
</tbody>
</table>

It also possesses the features of word net for better understanding the contextual information.

• **Step 1.** Input a text paragraph.
• **Step 2.** Divide and conquer:
  1. Divide the text into n- grams on the basis of full stop.
  2. Further Sub divide the n– grams into sub grams on the basis of Separators like comma, semi colon, or other conjunctive words (ਅਂ, ਦੋ ਲਹ, ਵਿਅਂਤਿਆਂ, ਦੀ) 
  3. Use tree data structure with 3 levels and height=2. Parent node as input text. Internal nodes as grams and leaf nodes as sub grams.
• **Step 3.** Preprocessing phase:
  1. Remove stop words
  2. Remove extra symbols
  3. Perform Stemming
• **Step 4.** Feature Extraction phase:
  1. Extracting Keywords: nouns, adjectives, adverb and verb.
• **Step 5.** Use Subjective Lexicon:
  1. Assign polarities to all keywords having the range 0.0 to 1.0
• **Step 6.** Remove the objective information:
  1. The sub gram which does not contain any polarity is considered as objective information having neutral polarity.
• **Step 7.** Compute the overall polarity of a sub gram:
  1. Sum up the positive and negative polarity
  2. Choose the dominating polarity for the respective sub gram.
• **Step 8.** If dominating polarity is positive then follow the rules below:
  1. Positive polarity must have value at least 0.5 more than the negative. If this is not true then assign the negative polarity to the sub gram.
  2. If value of negative and positive polarity is equal, then assign negative polarity to the sub gram.
• **Step 9.** Handling Negations: The sub gram that contains negation words (ਨਸੀਂ, ਨਾ), invert the polarity of that sub gram.
• **Step 10.** Final Output using iterative Process and Bottom up approach:
  1. Repeat step 8 for each gram by taking its respective child sub grams.
  2. Again Repeat step 8 for the parent node i.e input text. Polarity of the parent node determines the overall polarity of the text.

### 3.3. Negation Handling

There are certain words which are categorized as negation words like-“ਨਸੀਂ”, “ਨਾ”, “ਨਾ ਦਾ”, “ਨਾ ਭਾਵ”, “ਨਾ ਤਿਖਤਲ”, “ਨਾਂ”। Polarity of the sentence gets inverted due to the presence of these words. A list of these kinds of words is maintained to face this issue.

### 3.4. N-grams

In this technique, Following steps are carried out:

1. Dataset is divided into Training set and Testing set.
2. Using the Training set, we developed an N-gram model.
3. For each input from Testing set, create the trigrams
4. Match these trigrams with trigrams of training data using the N-gram model
5. If matches then increment the value of trimatched (variable name) else create the bi- grams and repeat the previous step for bigrams and unigrams and increment the value of respective match.

### 3.5. Feature Set

Features using subjective lexicon and n-grams are constructively shown in Table 3.

<table>
<thead>
<tr>
<th>Feature Set</th>
<th>Feature</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subjective Lexicon</td>
<td>Positive</td>
</tr>
<tr>
<td></td>
<td>Negative</td>
</tr>
<tr>
<td>n-grams</td>
<td>No. of unigrams(Positive)</td>
</tr>
<tr>
<td></td>
<td>No. of bigrams(Positive)</td>
</tr>
<tr>
<td></td>
<td>No. of trigrams(Positive)</td>
</tr>
<tr>
<td></td>
<td>No. of unigrams(Negative)</td>
</tr>
<tr>
<td></td>
<td>No. of bigrams(Negative)</td>
</tr>
<tr>
<td></td>
<td>No. of trigrams(Negative)</td>
</tr>
</tbody>
</table>

Increase in the number of features directly affects the increase in the accuracy of system shown in Figure 1. So, we used eight features for maximum coverage. Positive and negative features are taken in terms of subjective lexicon and n-grams.
The International Arab Journal of Information Technology, Volume 14, No. 5, September 2017

3.6. Corpus Generation

Punjabi dataset is generated by collecting data from Punjabi newspapers [3, 4, 10, 11, 12, 16, 17, 23, 30, 31, 33, 34, 35, 38, 39] and Blogs [1, 9, 36, 40, 41]. Statistics of Resultant Dataset is given in Table 4.

Table 4. Statistics: punjabi corpus.

<table>
<thead>
<tr>
<th></th>
<th>Training</th>
<th>Testing</th>
<th>Newspapers</th>
<th>Blogs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of documents</td>
<td>600</td>
<td>284</td>
<td>721</td>
<td>163</td>
</tr>
<tr>
<td>Number of sentences</td>
<td>31000</td>
<td>14200</td>
<td>36450</td>
<td>8150</td>
</tr>
<tr>
<td>Number of sentences per document(avg)</td>
<td>50</td>
<td>50</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>Number of words(total)</td>
<td>21000</td>
<td>99400</td>
<td>252350</td>
<td>57050</td>
</tr>
<tr>
<td>Number of words per document</td>
<td>359</td>
<td>359</td>
<td>350</td>
<td>350</td>
</tr>
</tbody>
</table>

3.7. Human Annotation

For accuracy of polarity assigned in the Subjective Lexicon, we hired 3 human annotators who have provided their agreement/disagreement regarding the polarity.

4. Methodology

Process flow of system is described by following Algorithm given below.

Algorithm:

1. Compute the positive and negative score using subjective lexicon described in Section B. Compute the no. of positive and negative trigrams, bigrams, and unigrams using n-gram modelling explained in Section E.
2. Compute the no. of positive and negative trigrams, bigrams, and unigrams using n-gram modelling explained in Section E.
3. Devise the feature vector which will include the information of step 1 and 2. The design of Feature Vector is explained in Table 3.
4. This Feature Vector is used for Training and Testing Support vector machine by using Training and Testing datasets.

5. Result Evaluation

Implementation of system is done using Java. Figure 3 shows the user interface, input and output of the system. Input is Punjabi text in Unicode format or File can be browsed for input which should be saved by setting encoding is equal to UTF-8. Our System takes the input and provide the result corresponding to all combinations of Lexicon and N-gram and hybrid system. Output also include the Graph which shows the comparison of processing time taken by all combinations and Hybrid System. Graph concludes that Hybrid System always takes lowest processing time. Results of our technique are evaluated to compare with the following techniques in Figure 2:

- Hindi Subjective Lexicon
- Hindi Sentiwordnet
- Bi-lingual Dictionary
- Translated Dictionary

![Figure 1. Increasing accuracy with increase in number of features.](image1)

![Figure 2. Comparison of our system with the existing resources.](image2)

![Figure 3. Snapshot showing Output of the System.](image3)

Results are compared by calculating parameters (Precision, Recall, F-measure and Accuracy). Table 6 shows the results computed and comparisons which conclude the better performance of our system. Confusion Matrix is created using SVM to conclude the performance of system shown in Table 5.

Table 5. Statistics: confusion matrix.

<table>
<thead>
<tr>
<th></th>
<th>Positive</th>
<th>Negative</th>
<th>Neutral</th>
</tr>
</thead>
<tbody>
<tr>
<td>Positive</td>
<td>259</td>
<td>32</td>
<td>9</td>
</tr>
<tr>
<td>Negative</td>
<td>69</td>
<td>223</td>
<td>8</td>
</tr>
<tr>
<td>Neutral</td>
<td>56</td>
<td>21</td>
<td>207</td>
</tr>
</tbody>
</table>

Table 6. Accuracy results.

<table>
<thead>
<tr>
<th>Method</th>
<th>News</th>
<th>Blogs</th>
<th>Naive Bayes</th>
<th>SVM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hindi Sentiwordnet</td>
<td>67.50</td>
<td>68.04</td>
<td>58.42</td>
<td>61.11</td>
</tr>
<tr>
<td>Hindi Subjective Lexicon</td>
<td>81.28</td>
<td>71.4</td>
<td>58.6</td>
<td>61.56</td>
</tr>
<tr>
<td>Bi-lingual Dictionary</td>
<td>70.30</td>
<td>67.05</td>
<td>58.21</td>
<td>61.58</td>
</tr>
<tr>
<td>Translated Dictionary</td>
<td>62.30</td>
<td>61.05</td>
<td>58.38</td>
<td>61.6</td>
</tr>
<tr>
<td>Our System</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lexicon</td>
<td>54.2</td>
<td>56.1</td>
<td>55.43</td>
<td>60.02</td>
</tr>
<tr>
<td>Lexicon + Negation handling</td>
<td>67.50</td>
<td>59.2</td>
<td>62.3</td>
<td>71.34</td>
</tr>
<tr>
<td>Lexicon + Negation handling +Stemming</td>
<td>73.45</td>
<td>68.1</td>
<td>68.52</td>
<td>74.56</td>
</tr>
<tr>
<td>N-gram</td>
<td>30.31</td>
<td>32.31</td>
<td>32.9</td>
<td>33.53</td>
</tr>
<tr>
<td>N-gram + Negation Handling</td>
<td>34.1</td>
<td>41.21</td>
<td>41.84</td>
<td>42.1</td>
</tr>
<tr>
<td>N-gram+ Negation handling +Stemming</td>
<td>42.41</td>
<td>46.54</td>
<td>47.33</td>
<td>48.02</td>
</tr>
<tr>
<td>Lexicon+ N-gram+ Negation handling +Stemming</td>
<td>71.23</td>
<td>70.34</td>
<td>72.67</td>
<td>78.02</td>
</tr>
</tbody>
</table>
6. Conclusions and Future Work

The system developed as part of our research, proves better performance but still efficiency is compromised. The factors which contribute towards low efficiency, can be considered for future research-

1. Coverage by lexical resources: lexicon used by us is deprived of concept of understanding the contextual information. So, Machine learning can be used in the integration with the Subjective Lexicon for the training of the system.

2. Contextual issue-Meaning of word is independent from that of other word in the sentence so Lexicons fail at this particular point. This problem is also categorized as contextual dependency. To solve this problem dynamic prior polarity can be added.

3. Vocabulary mismatch- vocabulary composition differs due to Cultural anthropology which states that the difference in the culture based on location, society, environment, and people. This problem can be solved by using the concept of Morphological Analysis.

We have done our research by the using of subjective lexicon, support vector machine, n -gram modeling and combination of these. This research can be further extended by adding more features like:

- Word length
- Punctuation
- Vocabulary Richness
- Frequency of function words
- Phrase patterns

References

International Conference on Natural Language Processing, India, pp.1-6, 2010.


Amandeep Kaur has completed her Masters degree in Computer Science & Engineering from Panjab University, Chandigarh, in 2014. Her research interests are Natural Language processing, Image Processing, Machine learning, Computer Vision.

Vishal Gupta is Sr. Assistant Professor in Computer Science and Engineering at University Institute of Engineering and Technology, Panjab University Chandigarh. He has written around 70 research papers in international and national journals and conferences. He has developed a number of research projects in field of NLP and text Mining like keywords extraction system, automatic question answering and text summarization system etc.