IOSR Journal of Engineering (IOSRJEN) **www.iosrjen.org**
ISSN (e): 2250-3021, ISSN (p): 2278-8719
Special Issue || Dec. 2019 || PP 58-63
International Conference on Management Practices, Innovations & Research 2019

An Enhanced Approach for Classifying Twitter Emotions Using Machine Learning

Bireshwar Ganguly[1], Devashri Raich[2]
[1]*Assistant Professor, Dept. of IT, RCERT, Chandrapur*
[2]*Assistant Professor, Dept. of IT, RCERT, Chandrapur*

Abstract: *In this modern age, social media apps like Twitter, Facebook, Tumbler, and etc carries a significant role in everyone's life. Twitter is a worldwide blogging platform that has a lot of information that can be utilized for different assessment & analysis of views & opinions of people all across the globe. The amount of data accumulated on Twitter is very huge. This data is unstructured & raw as it is written by common man. Twitter Sentimental Analysis is the process of accessing tweets for a particular topic and predicts the sentiment of these tweets as positive, negative or neutral with the help of different machine learning algorithm. Analysis such as prediction, forecasts, evaluations, elections, marketing etc using emotion analysis is one such procedure of extracting vital information from this data and views. The aim of this paper is to classify twitter data into sentiments (positive or negative) by using different machine learning techniques on data collected for different people and classify its polarity. The proposed work takes as input a large corpus of documents like tweets or news articles and generates a vector space of typically several hundred dimensions. Each word in the corpus is assigned a unique vector in the vector space. The vector representation of words is to automatically incorporate several features that we would normally need to build ourselves. Since the technique relies on Machine learning to detect patterns, we can rely on it to detect multiple features on different levels of abstractions.*
Keywords: *Sentiment Analysis, Twitter, Machine Learning*

I. Introduction

The Sentiment is essentially about feelings; Attitudes, emotions and opinions. Sentiment analysis refers to the practice of using natural language processing techniques and text analysis to identify and extract subjective information from a piece of text. A person's opinions or feelings are largely subjective and not objective. It can be extremely difficult to accurately analyze the opinion or mood of a person from a text. Essentially, we want to gain an understanding of a writer's attitude towards a topic in a text and its polarity through the analysis of feelings in an analytical perspective of the text. either positive, negative or neutral.

In recent years, the interest of brands, companies and researchers in the analysis of feelings and their application to business analysis has continued to grow. Today's business world is looking for "Business Insight," as is the case in many data analysis streams. In terms of Sentiment analysis, we are talking about information about consumer behaviour, what customers want, what they like and what are not products, what are their purchase signals, what does it look like? decision-making process, etc., because in the end, customers who are satisfied with the work we built-in analyzer using the NRC Mood Dictionary to calculate the presence of eight different emotions and their respective weights in a text.

Twitter is an online microblogging tool that transmits more than 400 million daily messages. It contains a lot of information about almost every industry, from entertainment to sports and health to business. It is in its accessibility. It is easy to exchange and collect information. Twitter offers unmatched access to our legislators and celebrities, as well as the latest news. Twitter is also an important data source for large business models.

All of the above features make Twitter the best place to collect updated and real-time data to analyze and investigate real-world situations.

II. Literature Review

Go, Bhayani and Huang (2009) [2] they sort the tweets of a search term into negative or positive emotions. They automatically collect training data on Twitter. To collect positive and negative tweets, they ask Twitter for happy and sad emoticons.

• Cheerful emoticons are different versions of smiley faces, such as ":)", ":-)", ":)", ": D", "=)" etc.
• Sad emoticons contain frowns like ":(", ":-(", ":(", etc.

They try different functions (unigrams, bigramas and part of the language) and train their classifier with several machine learning algorithms (Naive Bayes, Maximum Entropy and Scalable Vector Machines) and compare them with a basic classifier counting the number of positive and negative classifiers. Words include a corpus

accessible to the public. They report that only bigigrams and partial language marking are not useful and that the Naïve Bayes classifier gives the best results

Pak and Paroubek (2010) [1] They find that the use of informal and creative language makes tweet analysis a slightly different task. They use previous work in hashtags and sentiment analysis to build their classifier. They use the Edinburgh Twitter corpus to find the most common hashtags. Manually classify these hashtags and then use them to classify tweets. In addition to using the n-gram and partial language functions, a set of functions is also created from the existing MPQA thematic dictionary and the Lingo Internet dictionary. They report that the best results are achieved with n-gram functions with lexical functionality, while the use of speech functions results in a loss of precision.

Koulompis, Wilson and Moore (2011) [5] they studied the usefulness of language features to detect the mood of Twitter messages. They evaluated the value of lexical resources and the existing characteristics that capture information about the informal and creative language used in microblogging. They approached the problem under supervision, but used existing tags in Twitter data to create training data.

Saif, He and Alani (2012) [6] they discuss a semantic approach to identify the entity that is discussed in a tweet, such as: They also show that the deletion of stop words is not a necessary step and can have adverse effects on the classifier. All prior techniques are based on the characteristics of n grams. It is not clear if it makes sense to use partial language marking. To improve accuracy, use several different methods to select features or use knowledge in microblogging. On the other hand, we improve our results through the use of more basic techniques used in the analysis of feelings, such as derivation, two-level classification and detection of negation, and levels of negation. Negative detection is a technique that has been widely studied in the analysis of feelings. Negation words such as "no," "never," "no," etc., can radically change the meaning of a sentence and, therefore, the feeling it expresses. Due to the presence of such words, the meaning of the neighboring words is reversed. These words must be in the context of negation. Much research has focused on recognizing the scope of negation. The degree of negation of an index can be changed from this word to the following punctuation.

Councilman McDonald and Velikovich (2010) [7] discuss a technique to identify clues of negation and their meaning in a sentence. Identify explicit negation clues in the text and for each word in the scope. Then they find left and right the distance to the next negative index.

III. Methodology

We use several feature sets and machine learning classifiers to determine the best combination to analyze Twitter's feelings. We also experiment with different pre treatment steps, such as punctuation, emoticons, specific Twitter terms and referral. We examine the following functions: unigrams, bigrams, trigrams and negation detection. Finally, we train our classifier with several machine learning algorithms: Naive Bayes, Decision Trees and Maximum Entropy.

1. Data Collection:

This is a collection of 5513 tweets collected for four different themes, namely Apple, Google, Microsoft, and Twitter. Sanders Analytics LLC collects and archives them by hand. Each entry in the corpus contains a Tweet-ID, a subject and a feeling label. We use the Python Twitter library to enrich this data by loading data such as tweet text, creation date, creator and more. for each Tweet ID. Each tweet is divided into four categories by an American. For the purposes of our experiments, we consider the same class as irrelevant and neutral. The illustration of the tweets in this corpus is presented as follows:

• Positive For a positive attitude towards the subject
• Positive To show missing, mixed or weak feelings about the subject.
• Negative Represent a negative attitude towards the subject.
• Not applicable for English texts or non-thematic comments.

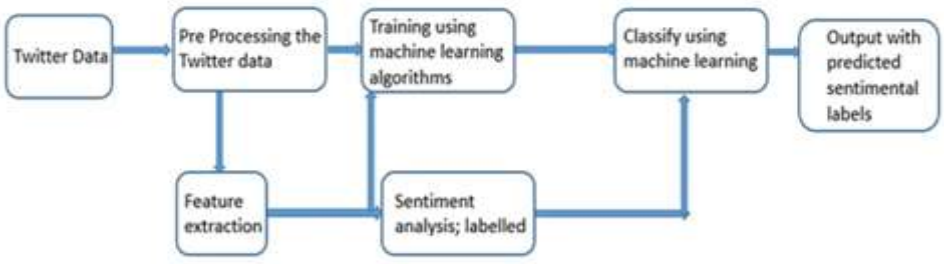

Figure 1: Schematic Block Representation of the Methodology

2. Pre Processing

User-generated content on the Web is rarely present in a form suitable for learning. It becomes important to standardize the text through a series of preprocessing steps. We use a complete set of preprocessing steps to reduce the size of the feature set and make it suitable for learning algorithms. Figure 2 shows different characteristics of microblogging. Table 3 shows the frequency of these functions by tweet, cut by the records. We also give a brief description of the steps prior to treatment.

2.1 Hash tags

A hashtag is a non-spatial word or phrase preceded by the hash symbol (#). He names the topics and the sentences that are currently on the trend topics. For example #iPad, #news
Regular expression: # (w +)
Replace the expression: HASH_1

2.2 Handles

Each Twitter user has a unique username. All information for this user can be displayed by typing the username preceded by "@". Therefore, they are like proper names. For example @ Apple
Regular expression: @ (w +)
Replace the expression: HNDL_1

2.3 URLs

Users often share hyperlinks in their tweets. Twitter truncates it using its internal URL shortening service. By using these links, Twitter can warn users when the link leaves their domain. From the point of view of text classification, a specific URL is not important. However, the presence of a URL can be an important feature. The regular expression to recognize a URL is quite complex because there can be different types of URLs. However, due to the Twitter shortening service, we can use a relatively simple regular expression.
Regular expression: (http | https | ftp): // [a-zA-Z0-9 \ ./] +
Replace the expression: URL

2.4 Emoticons

The use of emoticons is widespread on the Web, especially in microblogging sites. We identify the following emoticons and replace them with a word. Table 4 lists currently recognized emoticons. All other emoticons would be ignored.

2.5 Punctuation marks

Although any punctuation is not important from the standpoint of classification, some of them, such as: As a question mark or exclamation point, they also provide information about the feelings of the text. We replace each word limit with a list of relevant punctuation marks that exist at that time. Table 5 lists the currently identified score. We also remove single quotes that may be present in the text.

People often use recurring characters while using familiar phrases such as "I'm in a hurry," "We win, yaaayyyyy!" As the last preprocessing step, we replace the characters that are repeated more than twice with two characters.
Regular expression: (.) 1 {1,}
Replace the expression: 1 1

2.6 Reduction of feature space

It is important to know that these pretreatment steps reduce our functionality, otherwise they may be too scarce.
People often use recurring characters while using familiar phrases such as "I'm in a hurry," "We win, yaaayyyyy!" As the last preprocessing step, we replace the characters that are repeated more than twice with two characters.
Regular expression: (.) 1 {1,}
Replace the expression: 1 1

3. Stemming Algorithms

All stemming algorithms are of the following main types: elimination of affixes, random and mixed. The first type, the stem to eliminate Affix, is the most basic. They apply a set of transformation rules to each word to cut the commonly known prefixes and / or suffixes [8]. A trivial derivation algorithm would be to cut the words to the nth symbol. However, this is obviously not suitable for practical purposes.

J. B. Lovins described the first derivation algorithm in 1968. He defined 294 terminations, each associated with one of the 29 conditions, plus 35 transformation rules. For an accepted word, an ending with a satisfactory state is found and removed. Another famous stem, which is widely used, is described in the next section.

Martin Porter wrote a Stemmer, which was published in July 1980. This Stemmer was very common and became and remains the de facto standard algorithm for English derivation. It offers an excellent compromise between speed, readability and precision. About 60 rules are used, which are applied in 6 consecutive steps [9]. An important feature is that it is not a recursion.

4. Features

Several functions can be used to create a classifier for tweets. The most commonly used basic set of features is the word n-gram. However, tweets contain a lot of domain-specific information that can also be used for classification. We have experienced two functional sets:

4.1 Unigrams

Unigrams are the simplest functions that can be used for text classification. A tweet can be represented by a variety of words. However, we use the presence of unigrams in a tweet as a set of features. The presence of a word is more important than the frequency with which it is repeated. Pang et al. He discovered that the presence of unigrams leads to better results than repetition [1]. It also helps us avoid having to scale data, which can significantly reduce training time [2].

We also observe that the unigrams conform to the Zipf law. He says that in a natural language corpus, the frequency of a word is inversely proportional to its range in the frequency table. Figure 4 is a graphical representation of the logarithmic frequency versus the logarithmic range of our data set. A linear trend line fits the data well.

4.2 N-grams

N-gram refers to a sequence of n words. Probabilistic language models based on unigrams, bigrams and trigrams can be used successfully to predict the next word in a common word context. In the area of sentiment analysis, the performance of N-grams is unclear. According to Pang et al. Some researchers report that unigrams are better than bigrams to rank only movie critics, while others report that bigrams and trigrams provide a better polarity rating for product reviews [1].

As the order of n-grams increases, they tend to be increasingly scarce. According to our experiences, we find that the amount of bigramas and trigrams increases much faster than the amount of unigrams with the amount of tweets. Figure 4 shows the amount of n-grams versus the number of tweets. We can observe that the bigrams and trigrams increase almost linearly, while the unigrams increase logarithmically.

4.3 Negation Handling

The need to recognize negations in the analysis of feelings can be explained by the difference in the meaning of the phrases "it is good." "Not good" illustrate. However, the negations that occur in natural language are rarely so simple. The treatment of negation consists of two tasks: the recognition of explicit negation indices and the scope of the negation of these words.

Councilill et al. See if the detection of negation is useful for the analysis of feelings and to what extent it is possible to determine the exact extent of the negation in the text [7]. They describe a method to detect negation based on the left and right distance of a token to the following explicit negation index.

IV. Implementation

We train 90% of our data with different feature combinations and test them with the remaining 10%. We take the properties in the following combinations only unigrams, unigrams + bigrams and filtered trigrams, unigrams + refusal, unigrams + filtered bigrams and trigrams + negation. Then we train classifiers with different classification algorithms: Naive Bayes Classifier and Maximum Entropy Classifier.

Classifying a tweet can be done in two steps: first by classifying the "neutral" (or "subjective") tweets against "the target" and secondly by classifying the objective tweets as "positive" tweets versus "negative". We also train classifiers in 2 levels.

1. Naive Bayes

The Naive Bayes classifier is the simplest and fastest classifier. Many researchers [2], [4] claim to have obtained the best results with this classifier.

If we need to find the name of a particular tweet, we will determine the probabilities of all the names, taking this property into account, and then select the label with the maximum probability.

For the Naïve Bayes classifier, we specify the precision values for the recovery of different classes: Negative, Neutral and Positive. Solid marks indicate P-R values for one-step classifiers and hollow marks show the effect of using dual class classifiers. Different points apply to different sets of functions. We can see that the accuracy and recovery values are higher for a single step than for a double step

2. Maximum Entropy Classifier

This classifier determines a probability distribution that maximizes the probability of verifiable data. This probability function is established by the weight vector. The optimal value can be determined using the Lagrange multiplier method.

III. Results

In Naive Bayes the accuracy of unigram is the lowest with 69.53%. The accuracy increases (71.32%) or more than n-grams (74.43%). We see that when we use both negation detection and higher order n-grams, the accuracy is slightly lower than only higher order n-grams (72.23%). Therefore, we can conclude that the accuracies for the two-step classifiers are better than those of the corresponding simple steps.

The results of the formation of the maximum entropy classifier are as follows. The details follow a similar trend compared to the Naive Bayes classifier. Unigram is the lowest with 79.73%, and negation detection shows an increase of 80.96%. The maximum is reached with unigram, bigram and trigram in 75.22%, followed closely by n-gram and negation in 75.16%. Again, the details for accuracies for double step classifiers are much lower.

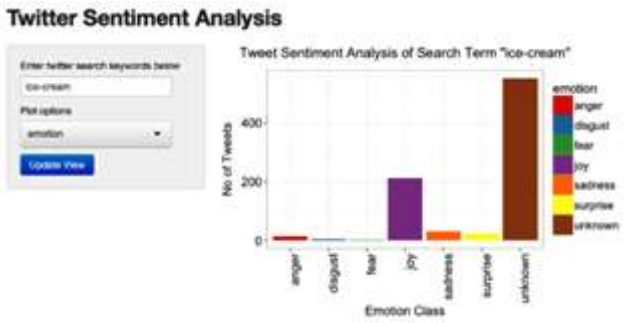

Fig 2: Emotion Class [13]

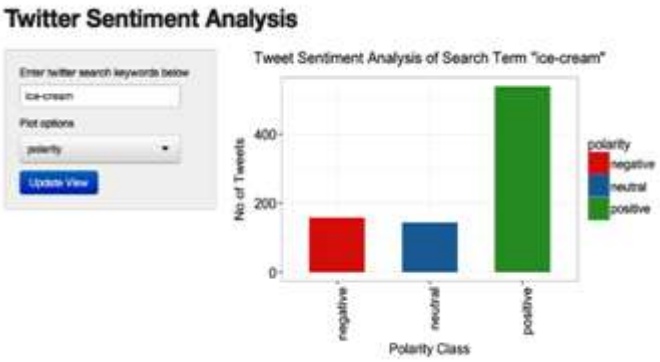

Fig 3: Polarity Class [13]

Hence, we see that the accuracy of the "neutral" class increases with the use of a double stage classifier, but with a significant decrease in recall and a slight decrease in the accuracy of the "negative" and "positive" classes.

IV. Conclusion & Future Scope

We create a sentiment classifier for Twitter using tagged records. We also study the relevance of using a two-step classifier and negation detection for sentiment analysis purposes.

Our basic classifier, which uses only unigrams, reaches an accuracy of approximately 70.00%. The accuracy of the classifier increases when we use negation detection or introduce bigrams and trigrams. Therefore, we can conclude that the recognition of negation and higher order n-grams are useful for classifying texts. However, if we use both n-gram detection and negation detection, the accuracy decreases slightly. We

also note that single-step classifiers perform two-step classifiers. In general, the Naive Bayes classifier is better than the maximum entropy classifier.
We obtain the best accuracy of 76% approx in Unigram + Bigram + Trigram, formed with the Naive Bayes classifier.

Review of support vector machines In several articles, the results were also discussed using support vector machines (SVM). The next step would be to test our SVM approach. However, Go, Bhayani and Huang reported that SVMs do not increase accuracy [2].

Futuristically creating a classifier for tweets in Hindi on Twitter, many users mainly use the Hindi language. The approach discussed here can be used to create a Hindi Sentiment classifier.

Improving results with semantic analysis If you understand the role of the names you are talking about, you can better classify a specific tweet. We can use semantic markers to achieve this. Such an approach is discussed by Saif, He and Alani [6].

References

[1]. Pak, Alexander, and Patrick Paroubek. "Twitter as a Corpus for Sentiment Analysis and Opinion Mining." LREc. Vol. 10. 2010.
[2]. Alec Go, Richa Bhayani, and Lei Huang. Twitter sentiment classification using distant supervision. Processing, pages 1-6, 2009.
[3]. Niek Sanders. Twitter sentiment corpus. http://www.sananalytics.com/lab/twitter-sentiment/. Sanders Analytics.
[4]. Alexander Pak and Patrick Paroubek. Twitter as a corpus for sentiment analysis and opinion mining. volume 2010, pages 1320-1326, 2010.
[5]. Efthymios Kouloumpis, Theresa Wilson, and Johanna Moore. Twitter sentiment analysis: The good the bad and the omg! ICWSM, 11:pages 538-541, 2011.
[6]. Hassan Saif, Yulan He, and Harith Alani. Semantic sentiment analysis of twitter. In The Semantic Web-ISWC 2012, pages 508-524. Springer, 2012.
[7]. Isaac G Councill, Ryan McDonald, and Leonid Velikovich. What's great and what's not: learning to classify the scope of negation for improved sentiment analysis. In Proceedings of the workshop on negation and speculation in natural language processing, pages 51-59. Association for Computational Linguistics, 2010.
[8]. Ilia Smirnov. Overview of stemming algorithms. Mechanical Translation, 2008.
[9]. Martin F Porter. An algorithm for suffix stripping. Program: electronic library and information systems, 40(3):pages 211-218, 2006.
[10]. Balakrishnan Gokulakrishnan, P Priyanthan, T Ragavan, N Prasath, and A Perera. Opinion mining and sentiment analysis on a twitter data stream. In Advances in ICT for Emerging Regions (ICTer), 2012 International Conference on. IEEE, 2012.
[11]. John Ross Quinlan. C4. 5: programs for machine learning, volume 1. Morgan kaufmann, 1993.
[12]. Steven Bird, Ewan Klein, and Edward Loper. Natural language processing with Python. " O'Reilly Media, Inc.", 2009.
[13]. https://github.com/bmatthie/xForce-sentiment