Received Date: 27/12/2019 Acceptance Date: 08/01/2020 Publication Date: 31/01/2020

Sentiment Analysis Using Multi Class Classification to Acronyms

[1]Mr. Ganesh T. Mahalkar, [2]Prof. P.D. Sathya, [3]Dr.S. Jagdish Kumar

[1]M.E Student, Department of Computer Sci. & Engineering, SYCET Aurangabad

[2]Associate Professor and HOD, Department of Computer Sci. & Engineering, SYCET Aurangabad

[3]Associate Professor, Department of Computer Sci. & Engineering, SYCET Aurangabad
Email – [1] ganeshmahalkar87@gmail.com

Abstract: *Growth in the area of different multi-classification and sentiment analysis has been rapid and aims to explore the opinions or text present on different platforms of social media through support vector machine-learning techniques with sentiment, personal analysis or polarity calculations. Sarcasm is a sort of sentiment where public expresses their negative emotions using positive word within the text. It is very tough for humans to acknowledge. In this way we show the interest in sarcasm detection of social media text, particularly in tweets [22] .We classify the tweets into 6 different classes; however the approach can be run to classify into more classes [1].Experiments show that our approach reaches an accuracy of classification equal to 65% and a precision level of sentimental tweets (other than neutral and sarcastic) equal to 72.58%. Nevertheless, the approach proves to be very accurate in binary classification (i.e., classification into "positive" and "negative") and ternary classification (i.e., classification into "positive", "negative" and "neutral"): in the former case, we reach an accuracy of 87.5% for the same dataset used after removing neutral tweets, and in the latter case, we reached an accuracy of classification of 83.0%. [1]To deal with these challenges, the contribution of this paper includes the approval of a hybrid approach that involves a sentiment analyzer that includes machine learning. Moreover, this paper also provides a comparison of techniques of sentiment analysis in the analysis of when the user uses the shortcut or acronyms like "ILU, 143(happy)*, *KMN (kill me now)*, *MYOB (mind your own business)*, *B3 (blah, blah, blah)*"*the machine unable to perform the classification, so in Proposed system may be able to machine learning algorithm can classify the acronyms.*

Key Words: *Twitter, sentiment analyzer, support vector machine etc.*

1. INTRODUCTION:

By social media activity to the presence of large amount of data available on web sites, various cooperative started taking interest in this as opinion mining this information can be very valuable to them. Sarcasm is a sort of sentiment where public expresses their negative emotions using positive word within the text. It is very tough for humans to acknowledge. In this way we show the interest in sarcasm detection of social media text, particularly in tweets. [22]. Social net-working websites have become a popular platform for users to express their feelings and opinions on various topics, such as events, or products. Social media channels have become a popular platform to discuss ideas and to interact with people worldwide area. Twitter is also important social media network for people to express their feelings, opinions, and thoughts. Users post more than 340 million tweets and 1.6 billion search queries every day. [22].This gives birth to an entirely different and expansive field of study known as Sentiment Analysis. This ecosystem presents a very rich, source of data to mine. However, due to the limitation in terms of characters (i.e. 140 characters per tweet), mining such data present lower performance than that when mining longer texts. In addition, classification into multiple classes remains a challenging task: binary classification of a text usually relies on the sentiment polarity of its components (i.e., whether they are positive or negative). However, when positive and negative classes are divided into subclasses, the accuracy tends to decrease remarkably.

Various names have given to this field as opinion mining, opinion extraction etc. However there is slight difference in meaning between these various terms. Before automatic mining of sentiments traditional survey techniques were highly biased as they were taken individually by users thus a need of an automatic system arose that can directly deal with hundreds of thousands of opinions hidden in users' posts in the form of reviews, online journal etc. Various applications of sentiment analysis are as in product reviews, movie reviews, business, politics, recommender system etc. Based on the opinion about a product or about different condition of a product, an organization can make changes accordingly. Similarly based on the opinion about a particular political party, government policies' changes can be made accordingly. Two main techniques used for sentiment analysis are machine learning based and lexicon based. Supervised, Unsupervised and Semi-supervised comes under Machine Learning. Supervised approaches e.g.

SVM[3][12][14][15][20][31][33], KNN[22][21], Naive Bayes[4][7][8] etc. requires a good quality training set and thus are highly domain dependent but provide better results if trained properly. Unsupervised approaches e.g. K-Means, Self-Organizing Maps (SOM) etc. do not make use of training set. Semi supervised approaches require partial identify of data and are of two types: a) Transductive Learning b) Inductive Learning Lexicon based approach makes use of dictionary consist of labelled words and with the help of these words, a text is intermediate whether it is subjective or objective[16][23][25]. This approach is further divided into a) Dictionary based which does not take into account the context of word within a text, and b) Corpus based which expands the dictionary with taking associations between different words into account. A complete survey in this field is provided in [9]. This research analyze sentiment of tweets [1][2][3][4][5][8]. As tweets are very unstructured in nature this research converts them into useful information so that better features can be used for machine learning. Hence in this research we provide a good data preprocessing to tweets followed by hybrid classifier. With the help of processed tweets or data features are generated and fed to the two machine learning algorithms KNN and SVM in a hybrid manner. Different feature have tried by authors for improving the results as in [2][3[4][5][18]. when the user uses the shortcut or acronyms like "ILU, 143(happy), KMN (kill me now), MYOB (mind your own business), B3 (blah, blah, blah)"the machine unable to perform the classification, so in Proposed system may be able to machine learning algorithm can classify the acronyms.

Fig. 1: Sentiment Analysis in the Twits

2. MOTIVATIONS AND RELATED WORK:

A. Motivations

Social networks and microblogging websites such as Twitter have been the subject to many studies in the recent few years. Automatic sentiment analysis and opinion mining present a hot topic of study. Sarcasm is a sort of sentiment where public expresses their negative emotions using positive word within the text. It is very tough for humans to acknowledge. In this way we show the interest in sarcasm detection of social media text, particularly in tweets.[22] Social networks present a huge source of data representing the opinions of a significant, yet totally random, proportion of users and customers who are using a product of a service. However, due to the informal language used, the presence of non-textual content and the use of slang words and abbreviations, classification of data extracted from such microblogging websites is rather a challenging task. Gheg et al. [6] defines *"Hidden Sentiment Identification"* which is the identification of the real feeling rather than the sentiment polarity, *"Handling Polysemy"* which is the existence of multiple meanings that might have different sentiment polarity for the same word, and *"Mapping Slangs"* which is the identification of the meaning and the polarity of slang words, among others as the most challenging tasks facing the sentiment analysis of short microblog texts.[1] the machine learning approach to sarcasm detection on Twitter in two languages English and Czech. First work is sarcasm detection on Czech language. They used the two classifier Maximum Entropy (MaxEnt) and Support Vector Machine (SVM) with different combinations of features on both the Czech and English datasets.[22].

On a related context, the state of the art proposed approaches are mostly focusing on the binary and ternary sentiment classification. In other words, they classify texts either into *"positive"* and *"negative"*, or into *"positive"*, *"negative"* and *"neutral"*. However, to study the opinion of a user, it would be more interesting to go deeper in the classification, and detect the sentiment hidden behind his post. Following two examples of tweets which are negative, however, reflect two completely different aspects:

- *"Damn damn.. no iPhone support for windows XP x64. There are some workarounds, but I can't figure this out."*

• "Nooooooooooo! My iPhone glass cracked :("

In the first example, the user is expressing his fury towards the absence of support of his phone on an operating system. However, in the second he is expressing some feeling of sadness because of a physical problem his phone faced. The first example shows some important information regarding the satisfaction of the user, therefore, it might be more important to study. However, in general, both information can be used, yet, they have to be distinguished from each other.

B. Related Work

Twitter data mining has been a hot topic of research in the last few years. Nature of the data mined varies widely depending on the aim and the final result expected. Consequently, the techniques used to process data and extract the needed information are different. Akcora et al. [7] proposed a method to determine the changes in public opinion over the time, and identify the news that led to breakpoints in public opinion. In a related context, Sriram et al. [8] proposed a method to classify tweets depending on their natures into a set of classes including private messages, opinions and event, etc.

However, most of the work has been focusing on the content of the tweets and how to extract opinions of users towards specific topics or objects. The work of Pang et al. [8] presented the pioneer work for the use of machine learning to classify texts based on their sentiment polarity. In their work, the authors used unigrams, bigrams and adjectives in different ways to classify a set of movie reviews into positive or negative. Other works iterated more on the idea, and new types of features have been used for the classification, depending on the aim and application: Boia et al. [9] and Manuel et al. Proposed two approaches that, respectively, rely on emoticons to detect the polarity of tweets and on slang words to assign a sentiment score to online texts. These two works proved how non-textual components can be used to detect the polarity of a text.

More recent works went deeper, and new models have been built: Gao et al. [12] proposed a recent approach that focus in the repartition or the frequency of sentiment classes in the set they analyze. Moving from classification to quantification, the authors concluded that using a quantification-specific algorithm presents a better frequency estimation than using regular classification-oriented algorithms.

Few works have been conducted on the multi-class sentiment analysis. Most of them focused on assessing the sentiment strength into different sentiment strength levels (e.g., *"very negative"*, *"negative"*, *"neutral"*, *"positive"* and *"very positive"*) or simply give numeric sentiment scores to the texts [13] [14]. Nevertheless, other works were conducted to classify texts into different sentiment classes: Lin et al. [15] [16] proposed an approach that classifies documents into reader-emotion categories. They relied on what they qualify of similarity features and word emotion features along with other basic features. The approach, although it shows some potential, is oriented towards the reader rather than the writer.

TABLE I: Structure of the Dataset Used

Class	Training set	Test set
Happiness	3000	300
Love	3000	300
Sadness	3000	300
Hate	3000	300
Sarcasm	3000	300
Neutral	3000	300
Total	18000	1800

3. PROPOSED APPROACH:

A. Problem Statement

Given a set of tweets, we aim to classify each one of them to one of the following 6 classes: *"happiness"*, *"sadness"*, *"love"*, *"hate"*, *"sarcasm"* and *"neutral"*.[1] Therefore, from each tweet, we extract a 4 set of features, refer to a training set and use support vector machine to perform the classification. All so the user uses the shortcut or acronyms like "ILU, 143(happy)**,** KMN (kill me now)**,** MYOB (mind your own business)**,** B3 (blah, blah, blah)**"**the machine unable to perform the classification, so in Proposed system may be able to support vector machine can classify the acronyms

B. Data

For the sake of this work, we manually collected and prepared 2 datasets as follow:

- Set 1: this set contains 18000 tweets which have been manually classified into the 6 classes, each containing 3000 tweets. This set is used for training. Therefore, in the rest of this work, it will be referred to as the *"training set"*.[1]
- Set 2: this set contains 1400 tweets. All tweets are manually checked and classified into the 6 classes. This set will serve as a test set. Therefore, in the rest of this work, it will be referred to as the *"test set"*.

The structure of the dataset used is shown in TABLE I.

Tweeter is the social media network, which is use for communication. Also used for share the opinions for the user throw the tweets. A tweet is collected by using tweeter API. The 1000 tweets are collected. Many current methods for text sentiment analysis contain various preprocessing steps of text. One of the most important goals of preprocessing is to enhance the quality of the data by removing noise. Another point is the reduction of the feature space size.[22]

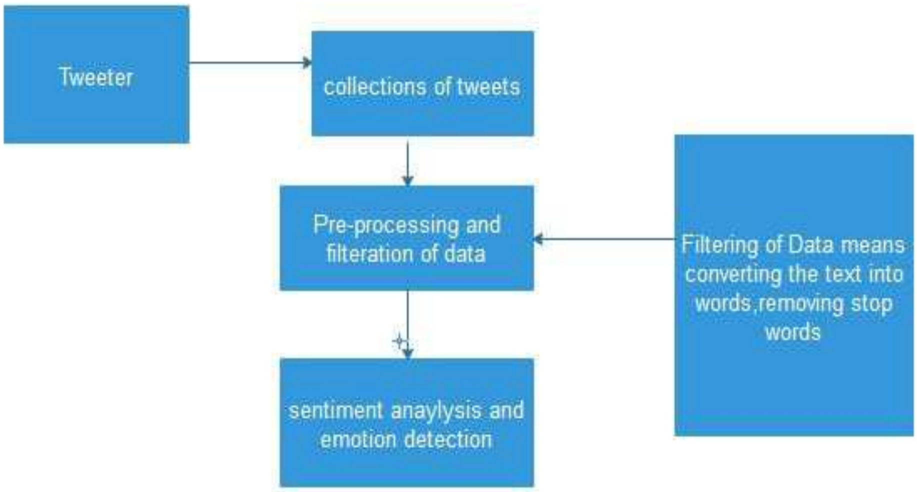

Fig. 2. System Architecture of Behavioral modeling

C. Features Extraction

Under different emotional conditions, humans tend to behave differently. This includes the way they talk and express their feelings. Therefore, it might be important to rely, not only on the vocabularies used, but also on the expressions and sentence structures used under the different conditions, to quantify and model these feelings. Therefore, in the rest of this section, we rely on these assumptions to extract the following four families of features:

1) Sentiment-based features: Sentiment-based features are ones based on the sentiment polarity (i.e., *"positive"*/*"negative"*) of the different components of tweets. We first extract emotional scores from words using SentiStrength. SentiStrength attributes sentiments scores to words, where negative words have scores varying from -1 (almost negative) 5 (extremely negative) and positive words have scores varying from 1 (almost positive) to 5 (extremely positive). We uses SentiStrength to extract the following features:[1]

- Total score of positive words denoted by PW
- Total Score of negative words denoted by NW (this score is positive)
- Number of highly emotional positive words (i.e., having score equal to or more than 3) denoted by N_{pw}
- Number of highly emotional negative words (i.e., having score equal to or less than -3) denoted by N_{nw}
- Ratio of emotional words $\rho(t)$ defined as

$$\rho(t) = \frac{PW(t) - NW(t)}{PW(t) + NW(t)}$$

(1) Where t is the tweet. In case the tweet does not contain any emotional word, ρ is set to 0.

We then add four more features by counting the number of positive, negative, joking (or ironic) and neutral emoticons. Joking emoticons are emoticons used sometimes with ironical or sarcastic statements (e.g., *":P"*). Hashtags also have emotional content. In some cases, they are used to disambiguate the real intention of the twitter conveyed in his message, particularly when he is being sarcastic. Therefore, we count also the number of positive and negative hashtags.

We then define 4 features that represent whether there is a sentiment contrast between the different components. By contrast we mean the coexistence of a negative component and a positive one within the same tweet: we extract such contrast between words, between hashtags, between words and hashtags and between words and emoticons, and use them as extra features.

2) ***Acronyms Based features***: In addition to sentiment-based features for acronyms all so find the accuracy of the result and classify in different classes. User used the shortcut method then classifier are cant classify that opinion so I have used the some acronyms to classify the method in TABLE II.[1]

TABLE II. Acronyms Meaning

Sr.No	Acronyms	Meaning
1	2F4U	Too Fast For You
2	2moro	Tomorrow
3	2nte	Tonight
4	ILU	I Love You
5	GN	Good Night
6	KMN	Kill Me Now
7	B3	Blah,Blah,Blah

3) Punctuation and syntax-based features: In addition to sentiment-based features, we extract a second set of features we qualify of punctuation and syntax-based features. A certain use of punctuation marks, the repetition of vowels or the employment of all-capital words may show how intense the sentiment of the person is. To detect such aspects, we extract the following set of features: number of exclamation marks, number of question marks, number of dots, number of all capital words and number of quotes.
We also add a sixth feature by checking if any of the words contains a vowel that is repeated more than twice (e.g., *"looooove"*). If such a word exists, the feature is set to *"true"*, otherwise, it is set to *"false"*.[1]

4) Unigram-based features: Since proposed by Pang et al. [10], unigrams and *n*-grams in general, have been used as basic features for sentiment analysis using machine learning. In the different approaches, unigrams are collected from the training datasets, and either the count or the presence of these unigrams are used as features for the classification. In our work, we make use of WordNet [20] to collect unigrams related to each sentiment class. We start with a set of seed words few in number for each class, and used WordNet to collect their synonyms and hyponyms down to a certain depth. We start with an initial set of seed words for each class (except the class "neutral"). The words selected are nouns, adjectives and verbs. We then collect the synonyms and hypernyms up to different depths. The number of words for each depth as well as the number/ratio of duplicated terms in different classes. To obtain as much terms as possible, while maintaining a low duplication ratio and keeping in mind that the deeper we go, the more we lose in the original meaning of the word, we set D_{hypo} to 1. The words are associated to the classes described, and are given the absolute value of scores returned by Santi Strength (if a word has a score equal to 0 in Santi Strength, we give it a score equal to 1).

5) Pattern-based features: The idea of our pattern-related features is inspired from our previous work [21], in which we proposed an approach that relies on Part of Speech tags (PoS-tags) to extract sarcastic patterns.[1]

TABLE II: Expressions Used to Replace the Words of EI and GFI

PoS-tag	Expression
"CD"	[CARDINAL]
"FW"	[FOREIGNWORD]
"UH"	[INTERJECTION]
"LS"	[LISTMARKER]
"NN", "NNS", "NNP", "NNPS",	[NOUN]
"PRP", "PRP$"	[INTERJECTION]
"MD"	[MODAL]
"RB", "RBR", "RBS"	[ADVERB]

"VB", "VBD", "VBG", "VBN", "VBP", "VBZ"	[VERB]
"WDT", "WP", "WP$", "WRB"	[WHDETERMINER]
"SYM"	[SYMBOL]

However, instead of dividing words into two categories, we divide them into three: a first one, referred to as *EI*, containing words which might have emotional content, a second one, referred to as *"CI"*, containing non emotional words whose content is important and a third one, referred to as *"GFI"*, containing the words whose grammatical function is important. If a word belongs to the first category, it is replaced by the corresponding expression shown in TABLE II along with its polarity (e.g., the word *"good"* would be replaced by *POS-ADJECTIVE*); if it belongs to the second, it is lemmatized and replaced by its lemma; and if it belongs to the third, it is replaced by the corresponding expression shown in TABLE II.

As mentioned above, the classification into categories is done based on the PoS-tag of the word. The list of part-of-speech tags and their category is given in TABLE III.[1]

TABLE III: Part-of-Speech Tag Categories

Class	PoS Tags
CI	"CC", "DT", "EX", "IN", 'MD", "PDT", "POS", "RB", "RBR", "RBS", "RP", "TO", "WDT", "WP", "WP$", "WRB"
GFI	"CD", "FW", "LS", "NNP", "NNPS", "PRP", "PRP$", "SYM", "UH"
EI	"JJ", "JJR", "JJS", "NN", "NNS", "VB", "VBD", "VBG", "VBN", "VBP", "VBZ"

We generate the vector of words for each tweet as defined. For example, the following PoS-tagged tweet *"He PRP is VBP dummy JJ , , why WP would VBD you PRP think VBP I PRP want VBP to TO go VB with IN him PRP !!!! ."* gives, among others, the following pattern vector [PRONOUN VERB NEG-ADJECTIVE . *why* VERB PRONOUN VERB PRONOUN POS-VERB *to* VERB *with* PRONOUN].[1]

We define a pattern as an ordered sequence of words. Patterns are extracted from the training set such as their lengths satisfy:

$$L_{min} \leq Length(pattern) \leq L_{max} \qquad (2)$$

where L_{min} and L_{max} represent respectively the minimal and maximal allowed length of patterns in *words* and *Length*(*pattern*) is the length of the pattern in *words*. The number of pattern lengths will be referred to $N_{L.}$ $(L_{max}-L_{min}+1)$.[1]

TABLE IV: Pattern Features

		Patterns Length L1 L2……Ln
Sentiment	1	F11 F12……F1n
	-	- - …… -
	-	- - …… -
	-	- - ……. –
Class	6	Fl1 Fl2…….Fln

Only patterns that appear at least N_{occ} times in our training set for the same class are kept; the others are discarded. We then divide the resulted patterns into N_F sets where:

$$N_F = N_L \times N_C \qquad (3)$$

Where N_L is the number of pattern lengths and N_C is the number of classes (7 in our case).

We create N_F features, as shown in TABLE IV. Each feature F_{ij} of the table represents the degree of resemblance of the tweet to the patterns of sentiment class *i* and length *j*. Therefore, given a tweet *t*, we calculate the resemblance degree *res*(*p*,*t*) of each pattern in the training set *p* to the tweet *t* [20]:

$$res(p,t) = \begin{cases} 1, & \text{if the tweet vector contains the pattern as it is, in the same order,} \\ \alpha \cdot n/N, & \text{if } n \text{ words out of the } N \text{ words of the pattern appear in the tweet in the correct order,} \\ 0, & \text{if no word of the pattern appears in the tweet.} \end{cases}$$

Given the *K* patterns that have the highest resemblance to the pattern *p* among the patterns extracted from the class *i* which have a length *j*, the value of the feature F_{ij} is

$$F_{ij} = \beta_j * \sum_{k=1}^{K} res(p_k, t) \quad (4)$$

Where β_j is a weight given to patterns of length L_j (regardless of their class). We give different weights for each length of pattern since longer patterns are more likely to have higher impact. F_{ij} as defined measures the degree of resemblance of a tweet *t* to patterns of class *i* and length *j*.

In our previous work [19], we demonstrated that the optimal values for N_{occ}, L_{min}, L_{max}, α and β_i are[1] as follows:

$$\begin{cases} Nocc = 2, \\ L_{min} = 3,\ L_{max} = 10, \\ \alpha = 0.03,\ \beta_n = (n-1)/(n+1),\ \forall n \in \{3,...,10\}. \end{cases}$$

On the other hand the parameter *K* has been introduced in this work since we noticed a high imbalance between the number of patterns for each class. Fig. 2 shows the classification accuracy using pattern-based features for different values.

4. EXPERIMENTAL RESULTS:

After the extraction of features, we run different test using *"Support Vector Machine"* [21] classifier. We use 4 Key Performance Indicators (KPIs) to evaluate the effectiveness of our approach: Accuracy, Precision, Recall and F-measure which is defined as follows:[1]

$$F\text{-measure.} = 2 \cdot \frac{precision \cdot recall}{precision + recall}$$

1) Accuracy: it represents the overall correctness of classification. In other words, it measures the fraction of all correctly classified instances over the total number of instances.
2) Precision: it represents the fraction of retrieved sarcastic tweets. that are relevant. In other words, it measures the number of tweets that have successfully been classified as sarcastic over the total number of tweets classified as sarcastic.
3) Recall: it represents the fraction of relevant sarcastic tweets that are retrieved. In other words, it measures the number of tweets that have successfully been classified as sarcastic over the total number of sarcastic tweets.
4) F1 score:
F1 =2 * (precision * recall/precision + recall) [22]
5) When the user uses the shortcut or acronyms like "ILU, 143(happy), KMN (kill me now), MYOB (mind your own business), B3 (blah, blah, blah)"the machine unable to perform the classification, so in Proposed system may be able to machine learning algorithm can classify the acronyms.

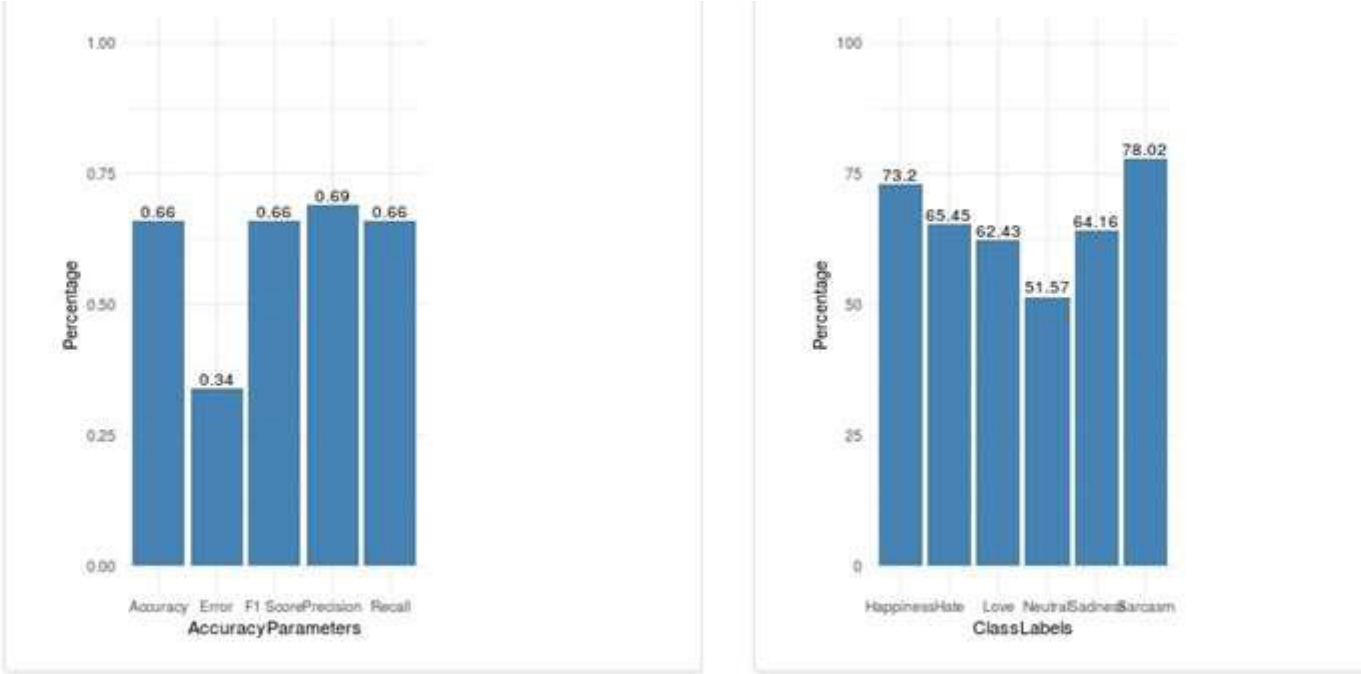

Fig. 3: Accuracy of Classification Using Support Vector Machine.

A. Multi-class Classification

We first run our experiment to detect the sentiment polarity of tweets. For this sake, we remove the tweets belonging to the classes *"neutral"* and *"sarcasm"*, and grouped the other classes into two main classes which are *"positive"* and *"negative"*. The former class contains tweets from the classes *"love"* and *"happiness"*, while the latter contains tweets from the classes *"hate"*, *"anger"* and *"sadness"*. TABLE V shows the results of classification. The accuracy obtained reaches 87.51%. Noticeably, the recall of positive tweets is the highest (i.e., 90.5%), however the precision of negative tweets is the highest (i.e., 92.2%). This means that tweets which are classified as negative are mostly negative. However, tweets which have positive polarity tend to be classified more correctly as shown in the confusion matrix presented in TABLE VI.

B. Ternary Classification

Despite its importance, binary classification supposes that the given data are already known to be emotional. However, Twitter contains many tweets which have no emotional polarity such as news tweets, etc. Therefore, in this subsection we TABLE VI: Binary Classification.

TABLE V: Ternary Classification Accuracy

class	Privies percentage	Current percentage
Negative	73.63	77.47
Neutral	59.75	51.93
Positive	68.45	82.93

Add neutral tweets as shown before in the description of our dataset. We then rely on the same set of features to classify the tweets. The results obtained are given in TABLE VII, and the confusion matrix of classification is given in TABLE VIII.

The obtained results show that the introduction of a third class decreases noticeably the accuracy to reach 83.0%. The new class (i.e., *"neutral"*) presents a low accuracy, but a very high precision rate. This can be explained by the fact that the amount of training data (i.e., number of tweets) for this class is lower than that for the other classes. Therefore, a tweet that meets the conditions of the class *"neutral"* can be easily detected by the classifier as *"neutral"*. However, not many of them meet the condition, and therefore, they are misclassified.

Overall, the results obtained are promising.

C. Multi-class classification

In this subsection, we use the 6 sentiment classes that we described in Section III. The classification results are given in TABLE IX, while the classification in 6 different classes.

TABLE VI: Multi-class classification Accuracy

Class	Percentage of Accuracy
Happiness	73.2
Hate	65.45
Love	62.43
Neutral	51.57
Sadness	64.16
Sarcasm	78.02

Despite the number of classes, the accuracy obtained is equal to 63%, with a precision that reaches 69.7%. More interestingly, some sentiments seem to be easier to detect than others. In particular, tweets belonging to the class *"happiness"* were classified with an accuracy equal to 83.1%. This shows that tweets belonging to this class are easily distinguished from other classes. This might be due to the fact that, contrarily to negative tweets, positive tweets belong to mainly two classes, easy to distinguish from each other. Negative tweets on the other hand are closer to each other. A typical example is given by the following tweet: *"Damn it.. I really hate when this happens. This crap doesn't want to work!!!"*. In this tweet, the user expresses both sentiments of anger and hate. However, since he explicitly uses the word *"hate"* the tweet would be classified as belonging to the class *"Hate"[1],* although it shows sentiments of anger more than hate.

5. CONCLUSION:

In this paper, a SVM based model is presented to update the classification right. The proposed method classified the tweets in positive, negative and neutral sentiments with whereas much of the literature in this field is associated with 2-way classification [11][12]. In existing system the machine is only able to find the 7 type of emotion "Happiness , Love, Sadness, Anger ,Hate, Sarcasm and neutral " but when the user uses the shortcut or acronyms like "ILU,143**,**(happy)**,** KMN(kill me now)**,** MYOB (mind your own business)**,**B3 (blah,blah,blah)"the machine unable to perform the classification, so in Proposed system may be able to machine learning algorithm can classify the acronyms. The comparative observations are taken against the SVM and KNN methods. The comparative results shows that the proposed model has improved the accuracy and f-measure of tweet class prediction. As number of features for learning the classifiers are limited in our approach, we will be using more features and better feature selection methods like Information capture, Chi-Square etc. in our future work. Our comparison with literature [12] shows that increasing our dataset with more tweets and features can also help in increasing reasonable accuracy and f-measure. Other machine learning methods in combined way can also be explored in the future.

REFERENCES :

1. Mondher Bouazizi "A Pattern-Based Approach for Multi-Class Sentiment Analysis in Twitter", Graduate School of Science and Technology, Keio University Yokohama, Japan.(2016).
2. B. O'Connor, R. Balasubramanyan, B. Routledge, and N. Smith, "From tweets to polls: Linking text sentiment to public opinion time series," in *Proc. Int. AAAI Conf. Weblogs and Social Media*, pp. 26–33, May 2010.
3. M. A. Cabanlit and K. J. Espinosa, "Optimizing N-gram based text feature selection in sentiment analysis for commercial products in Twitter through polarity lexicons," in *Proc. 5th Int. Conf. Inform., Intelligence, Syst. and Applicat.*, pp. 94–97, July 2014.
4. U. R. Hodeghatta, "Sentiment analysis of Hollywood movies on Twitter," in *Proc. IEEE/ACM ASONAM*, pp. 1401–1404, Aug. 2013.
5. J. M. Soler, F. Cuartero, and M. Roblizo, "Twitter as a tool for predicting elections results," in *Proc. IEEE/ACM ASONAM*, pp. 1194–1200, Aug. 2012.
6. K. Ghag and K. Shah, "Comparative analysis of the techniques for sentiment analysis," in *Proc. Int. Conf. Advances in Technology and Eng.*, pp. 1–7, Jan. 2013.
7. C. G. Akcora, M. A. Bayir, M. Demirbas, and H. Ferhatosmanoglu, "Identifying breakpoints in public opinion," in *Proc. First Workshop on Social Media Analytics*, pp. 62–66, July 2010.
8. B. Sriram, D. Fuhry, E. Demir, H. Ferhatosmanoglu, and M. Demirbas, "Short text classification in twitter to improve information filtering," in *Proc. 33rd Int. ACM SIGIR Conf. Research and development in information retrieval*, pp. 841–842, July 2010.
9. B. Pang, L. Lillian, and V. Shivakumar, "Thumbs up?: Sentiment classification using machine learning techniques," in *Proc. ACL-02 Conf. Empirical Methods in Natural Language Process.*, vol. 10, pp.79–86, July 2002.

10. M. Boia, B. Faltings, C.-C. Musat and P. Pu, “A :) is worth a thousand words: How people attach sentiment to emoticons and words in tweets,” in *Proc. Int. Conf. Social Computing*, pp. 345–350, Sept. 2013.
11. K. Manuel, K. V. Indukuri and P. R. Krishna, “Analyzing internet slang for sentiment mining,” in *Proc. 2nd Vaagdevi Int. Conf. Inform. Technology for Real World Problems*, pp. 9–11 Dec. 2010.
12. W. Gao and F. Sebastiani, “Tweet Sentiment: From Classification to Quantification,” in *Proc. IEEE/ACM ASONAM*, pp. 97–104, Aug. 2015.
13. Y.H.P.P. Priyadarshana, K.I.H. Gunathunga, K.K.A. Nipuni N.Perera, L. Ranathunga, P.M. Karunaratne, and T.M. Thanthriwatta, “Sentiment analysis: Measuring sentiment strength of call centre conversations,” in *Proc. IEEE ICECCT*, pp.1–9, March 2015.
14. R. Srivastava and M.P.S. Bhatia, “Quantifying modified opinion strength: A fuzzy inference system for Sentiment Analysis,” in *Proc. Int. Conf. Advanced in Computing, Communications and Informatics*, pp.1512– 1519, Aug. 2013.
15. K.H. Lin, C. Yang and H.Chen, “What emotions do news articles trigger in their readers?,” in *Proc. ACM SIGIR ’07*, pp. 733–734, July 2007.
16. K.H. Lin, ; C. Yang and H Chen, “Emotion Classification of Online News Articles from the Reader’s Perspective,” in *Proc. IEEE/WIC/ACM WI-IAT ’08*, vol.1, pp.220–226, Dec. 2008.
17. L. Ye, R. Xu and J. Xu, “Emotion prediction of news articles from reader’s perspective based on multi-label classification,” in *Proc. Int. Conf. Machine Learning and Cybernetics*, vol.5, pp. 2019–2024, July 2012.
18. W. Liang, H. Wang, Y. Chu and C. Wu, “Emoticon recommendation in microblog using affective trajectory model,” in *Proc. Annual Summit and Conf. Asia-Pacific Signal and Information Processing Association (APSIPA)*, pp.1–5, Dec. 2014.
19. C. Fellbaun, *WordNet: an Electronic Lexical Database*, Cambridge, Massachusetts, 1998.
20. M. Bouazizi and T. Ohtsuki, “Sarcasm detection in Twitter,” to be published in *IEEE Globecom*, Dec. 2015.
21. Ankita Gupta1 , Jyotika Pruthi2 , Neha Sahu3 ” Sentiment Analysis of Tweets using Machine Learning Approach” Dept. of Computer Science THE NORTHCAP UNIVERSITY, Sector 23A Gurgaon, Haryana, INDIA. (2017).
22. Pooja Deshmukh “Sarcasm Detection and User Behaviour Analysis” Student of ME (CSE) Department of Computer Science and Engineering, Deogiri Institute of Engineering and Management Studies, Aurangabad.
23. Prerna Chikersal, Soujanya Poria, and Erik Cambria” SeNTU: Sentiment Analysis of Tweets by Combining a Rule-based Classifier with Supervised Learning” School of Computer Engineering Nanyang Technological University Singapore – 639798(2015)
24. Mondher Bouazizi “All Your Products Are Incredibly Amazing!!!-Are They Really?” Graduate School of Science and Technology,Keio University Yokohama, Japan.(2015)
25. M.A. Cabanlit and K.J. Espinosa, “Optimizing N-gram based text feature selection in sentiment analysis for commercial products in Twitter through polarity lexicons,” in *Proc. 5th Int. Conf. Inform., Intelligence, Syst. and Applicant.*, pp. 94–97.(July 2014)
26. Mukwazvure, Addlight, and K. P. Supreethi. "A hybrid approach to sentiment analysis of news comments." Reliability, Infocom Technologies and Optimization (ICRITO)(Trends and Future Directions), 2015 4th International Conference on. IEEE, 2015.
27. Asmi, Amna, and Tanko Ishaya. "Negation identification and calculation in sentiment analysis." The Second International Conference on Advances in Information Mining and Management. 2012.
28. Hutto, Clayton J., and Eric Gilbert. "Vader: A parsimonious rule-based model for sentiment analysis of social media text." Eighth International AAAI Conference on Weblogs and Social Media. 2014.
29. Khan, Jawad, and Byeong Soo Jeong. "Summarizing customer review based on product feature and opinion." Machine Learning and Cybernetics (ICMLC), 2016 International Conference on. IEEE, 2016.
30. Neviarouskaya, Alena, Helmut Prendinger, and Mitsuru Ishizuka. "Semantically distinct verb classes involved in sentiment analysis." IADIS AC (1). 2009.
31. Kawathekar, Swati A., and Manali M. Kshirsagar. "Sentiments analysis using Hybrid Approach involving Rule-Based & Support Vector Machines methods." IOSRJEN 2.1 (2012): 55-58.
32. Revathy, K., and B. Sathiyabhama. "A hybrid approach for supervised twitter sentiment classification." International Journal of Computer Science and Business Informatics 7 (2013).
33. Go, Alec, Richa Bhayani, and Lei Huang. "Twitter sentiment classification using distant supervision." CS224N Project Report, Stanford 1 (2009): 12.