Sentiment Analysis for Abbreviated and Slang Words

Lavanya L
PES University
Dept of CSE
Bangalore,India
lavanyakumar2102@gmail.com

Jyosna S
PES University
Dept of CSE
Bangalore,India
hema.m456@gmail.com

Ashwini M Joshi
PES University
Dept of CSE
Bangalore,India
ashwinimjoshi@pes.edu

***Abstract*—Social media websites have emerged as a one of the platforms to raise users' opinions and influence the way any business is commercialised.Opinion of people matter a lot to analyse how the propagation of information impacts the lives in large scale network like Twitter.Sentiment analysis of the user tweets/reviews determine the polarity and inclination of the vast population towards specific interest, item or entity[2]. This analysis can be used to understand the general sentiment of the public with respect to one event. In this project, we exploited a fast and accurate generation of labels for a movie review data set[2]. The primary aim of this project is to provide a method for analyzing sentiment for the noisy movie review twitter data that include abbreviated and slang words.These days, the latest mobile devices and websites are interpreting mashup language based keyboards; this has enabled many users to express their opinions and views about products in 'Hinglish'. This paper delves into the design of generation of our own labels given a data set and perform sentiment analysis on the noisy data using three different classifiers Logistic Regression,SVM and Random Forest . Results classify user's perception via positive, negative and neutral as the sentiments for the tweets.**

***Index Terms*—sentiment analysis, tweets,positive,negative, neutral,labels, hindi-english words, abbreviated words,slang words.**

I. Introduction

In the past, traditional media, TV, radio, newspapers were used to dominate the globe as the source of reporting about events to the public. Now, as internet is growing bigger, its horizons are growing wider and opinions of people makes a major decision about a topic being important or not. A topic/event becomes trending if more and more users contribute their opinions and judgements thereby making it a valuable resource for online perception. Mining of such informal and homogeneous data is highly useful to draw conclusions in various fields. Though, the highly unstructured format of the opinion data available on web makes the mining process challenging.

Sentiment analysis relates to the problem of mining the sentiments from online available data and categorizing the opinion expressed by an author towards a particular entity into at most three preset categories: positive, negative and neutral [1].It is intended to serve as an application to understand the attitudes, opinions and emotions expressed within an online mention. The intention is to gain an overview of the wider public opinion behind certain topics. Nowadays sentiment analysis is an integral part of social listening and it is more than just a feature in a social analytic tool.Using all the machine learning techniques and natural language processing it becomes much easier to classify the opinions as positive, negative or neutral.

In this paper, we present a sentiment analysis process for noisy movie review twitter data set.Twitter is a micro-blogging site that is rapidly growing in terms of number of users.Though there is abundance of data for analysis of the sentiment,a lot of challenges are involved in terms of tonality, lexicon and grammar of the tweets. They normally tend to be highly unstructured and non-grammatical. Since we have taken a movie review data set, there is extensive usage of slang words, abbreviated words acronyms and out of vocabulary words found common. The categorization of such words becomes extremely tough for natural processor involved, hence our primary aim was to to all the pre-processing which includes all the above mentioned noisy data.This project also took care of eradicating all the Hinglish words from the data set which were considered as noise. Hinglish is a linguistic blend of Hindi and English and is spoken by upwards 350 million people in India .It became very essential to replace these words with the respective English words to make sure that actual content of the review containing the emotions of the user is not lost.

Further in the project, we generated our own labels for the twitter data set using various methods and different classifiers were used to predict the sentiment of each review which is discussed in detail in this paper.

The rest of the paper is organized as follows. In section II we discuss the existing work in the field of sentiment analysis via various research papers. Section III gives brief details about technologies used. Section IV describes the detailed methodology and implementation to carryout sentiment analysis. The challenges we came across and the problem we resolved is in section V,VI. Further, in section VII future work is discussed.

II. Literature Review

This section summarizes some of the scholarly and research works in the field of natural language processing and machine learning to analyse the sentiment of people's opinions. In

the current years, voluminous amount of research has been conducted in the field of sentiment analysis.[1]

In [3], authors have presented a logical approach of enhancing the accuracy of sentiment analysis by dealing with acronyms in a particular domain. They have performed opinion extraction from the provided acronym filled words using machine learning and lexicon based approach. This work has thus showed that the presence of acronyms in the text can compromise the performance of the system in identification of semantic score for the text.

In [4], authors have proposed a way to classify tweets as positive, negative or neutral. In the preprocessing, they have faced few new challenges of treating repeated characters in informal expression of words and the affect of contrast in determining the sentence polarity. They propose ways to select optimal classifiers among Decision Tree, K Nearest Neighbor, Support Vector Machine, and a Voting Classifier - a soft voting method has been proposed to combine results from various classifiers for solving Twitter sentiment analysis problem.

In [5] the authors have carried out the sentiment analysis of text containing Hindi-English words .In their paper they have used a porter stemmer that took care of Hindi-English word forms and distinction where the feature rows are build in Hindi-English tokenizer, Hindi-English stemmer characterization and frequency counts.The stemmer also had two parts a positive distinction and a negative distinction. Then they have performed the sentiment analysis classification using machine learning techniques, which are Naive Bayes and Multinomial Naive Bayes classification.

In [6], the paper presents the overall procedure to perform Opinion Mining process to categorize unstructured data of Twitter into positive or negative categories. Secondly they have used various methods like lexicon based approach, dictionary based and corpus based approaches to perform the opinion mining on twitter data. This lexicon based approach is found to be language specific for this paper. In future, the paper is to implement opinion mining for diverse languages as well.

In [7], the authors have a presented a step-by-step technique methodology for Twitter sentiment analysis. Two methods are used to measure variations in the public opinion, first is lexicon-based method, uses a dictionary of words with assigned to them semantic scores to calculate a final polarity of a tweet, and incorporates part of speech tagging.Second is machine learning approaches using supervised learning for classification. They have also combined lexicon and machine learning approaches as one of the features to improve the accuracy.

In [8] the authors have a way to handle the slang words in the tweeter data.In the first phase they give the input data which is a username or a hashtag,then analyse the number of tweets in the second phase. In the third phase the data is restored from the database and in the fourth phase the tweets are processed. The slang words lexicons are switched with their associated meaning and the feature extraction is done which gives polarity to each words and adding these polarities they have categorised the tweets .Then sentiment analysis is performed on these tweets using the Naive Bayes classifier.

In [9] the authors have done sentiment analysis on twitter data where they have proposed a method of pre-processing which relies on the bindings of the slang words on the other coexisiting words to check the significance of the slang word.They have used n-grams to find the binding and conditional random fields to check the significance of slang word.then the sentiment classification is done using the SVM to see the improvement in accuracy.

III. Methodology For Sentiment Analysis

The sentiment analysis of twitter data is an emerging field that needs much more attention for its opinions to be valued.Fig.1 shows the high level diagram of the steps to be carried out for the process of sentiment analysis. Firstly the collected dataset is pre-processed to perform all the necessary data cleaning including abbreviated words and slang words. Secondly, important features are extracted from the clean text using feature selection methods.Thirdly, we generate labels as positive, negative or neutral for the entire dataset as the dataset provided does not have the text labelled. Then we prepare the training and test data which can be used as the input to the classifier. Finally, the train data and the test data are fed to the built classifiers to classify the remaining data i.e test data. Each of the processing steps is discussed in detail in the following sub-sections.

Fig. 1 Sentiment analysis process of noisy data

A. Loading The Dataset

For our research we have taken a bollywood movie review dataset from kaggle which is publicly available and loaded the dataset using Pandas.

B. Generation of labels for the tweets

One of the challenges in our twitter dataset was that we had to produce labels for the tweets as there was no true label present in the dataset.So we used two methods to generate the labels for our tweets automatically. These tweets generated were used as an input to the classifier to find the sentiment of the tweet. The tweets generated were compared with our manually entered test data labels, to check how well are the labels being generated. In this paper we also compared which method of generated labels worked well on the classifier. The first method is POS tagging, which is a lexicon approach and the second one is TextBlob for processing all the textual data.

- Pos Tagging with the use of sentiwordnet. The first method that we have used here is generating labels using

the sentiwordnet which is a lexicon resource in which each WordNet synset is associated with three numerical scores objective,positive,negative.The typical use of the SentiWordNet is to enrich the text representation in opinion mining algorithms , adding information on the sentiment related properties of the terms in the text.The method used to develop the SentiWordNet is based on quantitative analysis of the glosses associated to synsets. The three scores are derived by combining the results produced by eight ternary classifiers all characterised by similar accuracy but different classification behaviour. So we have design a function the takes the dataset as the input ,then extracts the tweets.On each tweet tokenization is applied and then these tokens are checked if there are present in the SWN corpus if there are present then it assigns those tokens with positive and negative scores that is given by the SentiWordNet if not present the tokens are further lemmatised and checked now also if not present it further performs stemming on these tokens and checks and assigns the scores if still not present they are appended as missing words . At the last step the positive and negative scores of the tokens of each tweet is aggregated .Now for each tweet we compare the positive and negative scores and label them as positive if positive score is more than negative score else if negative score is more than positive score then the tweet is labelled as negative if both the scores are equally the tweet is labelled as neutral.

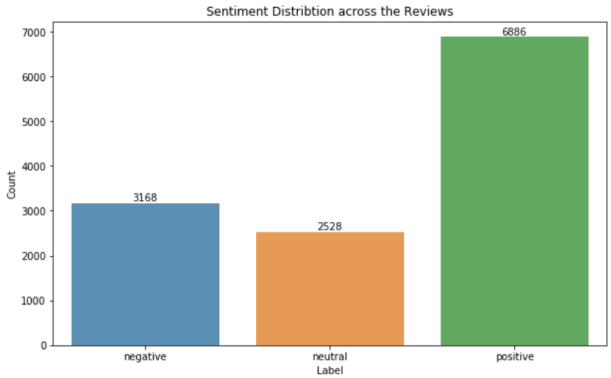

Fig. 2 Number of positive, negative and neutral labels from the text generated using POS Tagging.

- The other approach used for generating text to our dataset is TextBlob.TextBlob is a python library for processing of textual data.It provides a simple API dividing into common natural language processing tasks such as sentiment analysis, classification etc. Here we use the text blob for sentiment analysis. The output of this text blob is categorized into two - Polarity and Subjectivity. Polarity is a float value within the range [-1.0 to 1.0] where 0 indicates neutral, +1 indicates a very positive sentiment and -1 represents a very negative sentiment.Subjectivity is a float value within the range [0.0 to 1.0] where 0.0 is very objective and 1.0 is very subjective. Subjective sentence expresses some personal feelings, views, beliefs, opinions, allegations, desires, beliefs, suspicions, and speculations where as Objective sentences are factual.For a review/tweet textblob goes along finding the words and phrases it can assign polarty to, and it averages the score for a longer text like tweet. The fig 7. shows the number of positive,negative and neutral label in POS Tagging methods.

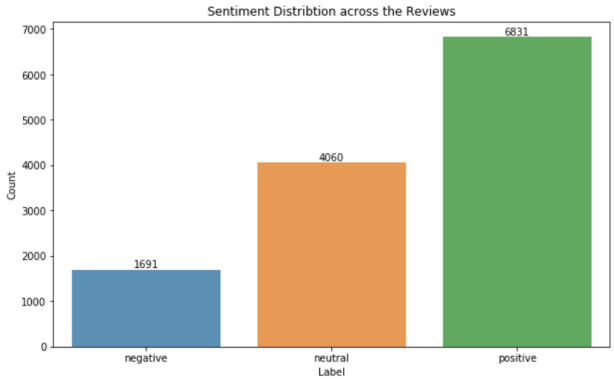

Fig. 3 Number of positive, negative and neutral labels from the text generated using TextBlob.

C. Comparison of the two techniques for label generation

The two methods used for generation of labels which were explained in a detailed way above. After all the pre-processing we found that TextBlob gave a better accuracy over POS tagging as POS tagging couldn't handle most of the words with their respective sentiments like TextBlob. Hence in the rest of the paper we have given input for the classifiers as the labels generated from TextBlob. The examples of sentiment orientation for the given text is shown in fig. 9.

	text_raw	label
0	love kabir superb movie story character sc...	positive
1	sweet teddy bear guy chicks magnet behave like...	positive
2	obsessed bekhayali happened	negative
3	character kabir needs psychological treatment ...	negative
4	beautiful person talented actor simplicity sa...	positive

Fig. 4 Review with semantic orientation

D. Data Preprocessing

Mining of twitter data is challenging task because it contains the user's raw opinion which are un-grammatical, informal and noisy. In order to apply the classifier, its very important to pre-process the data for better results. The pre-processing tasks include uniform casing, punctuation removal, removal of hashtags and other twitter notations, URLs, stopwords, replacement of slang words and abbreviated words.[4] The following steps show the pre-processing procedure.

1) Conversion of Strings to Lowercase:

- In this paper we start the text processing by normalizing the tweets by converting it to the lowercase by using the python string function (str.lower) which makes further processing steps easier.The function is used as shown in fig. 2.

```
df["text_raw"] = df["text_raw"].str.lower()
df["text_raw"].head()
```

```
0    @shahidkapoor i am in love with kabir @kabirsi...
1    that's because you are sweet teddy bear guy an...
2    i'm obsessed with bekhayali and kaise hua #kab...
3    the character of kabir needs a psychological t...
4    @advani_kiara you are such a beautiful.person ...
```

Fig. 5 Code snippet for lowercase

2) Expansion of the contracted words:

- The tweets in the dataset contains a lot of contracted words like i'm, she's ,there's etc which has to be decompressed with their respective expansions using dictionary that contains the mapping of the contracted words to their expansions which helps in text standardization. The code snippet for expanding contractions is shown in Fig 3, where the contracted words have been decompressed.

```
df['text_raw'] = df["text_raw"].apply(lambda text: expandContractions(text))
df["text_raw"].head()
```

```
0    @shahidkapoor i am in love with kabir @kabirsi...
1    that is because you are sweet teddy bear guy a...
2    i am obsessed with bekhayali and kaise hua #ka...
3    the character of kabir needs a psychological t...
4    @advani_kiara you are such a beautiful.person ...
```

Fig. 6 Code snippet for expanding contractions

3) Replacing the Abbreviated/Chat words:

- In this research paper one of our main task was to replace the abbreviated words in the tweets with the suitable english words. The task is to perform the identification of abbreviated words first and then replace it. For this purpose, we crawled different websites and in order to get the list of most commonly used acronyms. Fig 4 shows the snap of some of the acronyms and their definitions that are maintained in acronym dictionary, which are replaced with acronyms present in the text after the identification of these acronyms.

Acronyms	Long Form
Lol	laughing out loudly
tia	thank you in advance
gr8	great
rip	rest in peace
yolo	you live only once
Asap	as soon as possible
Bff	best friends forever

Fig. 7 Acronyms Dictionary

4) Removal of stopwords:

- There are some words in the dataset which are frequently occurring and they don't add any real value to the review/tweet. Such words must be removed, as it affects the entire performance of the model. The stopwords include "a","the","am etc which serves no purpose and this fature is implemented using a list stored in stopword file in nltk.corpus. We compare each word in the tweet with the list and delete the word in the tweet if it matches with the stopword list as shown in fig.

```
def remove_stopwords(text):
    """custom function to remove the stopwords"""
    return " ".join([word for word in str(text).split() if word not in stopwords])
df["text_raw"] = df["text_raw"].apply(lambda text: remove_stopwords(text))
df["text_raw"].head()
```

```
0    @shahidkapoor love kabir @kabirsingh superb mo...
1    sweet teddy bear guy chicks magnet behave like...
2             obsessed bekhayali kaise hua #kabirsingh
3    character kabir needs psychological treatment ...
4    @advani_kiara beautiful.person talented actor ...
```

Fig. 8 Code snippet for stopwords removal

5) Replacing the Hindi-English words:

- Another goal for us was to replace these Hindi-English words with correct english words so that we might not lose the important sentiments that might be conveyed by these words . So here in the same way as in the replacing of the contracted words we have made a dictionary that containing all the possible Hindi-English words mapped with the correct english words like for example"badmash":"naughty", "bas":"stop", "biwi":"wife" etc. So here each word in the tweet is checked for in the dictionary if present they are replaced with English words otherwise they are kept unchanged.

```
df['text_raw'] = df["text_raw"].apply(lambda text: hinglish_replace(text))
df["text_raw"].head()
```

```
0    @shahidkapoor love kabir @kabirsingh superb mo...
1    sweet teddy bear guy chicks magnet behave like...
2          obsessed bekhayali how happened #kabirsingh
3    character kabir needs psychological treatment ...
4    @advani_kiara beautiful.person talented actor ...
Name: text_raw, dtype: object
```

Fig. 9 Code snippet for stopwords removal

6) Removal of Punctuations:

- The Punctuations are removed from the tweets and have been replaced with empty strings which helps in removing the unwanted characters like @,#,* etc which don't add any value to the sentiment.This helps to reduce the clutter from the tweets in the dataset.

E. Feature Extraction

In feature extraction methods, we extract the different aspects such as adjectives, verbs and nouns and later these aspects are identified positive,negative or neutral to identify the polarity of a the whole sentence. TF-IDF is a feature vectorization method used in text mining to find the importance of a term to a document in the corpus. This feature is useful for a case where we need to find trending topics or to create word clouds. Its been used in this project for finding the sentiment of the word in the given tweet/review.

IV. Classification Algorithms

We consider the task of classifying a tweet as positive,negative or neutral. Several classifying algorithms are

tested in order to find the best performance one. In the simplified form, the text classification task can be described as follows: if our training dataset of labelled data is Ttrain = (t, l1), ..., (tn, ln), where each text ti belongs to a dataset T and the label li = li(di) is a pre-set class within the group of classes L = li, ..., lm, the goal is to build a learning algorithm that will receive as an input the training set Ttrain and will generate a model that will accurately classify unlabelled documents. Below are a few of the machine learning algorithms which we have used in this paper for the text classification.

A. Logistic Regression

So in our paper for sentiment classification we started with the simplest classification model, the Multinomial Logistic Regression.This model is a form of logistic regression that is used to predict a target variable for more than two classes. It is a modification of logistic regression using the Softmax function instead of sigmoid function. The softmax function squashes all the values to range [0,1] and the sum of the elements is one.

$$Softmax(x)_i = \frac{e^{x_i}}{\sum_{j=1}^{N} e^{x_j}}$$

The reason why we used the Multinomial Logistic Regression classification is, it is one of the efficient and simplest classification techniques as it does not require any input features to be scaled ,does not require any tuning and this technique is suitable when there are fixed number of categories in which the data needs to be classified. So we split the dataset into test and train set and trained the classifier using the training set and then tested the classifier with the test set.

B. SVM

Support Vector Machines is an algorithm that determines the best decision boundary between vectors that belong to a given group (or category) and vectors that do not belong to it. Vectors are (sometimes huge) lists of numbers which represent a set of coordinates in some space. So, when SVM determines the decision boundary we mentioned above, SVM decides where to draw the best “line” (or the best hyperplane) that divides the space into two subspaces: one for the vectors which belong to the given category and one for the vectors which do not belong to it.

We chose SVM for this paper because it solves the traditional text classification problem effectively, generally outperforming Naive Bayes as it is in favour of maximum margin. If we determine the given tweet/review with t, the hyper plane using h, and classes using a set Cj € {l, -1} into which the tweet has to be classified, the solution is written as follows equivalent to the sentiment of the tweet.

$$h = \sum i\, ai\, Ci\, tj,\ ai \geq 0 \tag{1}$$

The idea of SVM is to determine a boundary or boundaries that separate distinct clusters or groups of data. SVM performs this task constructing a set of points and separating those points using mathematical formulas. Fig. 10 illustrates the data flow of SVM.

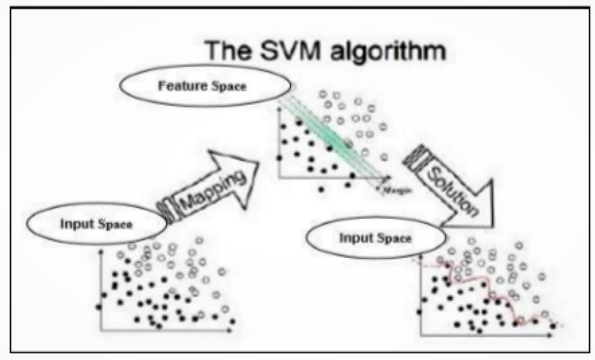

Fig. 10 Support Vector Machine Workflow

C. Random Forest Classifier

Random Forest Classifier is a meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and uses the averaging to improve the predictive accuracy and control over fitting.The sub-samples size are always same as the original input where the samples are drawn with replacement . For a given data point, each tree votes for a particular class label and the class label gaining the maximum votes will be assigned to that data point. The error rate of this classifier depends on the correlation or association among any two trees in the forest in addition to the strength of definite or individual tree in the forest. In order to minimize the error rate, the trees should be strong and the degree of associativity should be as less as possible.In the classifier tree, the internal nodes are represented as the features, the edges leaving a node are represented as tests on the feature’s weight, and the leaves are represented as class categories. It performs classification preliminary from the root node and moves incrementally downward until a leaf node is detected. The document is then classified in the category that labels the leaf node. This algorithm is used in many applications of speech and language processing.[1]

Below is the snippet of how the classifier gives the result based on results of all the trees.

We choose this classifier as it is impressive in versatility where the data need not be rescaled or transformed and can be easily used to handle categorical features . They are robust to outliers and since it fits a number of decision trees it provides low bias and moderate variance.

V. Evaluation Parameters

In common, the performance of sentiment analysis is measures using four indicators as follows: Accuracy, Precision, Recall and F1-score.[1] Accuracy is defined as all true predicted cases against all predicted cases. If we get 100% accuracy, then it means that all predicted cases exactly same as the actual cases. Precision is the percentage of relevant words to the total number of words identified by the system. Recall is

1. For $b = 1$ to B:

   (a) Draw a bootstrap sample $\mathbf{Z}^*$ of size N from the training data.

   (b) Grow a random-forest tree T_b to the bootstrapped data, by recursively repeating the following steps for each terminal node of the tree, until the minimum node size n_{min} is reached.

      i. Select m variables at random from the p variables.

      ii. Pick the best variable/split-point among the m.

      iii. Split the node into two daughter nodes.

2. Output the ensemble of trees $\{T_b\}_1^B$.

To make a prediction at a new point x:

Regression: $\hat{f}_{rf}^B(x) = \frac{1}{B}\sum_{b=1}^{B} T_b(x)$.

Classification: Let $\hat{C}_b(x)$ be the class prediction of the bth random-forest tree. Then $\hat{C}_{rf}^B(x) = majority\ vote\ \{\hat{C}_b(x)\}_1^B$.

Fig. 11 Random Forest Algorithm

the percentage of the number of relevant words identified by the system to the total number of relevant words.[3]

VI. Results and Analysis

We present our results for the positive ,negative and neutral classifications. The results act as first step for our classification approach. The dataset was collected and maintained by kaggle which is publicly available. First we generated labels to identify the sentiment of the text for feeding it into the classifier using TextBloB. Then we checked the accuracy of generated labels with the manually entered labels for test data and we found the accuracy to come out to be 83.38%. After which the acronyms, slang and informal words in text were mapped to its definitions i.e long forms.Then we fed this results as input to the three classifiers: Logistic Regression, Support Vector Machines and Random Forest classifier.

Firstly, for Logistic Regression we evaluated the result using information retrieval metrics such as Precision, Recall, Accuracy and F1-score. The results of the classifier as shown bellow in fig

Logistic Regression Accuracy Score -> 86.93579148124603

	precision	recall	f1-score	support
negative	0.51	0.81	0.62	258
neutral	0.94	0.81	0.87	1144
positive	0.91	0.91	0.91	1744
accuracy			0.87	3146
macro avg	0.79	0.85	0.80	3146
weighted avg	0.89	0.87	0.88	3146

Fig. 12 Logistic Regression Algorithm

Now for support vector machines, we calculated all the performance metrics which is shown in the figure below. The accuracy of SVM came out to be 87.44% being the best classifier for our model and the result is shown below.

In the next part we did try the review dataset on Random Forest Classifier. We used it in here as our third classifier because it is robust to outliers and it works really well for categorical data.The fig below shows the performance metrics for Random Forest Classifier which has an accuracy of 83%.

SVM Accuracy Score -> 87.44537147397695

	precision	recall	f1-score	support
0	0.61	0.70	0.65	291
1	0.90	0.89	0.89	842
2	0.93	0.90	0.91	1384
accuracy			0.87	2517
macro avg	0.81	0.83	0.82	2517
weighted avg	0.88	0.87	0.88	2517

Fig. 13 SVM Performance Metrics

RANDOM FOREST Accuracy Score --> 0.822949777495232

	precision	recall	f1-score	support
negative	0.98	0.20	0.33	414
neutral	0.78	0.93	0.84	987
positive	0.85	0.91	0.88	1745
accuracy			0.82	3146
macro avg	0.87	0.68	0.68	3146
weighted avg	0.84	0.82	0.80	3146

Fig. 14 Random Forest Performance Metrics

We note that the best results are reached through Support Vector Machines being applied as the second classifier for the classification process.Finally, our classification approach provides improvement in the accuracy by doing a lot of preprocessing for our dataset which holds a key role in getting better performance for the system.

VII. Conclusion

In this paper, we started to discuss how abbreviated words and the Hindi-English words were in abundant usage in different social media platforms like Twitter, Facebook etc., and how important it is to replace them with the appropriate English words instead of removing them so that during the sentiment analysis we won't lose the sentiment conveyed by these words .We elaborated on how we use a dictionary based method to replace these abbreviated and the Hindi-English words. Before these we had done some of the text pre-processing steps like removal of punctuation,removal of stop words,expansion of the contracted words . Then we discussed about how we generated label for our dataset which had no true label by using two methods one using the SentiWordNet and the other using the Text Blob . When we tested these two methods with our manually labelled data we saw that the text blob method was efficient for labelling and continued the rest with this labelling.Then we split our dataset into train and test data , and trained different classifier like the Multinomial Logistic Regression, Support Vector Machine and the Random forest. Using the test data we generated results of the classifiers in the form of four performance metrics accuracy , precision,recall and F1 score.Based on this result we concludes that the SVM classifier best classifies our tweets.

VIII. Future Work

The future opportunities in the domain of sentiment analysis include developing a technique to perform sentiment classifi-

cation that can be applicable for any data regardless of domain. Our paper has few incorrect results due to some ambiguous words. The proposed system does not detect the emotion of the words e.g "it was damn good", "i am obsessed with this song". "damn","obsessed" has been identified as negative word. Where it is not used as a negative word in the text but a figure of speech called hyperbole.This can be improved in the future by incorporating the hyperbole detection in the system. In addition, language diversity in social media data is a key issue which is required to be eliminated in future.

ACKNOWLEDGMENT

We would like to express our deep and sincere gratitude to our Chairperson, Shylaja S Sharath, Computer Science Department, PES University for giving the opportunity to do this research on sentiment analysis. We would also like to express our heartfelt thanks for the Computer science Department in our college for providing with all the resources throughout the project.This project would have been impossible without the college's guidance and support throughout the project.

REFERENCES

[1] M. Desai and M. Mehta, "Techniques for sentiment analysis of twitter data: a comprehensive survey", 2016 International Conference on Computing, Communication and Automation (ICCCA), 2016.

[2] Amandeep Kaur, Deepesh Khaneja, Khushboo Vyas, Ranjit Singh Saini, "Sentiment analsyis on Twitter using Apache Shark", 2016

[3] S. Malka, S. Jan and I. A. Shah, "Sentiment analysis based on abbreviations in text", Pakisthan Journal of Science, 71 (4 Suppl.): 2019, Page 230 - 234 ISSN: 2411 - 0930

[4] Huong Thanh Le *, Nhan Trong Tran, "A sentiment analyzer for informal text in social media", Hanoi University of Science and Technology - No. 1, Dai Co Viet, Hai Ba Trung, Hanoi, Viet Nam, 2018

[5] Jagmeet Singh1,Dr. Shashi Bhushan2 ,"Analysison Hinglish Opinion Using Multinomial Naive Bayes Algorithm", 2016.

[6] Ashlesha Wadurkar,Akash Bhad,Bhavana Rajput,Durgesh Pandey,Dr. Sachin V. Solanki,"A review of opinion mining on twitter using lexicon based approach",2019

[7] Olga Kolchyna, Th´arsis T. P. Souza, Philip C. Treleaven and Tomaso Aste,"Methodology for twitter sentiment analysis"

[8] A. Brahmananda Reddy, D.N.Vasundhara, P. Subhash, "Sentiment research on twitter data" ,International Journal of Recent Technology and Engineering (IJRTE),SSN:2277-3878,Volume-8 , Issue-2S11, September 2019.

[9] Tajinder Singh and Madhu Kumari , "Role of Text Pre-Processing in Twitter Sentiment Analysis", Twelfth International Multi-Conference on Information Processing-2016 (IMCIP-2016).

[10] Vishal A. Kharde and S.S. Sonawane ,"Sentiment Analysis of Twitter Data: A Survey of Techniques" , International Journal of Computer Applications (0975 – 8887) Volume 139 – No.11, April 2016.

[11] Apoorv Agarwal, Boyi Xie , Ilia Vovsha , Owen Rambow, Rebecca Passonneau "Sentiment Analysis of Twitter Data" , Department of Computer Science Columbia University New York, NY 10027 USA.

[12] Shubham Goyal ,"Review Paper on Sentiment Analysis of Twitter Data Using Text Mining and Hybrid Classification Approach",2017 IJEDR , Volume 5, Issue 2 , ISSN: 2321-9939.

[13] Asiri Wijesinghe, "Sentiment analysis on movie review", 2015

[14] Khalid N. Alhayyan, Imran Ahmad ,"Discovering and analyzing important real-time trends in noisy twitter streams ",ISSN: 1690-4524, SYSTEMICS, CYBERNETICS AND INFORMATICS, VOLUME 15 - NUMBER 2 - Year 2017.

[15] AdityaBohra,DeepanshuVijay,VinaySingh,SyedS.Akhtar, Manish,Shrivastava,"A dataset of hindi-english code- mixed social media text for hate speech detection",International Institute of Information Technology Hyderabad, Telangana, India.