Sentiment Analysis of Hindi-English Code-mixed Social Media Text

***Abstract*—This paper discusses the different classifiers that can be used for sentiment analysis of twitter data, to classify the tweets as positive or negative. The challenge of bilingual comments (English and Hindi in the Latin alphabet) is focused on here. An existing labelled Kaggle dataset is used for this study. Forty thousand rows of this dataset are randomly selected and then cleaned. Adjectives, adverbs and abstract nouns are selected as features and extracted for each cleaned tweet. Then seven different classifiers, namely, Naïve Bayes', Multinomial Naïve Bayes', Bernouille's Naïve Bayes', Logistic Regression, Stochastic Gradient Descent, Support Vector Machines and Maximum Entropy classifiers are trained on 85% of the dataset. A hybrid model is created using these seven classifiers by implementing the voting based ensemble model. Then a function is created which used TextBlob to identify the language of the word and in case, the language is 'hi', i.e. Hindi, then the Google Machine Translator is used to convert that word to its English form. The function also tries to handle the challenge of language ambiguity which occurs when a word exists in both the English and Hindi dictionaries. The translated final string is passed to another function that uses the string as a test case for the seven base classifiers and the hybrid model and the sentiment predicted by each of these classifiers is printed along with the extracted features and the translated final tweet/string given by the user.**

Keywords— twitter, dataset, Naïve Bayes', Logistic Regression, Stochastic Gradient Descent, Support Vector Machine, Maximum Entropy, hybrid, accuracy, textblob

I. Introduction

Sentiment Analysis is the interpretation and classification of emotions (positive, negative and neutral) within text data using text analysis techniques [19]. It combines natural language processing (NLP) and machine learning techniques to assign weighted sentiment scores [18].

It allows us to gain an overview of the wider public opinion behind certain topics. Some of the most popular types of sentiment analysis are Fine-grained Sentiment Analysis, Emotion detection, Aspect-based Sentiment Analysis and Multilingual Sentiment Analysis [19].

This paper focuses on bilingual sentiment analysis, which is a subset of multilingual sentiment analysis. This involves a lot of pre-processing and resources. On social media websites like Twitter, users comment in various languages and in different alphabets. This makes it a challenge to analyse the sentiment of the text and is referred to as code-mixing.

Code in sociolinguistics refers to a language or a language variety [21]. Code-mixing can be defined as simply mixing of two or more varieties of the same language or of different languages altogether. Out of the codes that are mixed, the one whose structure or syntax is followed is generally called the matric language [20]. Language contact and bilingualism are the prime causes of code mixing. Physical interaction of the speakers, social media interaction and academic or non-academic reading in a non-native language facilitate language contact [21]. Example of Code-Mixed Hindi-English text is – "tu apne saath college bag leja raha hai?" or "arey waah! I am very proud of you".

Bearing these thoughts in mind, this paper provides a method in which sentiment analysis can be done for multiple languages. In particular, the paper focuses on two languages – English and Hindi, written in the Latin alphabet.

Data is taken from an existing Kaggle dataset for this study [17]. Forty thousand rows are randomly selected from the dataset rows with label as '0' and '4' equally, which represent 'negative' and 'positive' respectively. This dataset is then pre-processed, i.e. cleaned and features are extracted.

Seven different base models – Naïve Bayes', Multinomial Naïve Bayes', Bernouille's Naïve Bayes', Logistic Regression, Stochastic Gradient Descent, Support Vector Machines, Maximum Entropy - are trained on 85% of the final dataset, i.e. 34,000 rows and tested on the remaining 6,000 rows to evaluate each of their performances The performance of these models is elevated using a hybrid model on the seven classifiers. The hybrid model created is a voting based ensemble model. The performance of these classifiers is measured by calculating their accuracies, precision, recall and F1-scores.

A function is created using the functionality of TextBlob and Google Translate to handle the translation of the words in a tweet if required, which calls another function on the translated final text for testing it on the eight classifiers. These classifiers predict the sentiment of the text. The output displays the translated text passed into the function, the features extracted from it and then the sentiment predicted by each of the classifiers.

There are no constraints as such that hinder the process of the translation. Some assumptions worth noting are that during the course of the study, out of all the parts of speech that are present, it has been assumed that adjectives, adverbs, and abstract nouns are the main parts of a sentence that influence the sentiment that is being conveyed in that particular sentence. This is the primary reason words that are obtained from the lemmatization of sentences are compared with adverbs, adjectives, and abstract nouns only.

The next part discusses the 'Background and Related Work,' followed by 'Data' that describes the 'Dataset' and 'Data Preprocessing'. These are followed by the 'Proposed

System', 'Implementation' and the 'Hybrid Model'. The 'Classifier Evaluation' and 'Results and Discussion' are then presented, which proceeds onto the 'Bilingual Sentiment Analysis' which handles the code-mixing challenge. Finally, the 'Conclusion and Future Work' ends the paper with 'Acknowledgment' and 'References'.

II. Background and Related Work

Sentiment analysis (SA) using code-mixed data from social media has several applications in opinion mining ranging from customer satisfaction to social campaign analysis in multilingual societies. Advances in this area are impeded by the lack of a suitable annotated dataset. The authors of [1] introduce a Hindi-English (Hi-En) code-mixed dataset for sentiment analysis and perform empirical analysis comparing the suitability and performance of various state-of-the-art SA methods in social media. They also introduce learning sub-word level representations in LSTM (Subword-LSTM) architecture instead of character-level or word-level representations. This linguistic prior in their architecture enables us to learn the information about sentiment value of important morphemes. This also seems to work well in highly noisy text containing misspellings as shown in our experiments which is demonstrated in morpheme-level feature maps learned by our model. Also, they hypothesize that encoding this linguistic prior in the Subword-LSTM architecture leads to the superior performance. Their system attains accuracy 4-5% greater than traditional approaches on our dataset, and also outperforms the available system for sentiment analysis in Hi-En code-mixed text by 18%.

Aditya Joshi et al [1] mainly focuses on introducing a constantly learning sub-word level representation in LSTM (Subword-LSTM) architecture instead of character-level or word-level representations. This linguistic prior in their architecture enables us to learn the information about sentiment value of important morphemes. Some points to stress upon are that it works well in noisy text containing misspelling of words as well.

In the present communication-based society, no natural language seems to have been left untouched by the trends of code-mixing. For different communicative purposes, a language uses linguistic codes from other languages. This gives rise to a mixed language which is neither totally the host language nor the foreign language. The mixed language poses a new challenge to the problem of machine translation. It is necessary to identify the "foreign" elements in the source language and process them accordingly. The foreign elements may not appear in their original form and may get morphologically transformed as per the host language. Further, in a complex sentence, a clause/utterance may be in the host language while another clause/utterance may be in the foreign language. Code-mixing of Hindi and English where Hindi is the host language, is a common phenomenon in day-to-day language usage in Indian metropolis. The scenario is so common that people have started considering this a different variety altogether and calling it by the name Hinglish. In R. Mahesh et al [2], a mechanism for machine translation of Hinglish to pure (standard) Hindi and pure English forms is presented.

A different angle to look at speech is to consider the negative aspect of words that are used as well. Hate speech detection in social media texts is an important Natural language Processing task, which has several crucial applications like sentiment analysis, investigating cyber bullying and examining socio-political controversies. While relevant research has been done independently on code-mixed social media texts and hate speech detection, in [3], Aditya Bohra et al work is the first attempt in detecting hate speech in Hindi English code-mixed social media text. In this paper, they analyze the problem of hate speech detection in code-mixed texts and present a Hindi-English code-mixed dataset consisting of tweets posted online on Twitter. The tweets are annotated with the language at word level and the class they belong to (Hate Speech or Normal Speech). A supervised classification system is also proposed for detecting hate speech in the text using various character level, word level, and lexicon-based features.

Code-mixing is a linguistic phenomenon frequently observed in user generated content on social media, especially by multilingual users. Apart from the inherent linguistic complexity, the analysis of code-mixed content poses complex challenges owing to the presence of spelling variations, transliteration and non-adherence to a formal grammar. However, for any downstream Natural Language Processing task, tools that are able to process and analyze code-mixed data are required. Currently there is a lack of publicly available resources for code-mixed Hindi-English data, while the amount of such text is increasing everyday. In the study in [4], Sakshi Gupta et al focus is on creation of a dataset that has codemixed Hindi-English sentences along with the associated language and normalisation labels. To the best of their knowledge, their work is the first attempt at the creation of a linguistic resource for this language pair, which is also made public. In this work, an empirical study detailing the construction of a language identification and normalisation system designed for this language pair is also presented.

To gain some reference from another scripting language, the Roman script is a good touchstone to use in this particular scenario. In order to determine the sentiment polarity of Hinglish text written in Roman script, the authors of [5] experimented with different combinations of feature selection methods and a host of classifiers using term frequency-inverse document frequency feature representation. They carried out in total 840 experiments in order to determine the best classifiers for sentiment expressed in the news and Facebook comments written in Hinglish. Some conclusions are a triumvirate of term frequency-inverse document frequency-based feature representation, gain ratio-based feature selection, and Radial Basis Function Neural Network as the best combination to classify sentiment expressed in the Hinglish text.

All these papers discussed here look at problems of Machine Translation of Code-Mixed Data [2] or Hate-Speech Detection [3] or Resource Creation [4] for such Data. They do not handle the specific problem of sentiment analysis of Hindi-English Code-Mixed Social Media Text. The few papers that handle this take a different and more

complex approach that is not completely necessary. For example, [5] converts the Hindi text written in the Latin Alphabet in the test case to Hindi written in the Devanagari Alphabet and then it performs sentiment analysis for Hindi Language separately and English language separately. This just adds an extra step and makes it rather complicated.

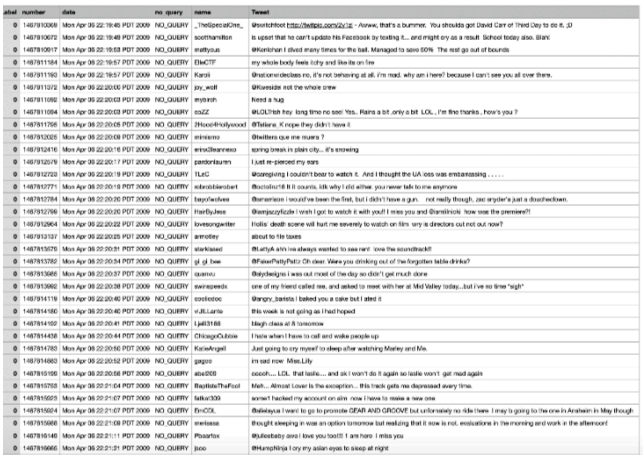
This knowledge gathered from all this literature review sparked the problem statement for Bilingual Sentiment analysis, which was taken as the title for this study.

III. Data

A. *DATASET*

The dataset that was used was obtained from “Kaggle” called the Sentiment140 dataset. It contains 1,600,000 tweets extracted using the twitter API. The tweets have been annotated (0 = negative, 2 = neutral, 4 = positive) and they can be used to detect sentiment. But only 40000 rows where randomly selected from this dataset, with equal distribution of positive and negative tweets, i.e. neutral tweets were ignored as this study focuses on binary classification.

Given below is a snippet of the dataset that was used for this study:



abel	number	date	no_query	name	Tweet
0	1467810369	Mon Apr 06 22:19:45 PDT 2009	NO_QUERY	_TheSpecialOne_	@switchfoot http://twitpic.com/2y1zl - Awww, that's a bummer. You shoulda got David Carr of Third Day to do it. ;D
0	1467810672	Mon Apr 06 22:19:49 PDT 2009	NO_QUERY	scotthamilton	is upset that he can't update his Facebook by texting it... and might cry as a result School today also. Blah!
0	1467810917	Mon Apr 06 22:19:53 PDT 2009	NO_QUERY	mattycus	@Kenichan I dived many times for the ball. Managed to save 50% The rest go out of bounds
0	1467811184	Mon Apr 06 22:19:57 PDT 2009	NO_QUERY	ElleCTF	my whole body feels itchy and like its on fire
0	1467811193	Mon Apr 06 22:19:57 PDT 2009	NO_QUERY	Karoli	@nationwideclass no, it's not behaving at all. i'm mad. why am i here? because I can't see you all over there.
0	1467811372	Mon Apr 06 22:20:00 PDT 2009	NO_QUERY	joy_wolf	@Kweside not the whole crew
0	1467811592	Mon Apr 06 22:20:03 PDT 2009	NO_QUERY	mybirch	Need a hug
0	1467811594	Mon Apr 06 22:20:03 PDT 2009	NO_QUERY	coZZ	@LOLTrish hey long time no see! Yes.. Rains a bit ,only a bit LOL , I'm fine thanks , how's you ?
0	1467811795	Mon Apr 06 22:20:05 PDT 2009	NO_QUERY	2Hood4Hollywood	@Tatiana_K nope they didn't have it
0	1467812025	Mon Apr 06 22:20:09 PDT 2009	NO_QUERY	elmiomo	@twitters que me muera ?
0	1467812416	Mon Apr 06 22:20:16 PDT 2009	NO_QUERY	erinx3leannexo	spring break in plain city... it's snowing
0	1467812579	Mon Apr 06 22:20:17 PDT 2009	NO_QUERY	pardonlaurrin	I just re-pierced my ears
0	1467812723	Mon Apr 06 22:20:19 PDT 2009	NO_QUERY	TLeC	@caregiving I couldn't bear to watch it. And I thought the UA loss was embarassing . . . . .
0	1467812771	Mon Apr 06 22:20:19 PDT 2009	NO_QUERY	robrobbinsbart	@bostonchi8 It ll counts, idk why I did either. you never talk to me anymore
0	1467812784	Mon Apr 06 22:20:20 PDT 2009	NO_QUERY	bayofwolves	@sarahlaux I would've been the first, but I didn't have a gun. not really though, zac snyder's just a douchebown.
0	1467812799	Mon Apr 06 22:20:20 PDT 2009	NO_QUERY	HairByJess	@iamjazzyfizzle I wish I got to watch it with you!! I miss you and @iamlilnicki how was the premiere?!
0	1467812964	Mon Apr 06 22:20:22 PDT 2009	NO_QUERY	lovesrepetter	Hollis' death scene will hurt me severely to watch on film wry is directors cut not out now?
0	1467813137	Mon Apr 06 22:20:25 PDT 2009	NO_QUERY	amotley	about to file taxes
0	1467813579	Mon Apr 06 22:20:31 PDT 2009	NO_QUERY	sterkland	@LettyA ahh ive always wanted to see rent love the soundtrack!!
0	1467813782	Mon Apr 06 22:20:34 PDT 2009	NO_QUERY	gi_gi_bee	@FakePattyPatz Oh dear. Were you drinking out of the forgotten table drinks?
0	1467813866	Mon Apr 06 22:20:37 PDT 2009	NO_QUERY	quarecu	@skydesigns I was out most of the day so didn't get much done
0	1467813992	Mon Apr 06 22:20:38 PDT 2009	NO_QUERY	swinspeedx	one of my friend called me, and asked to meet with her at Mid Valley today...but i've no time *sigh*
0	1467814119	Mon Apr 06 22:20:40 PDT 2009	NO_QUERY	cooliodoc	@angry_barista I baked you a cake but I ated it
0	1467814180	Mon Apr 06 22:20:40 PDT 2009	NO_QUERY	viJILLante	this week is not going as I had hoped
0	1467814337	Mon Apr 06 22:20:41 PDT 2009	NO_QUERY	ljelli1166	blegh class at 8 tomorrow
0	1467814458	Mon Apr 06 22:20:44 PDT 2009	NO_QUERY	ChicagoDubbie	I hate when I have to call and wake people up
0	1467814783	Mon Apr 06 22:20:50 PDT 2009	NO_QUERY	KaieArgell	Just going to cry myself to sleep after watching Marley and Me.
0	1467814883	Mon Apr 06 22:20:52 PDT 2009	NO_QUERY	gagoo	im sad now Miss Lily
0	1467815106	Mon Apr 06 22:20:56 PDT 2009	NO_QUERY	abe000	oooh.... LOL that looks... and ok I won't do it again so lastie won't get mad again
0	1467815768	Mon Apr 06 22:21:04 PDT 2009	NO_QUERY	BaptisteTheFool	Meh... Almost Lover is the exception... this track gets me depressed every time.
0	1467815923	Mon Apr 06 22:21:07 PDT 2009	NO_QUERY	falluc309	somet hacked my account on aim now I have to make a new one
0	1467815924	Mon Apr 06 22:21:07 PDT 2009	NO_QUERY	ElmCOL	@alielayus I want to go to promote GEAR AND GROOVE but unfortunately no ride there. I may b going to the one in Anaheim in May though
0	1467815988	Mon Apr 06 22:21:09 PDT 2009	NO_QUERY	enelaaa	thought sleeping in was an option tomorrow but realizing that it now is not. evaluations in the morning and work in the afternoon!
0	1467816140	Mon Apr 06 22:21:11 PDT 2009	NO_QUERY	Phaarfox	@juliebaby aaw i love you too!!! I am here. I miss you
0	1467816666	Mon Apr 06 22:21:21 PDT 2009	NO_QUERY	jooo	@HumphriesJ I cry my asian eyes to sleep at night

Fig. 1. Snapshot of the Dataset

It contains six fields or columns or variables, which are described as follows:

- Label: The polarity of the tweet, that is the sentiment of the text. It has 3 major classifications that signify a different polarity. “0” indicates a negative polarity, “2” indicates a neutral polarity and “4” indicates a positive polarity.
- number: The id of the tweet. An example is “1467810389”.
- date: The date of the tweet, i.e. when the user tweeted. For example, “Mon April 06 22:19:45 PDT 2009”.
- no_query: The query. If there is no query, then this value is NO_QUERY. Only the rows with flag as NO_QUERY are useful for this study.
- name: The user that tweeted, more specifically the user handle used by the Twitter user who tweeted. For example, “HairByJess”. This is the user handle.
- Tweet: The text of the tweet. An example can be “I am very happy. You are an amazing and excellent person”. This is the most important field for this study along with the ‘Label’ field, as this is the text that will be passed into the classifiers.

B. *DATA PREPROCESSING*

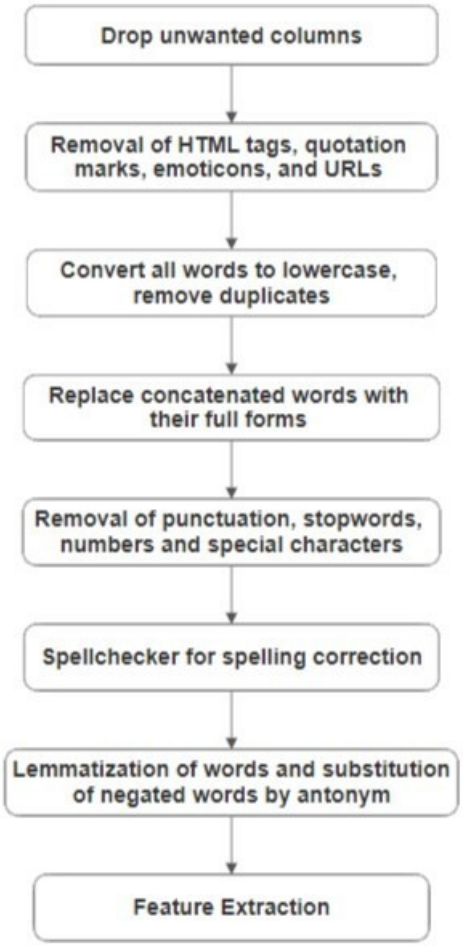


Fig. 2. Flowchart depicting Preprocessing Algorithm

In order to make sure that the work was carried out in the most efficient manner, the dataset used needed to be pre-processed before actual usage. First, only a handful of columns were used from the set of 6 columns. These columns used were “Tweet” and “Label”. The remaining columns are discarded as they were not needed.

The snapshot of the dataset after removal of unwanted columns is as follows:

```
data
```

	Tweet	label
0	@switchfoot http://twitpic.com/2y1zl - Awww, t...	negative
1	is upset that he can't update his Facebook by ...	negative
2	@Kenichan I dived many times for the ball. Man...	negative
3	my whole body feels itchy and like its on fire	negative
4	@nationwideclass no, it's not behaving at all....	negative
...	...	...
39996	My GrandMa is making Dinenr with my Mum	positive
39997	Mid-morning snack time... A bowl of cheese noo...	positive
39998	@ShaDeLa same here say it like from the Termi...	positive
39999	@DestinyHope92 im great thaanks wbuu?	positive
40000	cant wait til her date this weekend	positive

Fig. 3. Dataset after dropping unwanted columns

Next, any HTML tags present in the text were removed with the help of BeautifulSoup package. Quotation marks (‘, “), punctuations, numbers and special characters were removed as they are unnecessary for the sentiment analysis of the text. If any emoticons were used in the tweet, they were converted into their equivalent emotion that they are intended to signify, and the emojis were removed. Then, all the words in the text are converted to lower case and repeating words in the same tweet are removed. Concatenated words, i.e. words which were joined such as “Can’t” were expanded to “Can not”. The data cleaning process also included the removal of URLs and user handles or tags (“@” symbol followed by the account handle). Stopwords, other than ‘not’ and the word ‘is’ were also removed from the text. The cleaned tweet contained corrected spellings of all words, because of the use of the ‘SpellChecker’ package. All words with length equal to one were also removed. Next, the words were lemmatized, and words preceded by ‘not’ were substituted by their antonyms to represent the negation effect. These were all some data cleaning steps that were important to the study to function effectively. Finally, the dataset contained the cleaned tweets.

The following snippet is a showcase of the dataset used after cleaning the tweet in the above-mentioned methods:

Tweet	label	cleaned_tweets
@switchfoot http://twitpic.com/2y1zl - Awww, t...	negative	aww bummer should get david carr third day ...
is upset that he can't update his Facebook by ...	negative	upset update casebook test might cry result ...
@Kenichan I dived many times for the ball. Man...	negative	dive many time ball manage save rest go bound
my whole body feels itchy and like its on fire	negative	le body feels itchy like fire
@nationwideclass no, it's not behaving at all....	negative	behaving mad see
...	...	...
My GrandMa is making Dinenr with my Mum	positive	grandma make dinner mum
Mid-morning snack time... A bowl of cheese noo...	positive	mid morning snack time bowl cheese noodle yum
@ShaDeLa same here say it like from the Termi...	positive	say like terminator movie come like
@DestinyHope92 im great thaanks wbuu?	positive	great thanks but
cant wait til her date this weekend	positive	cant wait til date weekend

Fig. 4. Dataset after cleaning of tweets

Certain features, like adjectives, abstract nouns, and adverbs were focused on and the rest of the words were removed as they did not add any value to the sentiment. This was done as part for the feature identification and extraction. This was implemented by checking each word in the cleaned tweet with the words in a file, which was prefilled with most common adjectives, adverbs and abstract nouns. If the word was not present in this file, it was not chosen as a feature and if it matched a word in this file, it was selected as one of the features. All these features of each text was stored in another column. This method was used because the existing method is not completely accurate and is outdated. For example, the word ‘clever’ was being identified as a ‘noun’ by the pos_tag function provided by the nltk library.

These features were then formatted in such a way that is accepted by the pre-existing classifier functions as input for training and testing purposes.

Given below is snippet of the dataset with the classification and segregation of adverbs, adjectives and abstract nouns:

```
In [53]: #spliting into train sets for training
         training_set=[]
         count=0
         for i in (X_train["cleaned_tweets"]):
             training_set.append((i.split(),Y_train[count]))
             count+=1

         def list_to_dict(words_list):
             return dict([(word, True) for word in words_list])

         training_set_formatted = [(list_to_dict(element[0]), element[1]) for element in training_set]
         training_set_formatted

Out[53]: [({'ill': True, 'playful': True}, 'positive'),
          ({}, 'negative'),
          ({'happy': True, 'hard': True}, 'positive'),
          ({'officially': True, 'happy': True, 'hard': True}, 'negative'),
          ({'live': True}, 'negative'),
          ({'important': True, 'busy': True}, 'negative'),
          ({'yesterday': True}, 'negative'),
          ({'green': True}, 'positive'),
          ({'well': True}, 'positive'),
          ({}, 'positive'),
          ({}, 'positive'),
          ({'blank': True}, 'negative'),
          ({'exactly': True}, 'positive'),
          ({'good': True}, 'positive'),
          ({'ill': True}, 'negative'),
          ({}, 'negative'),
          ({'live': True}, 'positive'),
          ({'jealous': True}, 'negative'),
          ({}, 'negative'),
          ({'good': True}, 'positive'),
```

Fig. 5. Feature extraction of the cleaned tweets

The final dataset used for the study is given below. The data is successfully pre-processed, cleaned, and features have been extracted and it is ready for use in this study:

```
In [20]: data
```

Out[20]:

	Tweet	label	cleaned_tweets
0	@switchfoot http://twitpic.com/2y1zl - Awww, t...	negative	third playful
1	is upset that he can't update his Facebook by ...	negative	upset
2	@Kenichan I dived many times for the ball. Man...	negative	
3	my whole body feels itchy and like its on fire	negative	itchy
4	@nationwideclass no, it's not behaving at all....	negative	mad
...	...	...	...
39996	My GrandMa is making Dinenr with my Mum	positive	
39997	Mid-morning snack time... A bowl of cheese noo...	positive	
39998	@ShaDeLa same here say it like from the Termi...	positive	
39999	@DestinyHope92 im great thaanks wbuu?	positive	great
40000	cant wait til her date this weekend	positive	

40001 rows × 3 columns

Fig. 6. Final Dataset after cleaning and feature extraction

It can be seen that this dataset has three columns, namely ‘Tweet’, ‘Label’ and ‘cleaned_tweets’. The first two are the ones taken from the original dataset. The last column is a derived column from all the preprocessing and contains just the features extracted from each of the tweet texts, as string.

IV. Proposed System

To explain the basic system that is used during the course of this study, we can take assistance from the flowchart given below:

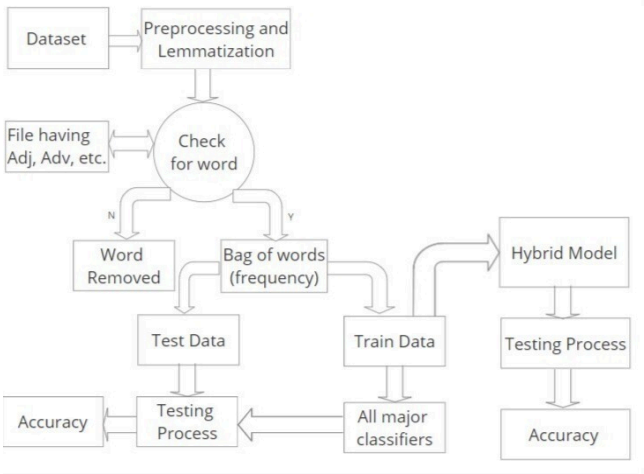


Fig. 7. Block Diagram depicting the flow

Initially, the dataset is cleaned as mentioned under 'Data Preprocessing' section under the heading 'Data'. Unwanted columns, extra rows and stop words along with useless symbols are discarded. Then the sentences undergo the process of lemmatization, where each sentence is broken down into words. These words are then referenced and checked against a file containing commonly used adverbs, adjectives and abstract nouns. If these words are matched against words in the file, the word is put into a bag of words for the collection. If the words are not matched, they are removed and discarded. These words along with their sentiment are split into test and train data with a ratio of 85%. This approximates to 34,000 tweets of training data and 6,000 tweets of testing data. The training data is then passed into all the major classifiers that were used, namely Naive Bayes' Classifier, Multinomial Naive Bayes' Classifier, Bernoulli's Naive Bayes' Classifier, Logistic Regression, Stochastic Gradient Descent Classifier, Support Vector Machines Classifier, and the Max Entropy Classifier. Following this step, the testing data is passed into the respective trained models and determine the resulting accuracies of each classifier. From the above step, the classifier that has the highest accuracy will act as a benchmark accuracy for the study. In order to reach a higher level of accuracy, a hybrid model was constructed. This hybrid model is an ensemble of all the classifiers used previously where the mode of the predicted labels is used as the final classification. Following the suit, the testing data is passed on to the hybrid model and the accuracy is determined. The aim is to obtain an optimal accuracy result that is higher than the previously set benchmark.

After this, a translation mechanism is used to handle the challenge of the Hinglish text using the Google Translator Machine and a function is created that takes in text as input and translates it if required and tests it with each of the seven trained base models and the hybrid model and the sentiment predicted by each of these is printed as the output along with the features.

V. Implementation

A classifier is any algorithm that sorts data into labelled classes, or categories of information [15]. In other words, classification is the process of predicting the class (sometimes are called as targets or labels) of given data points [16]. For example, spam detection in email service can be identified as a binary classification problem, where the two classes are spam and not spam [16]. Similarly, we are classifying tweets here as positive or negative.

The following seven classifiers were made to learn from the data given to it, also called the training data and make new classifications on the test data -

A. Naïve Bayes' Classifier

Naïve Bayes' classifiers are a collection of classification algorithms based on Bayes' Theorem. It is a family of algorithms where every pair of features being classified are independent of each other [14].

Assumptions of Naïve Bayes' classifiers are that each feature must be independent and equal. The formula used which is derived from Bayes' Theorem is as follows [14]:

$$y = argmax_y P(y) \prod_{i=1}^{n} P(x_i|y) \quad (1)$$

Some advantages of this classifier are [22]:

- Easy to implement.
- Fast.
- Does not require much training data.

Use Cases [22]:

- Document Classification
- Spam Filters
- Sentiment Analysis
- Disease Predictions

B. Multinomial Naïve Bayes' Classifier

Multinomial Naïve Bayes' Classifier estimates the conditional probability of a particular word, given a class as the relative frequency of term t in documents belonging to class (c) [23]. The variation takes into account the number of occurrences of term t in training documents from class (c), including multiple occurrences [23].

It is a specialised version of Naïve Bayes' that is designed more for text documents and explicitly models the word counts and adjusts the underlying calculations to deal with it [23].

The formula used is [23]:

$$p(C_k) = \frac{(\sum_i x_i)!}{\prod_i x_i!} \prod_i pki^{x_i} \quad (2)$$

C. *Bernoulli's Naïve Bayes' Classifier*

The Bernoulli Naïve Bayes' classifier assumes that all our features are binary such that they take only two values (example – a nominal categorical feature that has been one-hot encoded) [23].

The formula derived is [23]:

$$p(C_k) = \prod_{i=1}^{n} p_{k_i}^{x_i}\left(1 - p_{k_i}\right)^{(1-x_i)} \quad (3)$$

In the above formula, p_{k_i} is the probability of a class C_k generating term x_i.

This event model is especially popular for classifying short texts [23]. It has the benefit of explicitly modelling the absence of terms [14].

D. *Logistic Regression*

Logistic Regression is a statistical method for analysing a dataset in which there are one or more independent variables that determine an outcome [14]. The outcome is measured with a dichotomous variable (in which there are only two outcomes) [22].

Fig. 8. Logistic Regression [18]

Some advantages are [22]:

- Performs well when data is linearly separable.
- Less prone to overfitting.
- Easy to implement and interpret.

Use Cases [22]:

- Identifying risk factors for diseases
- Word Classification
- Weather Prediction
- Voting Applications

E. *Stochastic Gradient Descent Classifier*

Stochastic Gradient Descent (SGD) is a simple yet very efficient approach to discriminative learning of linear classifiers under convex loss functions such as (linear) Support Vector Machines and Logistic Regression [14].

Stochastic gradient descent refers to calculating the derivative from each training data instance and calculating the update immediately [22]. It is particularly useful when the sample data is in a larger number. It supports different loss functions and penalties for classification [22].

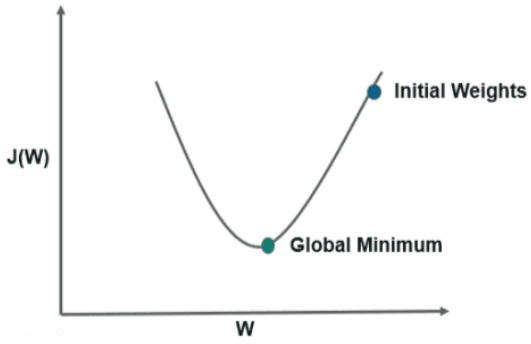


Fig. 9. Stochastic Gradient Descent [22]

Only recently received a lot of attention even though it has been around for a long time, some of its advantages are as follows [22]:

- Efficiency.
- Ease of implementation.

F. *Support Vector Machines*

Generally, Support Vector Machines (SVM) is considered to be a classification approach, but it can be employed in both classification and regression problems [14]. It can easily handle multiple continuous and categorical variables [14].

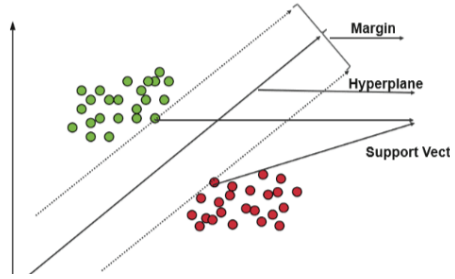


Fig. 10. Support Vectors [22]

Some advantages of SVMs are as follows [22]:

- Effective in high dimensional spaces.
- Effective where no. of dimensions > no. of samples

G. *Maximum Entropy Classifier*

The Maximum Entropy (MaxEnt) classifier is a probabilistic classifier which belongs to the class of

exponential models [14]. It does not assume that the features are conditionally independent of each other.

The MaxEnt is based on the Principle of Maximum Entropy and from all the models that fit our training data, selects the one which has the largest entropy [14].

Some advantages are [22]:

- They do not overfit.
- Provide a better accuracy than other classification algorithms.

VI. Hybrid Model

Hybrid machine learning models combine strengths of various models or classifiers. Hybrid models bring the best from various machine learning methods or classifiers.

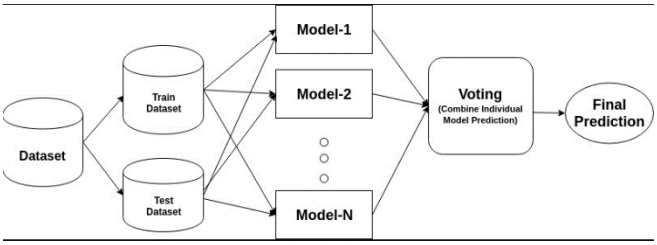


Fig. 11. Representation of a Hybrid Model [13]

In this study, a hybrid model is used by combining all the seven classifiers (Naïve Bayes', Multinomial Naïve Bayes', Bernoulli's Naïve Bayes', Logistic Regression, Stochastic Gradient Descent, Support Vector Machines and Maximum Entropy). Through this hybrid machine, it is intended to improve the classification accuracy of the classifiers. A voting-based ensemble model was devised as the hybrid model here.

A. *What is an ensemble method?*

Ensemble models in machine learning combine the decisions from multiple models to improve the overall performance [24]. They are a divide and conquer approach [24]. In other words, ensemble modelling is a process where multiple diverse base models are used to predict an outcome [12].

The approach seeks the wisdom of the crowds in making a prediction [12]. Even though the ensemble model has multiple based models (seven models here), it acts and performs as a single model.

B. *Why ensemble models?*

The motivation for using ensemble models is to reduce the generalization error of the prediction. As long as the base models are diverse and independent, the prediction error decreases when the ensemble approach is used [24].

The main causes of error in learning models are due to noise, bias and variance [12]. Ensemble models help to minimize these factors. These methods are designed to improve the stability and the accuracy of machine learning algorithms and classifiers.

Their advantages are [24]:

- More accurate prediction results.
- Stable and more robust model.
- Ensemble models can be used to capture the linear as well as the non-linear relationships in the data.

VII. Classifier Evaluation

The most important part after the completion of any classifier is the evaluation to check its accuracy and efficiency. There are a lot of ways in which a classifier can be evaluated. Here,

- Class 1: Positive
- Class 2: Negative

Definition of some terms [11]:

- Positive (P): Observation is positive.
- Negative (N): Observation is not positive (i.e. negative).
- True Positive (TP): Observation is positive and is predicted to be positive.
- False Positive (FP): Observation is negative but is predicted positive.
- True Negative (TN): Observation is negative and is predicted to be negative.
- False Negative (FN): Observation is positive but is predicted negative.

Now, the evaluation methods used here are listed below:

A. *Accuracy*

Accuracy is a ratio of correctly predicted observations to the total observations [22].

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$$

[11] (4)

However, there are problems with accuracy. It assumes equal costs for both kinds of errors. A 99% accuracy can be excellent, good, mediocre, poor or terrible depending upon the problem [11].

B. *F1-Score*

It is the weighted average of precision and recall [22]. In other words, the F1-Score conveys the balance between the precision and the recall. It helps to have a measurement that represents both precision and recall [11]. It is also called the F-Score or the F-Measure.

$$F1\text{-}Score = \frac{2*Recall*Precision}{Recall+Precision}$$

[11] (5)

It is calculated as the harmonic mean in place of arithmetic mean as it punishes the extreme values more. The

F1-Score will always be nearer to the smaller value of Precision or Recall [11].

C. *Precision*

Precision is the fraction of relevant instances among the retrieved instances. It is basically used as the measure of relevance [22].

$$Precision = \frac{TP}{TP+FP}$$

[11] (6)

To get the value of precision we divide the total number of correctly classified positive examples by the total number of predicted positive examples. High Precision indicates an example labelled as positive is indeed positive (a small number of FP) [11].

D. *Recall*

Recall is defined as the ratio of the total number of correctly classified positive examples divide to the total number of positive examples. In other words, it is the fraction of relevant instances that have been retrieved over the total number of instances [22].

$$Recall = \frac{TP}{TP+FN}$$

[11] (7)

High Recall indicates the class is correctly recognized (a small number of FN) [11]. It is used as the measure of relevance [22].

E. *Confusion Matrix [11]*

- A confusion matrix is a summary of prediction results on a classification problem.
- The number of correct and incorrect predictions are summarized with count values and broken down by each class.
- In other words, it shows the ways in which a classification model is confused when it makes predictions.
- It gives us insight not only into the errors being made by a classifier but more importantly the types of errors that are being made.

	Class 1 Predicted	Class 2 Predicted
Class 1 Actual	TP	FN
Class 2 Actual	FP	TN

Fig. 12. Confusion Matrix Depiction [11]

VIII. Results and Discussions

The Accuracy, F1-Score, Recall and Precision of the seven base classifier models are as follows:

A. *Naïve Bayes' Classifier:*

	precision	recall	f1-score
positive	0.59	0.78	0.67
negative	0.68	0.46	0.55
accuracy			0.62
macro avg	0.64	0.62	0.61
weighted avg	0.64	0.62	0.61

Fig. 13. Performance of Naïve Bayes' Classifier

B. *Multinomial Naïve Bayes' Classifier:*

	precision	recall	f1-score
positive	0.59	0.79	0.67
negative	0.69	0.46	0.55
accuracy			0.62
macro avg	0.64	0.62	0.61
weighted avg	0.64	0.62	0.61

Fig. 14. Performance of Multinomial Naïve Baiyes' Classifier

C. *Bernouille's Naïve Bayes' Classifier:*

	precision	recall	f1-score
positive	0.59	0.79	0.67
negative	0.69	0.46	0.55
accuracy			0.62
macro avg	0.64	0.62	0.61
weighted avg	0.64	0.62	0.61

Fig. 15. Performance of Bernouille's Naïve Baiyes' Classifier

D. *Logistic Regression:*

	precision	recall	f1-score
positive	0.59	0.78	0.67
negative	0.68	0.47	0.55
accuracy			0.62
macro avg	0.64	0.62	0.61
weighted avg	0.64	0.62	0.61

Fig. 16. Performance of Logistic Regression Model

E. *Stochastic Gradient Descent Classifier:*

	precision	recall	f1-score
positive	0.69	0.40	0.51
negative	0.58	0.82	0.68
accuracy			0.61
macro avg	0.64	0.61	0.59
weighted avg	0.63	0.61	0.59

Fig. 17. Performance of SGD Classifier

F. Support Vector Machines Classifier:

	precision	recall	f1-score
positive	0.58	0.79	0.67
negative	0.68	0.44	0.53
accuracy			0.61
macro avg	0.63	0.62	0.60
weighted avg	0.63	0.61	0.60

Fig. 18. Performance of SVM Classifier

G. Maximum Entropy Classifier:

	precision	recall	f1-score
positive	0.68	0.40	0.51
negative	0.58	0.82	0.68
accuracy			0.61
macro avg	0.63	0.61	0.59
weighted avg	0.63	0.61	0.59

Fig. 19. Performance of MaxEnt Classifier

H. Hybrid Model:

	precision	recall	f1-score
positive	0.59	0.79	0.67
negative	0.69	0.46	0.55
accuracy			0.62
macro avg	0.64	0.62	0.61
weighted avg	0.64	0.62	0.61

Fig. 20. Performance of the Hybrid Model

The accuracies of all the seven base classifiers and the hybrid classifier together are as follows:

TABLE I. CLASSIFIER ACCURACIES

Classifier	***Accuracy***
Naïve Bayes'	62.0729
Multinomial Naïve Bayes'	62.2062
Bernouille's Naïve Bayes'	62.2062
Logistic Regression	62.2562
Stochastic Gradient Descent	61.2397
Support Vector Machines	61.3897
Maximum Entropy	61.3897
Hybrid Model	62.2563

Comparing the above-mentioned accuracies of all these classifiers, the following line plot can be derived:

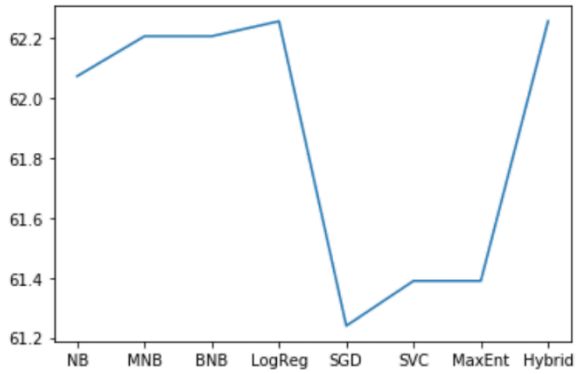


Fig. 21. Comparison of Classifier Accuracies Graph

It can be observed from the Accuracy graph that the hybrid model provides the best accuracy and the Logistic Regression Classifier is the best base model with the most accuracy when compared to the other base models, while Stochastic Gradient Descent Classifier has the least accuracy.

The confusion matrix data before and after normalization, that is TP, TN, FP, FN values are as follows:

```
Confusion matrix, without normalization
[[2306  694]
 [1595 1406]]
Normalized confusion matrix
[[0.77 0.23]
 [0.53 0.47]]
```

Fig. 22. Confusion Matrix Data for Hybrid Model

Confusion Matrix before normalization:

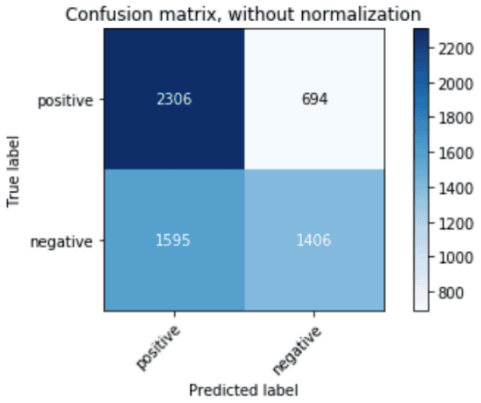


Fig. 23. Confusion Matrix, before normalization

Confusion Matrix after normalization:

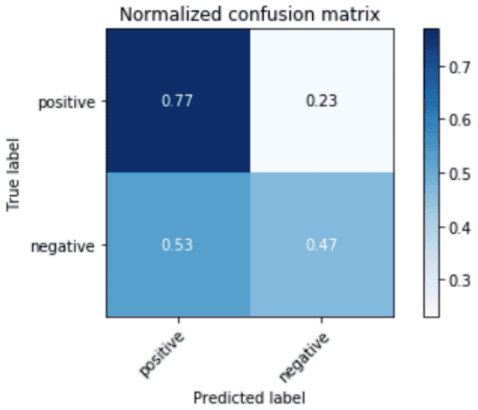


Fig. 24. Confusion Matrix, after normalizaiton

IX. Bilingual Sentiment Analysis

A translation mechanism is used to handle the challenge of the Hinglish text using the Google Translator Machine and a function is created that takes in text as input and translates it if required and tests it with each of the seven trained base models and the hybrid model and the sentiment predicted by each of these is printed as the output along with the features.

The final master function also used the package TextBlob to check whether the language of the word is Hindi or English and accordingly passes the word to the Translator function. The final string is passed onto the text_classify function which returns the desired output.

The screenshots of some of the test tweets along with the sentiment predicted by the eight classifiers are as follows:

- Text: “arrey waah! I’m very proud of you”

```
func("arrey waah! I'm very proud of you")
```

```
['proud']
Tweet given by user :  Oh wow! E'm Very Proud OF U
naive bayes classifier
This Tweet is  positive
------------------------------
Multinomail naive bayes classifier
This Tweet is  positive
------------------------------
Bernouli classifier
This Tweet is  positive
------------------------------
Bernouli LogisticRegression_classifier
This Tweet is  positive
------------------------------
SGD classifier
This Tweet is  positive
------------------------------
SVC classifier
This Tweet is  positive
------------------------------
Max Entropy classifier
This Tweet is  positive
------------------------------
Hybrid model
This Tweet is  positive
```

Fig. 25. Test Case 1

It can be observed that all the classifiers classify the text as positive, which is correct.

- Text: “tum log pagal ho”

```
func("tum log pagal ho")
```

```
['mad']
Tweet given by user :  Are you mad?
naive bayes classifier
This Tweet is  negative
------------------------------
Multinomail naive bayes classifier
This Tweet is  negative
------------------------------
Bernouli classifier
This Tweet is  negative
------------------------------
Bernouli LogisticRegression_classifier
This Tweet is  negative
------------------------------
SGD classifier
This Tweet is  negative
------------------------------
SVC classifier
This Tweet is  negative
------------------------------
Max Entropy classifier
This Tweet is  negative
------------------------------
Hybrid model
This Tweet is  negative
```

Fig. 26. Test Case 2

Here, the predicted sentiment by all the eight classifier is shown as negative. It can be observed that even though the word ‘log’ has meaning in both English and Hindi, here it is

taken as Hindi, which is correct according to the context here.

- Text: “tum pagal ho”

```
In [47]: func("tum pagal ho")

['mad']
Tweet given by user :  you are mad
naive bayes classifier
This Tweet is  negative
------------------------------
Multinomail naive bayes classifier
This Tweet is  negative
------------------------------
Bernouli classifier
This Tweet is  negative
------------------------------
Bernouli LogisticRegression_classifier
This Tweet is  negative
------------------------------
SGD classifier
This Tweet is  negative
------------------------------
SVC classifier
This Tweet is  negative
------------------------------
Max Entropy classifier
This Tweet is  negative
------------------------------
Hybrid model
This Tweet is  negative
```

Fig. 27. Test Case 3

This tweet is classified as a negative tweet by the classifiers because the tweet given by user when translated to English is “you are mad” which has a negative sentiment.

- Text: “tum kharab ho”

```
In [48]: func("tum kharab ho")

['bad']
Tweet given by user :  You're bad
naive bayes classifier
This Tweet is  negative
------------------------------
Multinomail naive bayes classifier
This Tweet is  negative
------------------------------
Bernouli classifier
This Tweet is  negative
------------------------------
Bernouli LogisticRegression_classifier
This Tweet is  negative
------------------------------
SGD classifier
This Tweet is  negative
------------------------------
SVC classifier
This Tweet is  negative
------------------------------
Max Entropy classifier
This Tweet is  negative
------------------------------
Hybrid model
This Tweet is  negative
```

Fig. 28. Test Case 4

It is classified as a negative tweet by all the classifiers as the final translated string is “You’re bad”.

- Text: “you are not accha”

```
In [942]: func("you are not accha")

['evil']
Tweet given by user :   you are not good
naive bayes classifier
This Tweet is  negative
------------------------------
Multinomail naive bayes classifier
This Tweet is  negative
------------------------------
Bernouli classifier
This Tweet is  negative
------------------------------
Bernouli LogisticRegression_classifier
This Tweet is  negative
------------------------------
SGD classifier
This Tweet is  negative
------------------------------
SVC classifier
This Tweet is  negative
------------------------------
Max Entropy classifier
This Tweet is  negative
------------------------------
Hybrid model
This Tweet is  negative
```

Fig. 29. Test Case 5

X. Conclusion And Future Work

Some key features that were brought out during the course of this study helped us to get a broader understanding of the subject and gave us the necessary encouragement to dive deeper. The study conducted is unique in multiple ways which act as a differentiation factor when compared to similar work. The methodology followed, helped to create an aggregated model consisting of all the classifiers used during the process. The ensemble model created worked to our advantage as it provided the highest accuracy of 61.856% compared to the classifiers individually. TextBlob was used to detect the language of the words in a sentence. If the word was in Hindi, Google Translate was used to directly convert it into English. Using this approach of both the platforms, increased the accuracy significantly when compared to using them individually. On a personal note, we were able to get familiar with the usage and implementation of different classifiers. We also got the opportunity to create an ensemble model to give us optimal results. We also understood which classifiers work when used on a certain type of data. This also helped us to familiarize ourselves with the advantages and disadvantages of each particular classifier. If we were to look towards the future to enhance the study, we could find a larger and better dataset to work with. We can try to use better translation techniques and give a try at more complex machine learning models for the classification of text. The complexities of each of these models can also be found and compared to give the best efficiency. Furthermore, this study can be extended to multiple regional languages and can be translated into a multilingual sentiment analysis problem, making it more generic.

Acknowledgment

The satisfaction and euphoria that accompany the successful completion of any task would be but incomplete without mentioning the people who made it possible, whose constant guidance crowned our efforts with success. With a profound sense of gratitude, we acknowledge the guidance and support extended by Dr. Shylaja S. S., HOD, CSE Department, PES University. We would also like to thank our Chancellor, Dr. M. R. Doreswamy, Pro-chancellor, Prof. Jawahar D. and Vice-chancellor, Dr. J. Suryaprasad, PES University for providing us with this opportunity and the necessary facilities to carry out our project work.The guidance we received gave us the environment to enhance our knowledge, skills and to reach the pinnacle with sheer determination, dedication and hard work.

References

[1] Aditya Joshi, Ameya Prabhu Pandurang, Manish Shrivatsava and Vasudeva Varma, “Towards Sub-Word Level Compositons for Sentiment Analysis of Hindi-English Code Mixed Text,” 26th International Conference on Computational Linguistics, December 2017.

[2] R. Mahesh, K. Sinha and Anil Thakur, “Machine Translation of Bilingual Hindi-English (Hinglish) Text,” January 2005.

[3] Aditya Bohra, Deepanshu Vijay, Vinay Singh, Syed S. Akhtar and Manish Shrivatsava, “A Dataset of Hindi-English Code-Mixed Social Media Text for Hate Speech Detection,” 16th Annual Conference of the North American Chapter of the Association for Computational Linguistics, June 2018.

[4] Sakshi Gupta, Piyush Bansal and Radhika Mamidi, “Resource Creation for Hindi-English Code Mixed Social Media Text,” July 2016.

[5] Kumar Ravi and Vadlamani Ravi, “Sentiment classification of Hinglish text,” March 2016.

[6] Sahil Swami, Ankush Khandelwal, Vinay Singh, Syed S. Akhtar and Manish Shrivastava, “A Corpus of English-Hindi Code-Mixed Tweets for Sarcasm Detection,” 19th International Conference on Computational Linguistics and Intelligent Text Processing, March 2018.

[7] Alexandra Balahur and Marco Turchi, “Multilingual Sentiment Analysis using Machine Translation”, July 2012.

[8] Puneet Mathur, Rajiv Shah, Ramit Sawhney and Debanjan Mahata, “Detecting Offensive Tweets in Hindi-English Code-Switched Language,” Proceedings of the Sixth International Workshop on Natural Language Processing for Social Media, July 2018.

[9] Hitesh Nankani, Hritwik Dutta, Harsh Shrivastava, P.V.N.S. Rama Krishna and Debanjan Mahata, “Multilingual Sentiment Analysis,” January 2020.

[10] Nurendra Choudhary, Rajat Singh, Vijjini Anvesh Rao and Manish Shrivastava, “Twitter corpus of Resource-Scarce Languages for Sentiment Analysis and Multilingual Emoji Prediction,” Proceedings of the 27th International Conference on Computational Linguistics, August 2018.

[11] https://www.geeksforgeeks.org/confusion-matrix-machine-learning/

[12] https://www.datacamp.com/community/tutorials/ensemble-learning-python

[13] https://towardsdatascience.com/advanced-ensemble-classifiers-8d7372e74e40

[14] https://analyticsindiamag.com/7-types-classification-algorithms/

[15] https://deepai.org/machine-learning-glossary-and-terms/classifier

[16] https://towardsdatascience.com/machine-learning-classifiers-a5cc4e1b0623

[17] https://www.kaggle.com/kazanova/sentiment140

[18] https://dataaspirant.com/2017/03/02/how-logistic-regression-model-works/

[19] https://monkeylearn.com/sentiment-analysis/

[20] http://languagelinguistics.com/2017/06/27/code-mixing-sociolinguistics/

[21] http://languagelinguistics.com/2018/02/06/difference-code-mixing-code-switching/

[22] https://www.edureka.co/blog/classification-in-machine-learning/

[23] https://en.wikipedia.org/wiki/Naive_Bayes_classifier#Multinomial_naive_Bayes

[24] https://towardsdatascience.com/simple-guide-for-ensemble-learning-methods-d87cc68705a2