Emotion Detection of Twitter Post using Multinomial Naive Bayes

Nazia Anjum Sharupa[1], Minhaz Rahman[2], Nasif Alvi[3], M. Raihan[4], Afsana Islam[5] and Tanzil Raihan[6]
Department of Computer Science and Engineering, North Western University, Khulna, Bangladesh[1-6]
Computer Science and Engineering Discipline, Khulna University, Khulna, Bangladesh[3]
Emails: nasxotic@gmail.com[1], minhaz.rtk@gmail.com[2], nasif.cse12@gmail.com[3], raihanbme@gmail.com[4], mraihan@nwu.edu.bd[4], afsanatultuli@gmail.com[5] and tanzilraihan1995@gmail.com[6]

***Abstract*—In research fields, emotion analysis and opinion mining using data from different platforms are up burning field. In this paper, we tried to represent sentiment of twitter data on core text. But tweets can only be in 140 characters, with lots of noise. Tweets contain few words which is in short forms, ambiguous and noisy, so it is hard to figure out the user's sentiments. So, it becomes very difficult to have the right opinion with these noise and short forms of words. The main job is to preprocess the data and then extract the features from there. But preprocessing demands, different theories, methods, steps which always varies. Our goal is to improve the outcomes using Naive Bayes classifier and an almost a good trained data set. Finally, we have our average accuracy for happy class 60%, surprise class 61%, relief class it is 71% and worry class has the highest 81%, by using unigram model for preprocessing. On the other hand, using unigram with POS tag model we have average accuracy of 63% same for happy and surprise class, 72% for relief and 83% for worry class.**

***Keywords*—Sentiment Analysis, Data Preprocessing, Naive Bayes, Unigram, POS tag, Stop words, Emotions, Feature extraction.**

I. Introduction

In the era of the online function, every topic is discussed and mentioned in various ways. In day to day life people are tweeting publishing and even sharing their thoughts. That's what makes microblogging so popular and faster way for these facts [1]. Sentiment analysis is known as interpretation of human thoughts. The classification of emotions are like positive, negative and neutral within text data can be determined using text analysis techniques. The users of twitter in monthly basis are millions from first quarter of 2010 to third quarter of 2016 where each quarter represents 4 months of a year [2]. When people are sharing their opinion and sentiment towards the society or any specific sector data analyst are getting options for an evaluation. Sentiment analysis, emotion detections, facial expression recognition etc. are burning sectors are to be done now a days. By keeping this in mind, our thesis paper is based on sentiment analysis of tweeter data. We have only analyzed row test. Because as we know, tweets have to 140 characters. Too small size for sentiment expression. So, people are using short form of words, emoticons, acronym etc. which are not grammatically right but with the trend. We have classified tweets in four classes like happy, relief, surprised, worry. We have tried to classify the basic and mostly used sentiments of tweets. In this research, we have applied two models like unigram and unigram with pos tags which is for feature extraction and used a classifier called Naive Bayes for analysis emotions.

Another part of the article is arranged as follows: in section II and III, the related works and methodology have been elaborated. In section IV the outcome of this analysis has been discussed with the impulsion to justify the significance of this exploration. Finally, this research paper is resolved with section V.

II. Related Works

The authors of this paper Abu Zonayed et al. [3] explained multiple theories for emotion analysis of core text from tweets. They had basic emotions surprise, neutral sadness, disgust and happiness. They have chosen unigram model, POS tag model and used NLTK. They have obtained 81% accuracy in Unigram and 79.5% in Unigram with POS tagging. Nikolaos et al. [4] they proposed a distributed algorithm which is implemented in Spark. In their paper sentiment classification framework are done with the combinations of Pattern Features, Punctuation Features, Word and N-Gram Features. The level of time reduction reaches 17%. In this paper Komalet et al. [5] they showed a way of analysis twitter data by using Hadoop. In this paper POS detects any of these terms like verb, adjective or a noun. Then they figured out the POS from given preprocessed data by applying Stanford NLP. In this paper Muqtar et al. [6] they showed opinion mining and clustered the data like reviews, blogs, comments, articles and so on which is usually user generated. They discussed systems like featuring, clustering that are data extraction, data cleaning and normalization, spectral clustering, K-means clustering, feature selection, feature vector, Hierarchical clustering. In the paper Bholane et al. [7] they proposed sentiment analysis of twitter data using SentiStrength support vector machine (SVM) and Twitter Sentiment. Individually 62.3% and 57.2% respectively. With SVM they increased the accuracy to 23.24%. In this paper Pak et al. [8] they have done linguistic types of analysis of corpus and clarified a system for building a classifier for sentiment. They had sentiment classifier using multinomial Naive Bayes classifier. Naive Bayes classifier performed better for achieving best accuracy. In case of accuracy bigrams performed better than unigram and trigrams. The authors of this paper Agrawal et al. [9] works with two ways classification. Binary classification like a 3-way classification of positive, negative and neutral and 2-way classification of positive and negative classes. They obtain a minimum accuracy of 75.1% and 75.39% maximum with the combination of Unigram model and Senti-features.

11th ICCCNT 2020
July 1-3, 2020 - IIT - Kharagpur
Kharagpur, India

Purver et al. [10] mainly worked with twitter data on hashtags and emoticons. In their paper, they showed six basic emotions and explained hashtags and emoticons. After completing the survey by 492 individuals, they got accuracy of nearly 50-80% which technically varies by experiments. Deebha et al. [11] proposed a lexicon-based technique which consists of a collection of negation words, negative and positive. They judge a whole sentence by a single word. Ex. Good, Bad, Okay etc. Zhao et al. [12] discussed neural network model for tweets with negative or positive sentiments. They used a BoW-SVM model, BoW-LR model and GloVe-SVM model and obtain 66.49%, 74.12% and 81.95% accuracy respectively.

In the previous works, emotions like sadness, happiness and surprise are mentioned in majority. As it is cleared that these and neutral are the most common emotions which are experimented with. Besides the Naive Bayes theory, unigram and unigram with POS tags models are used in several times for good outcomes. In that case for a new paper, there is a need of new or some unique emotions to work with. Also, Naive Bayes theorem can be used for this kind of need. By keep this mind, our research has come up with new emotions like worry and relief including happiness and surprise parallelly. As it is mentioned earlier, these theories come with good results and so we have used the unigram and unigram with POS tags models along with Naive Bayes theorem with good percentages of accuracy.

III. Methodology

Most of the research is about 3-way classifications in previous years. We are working on 4-way classifications. Now, this is the discussion about our framework on emotions analysis using tweets. Fig. 1 is our proposed framework on emotion analysis.

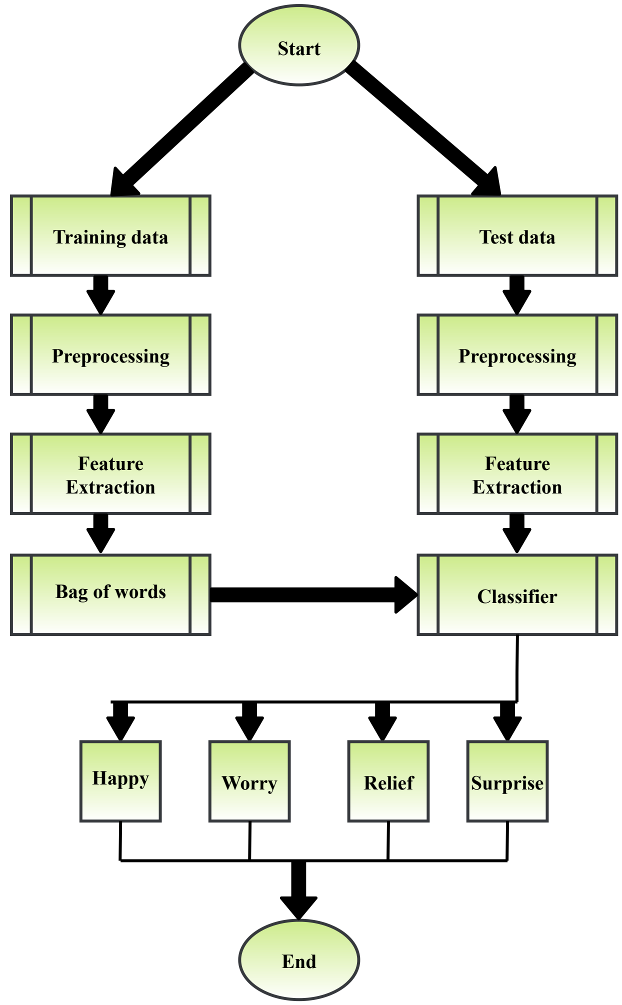

Fig. 1: System Architecture

A. Data Preparation

Nowadays, emotions analysis on tweets is an interesting field for researches and there are available data online. Twitter data can be used from Twitter APIs. As the tweets are too noisy so it has to processed before starting implementation. More than 1600000 tweets together having three polarities like neutral, negative and positive is sentiment140 which provides human labeled corpus. We have labeled manually after using this site.

B. Preprocessing

In data preparation of getting that label data, we have pre-processed our data. Fig. 2 represents the steps of preprocessing of noisy data set.

1) Acronym Expansion: We have generated a dictionary where we kept acronyms and used their English expansion along with. In social media, some mostly used acronyms were collected [13]. Tweets cannot exceed the limit of 140 characters, users use short form to represent their opinion.

2) Removing Emoticons and Symbols: In preprocessing we have removed the emoticons and different kind of symbols used in the tweets. We want to identify the emotion of a tweet by using only the text. For example,'no new good news for 3 months. *sniffs* I'm going to be so worried ☹ ', here the symbol '*', '*'will be removed and the emoticon '☹'will also be deleted.

3) Word Correction: We have seen in tweets there are some repeated alphabets like 'niceeeeeeeeeeeeeee', 'thanksssssssssssss', 'gooddddddddddd', etc. To correct the words like these we have converted the repeated sequence of characters into two characters like 'nice', 'thanks', 'good'.

4) Punctuation Removal: In preprocessing of this step, we can remove the punctuation marks. Because any sentiment or emotion does not represent punctuation marks in a text. For example: full stop '.', '?', '!', ':', '*'etc.

5) Hashtag Removal: By clicking the # symbol named hashtag, users can view and experience other tweets or points of viewers which contains the same type keywords or topic [14]. In the hashtag removal process all hashtags marked with the sign (#) in front of unspaced phrase are removed from the entire text.

6) @username Removal: In tweeter medium, when we want to mention another usernames, the @ sign is used [14] . Then any kind of emotions do not mention by these usernames. Then these usernames had removed.

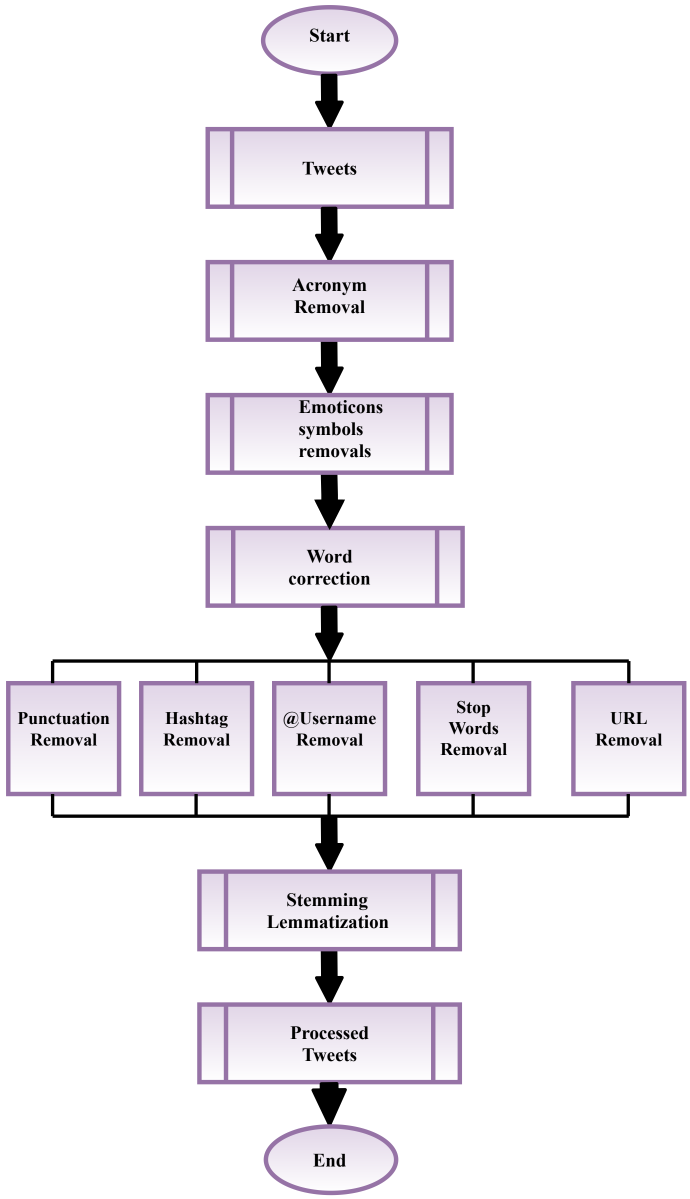

Fig. 2: Block Diagram of Preprocessing

7) Stop Words Removal: There is no limitation of stop words. 153 stop words defined in python. That stop words are **their**, **that**, **with**, **into**, etc. We can remove this stop words from our tweets.

8) Uniform Resource Locator (URL) Removal: Nowadays user also post URL or link along with their tweets. That will be like @username these links do not represent any sentiment or opinion. That's why we can remove all the links or URLs from tweets.

9) Stemming and Lemmatization: By reducing inflectional forms lemmatization and stemming is used to find out the raw or root form of the marked words. They both generate the root form of the inflected words. For example, **book**, **books**, **book's** these all words will be reduced into **book**.

TABLE I: Some POS Tags from feature extraction.

Short Form	Full Form
NN	Proper singular Noun
VB	Verb in base form
VBP	Verb in present tense as not third person singular
VBG	Verb in present participle or gerund
JJR	Adjective as comparative
JJS	Adjective as superlative
UH	Interjection
RBR	Adverb as comparative

C. Unigram Model

In Unigram we assume that the occurrence of each word is independent of its previous word. Hence each word becomes a gram(feature) here. A unigram model in known as the combination of several one-state finite automata. It splits different terms based on probabilities in a text. [3] Like this:

$${}_S P\left(x_1, x_2 \cdots x_n\right) = P\left(x_1\right) P\left(x_2 \mid x_1\right) \cdots P\left(x_n \mid x_{n-1} \cdots x_1\right)$$

Unigram model simply throws away all context with conditionings and estimate each term individually like this:

$$P_{uni}\left(x_1, x_2 \cdots x_n\right) = P\left(x_1\right) P\left(x_2\right) \cdots P\left(x_n\right)$$

Many researches have proven the better efficiency and great performance of unigram model in emotion analysis fields. This the reason why we have chosen this model for our feature extraction part of implementation.

D. Unigram Model along with POS tagging

POS tagging is the process of naming words based on its definition according to part of a speech. To annotate each word with its corresponding POS tag, we have used unigram model and Natural Language Tool Kit (NLTK), which is a library and built in program for symbolic and statistical Natural Language Processing (NLP) written for Python (programming language) [3][17]. Table I is the POS tags from our feature extraction:

1) Bag-of-Words Model: This model is simply representing what it is called. In this model, such as a sentence is symbolized as the bag of its words. So, this Bag-of-Words model turns every sentence into unsorted list of words. We stored these words as the features for the classifier. [3] For example with a simple text:

'Working so hard with hope'

The unigram model will convert the sentence into following list of words:

[**'Working'**, **'so'**, **'hard'**, **'with'**, **'hope'**]

E. Naive Bayes Classifier

Naive Bayes is one of the simple models which performs well in text classification. Pang et al. [15], Pak and Paroubek [8] have showed the better performance of Naive Bayes classifier in sentiment analysis as well as text classification.

Let's say, when document= **d** and also class= **c** then Naive Bayes classifier will be like:

$$c_{MAP} = \arg\max c\epsilon CP\left(c \mid d\right)$$

where, ('maximum posteriori') MAP = mostly class. features = x1,x2,x3,..,xn of Document 'd'. let's say, Feature probabilities P(xi—c) are independent and P(xi—c) given 'c'.

$$= \arg\max c\epsilon CP\left(d \mid c\right)P\left(c\right)P\left(d\right)$$

$$= \arg\max c\epsilon CP\left(d \mid c\right)P\left(c\right)$$

because, P(d) has nothing to do in finding c [4].

$$= \arg\max c\epsilon CP\left(x1, x2, x3 \cdots xn \mid c\right)P\left(c\right)$$

$$P\left(x1, x2 \cdots xn \mid c\right) = P\left(x1 \mid c\right)P\left(x2 \mid c\right)\cdots P\left(xn \mid c\right)$$

$$CNB = \arg\max c\epsilon CP\left(c\right)\Pi P\left(x \mid c\right)x\epsilon X$$

1) Multinomial Naive Bayes: A upgrade version of Naive Bayes and it captures frequency of given words information from documents. By using the frequencies from the data,

$$P\left(xi \mid cj\right) = count\left(xi, cj\right)\sum count\left(x, cj\right)w\epsilon V$$

where, v is the vocabulary.

2) Laplace (add-1) Smoothing: Zero probabilities should not be considered, whatever the other evidence can be if there is no training document with the certain word *xi* then the calculation will be an error in Naive Bayes [15]. So, both in presence or absence of the particular words, multinomial Naive Bayes manipulates the word counts and add one smoothing to adjusts the underlying calculations. This technique is known as the Laplacian correction or Laplace estimator and also add one smoothing [3].

$$P\left(xi \mid c\right) = count\left(xi, c\right) + 1\sum\left(count\left(x, c\right) + 1\right)w\epsilon V$$

F. Dataset

We had 40000 tweets using Sentiment140 website. Then we selected total 3200 and where 800 tweets each for classes happy, surprise, relief, worry. Now we have trained data. Then, we have 800 as test data and 2400 tweets as training data. Using unigram with POS tagging, we had total 91243 individual words.

$$|V| = 91243$$

99033 is the number of terms contained in vocabulary V. Using unigram method, we had total 5006 individual words.

$$|V| = 99033$$

TABLE II: Average accuracy (4-way classification)

Feature Extraction Model	Accuracy				Average Accuracy
	Happy	Surprise	Relief	Worry	
Unigram	60%	61%	71%	81%	**68.25%**
Unigram with POS tag	63%	63%	72%	83%	**70.25%**

TABLE III: Precision, recall and F-score for unigram

Class	Precision	Recall	F-score	Accuracy
Happy	62%	80%	69.8%	67.5%
Worry	81%	61.1%	69.6%	
Relief	71%	64.6%	67.6%	
Surprise	56%	70%	62.2%	

IV. Outcomes

We have tested a 4-way classification (happy, surprise, relief and worry) with a group of data. For each of the classification experimental results given below.

A. 4-way Classification

Table II shows the accuracy of our both models of 4-way classification like happy, worry, relief, surprise.

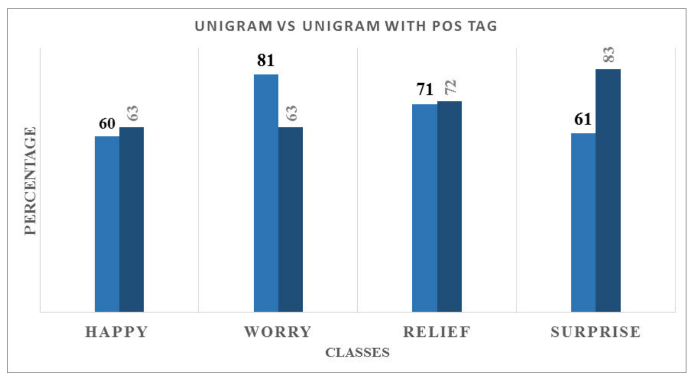

Fig. 3: Graphical Representation of Results (Unigram vs Unigram with POS Tags)

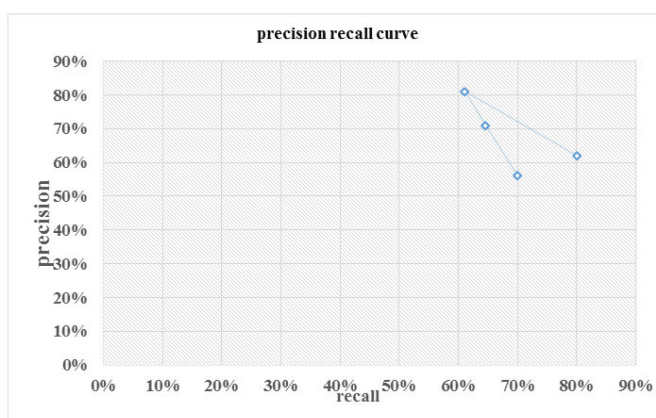

Fig. 4: Graphical Representation of Precision Recall curve (Unigram model))

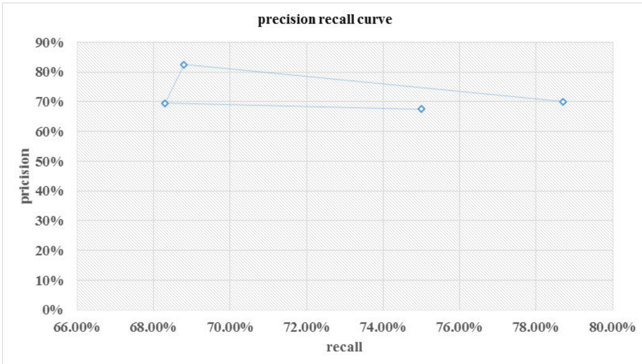

Fig. 5: Graphical Representation of Precision Recall curve (Unigram with POS Tags))

TABLE IV: Precision, Recall and F-score for 4-way classification for unigram with pos tag

Class	Precision	Recall	F-score	Accuracy
Happy	70%	78.7%	74.1%	72.3%
Worry	82.5%	68.8%	75%	
Relief	69.5%	68.3%	69%	
Surprise	67.5%	75%	71.1%	

B. Discussions

We have experimented with Naive Bayes algorithm and extracted the features for 4-way classification. Fig. 3 represents the accuracy of unigram and represents the accuracy of unigram with POS tags. We have classified sentiment into happy, worry, relief and surprise. We have considered the highest caring value belongs to that defined class. If the algorithm finds out the document is caring higher value in **happy** class then the document will count as happy sentiment. After that, we have used PR (precision recall curve) curve. The recall and precision have been calculated for those values. Here, that takes true values as input and probabilities of positive class as output. The recall-precision values as a curve showed in Fig. 4 and 5 curve.

Precision-Recall curve turns out useful measure for success prediction of our four classes. A recall-precision curve is done by plotting the precision in y-axis and the recall in x-axis using the values from the table. The high area represents high recall and high precision in the curve. Our PR curve created by TP/(TP+FN) and TP/(TP+FP) on y-axis and on x-axis respectively.

We have our average accuracy on unigram model 68.25% and unigram with POS tag model 70.25%. Accuracy comparison of our both approaches has shown in Fig. 3. First, we have preprocessed our data set which was simply twitter posts. Then we have manually labeled some data and ready train data set and test data set. We have applied unigram with POS tags and unigram model for preprocessing our data set. After that we have used these raw and tokenized data set for classification using Naive Bayes. Through this procedure we have reached on Table II. Then we calculated precision, recall, f-score and accuracy in Table III for unigram and Table IV for unigram with POS tags. In the part of Precision, it has showed how much good the model is at predicting positive class. So, Precision can also be known as the positive predictive value. Recall is the ratio of true positives which is divided by sum of the true positives and false negatives. Recall is the number of positive class predictions in our dataset. F-score shows single score to do the balance of both the concerns of precision and recall.

The major objectives of our research, collecting tweets data and then manually labeling those. Preparing training data set and preprocess data using unigram and unigram with POS tag. Extracting features and then use Naive Bayes. Classify in 4-way Determine whether a tweet represents happiness, worry, surprise or relief. Showing the accuracy and to compare the accuracy of different models.

V. Conclusion

Now-a-days microblogging sites have solid effect on personal to political lives. It has been a part of our daily life. So, sentiment analysis is a burning demand and rising issue. So, we wanted to do emotion analysis for twitter data using Naive Bayes classifier. We have our average accuracy for happy class which is 60%, for surprise class we have 61%, for relief class it is 71% and worry class has the highest 81%, by using unigram model for preprocessing. On the other hand, using unigram with POS tag model we have average accuracy of 63% same for happy and surprise class, 72% for relief and 83% for worry class. Besides we have calculated precision, recall, f-score and accuracy where we have accuracy of 67.5% by using unigram model and 72.25% by using unigram with POS tag model which is nearly equal to average accuracy. It would be better if we could do the experiment with a huge training data. Another problem is to find the correct emotion. This limitation will be solved in future. Our research shows that if we can use two or more algorithms combined and train more data, the results will be more satisfying. Comparing with similar researches, we can assure there is not much difference in our result from theirs using core text. It would be better if we could do the experiment with a huge training data and find more than two unique emotions. Another problem is to find the exact correct emotion. Worry and relief kinds of emotions are opposite to each other but happy and surprised are nearly equal as one can be happy to be surprised. This kind of situations weren't handled well in our paper. Our research shows that if we can use two or more algorithms combined and train more data, the results will be more satisfying. We could use more emotions to find out then. This limitation can be solved in future work. Comparing with similar researches, we can assure there is not much difference in accuracy of our results from theirs using core text.

References

[1] "Number of monthly active Twitter users worldwide from 1st quarter 2010 to 3rd quarter 2016 (in millions)", [Online]. Available:http://www.statista.com/statistics/282087/number-of-monthly-active-twitter-users/ [Accessed 07 July 2019].

[2] "Twitter: number of active users 2010-2019 — Statista", *Statista*, 2019. [Online]. Available: https://www.statista.com/statistics/282087/number-of-monthly-active-twitter-users/. [Accessed: 09 July 2019].

[3] A. Riyadh, N. Alvi and K. Talukder, "Exploring Human Emotion Via Twitter"in *20th International Conference on Computer and Information Technology (ICCIT)*, 2017.

[4] N. Nodarakis, S. Sioutas, A. Tsakalidis and G. Tzimas, “Large Scale Sentiment Analysis on Twitter with Spark”in *Proceedings of the EDBT/ICDT 2016 Joint Conference*, 2016.

[5] K. Sutar, S. Kasab, S. Kindare and P. Dhule, “Sentiment Analysis: Opinion Mining of Positive, Negative or Neutral Twitter Data Using Hadoop”, *IJCSN International Journal of Computer Science and Network*, vol. 5, no. 1, 2016.

[6] M. Unnisa, A. Ameen and S. Raziuddin, “Opinion Mining on Twitter Data using Unsupervised Learning Technique”, *International Journal of Computer Applications*, vol. 148, no. 12, 2016.

[7] B. Savita D and P. Gore, “Sentimental Analysis on Twitter Data Using Support Vector Mechine”, *International Journal of Computer Science Trends and Technology (IJCST)*, vol. 4, no. 3, 2016.

[8] A. Pak and P. Paroubek, “Twitter as a Corpus for Sentiment Analysis and Opinion Mining”in *International Conference on Language Resources and Evaluation*, vol. 10, no. 2010.

[9] A. Agarwal, B. Xie, I. Vovsha, O. Rambow and R. Passonneau, “Sentiment Analysis of Twitter Data”, in *Proceedings of the Workshop on Language in Social Media (LSM 2011) Association for Computational Linguistics*, 2011, pp. 3038, 2011.

[10] M. Purver and S. Battersby, “Experimenting with Distant Supervision for Emotion Classification”in *13th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 482491, 2012.

[11] D. Mumtaz and B. Ahuja, “A Lexical Approach for Opinion Mining in Twitter”, *International Journal of Education and Management Engineering*, vol. 6, no. 4, pp. 20-29, 2016.

[12] Z. Jianqiang, G. Xiaolin and Z. Xuejun, “Deep Convolution Neural Networks for Twitter Sentiment Analysis”, *IEEE Access*, vol. 6, pp. 23253-23260, 2018.

[13] R. Menaha and V. Jayanthi, “A Survey on AcronymExpansion Mining Approaches from Text and Web”, *Smart Intelligent Computing and Applications*, pp. 121-133, 2018. Available: 10.1007/978-981-13-1921-1_12.

[14] “Glossary”, *Help.twitter.com*, 2020. [Online]. Available: https://help.twitter.com/en/glossary. [Accessed 18 Sep. 2019].

[15] Pang, Bo, Lillian Lee, and Shivakumar Vaithyanathan. “Thumbs up? sentiment classification using machine learning techniques”in *Proceedings of the ACL-02 conference on Empirical methods in natural language processing*, Vol. 10, pp. 79-86.

[16] “NLTK Book”, *Nltk.org*, 2020. [Online]. Available: https://www.nltk.org/book/. [Accessed: 09- Mar- 2020].