




Sentiment Analysis of Hinglish Text and Sarcasm Detection

Abhishek Gupta¹, Abinash Mishra², and U. Srinivasulu Reddy² 

¹ Department of Computer Science and Engineering, Indian Institute of Information Technology, Tiruchirappalli, National Institute of Technology, Tiruchirappalli Campus, Tiruchirappalli, India
abhi.mittal021@gmail.com

² Machine Learning and Data Analytics Lab, Department of Computer Applications, National Institute of Technology, Tiruchirappalli, Tiruchirappalli 620015, India
{405117002,usreddy}@nitt.edu

Abstract. Today the term “Sentiment Analysis” is no newer to the world. It falls under the umbrella of Natural Language Processing which is a very interesting and creative field of artificial intelligence. One important aspect which needs to take in consideration before going for Sentiment Analysis is the kind of the language of the data which is supposed to be processed. People in urban areas of the northern part of India used to communicate in the mixed language of Hindi- English which is commonly termed as “Hinglish”. While doing sentiment analysis one needs resources like Dictionary containing polarity for words, part of speech tagger for both of the languages. A lot of resources were developed for the English language, but this does not hold true for Hinglish. The aim of this research is not only to carry out sentiment analysis and sarcasm detection but also to contribute to the resource development for the Hinglish language. In this paper, Sentiment Analysis is done to classify sentences as positive, negative, sarcastic and non-sarcastic. This is done using extended sentiwordnet 3.0 and naïve Bayes classifier. From the current study it is analyzed that, sentiment analysis using SentiWordNet gives a better precision than the Naïve Bayes whereas the latter successfully classified the sentences into sarcastic and non-sarcastic.

Keywords: Hinglish · Hindi SentiWordNet · English SentiWordNet · Hinglish SentiWordNet · Code switch · Part of speech · Lexicon · Naïve Bayes Classifier

1 Introduction

1.1 Sentiment Analysis

Sentiment Analysis is very useful in today’s era of advanced learning and technology. It has gain popularity due to its heavy potential to break into a person’s inner world via extracting his/her emotions from the data which he/she continuously posting on social platforms like Facebook, twitter etc. If someone gets the set of emotions a person carrying with him, it becomes a cakewalk for that person to predict his/her opinion about something more accurately and that’s what the Sentiment Analysis is all about. This

analysis of emotions has vast fields of applications ranging from enhancing selling on E-Commerce platforms to preparing influential political agenda for elections. While dealing with the analysis of emotions, one should not overlook the very aspect of the psychology of human being that [1] whenever a person stands at the peak of exhibiting his emotions, he expresses them in his mother tongue. This data is very rich in emotions. A lot of people on social media use other languages than English and there is a lot of transliteration involved due to easy typing. Hindi is such language which is widely being spoken in the northern part of the country [2]. Particularly in urban areas of the country, people consciously or unconsciously, frequently use English words while communicating in Hindi. This frequent switching between Hindi and English commonly known as “Hinglish”. Hinglish example: “Aaj ka movie show houseful hai”. It’s Hindi containing English words written in Roman Script, which when translated in English means “Today’s movie show is houseful”. This Hinglish data, available on various social networking platforms, contains valuable information. One can exploit it using various text analysis means like sentiment analysis and sarcasm detection. This kind of diverseness in the language is another reason which prompts researchers to go for Hinglish sentiment analysis.

1.2 Sarcasm Detection

Sarcasm is something that diverts the sentiment and meaning of an utterance from its literal meaning. So, it is very important to detect sarcasm while doing sentiment analysis. Some of the main factors that constitute a sarcasm are a change in tone while speaking, facial expression, body movements, use of over intensified words etc. This is the reason that it is as easy to detect it in oral communication as tough in written form of communication. This is why sarcasm detection, in order to classify the data, involves a lot of groundwork on the text like extraction of lexical and syntactic features in the text. For example, leveraging certain lexical features like emoticons, interjections and N-grams in this regard. In the current work, classifier was trained on such features to filter sarcastic sentences from non-sarcastic ones.

2 Literature Survey

2.1 Sentiment Classification

Pandey et al. [3] used HindiSentiWordNet (HSWN) to find the overall sentiment associated with the document of Hindi movie reviews. They improved the existing HSWN by adding missing sentimental words related to Hindi movie domain. Subramaniam Seshadri et al. [4] added Hinglish words to knowledge base along with the English word in a bid to improve the result of sentiment analysis and higher accuracy. However, they limit their research work to Hinglish dictionary improvement and didn’t cover the parts of speech tagging for Hinglish sentences. Mulatkar [5] used the Word Sense Disambiguation algorithm to found out the correct sense of words. Gupta et al. [6] used a pre- annotated corpus and additional inclusion of phrases, checking for the overall polarity of the review with negation handling. The authors successfully classified movie

reviews, which are in Hindi language, as positive, negative and neutral. However, their approach didn't classify the movie reviews written in Hinglish language which amount a big chunk of movie reviews posted on social media regularly. Kaur et al. [7] did an extensive study of many machine learning methods such as Support Vector Machine (SVM), Naive Bayes, Decision Tree and showed that these methods are suitable while classifying literary artworks especially poetry. Yadav and Bhojane [8] proposed a system for sentiment analysis of Hindi health news, which used their own corpus to find the overall sentiment associated with the document. They used a Neural Network to train the polarity words stored in the database to make the processing faster.

From the above discussion, it is clear that accuracy of sentiment analysis depends upon the reliable resources such as Sent WordNet, pre-annotated corpus etc. In the current work, we developed a Hinglish SentiWordnet by mixing English and Hindi Sent Wordnet. Besides we also developed a part of speech tagger for Hinglish code-switch language.

2.2 Sarcasm Detection

Bouazizi et al. [9] performed sarcasm detection on Twitter data. First, they classified the features which are useful in sarcasm detection in four different sets and then, based on their presence in sentences and pattern of their occurring, classified sentences as sarcastic and non-sarcastic. Four sets of features which were proposed by them are pattern, sentiment-focused, syntactic & semantic and punctuation-focused features. The authors successfully filtered out the sarcastic sentences from dataset. However, their work is suitable for English language. For a code-switch language like Hinglish, there is a need to develop parts of speech tagger to identify the suitable features for sarcasm detection. Bindra et al. [10] used different Twitter tags as sentiment labels. A Twitter tag can be a reference to a specific user, hashtags or URLs. For example, "@" is used to tag a user like @Sachin. "#" is used for a hashtag like #sarcasm, #sad, #wonderful etc. These annotations used to develop the corpus for sarcasm and sentiment classification. They used the Twitter API to collect such sentences. Logistic regression (LogR) and support vector machine with sequential minimal optimization (SMO) were two classifiers they used in their experiment. Bharti et al. [11] proposed a Hadoop based framework that captured real-time sentences and processed them with a set of algorithms which identified sarcastic sentiment effectively. Apache Flume was used for capturing sentences in real time. For processing these sentences stored in the HDFS, they used Apache Hive. Further, Natural Language Processing (NLP) techniques like POS tagging, parsing, text mining and sentiment analysis were used to identify sarcasm in these processed sentences. However, such resources are neither well developed nor openly available for Hinglish language.

From the above discussion, it is clear that the presence of special symbols, exclamatory signs, hyperbolic utterances leads to sarcasm in a sentence. So, identification of such pieces of evidence is important which would be served as features to classifier for sarcasm detection.

In consideration to the above discussion, sentiment classification and sarcasm detection namely Dictionary approach and Machine Learning techniques were chosen for sentiment classification and sarcasm detection respectively. It is also evident from the

above discussion that both the approaches have pros and cons. Lexical analysis can be used directly on data and does not require any pre-annotated data. For sarcasm detection, machine learning techniques are good. This required pre-annotated data as well as classified sets of features so that classifier can be trained to detect sarcasm.

3 Proposed Work

In this paper, authors aimed to carry out sentiment analysis and sarcasm detection for Hinglish sentences. To accomplish the same, authors suggested a hybrid approach which combines the idea of sentiment analysis with sarcasm detection. They first carried out sentiment analysis using dictionary-based approach and then the result of the same supplied as a test data to Naïve Bayes classifier for sarcasm detection. The final output is the combined result of the dictionary-based approach and the Naïve Bayes classifier. The authors developed resources like a Hinglish SentiWordNet for sentiment analysis and part of speech tagger for tagging Hinglish code-switch language sentences. The proposed solution is a hybrid of sentiwordnet based approach, for classifying sentiments of Hinglish text, and the Naïve Bayes classifier for sarcasm detection in Hinglish text.

3.1 Hinglish SentiWordNet Approach

The English sentiwordnet (ESWN) is extended by adding the Hindi sentiwordnet (HSWN) to it. To accomplish this, at first transformation of HSWN is done so that it becomes compatible with ESWN. After that ESWN is appended by adding transformed HSWN. Finally, more Hinglish words are added into extended SentiWordNet for better precision. Figure 1 shows the flow chart of the proposed method.

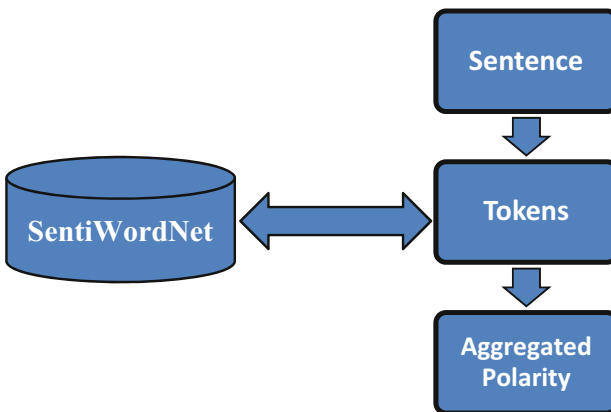


Fig. 1. Flow diagram for lexicon-based method.

In this approach when a sentence is fed to the proposed system, firstly the sentence is splitted into tokens. Then, each token is looked upon in the extended sentiwordnet. Sentiwordnet contains the sentiment score for each word. The Proposed system first

checks for the category of the current token i.e. token is an adjective (a), noun (n), adverb (r) or a verb (v). After that, it looks for its sentiment score. Based on the score it returns the sentiment category of the token as either positive or negative. In this way proposed system performs for all tokens of a sentence. At last the proposed system takes the sum of all sentiment scores corresponding to all tokens of a sentence. If the total score is greater than zero, then classified the sentence as positive. If the total score is less than zero, then classified the sentence as negative.

3.2 Naïve Bayes Classifier Approach

The Naive Bayes classifier is one the important technique in machine learning. It is based on the Bayes theorem which calculates a conditional probability for an event to occur given the historical data of another event which already occurred. Thus, it predicts the occurrence of an event in light of another previously occurred event. In this way, it can be applied to predict whether a sentence is sarcastic or not.

Formula based on Bayes theorem is given below:

$$P(X \text{ and } Y) = P(X) * P(Y|X) \quad (1)$$

Here,

$P(X)$ = Probability of event X

$P(X \text{ and } Y)$ = Probability of event X and Y

$P(Y|X)$ = Probability of event Y given event X

Figure 2 shows the proposed approach for Sarcasm detection.

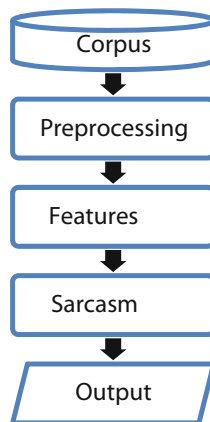


Fig. 2. Flow chart for sarcasm detection.

Preprocessing. The gathered data contains URLs and other non-useful data like a hash-tag (#), annotation (@) which needs to be omitted. Some special characters are also present in the data, which are very useful for the detection of the emoticons. So before removing unwanted things first detect the emoticons present in dataset and replace them

with one of the two generic words i.e. “zxp” and “zxn”. Here “zxp” symbolize all emoticon with positive sentiment and “zxn” symbolize all emoticon with negative sentiment. Dataset also contains sentences with #sarcasm which is also useful in case of sarcasm detection. So that detect this special hashtag in all sentences and replace it with the generic word “zxs”. Now remove all other special characters and unwanted data.

Feature Selection. Part of speech tags are used as the features for the classification model. To extract part of speech tags, first split sentences into tokens and then store them in a list. Now read the part of speech tags of required features and store them in another list in the same order in which they are present in the sentence. Then, return this list to the classification model for learning.

Part of Speech Tagging. Part of speech of a sentence are very useful in identifying the pattern on which sarcasm detection is based. These parts of speech served as features to the classifier. For code-switch languages, tagging part of speech is a multiple fold process. For Hinglish code-switch language, the author suggests a two-fold process for parts of speech tagging. In this process, the author first tagged the sentences with Stanford NLP POS Tagger. This tagger tagged the part of the English language of a sentence with appropriate tags and the parts of the Hindi language as FW (Foreign Word). After that author process the output from the first level tagging and replace the FW tag with appropriate language tag based on an algorithm. According to this algorithm, developed system looks at each and every token tagged as FW and replace FW tag with appropriate language tag based on Hindi grammar rules defined for verbs, nouns, adjectives etc. For example, if a token ends with “ta”, “ti”, or “te” than it would tag as verb. It tags all words which are in uppercase as CAPS, all occurrences of “zxs” as SAR, all occurrences of “zxp” as HBP, all occurrences of “zxn” as HBN, all Hindi noun word as NN and all Hindi verbs as VB.

Sarcasm Detection. The Pre-labelled data is used to train the classifier. The proposed system extract features from both of the files as discussed above and pass it to Naïve Bayes classifier during the learning phase. In this way, training of classifier is done. The classifier lookout for the presence of features which amount to sarcasm in test data as per learning and labelled each sentence accordingly as Sarcastic or Not- Sarcastic.

3.3 Performance Measure

To measure the performance of the above experiment, this paper uses confusion matrix and F- score. From the confusion matrix, accuracy and precision can be calculated. F- score is used to measure the performance of Naïve Bayes classifier. It equals to the harmonic mean of precision and recall. Formula to calculate F- score is given below:

$$F\text{- Score} = 2 * (P * R)/(P + R) \quad (2)$$

Here,

P = Precision, indicates the relevancy between correctly classified sentiments out of total classified sentiments. Mathematically,

$$P = (\text{True Positive})/(\text{True Positive} + (\text{False Positive})) \quad (3)$$

R = Recall, indicates the relevancy between correctly classified sentiments out of total existing corresponding sentiments. Mathematically,

$$R = (\text{True Positive}) / ((\text{True Positive}) + (\text{False Negative})) \quad (4)$$

4 Results

4.1 Sentiment Classification

The test was carried out by giving dataset as input to the proposed system. The proposed system breaks the sentence into tokens. Then it looks for the respective sentiment score. Based on the score it returns the sentiment category of each token. Finally, the polarity of the sentence arrived at by adding all the corresponding sentiment scores of each token. The performance of Hinglish Sentiwordnet for the classification of Hinglish sentences as positive or negative is given in Table 1.

Table 1. Confusion table of sentiment analysis.

N = 1000	Actual positive	Actual negative
Predicted positive	534	27
Predicted negative	36	403

The performance of Hinglish SentiWordnet is evaluated and expressed in Table 2.

Table 2. Performance of Hinglish SentiWordNet.

Data size	Precision	Recall
1000	95.18	93.68

Analyzing Tables 1 and 2, it is clear that the proposed Hinglish SentiWordnet performs well for classifying Hinglish Data.

4.2 Sarcasm Detection

The test was conducted using the Naive Bayes classifier and the following results were harvested. The test data was passed to the proposed system. Then said classifier lookout for the presence of features which amount to sarcasm as per learning and labelled each sentence accordingly as Sarcastic or Not- Sarcastic. The performance of Naïve Bayes classifier for sarcasm detection is given in Table 3.

Table 3. Confusion table for sarcasm detection.

N = 1000	Actual sarcastic	Actual non-sarcastic
Predicted sarcastic	199	16
Predicted non-sarcastic	4	781

The performance of Naïve Bayes Classifier is evaluated and expressed in Table 4.

Table 4. Performance of naïve bayes classifier.

Data size	F-score
1000	95.20

Analyzing Tables 3 and 4, it is clear that the proposed Naïve Bayes Classifier performs well for classifying Hinglish Data.

4.3 Combined Result of Sentiment Analysis and Sarcasm Detection

The author arrived at the hybrid result by combining the above said approaches in a sequential manner i.e. first performed the sentiment analysis using the Hinglish Senti Wordnet on test data and classified the sentences in two categories named positive and negative. Then this classified data is supplied as test data to train Naïve Bayes classifier

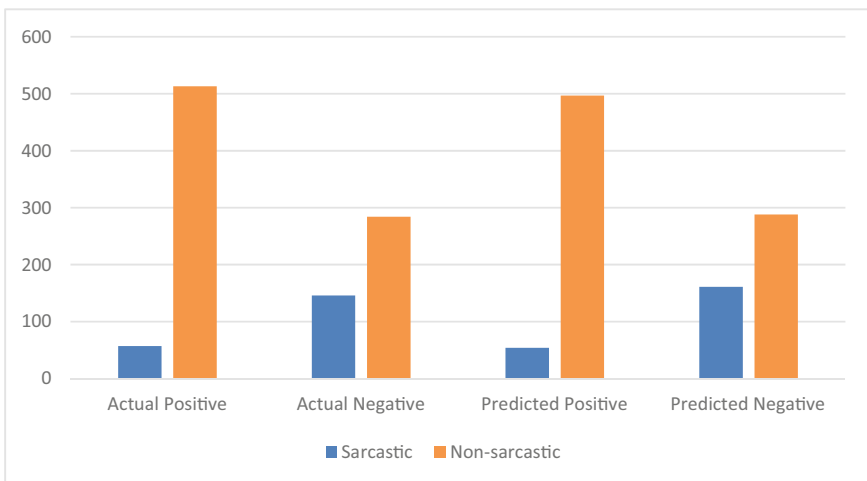


Fig. 3. Combined result of sentiment analysis and sarcasm detection.

for sarcasm detection. The trained Naïve Bayes classifier successfully labeled these sentences as sarcastic or non-sarcastic based on the features present in them. The final output is a combination of both approaches *i.e.*, Hinglish sentences get classified out of these four following categories: positive-sarcastic, negative-sarcastic, positive-non-sarcastic and negative-non-sarcastic. The combined result showed how many sentences are sarcastic and non-sarcastic in nature out of positive and negative sentences. The combined result is shown using a Bar Graph in Fig. 3. Also, the plot explains the actual and predicted class with respect to the sarcasm detection.

The above resulted graph reveals interesting conclusions. Originally, out of total sarcastic sentences, 28.08% sentences are in positive nature and remaining 71.92% sentences are in negative nature. The statistics of predicted data also establishes the same thing. In the resulted data, out of total predicted sarcastic sentences, 25.12% sentences are in positive nature and remaining 74.88% sentences are in negative nature. This result establishes the general inclination of sarcasm towards negative sense.

5 Conclusion

This work has analyzed the Hinglish language data. The author proposed a dictionary-based method to analyze sentiments. The author also proposed a machine learning based method to filter such sentences into two categories namely Sarcastic and Non-Sarcastic.

The author suggests that the English SentiWordNet can be extended by appending its content with the content of Hindi SentiWordNet. The author shows that such Extended SentiWordNet proved useful in sentiment analysis of Hinglish sentences. The system developed by the author breaks Hinglish sentences into tokens to check their polarity. The Hinglish SentiWordNet returns the polarity of each token which later aggregated to get the overall polarity of a sentence. In this way proposed system able to classify Hinglish sentences as positive or negative. The proposed study showed a better measure of F-score **95.20%**, precision **95.18%**, and a recall value of **93.6%**.

To categorize the sentence into sarcastic and non-sarcastic authors trained the naïve Bayes classifier with different sets of features which are responsible for the detection of sarcasm in such sentences. Once it analyzes the sentence, it annotates the sentences with a tag (sarcastic or non-sarcastic).

The limitation of the techniques discussed above guides the author towards future explorations. More concretely, it will be beneficial to incorporate the context, in which sentences are utter, in sentiment analysis and sarcasm detection. This kind of context-aware analysis significantly improves the performance of the proposed system. The context, in which sentences are utter, affects the polarity of sentences according to its sentimental nature. Sometimes it makes a normal comment a sarcasm if its sentimental nature is directly opposite to that of said comment.

The performance can further be improved by applying deep learning architecture and its variant towards the improvement in measure of F-score. Also, penalize algorithm can be implemented in order to reduce the miss-classification error, in turn the performance can further be improved from the existing.

References

1. Puntoni, S., de Langhe, B., Van Osselaer, S.M.J.: Bilingualism and the emotional intensity of advertising language. *J. Consum. Res.* **35**(6), 1012–1025 (2009)
2. Sailaja, P.: Hinglish: code-switching in Indian English. *ELT J.* **65**(4), 473–480 (2011)
3. Pandey, P., Govilkar, S.: A framework for sentiment analysis in hindi using hswm. *Int. J. Comput. Appl.* **119**(19), 23–26 (2015)
4. Shehadri, S., Lohidasan, A., Lokhande, R., Save, A., Nagarhalli, T.P.: A new technique for opinion mining of hinglish words. *Int. J. Innov. Res. Sci. Eng. Technol.* **4**(8), 7184–7189 (2015)
5. Mulatkar, S., Bhojane, V.: Sentiment classification in Hindi. *IOSR J. Comput. Eng.* **17**(4), 100–102 (2015)
6. Gupta, A., Sonavane, D., Attarde, K., Shelar, N., Mate, P.: Sentiment analysis of movie reviews in Hindi. *Int. J. Tech. Res. Appl.* **41**, 31–33 (2016)
7. Kaur, J., Saini, J.R.: Emotion detection and sentiment analysis in text corpus: a differential study with informal and formal writing styles. *Int. J. Comput. Appl.* **101**(9), 1–9 (2014)
8. Yadav, M., Bhojane, V.: Design of sentiment analysis system for Hindi content. *Int. J. Innov. Res. Sci. Eng. Technol.* **4**, 12054–12063 (2015)
9. Bouazizi, M.: A pattern-based approach for sarcasm detection on Twitter. *IEEE J. Mag.* **4**, 5477–5488 (2016)
10. Bindra, K.K., Gupta, A.: Tweet sarcasm: mechanism of sarcasm detection in Twitter. *Int. J. Comput. Sci. Inf. Technol.* **7**(1), 215–217 (2016)
11. Bharti, S.K., Vachha, B., Pradhan, R.K., Babu, K.S., Jena, S.K.: Sarcastic sentiment detection in tweets streamed in real time: A big data approach. *Digit. Commun. Netw.* **2**(3), 108–121 (2016)