International Journal on Emerging Technologies **11**(2): 811-816(2020)

ISSN No. (Print): 0975-8364
ISSN No. (Online): 2249-3255

A Pattern based Approach for Sentiment Analysis using Ternary Classification on Twitter Data

E. Kannan[1] and Lakshmi Anusha Kothamasu[2]
[1]Professor, Department of Computer Science and Engineering,
VelTech Rangarajan Dr. Sagunthala R & D Institute of Science & Technology, Avadi 600062 (Chennai), India.
[2]Research Scholar, Department of Computer Science and Engineering,
VelTech Rangarajan Dr. Sagunthala R & D Institute of Science & Technology, Avadi 600062 (Chennai), India.

(Corresponding author: E. Kannan)
(Received 08 January 2020, Revised 06 March 2020, Accepted 07 March 2020)
(Published by Research Trend, Website: www.researchtrend.net)

ABSTRACT: With today's advances in technology, people can use numerous social platforms such as WhatsApp, Instagram, Facebook, etc. to share/express their opinions. This information is readily available for anyone, such as enterprises, authorities, and groups of people, which produce large amounts of data every day. Twitter has become extremely focused and manually classifying this data and identifying polarities between optimistic, negative, and neutral classes is very challenging. However, several current technologies are confined only in clustered settings. This volume of data, therefore, provides an immense opportunity that can be used to sense the inclination for such items. However, since nobody could waste infinite time reading such messages, an automatic decision-making solution is required. A couple of hundred messages at most can only be read. Preprocessing influences precision. In our work, we try to perform an extensive preprocessing on "tweets" and afterward the technique of multinomial Naive Bayes is applied for the polarity of the tweet to be classified where it will be positive, negative and neutral. If the tweet includes both positive and negative aspects, then perhaps the prevalent sentiment is chosen as final. It is evident that with dataset1, we achieved an accuracy of classification is 91% and for latter datasets2 and datasets3 to be 81%, 85% respectively. Our method suggested improved accuracy in case of simulation study as compared to the previous study.

Keywords: data collection, feature selection, machine learning, sentiment classifier, sentiment analysis, Twitter, text preprocessing, TCA.

I. INTRODUCTION

Social networking has now become an information base across all sorts of ages in the current digital environment. Almost anyone can express their feelings as opinions or as comments about everything else, such as videos, labels, goods, social events, etc., as a unified platform. Explosive growth in application development has contributed to a significant number of people having a different opinion and web perspectives. As we already know, Twitter has become quite famous among all social network networks. Many of the users world-wide often use Twitter as their medium to express their feelings or opinions on a variety of subjects [1]. Analyzing such kind of emotions can lead to fruitful information in fields such as social profiling and personalized marketing. However, it is not an easy task to be done, as the language used in Twitter is often informal that gives new challenges to analyze such unstructured text. Twitter has been a major information source and is a mini-blogging website, which is very well renowned because of its twitter posts. Twitter is one medium in which we can know what is happening in the world and what people are talking about right now. Twitter until November used to have only a length of 160 characters in which 140 characters used for the tweet and 20 characters for user name purpose.
The length was chosen in such a manner to fit it to the text message. However, in late September Twitter has doubled the length of a tweet to 280 characters long enough in about 40 languages except in Chinese, Japanese, or Korea-language tweets. However, initially, it was tested on a set of people and it came into effect for everyone from early November. Even though the length was increased, Twitter noted that certain 5 percent of people's tweets were much more than 140 characters but only 2 percent had more than 190 characters. Another one for her particular tweets regarded as tweets. The impact of the increase in length made people's engagement towards Twitter had increased a lot in which now they can send longer enough tweets very easily even though it was very often. So, the average length of the tweet has not changed, but the user's engagement and the followers have improved very much [4]. Sentiment analysis pertains generally to the need for text analysis, the processing of natural languages, and computer literature for the systemic identification, extraction, quantification, and practice of qualitative information [2].
In certain words, it distinguishes and classifies the differing views conveyed in a piece of text by machine, particularly to decide whether either user has a positive, negative or neutral attitude against one given subject, company, service, etc. Simply speaking, Sentiment Analysis can be described as somewhat of a way to systematically analyze online expressions. Throughout the company, it is quite important to study the social feelings of your name, service, or product when tracking personal conversations. It is often named the opinion mining by the technicians, as the method to assess the emotional tone under a series of words, always had to achieve an interpretation of the thoughts and feelings conveyed in an online sense. There are several challenges in performing Sentiment analysis. Starting with what type of dataset we are dealing with, how the sentiments are to be classified, what should be the evaluation metrics, and how the results are to be visualized? This is the most pretentious topic that so many authors have been studying, as it poses problems to improve accuracy.

Particularly over recent years, the issue of sentiment analysis has been researched extensively and current approaches are subject to some shortcomings. One challenge was that most solutions are limited to clustered contexts. In addition, sentiment analysis is focused on theory, natural language processing techniques, and methods of machine learning. Moreover, certain forms of methods involve the time and waste of several computing resources. The fundamental technologies are not appropriate or acceptable for opinion mining as the computational capacities of such systems vary considerably from the rapid growth of the data required.

Machine learning is one of the applications of Artificial Intelligence that the systems are provided with the ability to be learned automatically and improving themselves from experiences without being explicitly programmed. Machine learning focuses primarily on developing applications that obtain and use data for their learning purposes. The main objective is to enable machines to automatically learn and act without human involvement. Machine learning, in particular, is an analysis of specific algorithms and mathematical models that often the computer systems use to execute an efficient function, based solely on the trends and inferences, without using clear instructions. It will be generally used as an artificial intelligence subset [3]. In this paper, the main goal was "*on classifying a text into a ternary classification by machines for social network learning and microblogging"* that facilitates the users' discovery of messages with a particular theme or material.

II. BACKGORUND WORKS

There is been continuous research going on Twitter Sentiment Analysis for many years. Everyone intends to formulate an algorithm that can produce higher accuracy and improved F-measure on the test data with more number of tweets. Ahad *et al.,* (2018) have proposed a hybrid approach of using two machine learning algorithms namely K-Nearest Neighbors (KNN) and Support Vector Machines (SVM) as the classifiers and found hidden sentiment from tweets. The observations were obtained in terms of F-measure and accuracy. They implemented using weka and worked on several datasets and achieved an average accuracy of 76.17 with ternary classification [4]. Hota and Pathak (2018) proposed the source data derived from the Python Tweepy and used KNN classifier with either the correct value of variable k and selected n-gram modeling strategy for extraction of features and their research study showed 86% better accuracy over 81% of SVM accuracy [5].

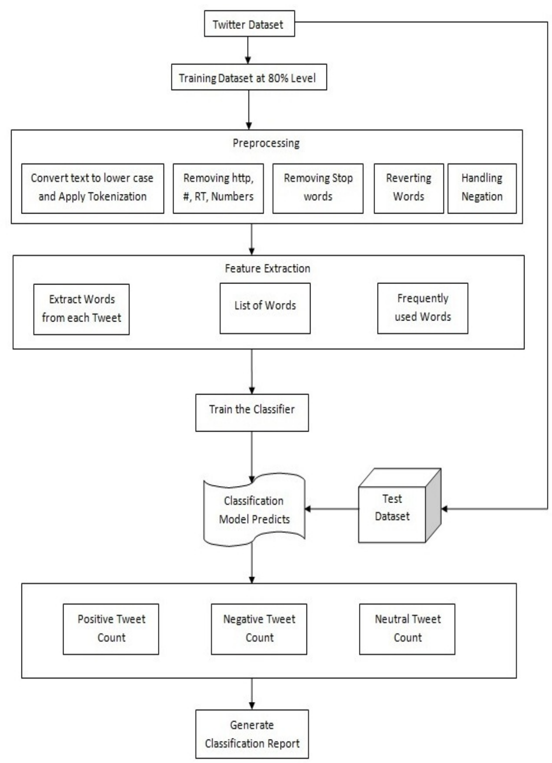
Zainuddin (2017) has carried out a study on hybrid sentiment classification on Twitter data. The proposed hybrid sentiment classification for Twitter by embedding Principle Component Analysis (PCA) feature selection method and achieved an accuracy of 76 percent [6]. Bouazizi & Ohtsuki (2017) have proposed a pattern-based approach for multiclass sentiment analysis in Twitter by using a random forest algorithm and achieved an accuracy of 70.1 percent [7]. Lalji and Deshmukh has proposed Twitter Sentiment Analysis using Hybrid Approach and achieved performance between 57.13 to 59.98 by using different training data size vary between 5000 to 25000 [8].

III. MATERIALS AND METHODS

We introduce our system where we could have clarified the process of collecting data interpreting emotions and classifying Twitter opinion. We also called the tweets, which numerous users have shared for their opinions throughout the form of hash tags and re-tweets. We implemented the Naïve Bayes multinomial classification on the train set with Python software and developed a classification framework. To order to acquire the exact result, the concept was checked on both the reference dataset. The relevant sections provide a detailed insight into the methods that assisted us in sentiment study. There are various stages involved in performing Sentiment Analysis on Twitter data. The below figure depicts the stage-by-stage procedure involved in analyzing the sentiment about the opinions expressed on Twitter.

A. Data Collection

We considered three datasets, in which three out of two datasets are collected from Sanders and one from the Kaggle repository. Dataset-1 contains 13871 tweets and several fields are tweet_id, candidate, sentiment, name, retweet_count, tweet_text, tweet_created, tweet_location and user_timezone, etc.



.

Fig. 1. Proposed Model.

Dataset-2 Contains 498 tweets and the fields are text and sentiment. Dataset-3 contains 997 tweets and several fields are sentiment, id, time, username, user-id and text. We selected only the text and sentiment fields among all these fields for our research purpose. We then further split this dataset into two separate datasets as train-set and test set in which 80% as train-set and 20% as test-set.

B. Data/Text Pre-Processing

After the Data Acquisition, a further step in either text or sentiment classification is the data/ text pre-processing. Few techniques are to be applied to the data to reduce the dimensionality and assist in the overall improvement of classification effectiveness [9].

We first convert all tweets into the lower case because all stop words are available in lower case. We will get effective features when we apply to stop words removing method on the tweet and then set the stop words to English from nltk corpus. We create a new stop words list such that we add nltk stop words list excluding words such as not, on', wouldn't, hasn't, etc to find accurate sentiments. We split the tweet into tokens using Regular expression tokenizer. Generally tweets contains special characters like @, #, (, :, !, -, $, and &. Regular expression tokenizer removes all special characters from the tweet and it removes delimiters where gaps equal to True and then cleaned the words by removing words with HTTP, words with #, RT, numbers, punctuations, non-textual content and stops words after we replace the repeated characters by using wordnet. Generally, public uses won't, don't, couldn't, can't while writing tweets instead of writing will not, do not, could not, and cannot. To overcome this problem, we used the negation replacement technique.

After the cleaning of the tweet, the processed words have only remained in each tweet and then we added this tweet and its corresponding sentiment to the dataset. We can find a stop words list at www.ranks.nl, acronyms at www.acronymfinder.com and www.internetslang.com [9, 10]. Once the preprocessing is completed, then we forward the tokens to feature selection.

C. Feature Selection

In general, any text classification framework includes feature selection or feature extraction step. In the feature selection step, the prominently used approach is the bag-of-words (BoW) approach. In this approach, every unique term in the collection is regarded as an individual feature. In our research, to retain the effectiveness of word context, we used nltk frequency distribution function to get the word features and placed in an empty dictionary. The justifications for using functional selection are: It allows machine-learning techniques to be trained faster, helps to reduce a model's complexity and facilitates the identification of polarities, enhances system accuracy and decreases overfitting. We can get a list of features from our datasets by using frequency distribution keys method. We can plot the most used words in the dataset by using a frequency distribution plot method.

D. Sentiment Classifier

In general, Sentiments can be broadly classified into either two classes with positive, negative or three classes with positive, negative and neutral. Further, Sentiments can be classified as bad, good, dislike, like, etc. One of the recent, classifications of sentiment classes are Fun, Happiness, Love, Neutral, Sadness, Anger, and Hate [18]. In our research, we have experimented with ternary classification using Multinomial Naïve Bayes classifier; works on the concept of Bayes theorem has a strong assumption that features are mutually independent. The classifier aims to establish the association of text with its related sentiment. The reason behind to choose Multinomial Naïve Bayes Classifier is, the Prediction Speed of Naïve Bayes algorithm is faster than Logistic Regression, Linear Support Vector Machine, Decision Trees, and Nearest Neighbor (Source: Mastering Machine Learning from MathWorks). We applied all these classifiers on three datasets and achieved the best results with the multinomial naïve Bayes model. For that, we placed multinomial naïve Bayes results in this paper. The Multinomial Naïve Bayes Classifier gives good results when the size of the input is high. Sentiment classes S*(positive, negative and neutral) is assigned to tweet T, where

$$S = \text{argmac}_s\, P_{NB}(S/T)$$

$$P_{NB}(S/T) := \frac{\left(P(S) \sum_{i=1}^{m} \mathrm{P}\left(\frac{\mathrm{f}}{\mathrm{s}}\right)^{n_i(\mathrm{T})} \right)}{\mathrm{P}(\mathrm{T})}$$

In this formula, f represents feature, $n_i(T)$ represents count of features found in tweet T, m represents total number of features. The parameters p(f/s) are obtained through maximum likelihood estimates.

We employed Multinomial Naive Bayes classifier on the train dataset and further we used this classifier on our test dataset to predict the accuracy.

E. Procedure

Step-1: Read Twitter dataset from the repository in the form of .csv file

Step-2: All the tweet and corresponding sentiment fields can be retained and other fields are removed from .csv files

Step-3: Split the dataset into the training dataset and test dataset in the ratio of 80:20 i.e. 80% of train dataset and 20% of test dataset from the Twitter data set.

Step-4: Convert text into lower, set the stop words-set to English, and apply pre-processing steps to avoid noisy data from train dataset

Step-5: The word features are extracted from the trained dataset using nltk frequency distribution method

Step-6: Train the train data set using Multinomial Naïve Bayes classifier

Step-7: Apply the classifier on 20% of the test dataset from the Twitter data

Step-8: The Positive, Negative and Neutral count can be calculated by comparing each of test record with the classifier

Step-9: The metrics like Precision, Recall, Accuracy, Error Rate, Specificity, and F1-score can be calculated using defined formulae.

Ternary Classification Algorithm (TCA):

Input: Data T

Output: Result R

Initialization:

(i) Let T = {T1, T2, T3 . . . Tn}, Total n data or tweet records with labels

(ii) Let train, test = 0.8, 0.2 of the original T

(iii) For each data T

//Pre-processing

a. Convert text into lower case

b. Perform tokenization-using RegExp_Tokenizer. It splits tweet into tokens and also removes special characters like @, (, !, : etc from tweet and removes delimiters where gaps=True.

c. Remove http, RT, numbers, # tag and stopwords from tokens

d. Used wordnet for removing repeated characters.

e. Replace negations by assigning patterns.

iv. Extract features using nltk distribution function.

v. Train Multinomial Naïve Bayes Classifier on train-set.

vi. Compare each of Test record with classifier

If match found with Positive

Pos = pos+1

If match found with Negative

Neg = neg+1

If match found with Neutral

Neu = neu+1

vii. Print Result of Pos count, Neg count and Neu count

viii. Calculate all measures like Precision, Recall Accuracy, Error Rate, Specificity and F1-score

IV. RESULT AND DISCUSSIONS

This section describes the results of our experiment along with a discussion about the results obtained. We considered 3 different datasets. For discussion purposes, we here quote them as Dataset1, Dataset2, and Dataset3 respectively. We first start discussing in detail about Dataset1 and then we summarize the Dataset2 and Dataset3 instead of elaborated discussion.

Table 1: Datasets Information.

Dataset Name	Total Tweets	Test Data Size (20% of Total)
Dataset1	13871	2775
Dataset2	498	100
Dataset3	997	200

To start with, Dataset1 contains a total of 13871 records with 21 different fields with id, tweet, sentiment, name, retweet, time, etc. We considered 80 percent as train data and 20 percent as test data i.e. a total of 2775 records considered as test data from Dataset1. The following table gives the details about Sentiment classification of actual versus predicted that our experiment resulted.

Table 2: Sentiment Table with Actual versus Predicted on Dataset1.

Sentiment	Actual	Predicted
[Negative]	1645	1670
[Positive]	445	463
[Neutral]	685	642

When identified with the inconsistencies occurred with the actual results against predicted results, we identified that some negative tweets were predicted to be positive and neutral, while some positive tweets were predicted to be negative and neutral and some neutral tweets were predicted to be negative and positive. Such wrong predictions are known as False Negative, False Positive and False Neutral respectively. We depict this information in the form of a confusion matrix as given below.

Table 3: Confusion Matrix of Dataset1.

Actual/ Predicted	Predicted			
Actual	Sentiment	[Negative]	[Positive]	[Neutral]
	[Negative]	1468	18	6
	[Positive]	58	428	4
	[Neutral]	144	17	632

Performance Evaluation of Dataset1: The performance of the classifier is analyzed on the basis of four effective measures Precision, Recall, Accuracy and F-measure. In addition to these 4 measures, we can also measure using Specificity and Error-rate.

The Following Parameters required for Evaluating Performance:

Total tweet count is 20% of actual Dataset = 2775

True Negative (T_{NEG}) = 1468

Predicted Positive for Negative (P_{NPO}) = 18

Predicted Neutral for Negative (P_{NEN}) = 6

True Positive (T_{POS}) = 428

Predicted Negative for Positive (P_{PON}) = 58

Predicted Neutral for Positive (P_{PNE}) = 4

True Neutral (T_{NEU}) = 632

Predicted Negative for Neutral (P_{NNE}) = 144

Predicted Positive for Neutral (P_{NEP}) = 17

Metrics used for evaluating Performance of Dataset1:

Precision: It is the measure of exactness or quality.

Precision Negative (P_{NEG}) = T_{NEG} / (T_{NEG} + P_{PON} + P_{NNE}) = 1468/(1468+58+144) = 0.88

Precision Positive (P_{POS}) = T_{POS} / (P_{NPO} + T_{POS} + P_{NNE}) = 428/(18+428+17) = 0.92

Precision Neutral (P_{NEU}) = T_{NEU} / (P_{NEN} + P_{PNE} + T_{NEU}) = 632/(6+4+632) = 0.98

Precision (P) = (P_{NEG} + P_{POS} + P_{NEU}) / 3 = (0.88+0.92+0.98)/3 = 0.93

Recall (Sensitivity): It is the measure of completeness or quantity.

It gives the true positive rate.

Recall Negative (R_{NEG}) = T_{NEG} / (T_{NEG} + P_{NPO} + P_{NEN}) = 1468/(1468+18+6) = 0.98

Recall Positive (R_{POS}) = T_{POS} / (P_{PON} + T_{POS} + P_{PNE}) = 428/(58+428+4) = 0.87

Recall Neutral (R_{NEU}) = T_{NEU} / (P_{NNE} + P_{NEP} + T_{NEU}) = 632/(144+17+632) = 0.80

Recall (R) = (R_{NEG} + R_{POS} + R_{NEU}) / 3 = (0.98+0.87+0.80)/3 = 0.88

Accuracy: It is the measure of correct classifications divided by total classifications.

Accuracy (A) = (T_{NEG} + T_{POS} + T_{NEU})/Total = (1468+428+632)/2775 = 0.91

F-Measure: It is the measure of test's accuracy considering both precision and recall. It is also known by F-Score and F1-Score. It is the harmonic average of precision and recall.

F-Measure (F) = 2 * (P * R)/(P+R) = 2*(0.93*0.88)/(0.93+0.88) = 0.91

Specificity: It is the measure of true negative rate.

Specificity Negative (S_{NEG})= (T_{POS}+P_{PNE}+P_{NEP}+T_{NEU})/(T_{POS}+P_{PNE}+P_{NEP}+T_{NEU}+P_{PON}+P_{NNE}) =1081/1283=0.84

Specificity Positive (S_{POS})= (T_{NEG}+P_{NEN}+P_{NNE}+T_{NEU})/(T_{NEG}+P_{NEN}+P_{NNE}+T_{NEU}+P_{NPO}+P_{NEP}) =2250/2285=0.98

Specificity Neutral (S_{NEU}) = (T_{NEG}+P_{NPO}+P_{PON}+T_{POS})/(T_{NEG}+P_{NPO}+P_{PON}+T_{POS}+P_{NEN}+P_{PNE}) =1972/1982=0.99

Specificity (S) = (S_{NEG} + S_{POS} + S_{NEU}) / 3 = (0.84+0.98+0.99)/3 = 0.94

Error-rate: It is the measure of faultiness/ incorrect classifications divided by the total classifications.

Error-rate (E) = (1-Accuracy) = 1-0.91 = 0.09

The other 2 datasets we experimented has 498(Dataset2) and 997(Dataset3) records respectively. We herewith give the summary of our results we got using Dataset2 and Dataset3 along with performance evaluation results.

Table 4: Sentiment Table with Actual versus Predicted on Dataset2.

Sentiment	Actual	Predicted
[Negative]	32	29
[Positive]	37	36
[Neutral]	31	35

Table 5: Confusion Matrix of Dataset2.

Actual/ Predicted	Predicted			
Actual	Sentiment	[Negative]	[Positive]	[Neutral]
	[Negative]	26	2	4
	[Positive]	1	30	6
	[Neutral]	2	4	25

Table 6: Sentiment Table with Actual versus Predicted on Dataset3.

Sentiment	Actual	Predicted
[Negative]	63	70
[Positive]	79	76
[Neutral]	58	54

Table 7: Confusion Matrix of Dataset3.

Actual/ Predicted	Predicted			
Actual	Sentiment	[Negative]	[Positive]	[Neutral]
	[Negative]	55	2	6
	[Positive]	4	71	4
	[Neutral]	11	3	44

Table 8: Performance evaluation of three Datasets.

Metric/Dataset	Dataset1	Dataset2	Dataset3
Precision	0.93	0.81	0.84
Recall	0.88	0.81	0.84
Accuracy	0.91	0.81	0.85
F-Measure	0.91	0.81	0.84
Specificity	0.94	0.91	0.93
Error Rate	0.09	0.19	0.15

Fig. 2 represents the accuracy comparison on three datasets what we used in our research. The accuracy of the classifier is directly proportional to size of the dataset.

All the three datasets are different and the dataset-2 and dataset-3, which we considered, have more of a lol (Laughing out Loud) type of conversations with many short types of words that are out of the box type to understand even for preprocessing. That is the reason they are misclassified more compared with our Dataset-1 due to which the accuracy has fallen down when compared with dataset-1.

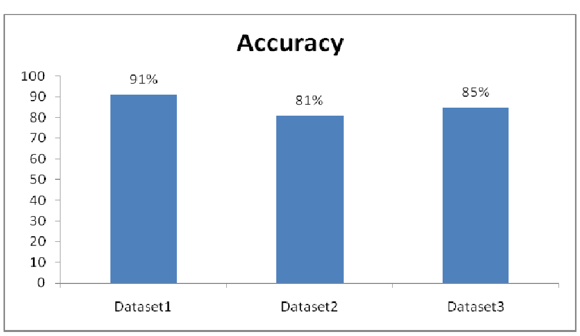


Fig. 2. Accuracy of Dataset1, Dataset2 and Dataset3.

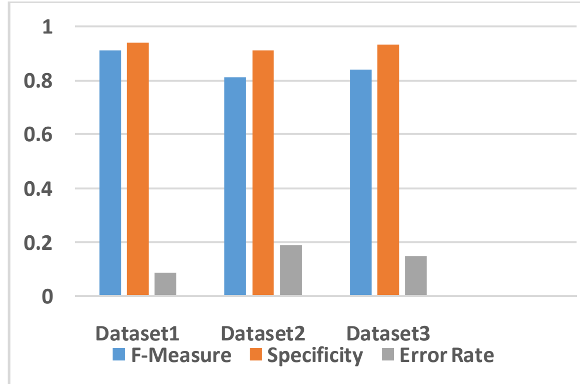


Fig. 3. Evaluation metrics of Dataset1, Dataset2 and Dataset3.

Fig. 3 depicts the evaluation parameters of three datasets for F-Measure, Specificity and error-metric.

Fig. 4 depicts the accuracy comparison of our experiment with Literature Survey experimental results. Literature Survey coded as L1, L2, L3, L4 and L5 respectively in the order they appear. This chart is presented only to show that our experiment have produced better accuracy and not to point any mistake in their work.

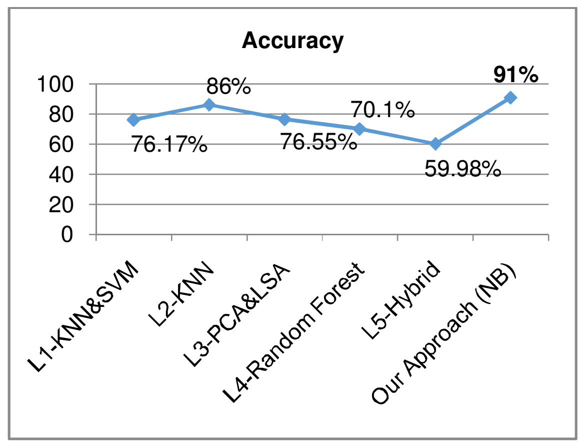


Fig. 4. Accuracy for Existing vs Proposed.

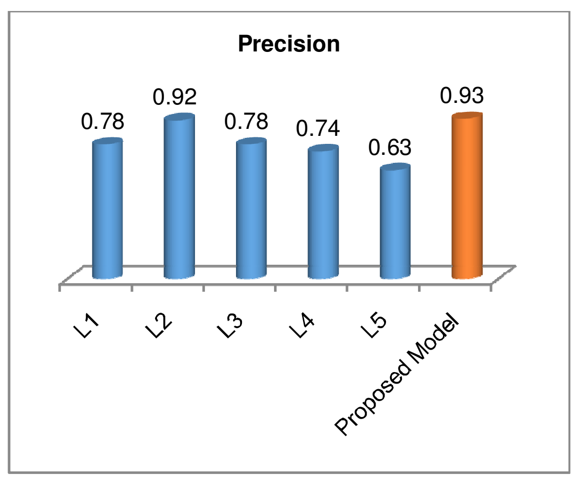


Fig. 5. Precision Comparison for Existing vs Proposed.

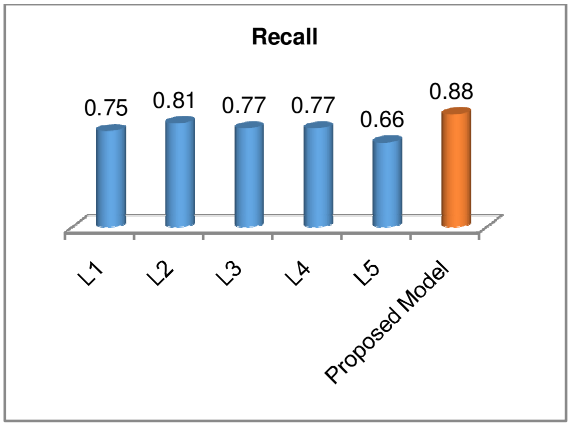


Fig. 6. Recall Comparison for Existing vs Proposed.

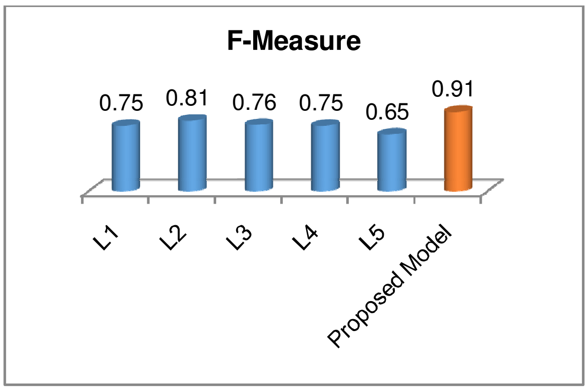


Fig. 7. F-Measure Comparison for Existing vs Proposed.

V. CONCLUSION

In this paper, we presented our approach of using Multinomial Naïve Bayes classifier using our methodology on three different datasets, which we named as Dataset1, Dataset2, and Dataset3 respectively and achieved an accuracy of 91%, 81% and 85% using ternary classification, which is a better one in terms of any of the existing studies that we referred on ternary classification. With our experiment, we conclude that as the dataset size increases the accuracy improves. We also experimented with varying train set and test set sizes. However, we got better results when considered 20% as test set size on the three datasets we experimented with.

VI. FUTURE SCOPE

We will continue our research on various other datasets and we try to add more feature processing mechanisms with improving accuracies on varied test sizes as well. In future, we would like to extend our work with multi classes than just three and come up with better and optimized accuracies.

Conflict of Interest. The authors declare no conflicts of interest

REFERENCES

[1]. Suresh, H. (2019). An innovative and efficient method for Twitter sentiment analysis. *International Journal of Data Mining, Modelling and Management*, *11*(1), 1-18.
[2]. Kumar, A., Sharma, S. and Singh, D. (2019). Sentiment Analysis on Twitter Data using a Hybrid Approach. *International Journal of Computer Sciences and Engineering*, *7*(5), 906-911.
[3]. Medhat, W., Hassan, A., & Korashy, H. (2014). Sentiment analysis algorithms and applications: A survey. *Ain Shams engineering journal*, *5*(4), 1093-1113.
[4]. Ahad, A., Yalavarthi, S. B., & Hussain, A. (2018). Multi-Level Tweets Classification and Mining using Machine Learning Approach. *Journal of Engineering and Applied Sciences, 13*(11), 3907-3915.
[5]. Hota, S., & Phatak, S. (2018). KNN classifier-based approach for multi-class sentiment analysis of twitter data. *International Journal of Engineering and Technology*.
[6]. Zainuddin, N., Selamat, A., & Ibrahim, R. (2018). Hybrid sentiment classification on twitter aspect-based sentiment analysis. *Applied Intelligence*, *48*(5), 1218-1232.
[7]. Bouazizi, M., & Ohtsuki, T. (2017). A pattern-based approach for multi-class sentiment analysis in twitter. *IEEE Access*, *5*, 20617-20639.
[8]. Lalji, T and Deshmukh, S, (2016). Twitter Sentiment Analysis Using Hybrid Approach. *International Research Journal of Engineering and Technology*, *3*(6), pp. 2887-90.
[9]. Lourdusamy, R., & Abraham, S. (2018). A Survey on Text Pre-processing Techniques and Tools. *International Journal of Computer Sciences and Engineering Open Access Survey Paper*, *6*.
[10]. Jianqiang, Z., & Xiaolin, G. (2017). Comparison research on text pre-processing methods on twitter sentiment analysis. *IEEE Access*, *5*, 2870-2879.
[11]. Vijayarani, S., Ilamathi, M. J., & Nithya, M. (2015). Preprocessing techniques for text mining-an overview. *International Journal of Computer Science & Communication Networks*, *5*(1), 7-16.
[12]. Sánchez-Mirabal, P. A., Torres, Y. R., Alvarado, S. H., Gutiérrez, Y., Montoyo, A., & Muñoz, R. (2014). UMCC DLSI: Sentiment Analysis in Twitter using Polirity Lexicons and Tweet Similarity. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)* (pp. 727-731).
[13]. Ahuja, R., Chug, A., Kohli, S., Gupta, S., & Ahuja, P. (2019). The impact of features extraction on the sentiment analysis. *Procedia Computer Science*, *152*, 341-348.
[14]. Shirbhate, A. G., & Deshmukh, S. N. (2016). Feature extraction for sentiment classification on twitter data. *International Journal of Science and Research*, *5*(2), 2183-2189.
[15]. Rahate, R. S., & Emmanuel, M. (2013). Feature selection for sentiment analysis by using SVM. *International Journal of Computer Applications*, *84*(5), 24-32.
[16]. Kübler, S., Liu, C., & Sayyed, Z. A. (2018). To use or not to use: Feature selection for sentiment analysis of highly imbalanced data. *Natural Language Engineering*, *24*(1), 3-37.
[17]. Ahmad, S. R., Bakar, A. A., & Yaakub, M. R. (2019). A review of feature selection techniques in sentiment analysis. *Intelligent Data Analysis*, *23*(1), 159-189. [18]. Bouazizi, M., & Ohtsuki, T. (2019). Multi-class sentiment analysis on twitter: Classification performance and challenges. *Big Data Mining and Analytics*, *2*(3), 181-194.

How to cite this article: Kannan, E. and Kothamasu, L. A. (2020). A Pattern based Approach for Sentiment Analysis using Ternary Classification on Twitter Data. *International Journal on Emerging Technologies*, *11*(2): 811–816.