International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 7, Issue 7, July 2017)

Comparative Study of Dictionary Based and Machine Learning Approaches for Hinglish Text Sentiment Analysis

Harpreet Kaur[1], Dr. Veenu Mangat[2], Nidhi[3]

[1]Research Scholar, [2,3]Assistant Professor, UIET, Panjab University, Chandigarh, India

***Abstract*— With the recent development of web 2.0, there has been a lot of increase in social networking and online marketing sites. The data or reviews obtained from these sites are analyzed for better human decision making. Sentiment analysis is a part of natural language processing which involves extraction of opinions or sentiments from reviews. Opinions can be classified into positive, negative or neutral. Most of the content on the internet is in the English language, but with the improved awareness of people, data in other languages is also increasing gradually. India is a country of many languages. Sentiment analysis of English is very popular but not much work has been done in Indian languages. These days, a ton of correspondence in online networking happens to utilize Hinglish content which is a mixture of two languages- Hindi and English. Hinglish is an informal language which is exceptionally famous in India as individuals feel greater talking in their particular language. In this paper, we present a dictionary-based approach for Hinglish text classification. We also implemented traditional machine learning classification algorithms such as SVM (Support Vector Machine), NB (Naïve Bayes) and ME (Maximum Entropy) for comparison. It is found that for Hinglish text, Dictionary based classification gives best accuracy results.**

***Keywords*— Sentiment Classification, Feature Extraction, Dictionary Development, Wordnet, Machine learning**

I. Introduction

Sentiment analysis is the process of analyzing people's emotions or opinions for entities like products, movies, news, etc. Data present on the internet is unstructured in nature. It needs processing like classification or clustering to provide useful information for future decision making. Sentiment analysis process involves Data Gathering, Text Pre-Processing, Feature Extraction, Feature Selection and Sentiment Classification which are explained in section 4. Some relevant terms in sentiment analysis are:

Subjectivity/Objectivity: Text which holds some conclusion or sentiment is called subjective text. For example "Rang de basanti bahut achi movie hai (Rang de basanti is a marvelous film)". On the other side, in the objective text does not hold any sentiment i.e. neutral. For example: " Raj Kumar film ka director hai (Raj Kumar is the director of the movie)".

For Sentiment analysis, we just require subjective content which can be additionally grouped into positive or negative. *Polarity:* Subjective text can be categorized as positive or negative which is called polarity of the text.

Sentiment level: Sentiment analysis can be performed at three levels-1) Document level in which the entire file is given positive or negative polarity. 2) Sentence level in which each sentence is checked to provide positive or negative polarity. Overall polarity of file is registered by numbering the positive and negative sentences or comments. Dominant part decides the general sentiment. 3) Phrase level in which expressions or aspects in a sentence are broke down to group as positive or negative. We perform sentiment analysis at document level.

Most of the existing work has been done in the English language. But nowadays in India, there is lots of increment in Hinglish content. The Hinglish content is a mixture of Hindi and English language. It is a Hindi sentence composed using the English language. For instance: "Film bahut achi thi (Movie was great)". This sentence has importance in Hindi language however it is composed using Roman English rather than Hindi Devanagri script. It is vital to break down this sort of content as a large portion of Indians remark or comment in this format only. An insignificant measure of work has been done in this content. This is also called code-mix sentiment analysis.

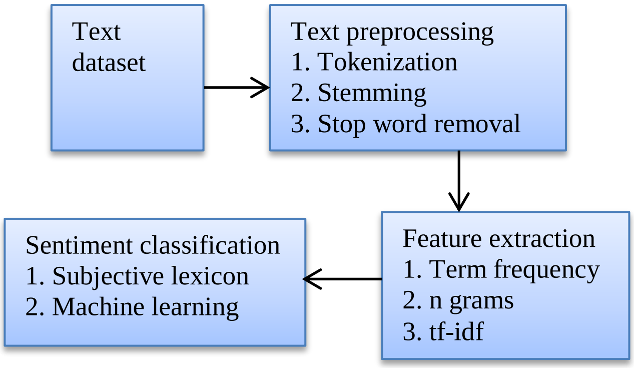


Figure 1. Sentiment Analysis Process

II. Literature Survey

2.1 Studies on Hindi text sentiment analysis

Sneha et al. [1] used WSD (word sense disambiguation) algorithm to correct sense disambiguation in text. Sense disambiguation is given to the system and it then finds the correct sense of word in a context. SVM is used for classification. Feature set includes term presence vs. term frequency and term position. SVM separate two classes- one is root words and other is affixes. Steps in algorithm are stop word removal, sorting of single column word line and then sorted list is stemmed and each word is compared with next 10 words. Assumption is that at least 10 morphological variations are present in the list. If word is present as substring in next word then it is divided into word + remaining characters of word. That substring is treated as root from and rest as affix.

Richa et al. [2] performed sentence-level sentiment analysis i.e. we get positive, negative and neutral sentences separately. They develop a Hindi dictionary to perform sentiment analysis of Hindi sentences. In it, the polarity of the sentence is found by a majority of opinion words that are present in the sentence whether they are positive or negative. Hindi movie review dataset is first preprocessed which involves stop word elimination and then it is given to a Hindi POS (Parts of Speech) tagger. The function of POS tagger is to identify adjective and adverbs in the sentence which holds the most of the sentiment. For dictionary preparation, a seed list of most frequently used Hindi opinion words is prepared. The output of POS tagger which is adjectives or adverbs are compared with this list. If there is a match, then their scores are noted. In case no match is found, then their synonym is considered and is matched with author constructed Hindi dictionary. Finally, majority words determine the overall polarity.

Pooja et al. [3] used HSWN (Hindi SentiWordnet) to find the sentiments related to Hindi movie review dataset. They performed document level sentiment analysis. Polarity of words in the reviews is fetched from HSWN and then aggregated to find the overall polarity of the review. Negation handling is also performed. Words which are not present in HSWN, their polarity is found with the help of synset replacement algorithm. This system has two objectives- Improving existing HSWN and Sentiment extraction. For the first objective, English SentiWordnet is used. Words which are not in HSWN are first translated to English and then their polarity scores are found. In second objective, sentiment is extracted by finding the overall polarity of the document that can be positive or negative or neutral. Preprocessing involves tokenization, negation handling and spell checking.

Deepali et al. [4] develop an improvised polarity lexicon to overcome the limitation of existing HindiWordnet which is very generic in nature. They work on datasets of hotel and movie domain. The built lexicon reflects context sensitivity and shows an improvement in the accuracy. Their system has two objectives: To build an improvised context sensitive polarity lexicon for specific domain and to improve the lexicon coverage by using the approach of HindiWordnet. Preprocessing involves tokenization, POS tagging and lemmatization (reduction to root word). Then tf-idf score of each opinion word is calculated and final polarity score is calculated. This is termed as Context specific polarity lexicon (CSPL). Then adverbs and adjectives are extracted from it. Their synonyms are extracted from HindiWordnet. If synonym is present in CSPL then there is no change in polarity score, otherwise extracted synonym is added to CSPL and same polarity score is assigned to it.

Mittal et al. [5] performed Hindi text sentiment analysis with main focus on Negation handling and discourse relation. They develop rules for negation presence for example to reverse the sentiment if words like “nahi”(not) etc. are found. To handle conjunctions like “fer bhi”(still), “magar”(but) , sentences after conjunction are given more importance than the before part. For example in review “movie ki starting achi thi magar aage ja kar bilkul bkwas ho gai” , text after conjunction(magar) is given more importance. They also developed an algorithm to improve existing HindiSentiWordNet (HSWN). For an adjective found in review data which is not present in HSWN, it is translated to English language and its polarity score is found from SentiWordnet. After that it is translated back to Hindi and added to HSWN with same polarity score.

Akshat et al. [6] performed Hindi sentiment classification using lexicon based approach. In this method, they first prepare a seed list using adjectives and adverbs from review data and then expand it using synonym and antonym relation. They build their own HindiSentiWordNet using word linking because HSWN is not publically available for use. They used graph based traversal to generate their lexicon. They assign polarity scores to lexicon words from. Their scoring algorithm adds all the scores of opinion words in a sentence and results in positive, negative or neutral outcome.

2.2 Studies on English text sentiment analysis

Pang et al.[12] performed sentiment classification using machine learning algorithms. They have used English movie review dataset for this purpose. Their dataset consists of 1301 positive and 752 negative movie reviews.

They have used n-gram feature set as a unigram, bigram and POS feature sets were also taken. Three machine learning algorithms Support Vector Machine, Naïve Bayes, and Maximum Entropy were used for classification. Their analysis shows that SVM is the best classifier for their dataset and Naïve Bayes gives least accuracy among three.

Moraes et al. [13] performed document-level sentiment classification in which the whole document is classified as positive or negative. They conduct a comparative study between SVM and Artificial Neural Network (ANN). They took benchmark IMDb movie review dataset developed by Pang et al.[1]. Their study shows that ANN outperformed SVM regarding classification accuracy. Also, training time of ANN is found to be larger than SVM.

Abinash et al. [14] performed sentiment analysis using n-gram feature set and machine learning algorithms such as SVM, NB, ME (Maximum Entropy) and SGD (Stochastic Gradient Descent). They extended the work of Pang et al.[1] on IMDb dataset by considering trigram features along with unigram and bigram features. To convert text into numerical values, they use CountVectorizer and Tf.idf (term frequency. inverse document frequency) methods. They concluded that as the value of n in n-gram increases the classification accuracy decreases.

Yang et al.[15] performed multiclass sentiment analysis i.e. three classes are there: positive, negative and neutral. They conducted several experiments to compare the performance of popular feature selection (document frequency, information gain, CHI statistics and gain ratio) and machine learning algorithms (support vector machine, radial basis function neural network, naïve Bayes, k nearest neighbor and decision tree). They work on publically available tweet dataset. Their study shows that gain ratio performs best among all the feature selection algorithms and support vector machine among classification algorithms.

Mittal et al.[16] performed an empirical comparison between prominent feature extraction methods. In their study, they have extracted four feature sets unigram, bigram, bi-tagged and dependency parsing tree-based features. Information gain and mRMR (minimum redundancy maximum relevancy) algorithm are used for feature selection. Their study shows that when four basis features sets are taken, mRMR gives better results than information gain. Also, naïve Bayes is superior to SVM regarding accuracy and execution time.

Qiang et al. [17] performed sentiment analysis on travel destinations sites reviews. Their aim is to compare various machine learning classifiers by varying training data sizes. They compare SVM, NB and N-gram based character language model.

N-gram based character language model outperforms both the classifiers. N-gram works on words but this model works on characters of words. Their study shows that with increasing training data, accuracy of all the classifier increases.

Moreo et al. [19] performed sentiment analysis on 500 news comments using lexicon based approach. Their work has two main objectives: 1) To detect the focus of user's comment in a multi domain environment 2) To detect sentiment of comments. They performed comment level sentiment analysis i.e. if a comment is composed of 3 sentences then all three sentences are analyzed separately. They performed filtering process to remove unwanted comments or advertisements. Polarity of comment is classified as negative, negative, neutral, positive and very positive.

Kang et al. [20] performed sentiment analysis on restaurant reviews. They take into consideration the problem that while classification of positive and negative reviews, there are more chances of positive reviews classification accuracy to appear 10% higher than negative review classification. This leads to decrease in average accuracy. To reduce this problem, they proposed an improved Naïve Bayes algorithm. Their study shows that by using improved Naïve Bayes algorithm along with unigram-bigram feature set narrow down the gap between positive and negative classification accuracy results to 3.6%.

Erik et al. [21] performed sentiment analysis on reviews written in English, Dutch and French languages. For training purpose, they used manually annotated positive, negative and neutral data. They use SVM, MNB (Multinomial NB) and ME for classification purpose. Results show that maximum accuracy is achieved for English text. Reason for low accuracy for Dutch and French is larger variety of language expressions and small training dataset.

Huaxia et al. [22] studied the effect of positive and negative tweets on movie sales. They studied how twitter's WOM (Word of Mouth) effect movie businesses by using popular machine learning algorithms. Their study shows that WOM does matter but it depends on how many followers does author have and also what WOM is about. More the number of followers more is the impact of tweet. Also, strongest reaction comes from tweets in which authors suggest to watch some movie.

Isa et al. [24] developed a lexicon model for deep sentiment analysis applications. Their model provides a detailed description of relations that can exist between actors in the reviews and actors who write the reviews.

This model gives subjective information about identity of actor in the review and orientation of his comment polarity. Special focus is given to the writer's own perspective and his views on what is happening in the review comment.

Maite et al. [23] used lexicon approach for sentiment extraction. Name of their system is Semantic orientation calculator. They use intensification and negation handling for better results. POS tagging is not only restricted to adjectives, other parts are also taken into consideration. The dictionaries have words annotated with their semantic orientation (positive or negative). Their system is not restricted to single domain. It is applicable to all domains.

2.3 Studies on Hinglish text sentiment analysis

Shanshank et al. [7] performed code mix sentiment analysis i.e. reviews having both English and Hindi words. Statistical method is used to find the sentiment in which if a number of positive words in the review is more, then the statement is tagged as positive and vice versa. Their methodology involves language identification in which words are tagged to corresponding languages as tag /H for Hindi and /E for English. Next step is spell correction e.g. guddd becomes gud. Next step is to handle ambiguous words. These are some words which are present in both languages (Hindi and English) like 'so', 'do', 'teen' etc. Sounds like 'awww', 'boo', 'opps' are also handled. Next step is to transliterate Roman Hindi to Devanagari Hindi. Then polarity of English and Hindi words is found using SentiWordnet and HindiSentiWordnet respectively.

Kumar et al. [8] performed sentiment classification using Hinglish news and Facebook dataset.

Their aim is to find the best pair of feature set and classifier algorithm. The first step is to pre-process dataset which includes Tokenization, Stop word removal and Document term matrix (DTM) preparation. Training data is created by human annotators manually as there is no lexicon available for Hinglish. Their study concluded that Tf.idf method for feature extraction, gain ratio method for feature selection, and Radial Basis Function (RBF) Neural Network gives the best accuracy for Hinglish text classification.

Rupal et al. [9] performed sentiment analysis on code mix sentences which include English, Tamil, Telugu, Hindi and Bengali language text. FIRE 2015 dataset is used which is a mixture of multiple languages. They used language identification and transliteration approach. After language identification, next step is to convert the script back to its original script. Then corresponding SentiWordnet are searched to calculate overall sentiment score. Statistical technique is used in which if number of positive words is more than negative words then sentence is given positive polarity.

Prashati et al. [11] performed sentiment analysis on tweets generated by Indian users which consists of English and Hinglish text. They use dictionary based approach in which two dictionaries are built- positive and negative. These dictionaries contain opinion words from the tweets which contribute in sentiment classification. Testing data is compared with dictionaries. If number of positive words in sentence is more than negative words, then sentence is considered as positive. They use word cloud to represent the summary of words in documents. In word cloud, most frequent terms appear bold and bigger.

Table 1:
Summary of research articles on Hindi, English and Hinglish text sentiment analysis

Author	Objectives	Algorithm	Dataset	Results
Research on Hindi text sentiment analysis				
Akshat et al.(2012)	1.They prepare lexicon by using a seed list created using adjectives and adverbs 2.They build their own HindiSentiWordNet using word linking	Lexicon based approach	Product reviews	Accuracy: Baseline 74.62 Baseline + NH 74.96 Baseline + Stem 78.27 Baseline + Stem + NH 79.03
Mittal et al.(2013)	1.Hindi text sentiment analysis with main focus on Negation handling and discourse relation 2. Improving existing HSWN	Lexicon based approach	Hindi movie reviews	Accuracy: Only Existing HSWN 50.45 With Improved HSWN 69.78 With Improved HSWN + Negation 78.39

				Improved HSWN +Negation+ Discourse 80.21
Richa et al.(2014)	Polarity detection of movie reviews by preparing seed list of Hindi words	Dictionary based approach	Hindi movie reviews	Accuracy:65%
Sneha et al.(2014)	Used word sense disambiguation algorithm to find correct sense of word in given context	SVM	Travel domain reviews	For given Hindi disambiguated list, corrected sense list is found
Pooja et al.(2015)	HSWN is used to find polarity of text at document level	Lexicon based approach	Hindi movie reviews	Improvement to existing HSWN
Deepali et al.(2016)	1.They build an improvised context sensitive polarity lexicon for given domain 2. Improving existing HSWN	Lexicon based approach	Hotel and movie Hinglish dataset	Accuracy: HSWN 52.5(movie) 46(hotel) CSPL 71(movie) 88(hotel) CSPL + HSWN 76.5(movie) 85(hotel) CSPLE 77(movie) 82.5 (hotel) CSPLE + HSWN 75(movie), 81.5(hotel)
Research on English text sentiment analysis				
Pang et al.(2002)	Classification using different feature selection and machine learning algorithms	SVM, NB, Maximum entropy(ME)	IMDb (Internet movie database)	Accuracy: Unigram: SVM(82.9), Bigram: ME(77.4), Unigram+Bigram: SVM (82.7)
Erik et al.(2008)	Sentiment analysis of multi-lingual text using well known machine learning algorithms	SVM, ME, MNB	English, Dutch, French reviews	Accuracy English 83 Dutch 70 French 68
Qiang et al.(2009)	Checking output variations by varying training dada size	SVM, NB, N Gram based computer language model	Travel destination site reviews	Accuracy 40 reviews: SVM(64.5), NB(51.3), N Gram (66.1) 240 reviews: SVM(80.5), NB(74.9), N Gram (78.3) 500 reviews: SVM(84.6), NB(80.1), N Gram (83.4)
Moreo et al.(2012)	To detect the focus of user's comment in a multi domain environment	Lexicon based approach	News comments	Accuracy Focus detection: 64%
Kang et al.(2012)	Improved algorithm to minimize the problem of accuracy gaps between classification of positive and negative	Improved Naïve bayes	Restaurant reviews	Gap between positive and negative classification accuracy results reduced to 3.6%
Isa et al.(2012)	Development of lexicon model for deep sentiment analysis and opinion minig	Extension of lexicon database for Dutch	Englsih sentences	Detailed description of relations between actors
Moraes et al.(2013)	Empirical comparison between ANN(Artificial neural network) and SVM	ANN and SVM	IMDb (Internet movie database)	Accuracy: ANN(90.3%), SVM(89.6%)

Huaxia et al.(2013)	Study of how positive and negative tweets effect the sales of movie tickets	NB	Movie tweets	1.More the number of followers more is the impact of tweet 2.Strongest reaction comes from tweets in which authors suggest to watch some movie.
Abinash et al.(2016)	Classification using different feature selection and machine learning algorithms	SVM, NB, ME, Stochastic gradient descent(SGD)	IMDb (Internet movie database)	Accuracy: SVM(88.94%), ME(88.48%), NB(86.23%), SGD(85.11%)
Mittal et al.(2016)	Comparison of feature selection methods for sentiment classification	Boolean Multinomial Naive Bayes(BMNB), SVM	Cornell movie review dataset	Accuracy: BMNB (92.5%), SVM (90.2%)
Yang et al.(2017)	Multiclass sentiment classification using different feature selection and machine learning algorithms	Decision tree(DT), SVM, RBFNN, NB, KNN	Movie review dataset	Accuracy: SVM(82.5%), RBFNN(77.9%), NB(75.8%), KNN(57.6%), DT(61%)
Research on Hinglish text sentiment analysis				
Shanshank et al.(2015)	Statistical approach is used for sentiment analysis of code mix(English+Hinglish) sentences	Lexicon based approach	FIRE 2014 and FIRE 2013 data	Precision : 80%
Kumar et al.(2016)	Classifying Hinglish dataset using different feature selection and machine learning algorithms	NB CART, JRip, SVM, Radial Basis Function Neural Network (RBFNet), Logistic Regression (LR), J48 (Decision Tree), Multi-layer Perceptron (MLP) and Random Forest(RB)	News and Facebook comments	Accuracy: RBFNet(86%), NB(65%), CART(70%), LR(78%), RF(79%), MLP(76%), SVM(79%), J48(51%), JRip(58%)
Rupal et al.(2016)	Statistical sentiment analysis on code mix sentences which include English, Tamil, Telugu, Hindi and Bengali language text.	Logistics regression(LR), SVM	FIRE 2015 dataset along with manually collected data	Precision: Bengali 0.43(LR), 0.59(SVM) English 0.61(LR) 0.79(SVM) Gujrati 0.17(LR), 0.09(SVM) Hindi 0.34(LR), 0.50(SVM) Kanad 0.42(LR) 0.47(SVM) Malayalam 0.38(LR), 0.39(SVM) Marathi 0.38(LR), 0.42(SVM) Tamil 0.65(LR), 0.78(SVM) Telugu 0.34(LR) 0.58(SVM)
Prashati et al.(2016)	Sentiment analysis of tweets generated by Indian users which consists of English and Hinglish text	Dictionary based approach	Hinglish tweets	Word cloud represents dominant sentiment

This chapter discusses latest studies in the field of Hindi, English and Hinglish sentiment analysis. The popular techniques for sentiment classification are machine learning algorithms such as SVM, NB, ME, ANN and lexicon based approach, dictionary based approach etc. Following figures provides a summarization of above study.

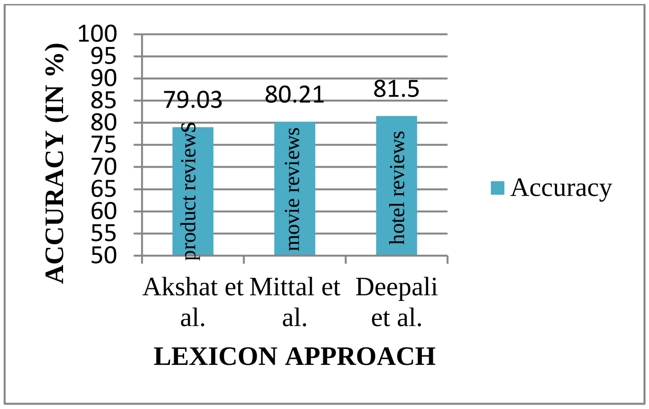


Figure 2: Summary of Hindi sentiment analysis using Lexicon approach

Figure 2 shows accuracy comparison of lexicon approach for different Hindi reviews datasets such as Product reviews, Movie reviews, Hotel reviews etc. The reason for difference between accuracies is due to use of different feature sets such as negation handling, discourse relation, stemming etc.

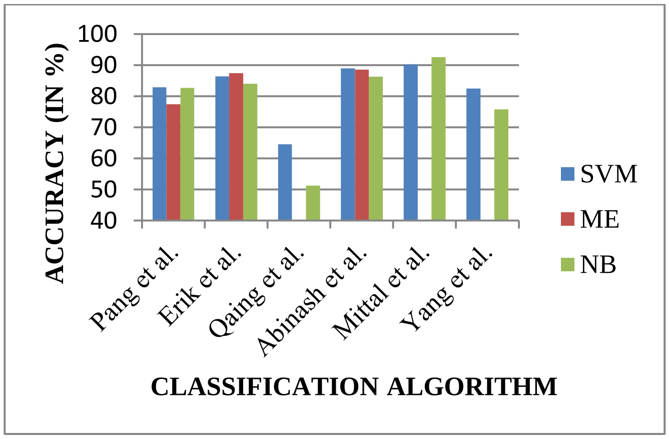
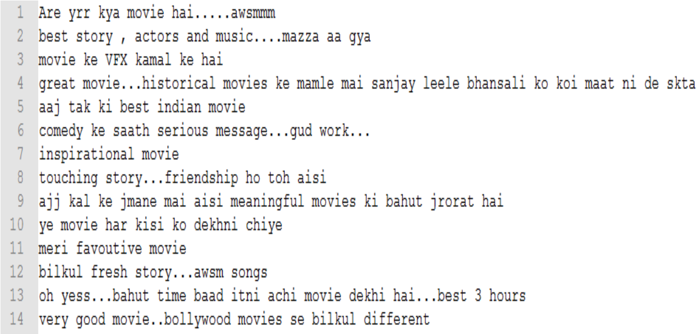


Figure 3: Summary of English sentiment analysis using machine learning algorithms

Figure 3 shows accuracy comparison of machine learning algorithms (SVM, ME, NB) for English text sentiment analysis. It can be seen from the figure that SVM is superior to other machine learning algorithms (ME and NB) for majority of datasets.

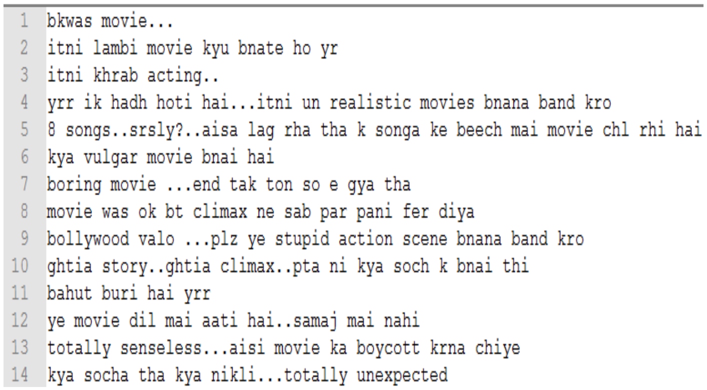
III. Proposed Approach

A. Dataset collection: Hinglish text sentiment analysis is not a very popular field. Therefore there are not many standard datasets present for sentiment analysis. Some available public datasets such as FIRE 2013, FIRE 2014, etc. are there but they have data that is a mixture of five or six other Indian languages along with Hinglish. We extracted some data that is a mixture of Hinglish and English only, but there is not much data for movie domain. Also, we manually collected data from various blogs, social networking sites such as Facebook and Twitter, YouTube, etc. We have collected 100 positive and 100 negative movie reviews. Some examples of our Hinglish dataset are shown in following diagrams.



```
Are yrr kya movie hai.....awsmmm
best story , actors and music....mazza aa gya
movie ke VFX kamal ke hai
great movie...historical movies ke mamle mai sanjay leele bhansali ko koi maat ni de skta
aaj tak ki best indian movie
comedy ke saath serious message...gud work...
inspirational movie
touching story...friendship ho toh aisi
ajj kal ke jmane mai aisi meaningful movies ki bahut jrorat hai
ye movie har kisi ko dekhni chiye
meri favoutive movie
bilkul fresh story...awsm songs
oh yess...bahut time baad itni achi movie dekhi hai...best 3 hours
very good movie..bollywood movies se bilkul different
```

Figure 4. Positive Hinglish movie reviews



```
bkwas movie...
itni lambi movie kyu bnate ho yr
itni khrab acting..
yrr ik hadh hoti hai...itni un realistic movies bnana band kro
8 songs..srsly?..aisa lag rha tha k songa ke beech mai movie chl rhi hai
kya vulgar movie bnai hai
boring movie ...end tak ton so e gya tha
movie was ok bt climax ne sab par pani fer diya
bollywood valo ...plz ye stupid action scene bnana band kro
ghtia story..ghtia climax..pta ni kya soch k bnai thi
bahut buri hai yrr
ye movie dil mai aati hai..samaj mai nahi
totally senseless...aisi movie ka boycott krna chiye
kya socha tha kya nikli...totally unexpected
```

Figure 5. Negative Hinglish movie reviews

*B. Dictionary Building Phase***:** As explained above, no Hinglish lexicon is available for sentiment analysis of the Hinglish text. For English and Hindi language, Wordnet and HindiWordnet are available respectively. These lexicon sources contain words related to English and Hindi. SentiWordnet [10] and HindiSentiWordnet are also developed to find the polarity of words. For example: consider a review like "Movie is awesome". Now to determine polarity of this review i.e. whether it is positive or negative, look for the score of word 'awesome' in SentiWordnet. In SentiWordnet, every word has two scores associated with it which are in the range of 0 to 1. This score indicates whether the word is positive or negative. Similarly, for Hindi words, polarity can be determined using HindiSentiWordnet. But no such lexicon is available for Hinglish text. Therefore our aim is to build a dictionary or lexicon for Hinglish movie reviews.

Table 2.
Extraction of opinion words from dataset

Movie reviews	Opinion words
best story, actors, and music....mazza aa gya	Best, mazza
movie ka screenplay bahut bkwas hai	Bkwas

Two separate dictionaries are created one for English data (which contain English opinion words like 'best' in the above example) and other for Hinglish data ('mazza', 'bkwas'). For dictionary creation, we first extract opinion words from the dataset as shown in Table 2. Then Hinglish and English opinion words are placed in separate files. Next step is to populate the dictionaries by using synonyms of these opinion words to maximize the corpus coverage as shown in Table 3. English dictionary is filled using synonyms of words from Wordnet. Similarly, Hinglish words are filled by their Hindi synonym in HindiWordnet.

Table 3.
Populating lexicon with synonyms of opinion words

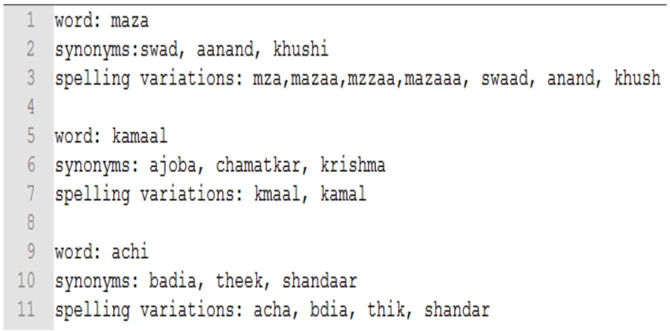
Opinion word	Synonyms	Sources used
Best	Fine, great, excellent	Wordnet
Maza	swad, aanand, khushi	HindiWordnet
Acha	badia, theek, shandaar	HindiWordnet

Additionally, there are many spelling variations of opinion words are found in movie reviews. To handle these kinds of variations, we add some spelling variants of these words to our dictionary as shown in Table 4.

Table 4.
Taking into account the spelling variations

Opinion word	Popular variations
Best	Besttt, bst
Maza	Mazza, mazaaa
Bakwas	Bkwas, bkwass

All possible variations will not be covered by our system, but the most frequently used will be covered. Also, our system is case-insensitive i.e. whether it is 'acha' or 'Acha', they will be considered as same. A snapshot of our Hinglish dictionary is depicted below:



```
word: maza
synonyms:swad, aanand, khushi
spelling variations: mza,mazaa,mzzaa,mazaaa, swaad, anand, khush

word: kamaal
synonyms: ajoba, chamatkar, krishma
spelling variations: kmaal, kamal

word: achi
synonyms: badia, theek, shandaar
spelling variations: acha, bdia, thik, shandar
```

Figure 6. Hinglish dictionary

C. Hinglish stop word list: Words which do not contribute to any sentiment are termed as stop words. Stop word list of English and Hindi text is available online. For Hinglish text, this list is manually created by using Hindi stop word list as shown in Table 5.

Table 5.
Preparation of stop word list for Hinglish text

Review	Opinion words	Stop words
best story, actors, and music....maza aa gya	Best, maza	And, aa, gya
movie ka screenplay bahut acha hai	Bahut acha	Ka, hai

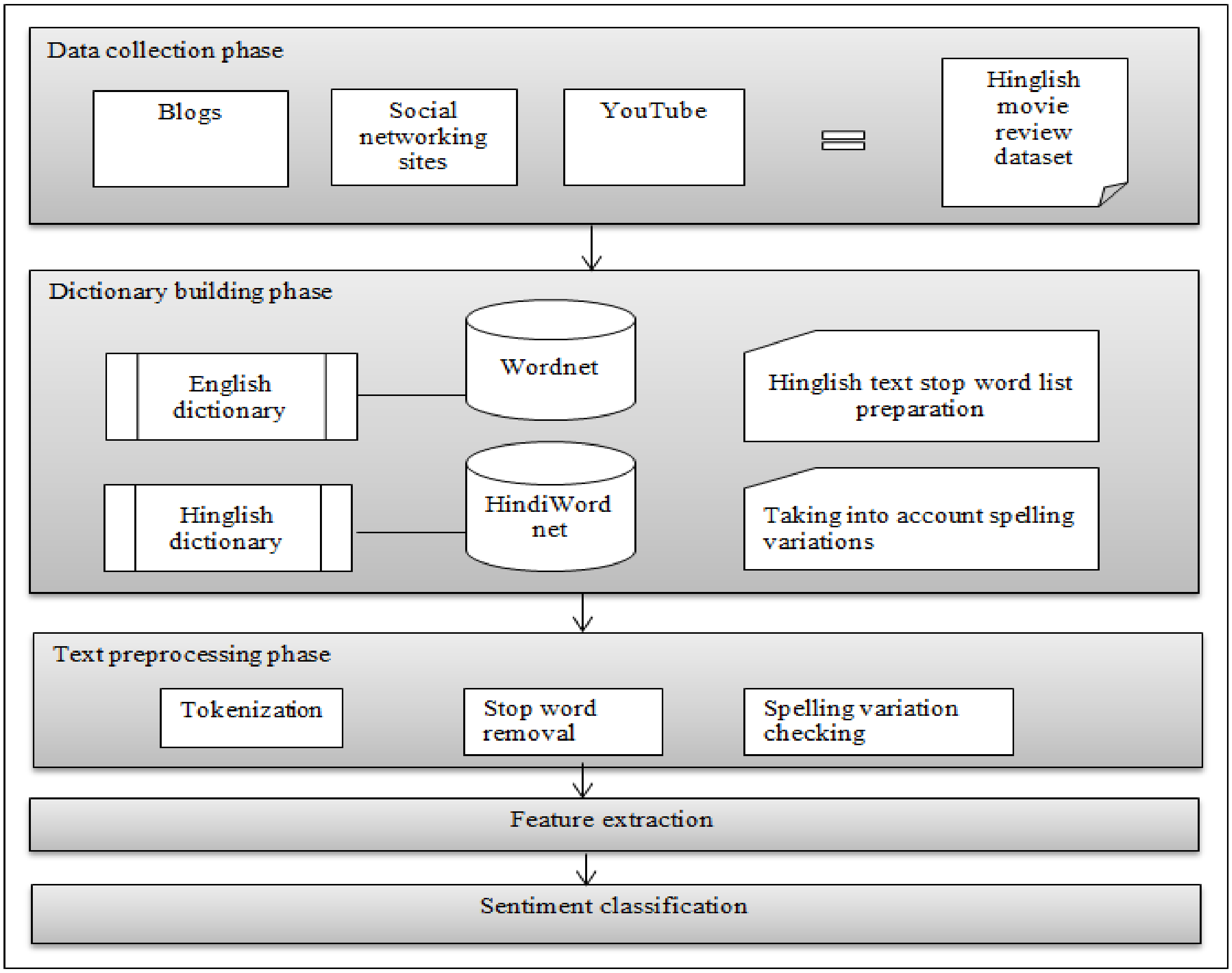


Figure 7. Proposed system for Hinglish text sentiment classification

All paragraphs must be indented. All paragraphs must be justified, i.e. both left-justified and right-justified.

IV. PROPOSED METHODOLOGY

After dictionary building module is completed, next step is to use these dictionaries for sentiment analysis as follows-

A. Text Preprocessing:

It involves following tasks:

Tokenization: The process of converting a text stream into individual tokens is called tokenization.

Stop word removal: Words which do not contribute to any sentiment are termed as stop words. Their removal is required as they decrease the overall accuracy. English stop word list is available online. For Hinglish text, we have used our own created Hinglish stop word list.

Spelling variation checking: A large number of spelling variations of opinion words are found in the reviews. To handle these exceptions, spelling variation list is used which is created in Section 3.

B. Feature Extraction:

It is the process of converting the text into some meaningful representations that make the relevant text available for sentiment analysis. The classifier is trained using the extracted features. Study of literature survey shows that tf.idf (term frequency. Inverse document frequency) feature set gives the best results for Hinglish text (using transliteration) [8]. As we are using dictionary based approach (without transliteration), we also use other feature sets such as unigram, bigram along with tf-idf.

N-gram: n-gram is a sequence adjacent n items from the given text. When n=1 it is called unigram and when n=2 it is called bigram. In this paper, we experimented with unigram and bigrams.

Tf.idf: It stands for Term Frequency. Inverse Document Frequency. This feature extraction technique is used to find the importance of a word to a document in the corpus. It increases as the number of occurrence of a word increases in a document and decreases as the number of occurrences of a word increases in the corpus. To compute tf.idf, we require two things: term frequency and inverse document frequency.

Term frequency: It gives the occurrences of the word in a document. It can be calculated as

$$\mathrm{Tf} = \left(\frac{\text{Occurrences of term in a document}}{\text{Total number of terms in a document}}\right) \qquad (1)$$

Inverse document frequency: It gives information about the importance of a term. While calculating tf, all frequent terms are considered as important, but we know that some terms like 'the', 'of', 'is' are frequent in every document and hence are less important. It can be computed as

$$\mathrm{Idf} = \mathrm{log\_e}\left(\frac{\text{Total number of documents}}{\text{Number of documents with term t in it}}\right) \qquad (2)$$

Then tf/idf is computed by multiplying these two.

$$\mathrm{Tf/idf} = \mathrm{tf} * \mathrm{idf} \qquad (3)$$

Negation handling: Negation means that review contains words such as not, don't, etc. These words reverse the sentiment of the review. For example: "Movie is not awsm". Now it contains the word 'awsm', the classifier can think that it is a positive review. Similarly, in Hinglish, we have this comment as "Movie badia nahi hai". To handle this, if the review contains these words, then sentiment scores are reversed.

C. Sentiment Classification: After feature extraction, we get feature scores which are used for training. We perform dictionary based sentiment classification using our dictionaries. We also have experimented with some machine learning classifiers such as SVM, Naïve Bayes, and Maximum Entropy, etc. and compare their results.

Machine learning algorithms

a. Support vector machine (SVM): It is a supervised learning classifier which aims at finding hyperplane for binary classification problem. Hyperplane divides document vectors into two classes and distance from the both classes is desired to be as large as possible. The problem of finding a hyperplane can be transformed into a constrained optimization problem. [18] Constrained optimization is the process of optimizing some objective function with respect to some variables which have some constraints on them. Here optimization problem is to maximize the margin between two hyperplanes. We know that equation of a line is $y = ax + b$ Where a is the slope and b is intercept, similarly we can create hyperplane equation. A hyperplane can be written as a set of points which satisfy following equation

$$v.x + b = 0 \qquad (4)$$

Where v and x are vectors and b is some point in plane. Suppose H_0 is the hyperplane separating the dataset and satisfying the above equation.

In the same way, we can find two other hyperplanes H_1 and H_2 which separate dataset and satisfy following equations

$$v.x + b = 1 \quad (5)$$

$$v.x + b = \ -1 \quad (6)$$

Next step is to ensure that no point should be present in between them. So we will only select that hyperplane which will satisfy following constraints. For every vector x_i either

$v.x_i + b \geq 1$ for x_i having class 1 i.e. positive or (7)

$v.x_i + b \leq -1$ for x_i having class -1 i.e. negative (8)

A number of SVM kernels have been developed such as RBF (Radial Basis Function) kernel, linear kernel, sigmoid kernel etc. We have used linear kernel which is the basic form of SVM.

b. Naïve Bayes (NB): It is a probabilistic classifier which is based on Bayes theorem to calculate that given set of features belongs to a particular label. This model assumes that features are independent of each other. After the estimation of parameters from training data, classification is carried on the test data by calculating posterior probabilities of classes.

$$P\left(\frac{c}{x}\right) = \frac{P\left(\frac{x}{c}\right).P(c)}{P(x)} \quad (9)$$

Where P(c) = prior probability of class, P(x) = prior probability of predictor, P(c/x) = prior probability of class given predictor, P(x/c) = likelihood which is the probability of predictor given class

$$P\left(\frac{C}{X}\right) = P\left(\frac{x_1}{c}\right).P\left(\frac{x_2}{c}\right).P\left(\frac{x_3}{c}\right)\dots P\left(\frac{x_n}{c}\right).P(c) \quad (10)$$

Once training is complete, we can use following to predict the class of test data

$$V(NB) = argmax\ P(Vi)\ \prod_{i=1}^{n} P\left(\frac{w}{Vi}\right) \quad (11)$$

Where V stands for class or value, Vi can be positive or negative, w stands for words in the review, argmax specifies that max value(probability is calculated for both Vi= positive and Vi= Negative) gives the class label.

c. Maximum entropy (ME): It is also a probabilistic classifier. Unlike NB, ME does not assume that all features are independent. It is based on the principle of maximum entropy.

This classifier selects the model (from all models that fits our training data) with largest entropy. This classifier is mainly used when we have no idea about prior distributions and are unable to make any assumptions. This model is used to predict the probabilities of different outcomes of dependent variables for some given independent variables.

Maximum entropy value in terms of exponential function can be calculated as follows

$$P\left(\frac{c}{d}\right) = \frac{1}{Z(d)}\exp\left(\sum\nolimits_f \gamma_{i.c} f_{i.c}(d,c)\right) \quad (12)$$

Where P(c/d) is the probability of document d belonging to class c, f_{ic} is the feature function of feature f_i and class c, γ_{ic} is the parameter to be estimated and Z(d) is the normalizing factor. Normalizing factor is calculated using following equation

$$Z(d) = \sum\nolimits_c \exp\left(\sum\nolimits_i \gamma_{i.c}\, f_{i.c}\,(d,c)\right) \quad (13)$$

Feature function is calculated using following equation

$$f_{i.c'}(d,c) = \begin{cases} 0, & if\ c = c' \\ \frac{N(d,i)}{N(d)}, & otherwise \end{cases} \quad (14)$$

Where N(d, i) is number of times word i appears in document d, N(d) is total number of words in document d.

D. Performance evaluation metrics

The performance metrics helps in evaluating the performances of machine learning algorithms. From classification perspective, terms like "True positive", "False positive", "True negative" and "False negative" are used to evaluate algorithm performance. True positive (TP) means the review is positive and classifier correctly classifies it positive whereas False positive (FP) means the review is positive, but classifier classifies it negative. Similarly, True negative (TN) says the review is negative, and classifier correctly classify it negative whereas False negative (FN) means the review is negative, but classifier classifies it positive. Using these terms, we can calculate following performance metrics

a. Precision: It is the measure of exactness of classifier result. It is the ratio of correctly labeled positive reviews to the total positive classified reviews

$$Precision = TP/(TP + FP) \quad (15)$$

4.2.2 Recall: It is the measure of completeness of classifier result. It is the ratio of correctly labeled positive reviews to the number of actually positive classified reviews

$$Recall = TP/(TP + FN) \quad (16)$$

4.2.3 Accuracy: It is the ratio of some correctly classified reviews to the total number of reviews.

$$Accuracy = (TP + TN)/(TP + TN + FP + FN) \quad (17)$$

4.2.4 F-Measure: It is the harmonic mean of recall and precision. It is used to optimize the system towards either precision or recall.

$$F - Measure = (\frac{2*Precision*Recall}{Precision+Recall}) \quad (18)$$

4.3 Experimental setup

Experiments are conducted on a PC (Personal computer) using Windows 10 operating system. In our experiments, Netbeans IDE (Integrated Development Environment) version 8.1 is used. Our dataset consists of 300 Hinglish movie reviews (150 positives and 150 negatives). We take 240 reviews (120 positive+120 negatives) for training purpose and 60 reviews (30 postive+30 negatives) for testing purpose. We make unigram, bigram and tf.idf feature set for feature extraction as explained in Section 4. We experiment with three popular machine learning algorithms SVM, NB and ME (described in section 4.1) and compare their results with our proposed Dictionary approach for sentiment classification.

V. RESULTS

According to above given experimental design, experiments are performed. We perform five-fold cross-validation on our dataset. Cross-validation is a validation technique for analyzing how the results of analysis generalize to an independent dataset. Results are shown in the following tables.

Experiment 1: To compare the Accuracy of classification algorithms for I1, I2, I3, I4 and I5

In this experiment, accuracy is computed for different classification algorithms in five iterations for Hinglish movie reviews. As concluded from Table 6, accuracy achieved by Dictionary based approach is 100% for I1 followed by SVM(67%), ME(62%) and NB (47%). For all the iterations, NB performs worst accuracy of 45% as compared to other machine learning algorithms.

Table 6.
Accuracy achieved by classification algorithms for five iterations

Accuracy					
Algorithm	**I1**	**I2**	**I3**	**I4**	**I5**
SVM	67.17	66.39	65.87	62.68	65.19
NB	47.52	46.42	45.86	45.77	45.96
ME	61.89	61.29	61.17	60.86	61.38
Dictionary approach	100	84.00	94.54	97.56	91.30

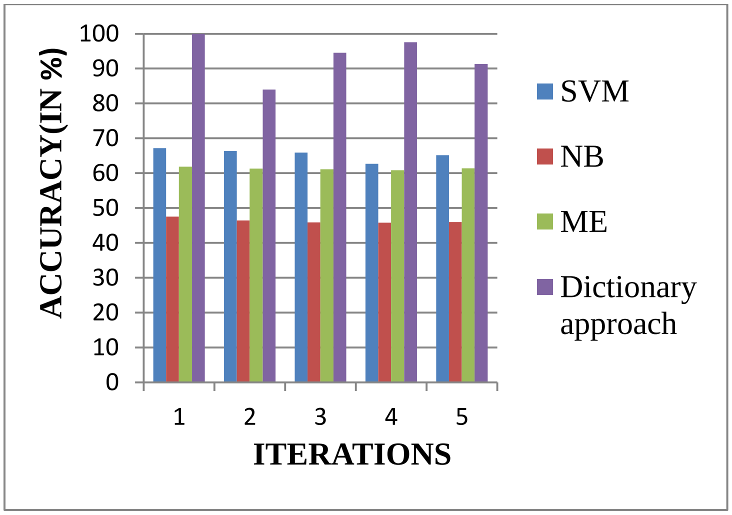


Figure 8. Comparison of Accuracy

Experiment 2: To compare the Precision of classification algorithms in I1, I2, I3, I4 and I5

Experimental results show that for Hinglish text, Dictionary based approach achieve 100% precision for four iterations (I1, I2, I4 and I5) whereas maximum precision achieved by SVM is 79%, ME is 72% for all iterations. NB gives half of the precision achieved by the proposed algorithm i.e. 50%.

Table 7.
Precision achieved by classification algorithms for five iterations

Precision					
Algorithm	I1	I2	I3	I4	I5
SVM	79.55	73.40	78.46	71.66	78.40
NB	49.75	49.75	49.15	49.80	48.75
ME	70.40	67.45	68.50	66.42	72.44
Dictionary approach	100	100	92.85	100	100

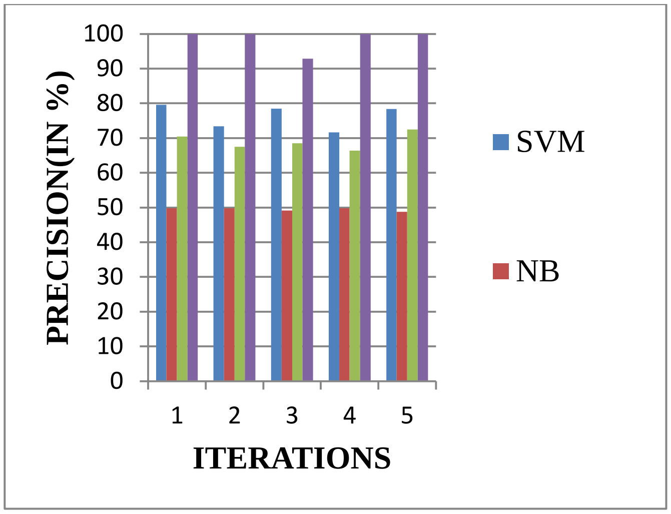


Figure 9. Comparison of Precision

Experiment 3: To compare the Recall of classification algorithms in I1, I2, I3, I4 and I5

The recall is the ratio of true positive to true positive and false negative classified. As depicted from Figure 10, Dictionary based approach achieves best recall values for all iterations followed by SVM, ME and NB. The minimum recall achieved by our proposed algorithm is 76% whereas other algorithms achieve 61%, 53% and 43% for SVM, ME and NB respectively.

Table 8.
Recall achieved by classification algorithms for five iterations

Recall					
Algorithm	I1	I2	I3	I4	I5
SVM	61.29	63.16	61.25	61.25	62.20
NB	46.40	46.40	43.40	43.20	43.40
ME	55.20	56.16	55.25	56.15	53.23
Dictionary approach	100	75.75	96.29	95.23	82.60

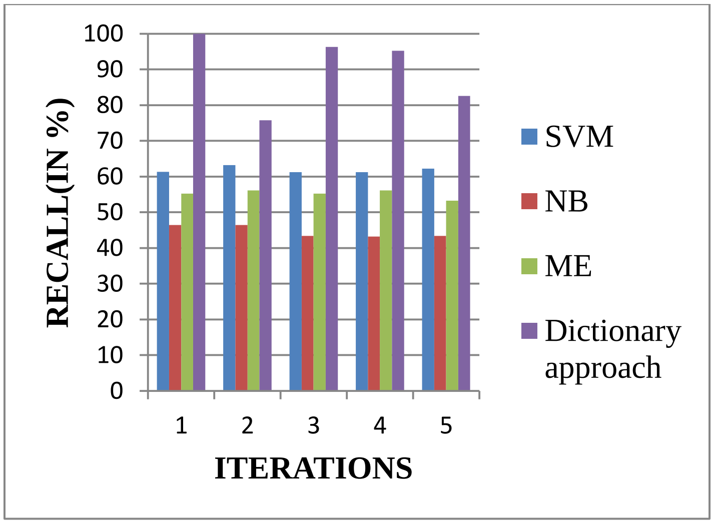


Figure 10. Comparison of Recall

Experiment 4: To compare the F-Measure of classification algorithms in I1, I2, I3, I4 and I5

In this experiment, F-Measure for Dictionary based approach is compared with other classification algorithms for all the iterations. It is found from Figure 11 that Dictionary based approach achieve F-Measure between 86-100% for different iterations. It is also found that, ME shows consistent behavior having F-Measure values around 61% for all the iterations.

Table 9.
F-Measure achieved by classification algorithms for five iterations

F-Measure					
Algorithm	I1	I2	I3	I4	I5
SVM	69.23	67.89	68.79	66.04	69.63
NB	48.01	48.01	46.09	46.20	45.91
ME	61.92	61.28	61.16	60.85	61.36
Dictionary approach	100	86.20	94.53	97.55	90.47

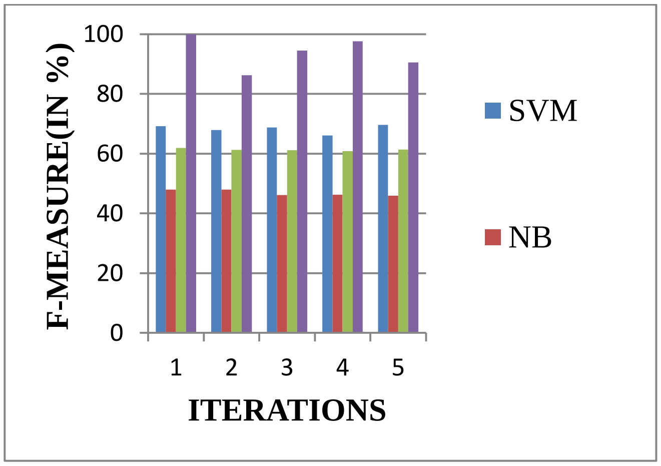


Figure 11. Comparison of F-Measure

Experiment 5: Summary of performance evaluation metrics for Hinglish text

In this experiment, average of accuracy, precision, recall and f-measure for five iterations is computed and compared with SVM, NB, ME and Dictionary approach. It is found that for Hinglish text, Dictionary based approach is superior to other three algorithms by achieving 93.48%, 93.75% accuracy and f-measure respectively. As observed from Table 10, NB is failed to achieve even 50% of average accuracy and f-measure for Hinglish text. Also, SVM performs better than ME by having average accuracy of 65.46% as compared to 61.31%.

Table 10.
Average of five iterations

Average of 5 iterations				
Algorithm	Accuracy	Precision	Recall	F-measure
SVM	65.46	76.29	61.83	68.31
NB	46.30	49.44	44.56	46.84
ME	61.31	69.04	55.19	61.31
Dictionary approach	93.48	98.57	89.97	93.75

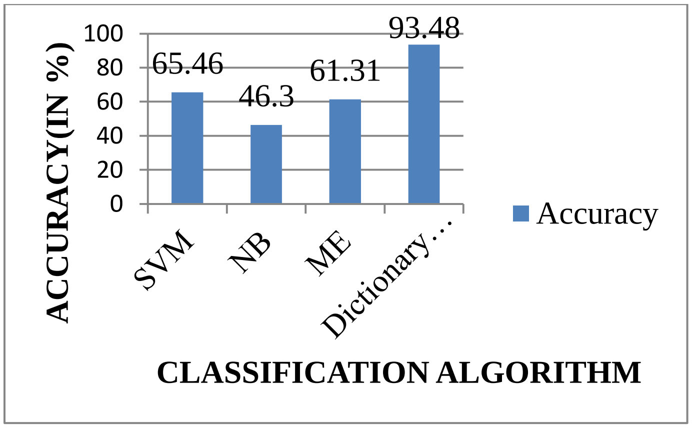


Figure 12. Comparison of average Accuracy of Classifiers

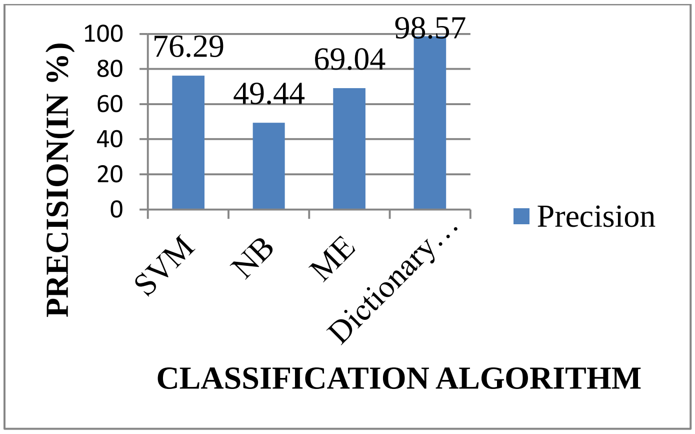


Figure 13. Comparison of average Precision of Classifiers

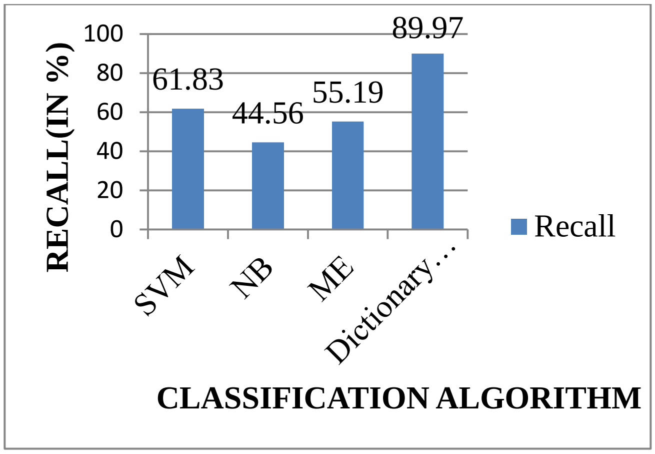


Figure 14. Comparison of average Recall of Classifiers

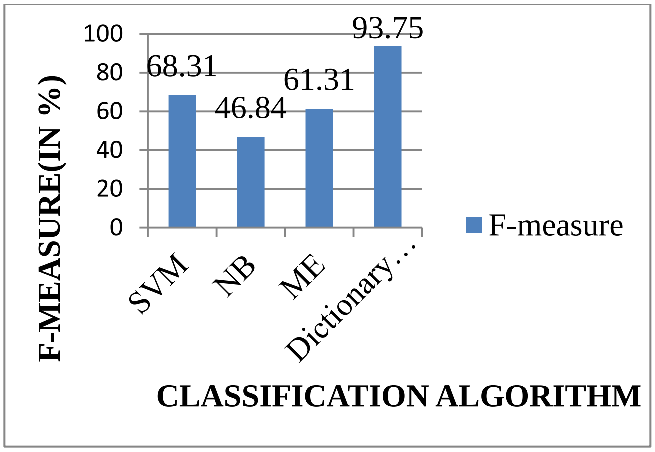


Figure 15. Comparison of average F- measure of Classifiers

It is analyzed from the above figures 12, 13, 14 and 15, Dictionary-based approach performs remarkably better than other three popular machine learning algorithms such as SVM, ME and NB for Sentiment Analysis of Hinglish text. Thus, following is the order of performance achieved by all the four classification algorithms for Hinglish text:

NB<ME<SVM<Dictionary

VI. Conclusion And Future Work

This paper provides an empirical comparison between Dictionary based approach and three modern machine learning algorithms for the purpose sentiment classification of Hinglish text.

Due to lack of resources such as datasets, stop word list, dictionaries for sentiment analysis consisting of positive and negative words, negation handling rules for Hinglish text sentiment analysis, we have created our own dataset of 300 Hinglish movie reviews by extracting reviews from different sites such as YouTube, Facebook, Twitter; build positive and negative dictionaries and Hinglish stop word list for our proposed algorithm. In dictionary based approach, all the above mentioned resources that we have created are properly utilized with negation handling rules for better performance of our algorithm. For performance measure, experiments are carried out on five iterations (as 5 fold cross validation is used) of our Hinglish movie review dataset and following performance evaluation metrics are used: Accuracy, Precision, Recall and F-Measure. Experimental study shows that for Hinglish text, Dictionary based approach gives significant results in terms of all the performance evaluation metrics for all the iterations as compared to machine learning algorithms (SVM, NB and ME).

Our system has some limitations also which can be incorporated in future work:

1. The Hinglish dataset is small in size, consists of 300 reviews only which can be further extended for efficient results.
2. Many reviews contain emoticons along with text which can also be taken into consideration in future.
3. Some reviews contain sarcastic text which can be misinterpreted by the algorithms and degrade their performance.
4. Hinglish text dictionaries can also be automated to incorporate new words for better sentiment analysis.

REFERENCES

[1] Mulatkar, S., 2014. Sentiment Classification In Hindi. International journal of scientific & technology research, vol. 3, no. 5.

[2] Sharma, R., Nigam, S. and Jain, R., 2014. Polarity Detection of Movie Reviews in Hindi Language. International Journal on Computational Science & Applications, vol. 4, no. 4, pp.49-57.

[3] Pandey, P. and Govilkar, S., 2015. A Framework for Sentiment Analysis in Hindi using HSWN. International Journal of Computer Applications, vol. 119, no. 19, pp. 23-26.

[4] Mishra, D., Venugopalan M. and Gupta D., 2016. Context Specific Lexicon for Hindi Reviews. Procedia Computer Science, vol. 93, pp. 554-563.

[5] Mittal N. and Agarwal, B., 2013. Sentiment Analysis of Hindi Review based on Negation and Discourse Relation. International Joint Conference on Natural Language Processing, pp. 45-50.

[6] Bakliwal, A, Arora, P. and Varma, V., 2012. Hindi Subjective Lexicon: A Lexical Resource for Hindi Polarity Classification. International conference on Language Resources and Evaluation.

[7] Pang, B., Lillian, L. and Shivakumar, V., 2002. Thumbs up?: Sentiment Classification Using Machine Learning Techniques. Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, vol. 10.

[8] Moraes, R., Valiati, J. and Gaviao Neto, W., 2013. Document-level sentiment classification: An empirical comparison between SVM and ANN. Expert Systems with Applications. vol. 40, no. 2, pp. 621-633.

[9] Tripathi, A., Agarwal, A. and Rath, S., 2016. Classification of sentiment reviews using n-gram machine learning approach. Expert Systems with Applications, vol. 57, pp. 117-126.

[10] Liu, Y., Bi, J. and Fan, Z., 2017. Multi-class sentiment classification: The experimental comparisons of feature selection and machine learning algorithms. Expert Systems With Applications, vol. 80, pp. 323–339.

[11] Agarwal, B. and Mittal, N., 2014. Prominent feature extraction for review analysis: an empirical study. Journal of Experimental & Theoretical Artificial Intelligence, vol. 28, no. 3, pp. 485-498.

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

[13] Moreo, A., Romero, M., Castro, J. and Zurita, J., 2012. Lexicon-based Comments-oriented News Sentiment Analyzer system. Expert Systems with Applications. vol. 39, no. 10, pp. 9166-9180.

[14] Kang, H., Yoo, S. and Han, D., 2012. Senti-lexicon and improved Naïve Bayes algorithms for sentiment analysis of restaurant reviews. Expert Systems with Applications. vol. 39, no. 5, pp. 6000-6010.

[15] Boiy, E., and Moens, M. , 2009. A machine learning approach to sentiment analysis in multilingual Web texts", Information Retrieval, vol. 12, no. 5, pp. 526-558.

[16] Rui, H., Liu, Y. and Whinston, 2013. A., Whose and what chatter matters? The effect of tweets on movie sales, Decision Support Systems, vol. 55, no. 4, pp. 863-870.

[17] Maks, I. and Vossen, P., 2012. A lexicon model for deep sentiment analysis and opinion mining applications, Decision Support Systems, vol. 53, no. 4, pp. 680-688.

[18] Sharma, S. and Rakesh Chandra, B., 2015. Sentiment analysis of code-mix script. International Conference on Computing and Network Communications, IEEE.

[19] Ravi, K. and Ravi, V., 2016. Sentiment classification of Hinglish text, International Conference on Recent Advances in Information Technology (RAIT), IEEE.

[20] Bhargava, R., Sharma, Y. and Sharma, S., 2016. Sentiment Analysis for Mixed Script Indic Sentences. International Conference on Advances in Computing, Communications and Informatics (ICACCI).

[21] Kanikar, P., Koppisetty, R., Govindan, S., Bhatand, S. and Virani, M., 2016. Semantic Analysis on Twitter Data Generated by Indian Users. International Journal of Scientific & Engineering Research, vol. 7, no. 9.