ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

*(An ISO 3297: 2007 Certified Organization)*

**Vol. 3, Issue 4, April 2015**

# Simulation of Opinion Mining In Hindi Language Based on Natural Language Processing

Hridesh Gupta[1], Pankaj Sharma[2]

M.Tech (CS&E), School of Computing Science and Engineering, Galgotias University, Greater Noida, U.P, India[1]

Assistant Professor, School of Computing Science and Engineering, Galgotias University, Greater Noida, U.P, India[2]

**ABSTRACT-** Microblogging is a very common mode of communication among internet users. Microblogs are real time content published by people and this content is generally laden with personal opinions about a variety of aspects in everyday life. This makes microblogs a rich source of data for opinion mining. We use a corpus from the popular microblogging website, Twitter [1]. We consider microblogs from the period before the Prime minister's elections in India in 2014 to analyze the collective sentiment of the microbloggers, against and in favor of each Prime Minister candidate. We classify the microblogs to positive and negative opinion classes and we use machine learning classification techniques achieve this and translator translate the all language reviews and microblogs convert to Hindi language.

**KEYWORDS:** Opinion Mining, Sentiment Analysis, Reviews, Naïve Bayes' Theorem, Hindi translator.

## I. INTRODUCTION

Microblogging is a very common and powerful mode of communication among internet users. Microblogs often reflect personal opinions and are very useful for opinion mining. We considered a corpus from twitter. Microblogs in twitter are called tweets. We choose tweets from the period of Prime Minister elections in India in 2014. We mainly try to classify the Tweets into political opinions against and in favor of the Prime Minister candidates, Narendra Modi and Rahul Gandhi. We used various feature selection techniques and classification algorithms. The best results were obtained by using n-gram features with support vector machines (SVM) classifier. The Status corpus used is manually annotated for sentiments and we use this as a gold standard for evaluation of precision, recall and f-score of our classification this and translator translate the all language reviews and microblogs convert to Hindi language using the Hindi translator.

## II. RELATED WORK

Sentiment analysis of Status is a growing area of research. Since twitter has a limit characters per post. Opinion table has been work done in sentiment analysis of twitter corpus by machine learning classification methods [Pak & Paroubek, 2010]. Opinion table has been work done specifically for political opinion mining from Twitter [Maynard & Funk, 2011]. Different from using different features and classifiers, there are variety of methods used like use of emoticons [Go et al., 2009], use of opinion reversal words etc for identifying sentiments. we have used some similar ideas in our data processing.

## III. PROPOSED ALGORITHM

Our approach consisted a variety of ideas borrowed from the zone of natural language processing, information retrieval and machine learning. The algorithm mainly designs of the following steps,

1. Data processing
2. Features
3. Training the classifier

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 4, April 2015

**Data processing**

The corpus has considerable amount of metadata such as date, time, identity number etc and to extract natural language content. We need to subject the data to a series of processing steps before the data can be used to extract features and train a classifier. Here is a sample of the raw data before data process.

315, 41199, AM 0:26:17, News Analysis: In Second Debate Modi Strikes Back (NY Times): Share With Friends.

**1. Stop words removal** - We used the Natural language Toolkit's (NLTK) [2] stopword corpus for the English language to remove the stop words from the paragraph. This helps eliminating the most common stop words from being included in the computation of n-grams and feature extraction.

**2. Stemming** - Twitter data is generally used with informal language and it includes internet jargons, slang and contemporary spellings. We were very frugal in stemming so as to not risk truncating words and losing out on probable features. We employed basic stemming (e.g. use of Bosnion).

**3. Spelling correction** - As Twitter users generally use unofficial language. There are often wrong spellings in tweets. We used Jazzy Open Source Magic Checker [3] to detect incorrect spellings in the tweets, paragraph and replace them with the closest word from the English dictionary.

**4. Entities** - The entities we used were Narendra Modi or Rahul Gandhi. But, these entites are addressed to by various names for e.g. Narendra Modi is generally self addressed as Mr. Prime Minister, Narendra Modi etc. So, we normalised this by replacing the possible names people use to address the entities by either Modi or Rahul.

**5. Emoticons mapping to sentiments** - There are a multitude of emoticons that are used repeatedly in Twitter. We used an approach inspired by the method used by [Go et al., 2009] and mapped some emoticons to positive and negative sentiments and discarded emoticons that are ambiguous or irrelevant to sentiments.

| Emoticons Mapping | |
|---|---|
| (+ve) sentiment | (-ve) sentiment |
| :)<br>:)<br>:D | :(<br>:( |

**6. Part of Speech tags** - We experimented with both NLTK and OpenNLP [4] Part Of Speech tagger with a heuristic that adjectives and/or adverbs are generally used to articulate opinions in natural language. The data was tokenized by spaces and the tokens were subject to the taggers. We also experimented by adverbs and words such as not, excluding all data sans the entity, adjectives, could not etc. which generally indicate a reversal of sentiment. He is equal to the route opinion reversing words were used by [Maynard & Funk, 2011]

**7. Filtering** - Tweets contained a lot of metadata and crop a bit of din which were removed. The following data was filtered,

- Identity numbers, date, time etc ,of the tweets.
- Irrelevant tags
- Hyperlinks
- #tags e.g. #msnbc2012
- Twitter handles e.g. @Pawan
- punctuation, special characters and digits

**8. Encoding** - There are a few tweets that are not in Hindi and other languages. These tweets contain UTF-8 encoded words for e.g. naive. These characters were excluded from the tweets and only ASCII encoded characters were hold in.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 4, April 2015

All content was also transformed to lowercase. These characters interfered with the classifiers.

After subjecting a tweet to this data processing process, we are left with only the natural language content of the tweet and the human annotated sentiment that is used by the classification algorithm in supervised learning. After data processing, the sample tweet we considered earlier about would be transformed to,
"news analysis second debate Modi strikes back ny times share friends top news,-1"

1for positive, -1 for negative are the annotations used for sentiment classification.

**9. Translator –** Translator is translating the all microblogs and tweets convert into Hindi language and microblogs & tweets are any language but translator translate into Hindi language. Using the Google translator.

## IV. FEATURES

**N-gram features** - The processed data is used to extract features that will be used to coach our classifier. We have experimented with Ngram. The data was tokenized by spaces using NLTK and these tokens were subject to NLTK to generate n-grams.

**Part Of Speech features** - Since the language used in Twitter is generally informal, part of speech tagging is not accurate for tweets. We used both NLTK Part Of Speech tagger and Open NLP Part Of Speech tagger along with a heuristic that adjectives and adverbs, JJ, JJR, JJS, RB, RBR and RBS in the Penn Tree bank tag set, are generally used to articulate opinions in natural language. So we further process the data by excluding all data saving. The entity, adjectives, adverbs and words such as not, could not etc which generally indicate a reversal of sentiment. This is similar to the route opinion reversing words were used by Maynard and Funk [Maynard & Funk, 2011]. After using part of speech taggers, we experimented with unigrams, bigrams and combination of unigrams and bigrams. We experimented with term recap-inverse document recap (tf-idf) where we considered only the most frequent terms ordered by tf-idf. We used the absolute approach of considering all the n-grams as features as well.

**Sentiment Lexicon features** - We used the terms in positive and negative opinion word lists [Hu & Liu, 2004] as features for classification.

## V. TRAINING THE CLASSIFIER

We experimented various combinations of features, classification algorithms and test options. As mentioned in the Feature extraction section, we extracted various feature sets and used these to construct the feature vectors. These were used to train the classifiers. We experimented with 4 different classifiers,

- Multinomial Naïve Bayes
- Logistic regression
- Random forest
- Support vector machines (SVM)

We used these classifiers from weka (weka is a machine learning algorithm), a popular suite of machine learning software [5] and Lib SVM, a library for Support Vector Machines [6]. We also experimented with one Vs all classification strategy with SVM. We experimented with the evaluation methods, 70-30 percent split and 10-fold cross validation.

## VI. PSEUDO CODE

**STEP 1:** Initialize P(positive) ←num – popozitii (positive)/ num_total_propozitii
**STEP 2:** Initialize P(negative) ←num – popozitii (negative) / num_total_propozitii
**STEP 3:** Convert sentences into words for each class of {positive, negative}: for each word in {phrase}
P(word | class) < num_apartii (word | class) 1 | num_cuv (class) + num_total_cuvinte
P (class)← P (class) * P (word | class)
Returns max {P(pos), P(neg)}
Convert sentence into english(all languages) to Hindi using google translator

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 4, April 2015

## VI. EXPERIMENTAL RESULTS

We used various combinations of features and classifiers and here are the different experimental results using both Weka and LibSVM. Table-1 shows the results of using 8000 most frequent unigrams without removal of stop words used with Naïve Bayes Multinomial classifier using a 70-30% evaluation method. We got no significant changes in results when we experimented with used Logistic regression and Random forest classifiers. Table-2 shows the results of using 10000 unigram features ordered by tf-idf using Naïve Bayes classifier. We used 10-deflect cross validation for the following results.

**Table 1:** Confusion matrix - 8000 most frequent unigrams without stopword removal as features using a Multinomial Naïve Bayes classifier

Narendra Modi

| Opinion | precision | Remember | f-measure |
|---|---|---|---|
| Positive | 63.9 | 06.3 | 11.4 |
| Negative | 46.6 | 95.1 | 62.6 |

Rahul Gandhi

| Opinion | purity | Remember | f-measure |
|---|---|---|---|
| Positive | 53.3 | 09.9 | 16.7 |
| Negative | 47.5 | 94.5 | 63.2 |

**Table 2:** Confusion matrix - 10000 unigram features ordered by tf-idf using Naïve Bayes classifier

Narendra Modi

| Opinion | Purity | Remember | F-measure |
|---|---|---|---|
| Positive | 44.8 | 50.5 | 43.8 |
| Negative | 23.9 | 20.4 | 33.4 |

Rahul Gandhi

| Opinion | Purity | Remember | F-measure |
|---|---|---|---|
| Positive | 59.0 | 65.8 | 70.2 |
| Negative | 34.9 | 31.1 | 18.7 |

We experimented with positive and negative terms in opinion word lists [Hu & Liu, 2004] as features for classification.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 4, April 2015

**Table 3:** Confusion matrix - Opinion word lists as features using Logistic regression classifier

Narendra Modi

| Opinion | Purity | Remember | F-measure |
|---|---|---|---|
| Positive | 62.0 | 44.6 | 23.4 |
| Negative | 36.6 | 29.0 | 32.5 |

Rahul Gandhi

| Opinion | Purity | Remember | F-measure |
|---|---|---|---|
| Positive | 24.8 | 24.5 | 20.5 |
| Negative | 61.6 | 75.5 | 68.0 |

The results obtained by employing part of speech tagging for feature selection are shown in Table-4.

**Table 4:** Confusion matrix - Parts of speech tags (adjectives/adverbs tags) used for feature selection and using Naïve Bayes classifier

**Narendra Modi**

| Opinion | Purity | Remember | F-measure |
|---|---|---|---|
| Positive | 57.7 | 13.0 | 15.4 |
| Negative | 23.7 | 06.7 | 03.9 |

**Rahul Gandhi**

| Opinion | Purity | Remember | F-measure |
|---|---|---|---|
| Positive | 40.8 | 11.8 | 23.4 |
| Negative | 59.8 | 85.5 | 48.0 |

We then trained the model on the entire training corpus and evaluated the test data by using the combination of unigrams and bigrams as features and SVM classifier employing a one Vs all classification strategy .Table-5 shows these results. We used LibSVM for this experiment.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 4, April 2015

**Table 5:** Confusion matrix - Test data - unigram & bigram features using SVM classifier and one-vs.-all classification strategy

**Narendra Modi**

| Opinion | Purity | Remember | F-measure |
|---|---|---|---|
| Positive | 41.58 | 48.79 | 45.05 |
| Negative | 46.86 | 55.29 | 50.72 |

**Rahul Gandhi**

| Opinion | Purity | Remember | F-measure |
|---|---|---|---|
| Positive | 33.70 | 39.74 | 36.60 |
| Negative | 54.79 | 57.17 | 55.95 |

## VIII.CONCLUSION

In this paper we have presented an all language corpus for opinion mining along with its Hindi translation. All language corpora are freely available for the research community. The Translator corpus is composed of all language reviews obtained from web pages related to movies and films. Then, we have generated the all language corpus, which is the Hindi translation of the using an automatic machine translation tool. Both corpora include a total of 500 reviews, 250 positives and 250 negatives. In addition, we have accomplished several experiments over the corpora using two different machine learning algorithms (SVM and Naïve Bayes) and applying a stemming process.

## REFERENCES

[1] Twitter, Microblogging website http://www.twitter.com/
[2] Natural Language Toolkit 2.0 (NLTK) http://nltk.org/
[3] Jazzy Open Source Spell Checker http://jazzy.sourceforge.net/
[4] OpenNLP, Apache Software Foundation http://opennlp.apache.org/
[5] Weka 3, machine learning software suite http://www.cs.waikato.ac.nz/ml/weka/
[6] Library for SVM, a library of SVM
[7] [Go et al., 2009] A. Go, R. Bhayani, L. Huang: Twitter Sentiment Classification using Distant Supervision, Technical report, Stanford Digital Library Technologies Project, 2009.
[8] [Maynard & Funk, 2011] D. Maynard, A Funk: Automatic detection of political opinions in Tweets, ESWC Workshops, pages 88 -99, 2011.
[9] [Pak & Paroubek, 2010] A. Pak & P. Paroubek: Twitter as a Corpus for Sentiment Analysis and Opinion Mining, LREC, 2010.
[10] [Liu, 2012] Bing Liu: Sentiment Analysis and Opinion Mining, Morgan & Claypool Publishers, 2012.
[11] [Hu & Liu, 2004] M. Hu and B. Liu, KDD, 2004.
[12] https://translate.google.co.in