International Journal of Control and Automation
Vol. 13, No. 4, (2020), pp. 412 – 421

Twitter Sentiment Analysis: A Novel Machine Learning Approach

Shubham Kumar[1], Akhilesh Kumar Srivastava[2*], Yaman Soni[3], Utkarsh Tyaghi[4], and Nikhil Kumar Singh[5]

[1, 3-5]Student, [2]Faculty,
[1, 3-5]Dept. of Electronics & Communication Engineering,
[2]Computer Science & Engineering Department
ABES Engineering College, Ghaziabad, U.P., 201009, India
shubham.16bec1054@abes.ac.in[1], akhilesh.srivastava@abes.ac.in[2]

Abstract

Sentiment Analysis is a technique used to classify the polarity of emotions (positive or negative) on a given text. Sentiment analysis is widely used in data mining [1] and information processing. Several studies have been done in the past to track the activities of the user on a social media platform like Twitter to gain general perception or response of a person toward a given entity such as products, services, organizations, individuals, political parties, etc. In this report, we have used two different machine learning techniques, Naïve Bayes and Support Vector Machine to do comprehension research on sentiment analysis.

Keywords: *Sentiment Analysis, Text Classification, text mining, machine learning*

1. Introduction

Twitter is a popular social networking site where members interact with each other and create messages known as "tweets". People express their feelings and emotion with their tweets. Twitter is the world's largest micro-blogging social networking platform with a huge user base, a statistical report from [12] shows there are currently 330 million active users with 500 million tweets per day who share their opinion through these tweets. That's why we chose the twitter dataset for our sentiment analysis work. We can analyze these tweets and gain a general opinion of a user. From opinion poll [2] to market strategy, and pandemic detections, sentiment analysis is playing a huge roll in business making nowadays. To use Twitter as a tool to read about public opinion is a widely used task nowadays. Sentiment analysis uses NLP and machine learning algorithms to detect the polarity of tweets. Also, with the recent advancement in machine learning and NLP techniques we can improve the accuracy of our mental predictions too. In sentiment analysis, the classification approach is used which uses an intelligent predictive algorithm to classify the polarity of a tweet by learning some pattern from the training data. Our model learns these patterns in the form of weight vector and these weight vectors are further used to classify the polarity of future tweets.

Classification uses two phases [fig. 2], the first one is the training phase in which the model is trained with the training data and some pattern is learned by the model. The second one is the testing phase in which the model is tested on test data and the accuracy of the model is recorded. We are using the dataset from an online data respiratory known as Kaggle, a public dataset platform where community members can share the dataset. We had two dataset training and

ISSN: 2005-4297 IJCA

testing data. We used training data to train our model which had labeled tweet as positive or negative. And we used test data (unlabeled tweet) to predict the sentiment of the tweet and recorded the accuracy of the model.

In this paper we have used two different classification algorithms, Naïve Bayes and Support Vector Machine with different featurization techniques like bag of words (bow) and term frequency-inverse document frequency (tfidf). After comparing the different model and their accuracy, the best one is used for sentiment analysis. And the work is done by open-source machine learning tools and software.

The data provided comes with the ‘URLs’, ‘brackets’, ‘usernames’, and ‘hashtag’ that need to be processed and converted into a standard format suitable for analysis and that can be feed into our model. We also need to extract only useful words from our document corpus to form word vector representation. We can remove stopwords like 'the', 'is', 'am', 'are', 'you' etc. which do not contribute to sentiment analysis. We trained our model with various machine learning algorithms with a high level of hyperparameter tuning to perform comprehensive research.

2. Related work

Sentiment analysis is a widely used domain nowadays, and relevant work has been done in recent few years. The research has been done on different datasets with different sizes, domains, corpus, languages, locations, etc. In [11], they took the unstructured data from the web, discussed the previous methods and approaches. Currently, most researcher focuses on subjective data and overlook the objective data, they proposed a methodology to classify the sentiment based on subjective as well as the objective of the statements. The research conducted by [7] has done great work by classifying the tweets into two ways, a 2-way of classifying tweets into the positive or negative and 3-way task of classifying tweets into positive, negative or neutral. In terms of language, the work conducted by [8] shows the work done on sentiment classification in both English and Arabic languages. The work done by [9] aims to extract the relevant topics of the campaign in the USA election and the names mentioned of the candidates. A model was built to observe the sentiment of the public on the candidates who were running in the election along with the topics mentioned by the public. In [13], Pang, Lee and Vaithyanathan worked on a movie review dataset to perform sentiment analysis. With 83 % accuracy, it was observed that machine learning model can perform better than simple counting methods in a very easy and simple way.

In [14], they proposed a work to establish a pattern between stock price and public statement on twitter For training set they prepared 15,000 high-frequency tokens from their data along with sentiment score termed as Total Sentiment Index. These tokens were categorized into positive and negative tokens based on the total sentiment index. TSI represented the relative sentiment of the tokens based on the number of times they appeared in positive and negative tweets. The proposed method to calculate TSI of a token was :

$$TSI = [\, p - (tp / tn) \times n \,] / [\, p + (tp / tn) \times n]$$

ISSN: 2005-4297 IJCA

p : count of tokens that occured in positive tweets

n : count of token that occured in negative tweets

tp : total number of positive tweets

tn : total number of negative tweets.

3. DATA DESCRIPTION

The data are given (training) is in the form of CSV file which contains 3 columns namely 'id', 'tweets', and ‘label’ where id is a unique integer and gives the unique index of tweets, the label is either 0 (negative) or 4 (positive) and tweet is in the form of the string enclosed in “ ”. Similarly, the test dataset comes with 'id', and 'tweet' column only. Our study is conducted on a dataset of 1.04 million tweets out of which nearly 76.3 % tweets were negative and 23.7% tweets were positive. It is observed that in-proportion in data class makes it difficult for most machine learning models to predict a new instance of the minority class. This problem can be solved in two ways eighter by upsampling (increasing the instance of minority class) or by downsampling (decreasing the size of majority class). We applied the downsampling [3] process on the majority class to avoid the biased result of the model towards the majority class. Finally, we worked on a 0.49 million dataset with an equal number of positive and negative tweets.

4. Methodology and Implementation

4.1. Preprocessing

The data in raw form is noisy in most cases, so we need to take care of it. The tweet data comes with URLs, hashtags, mentions (a reference to another user), brackets, Html tags, and emotions (emojis). The words also contain extra punctuations, repeated letters (like ‘yum’ as ‘yummyy’. The words with misspelling and repeated letter can be taken care of, emotions (emojis) can be utilized to predict sentiments while URLs and user mention can be completely ignored.

Here, the Preprocessing of data takes the following steps in the order given below: -

- First of all, we need to remove the Html tags.
- Removing punctuations ['"?!,. ():;] and special characters like ‘-‘, ‘:’, ‘,’, ‘\’ etc.
- Accept only those words which are made up of English letters and are not alpha- numeric.
- Normally words that are less than two characters do not give any sense, so better to remove them.
- Converting all the word to lowercase.
- Stopwords only increase the data size, so better to remove them too.

- Finally stemming the word to stem word (Snowball Stemming).

4.1.1. URLs

The user often put URLs to validate their tweet, but these URLs do not really contribute to determining sentiment, so we simply replace these URLs with a string 'URLs'.The expression used to match these URLs is (http\S+"),and (www\S+). We used python inbuilt library re (Regular Expression) to handle these URLs, which works as, tweet = re.sub(r"http\S+", "URL", tweet).

4.1.2. User Mention

Sometimes, the user mention other users in their tweets which does not help to decide the sentiment, so we can replace them by a string ‘user mention’.

The expression used to handle these mention is tweet = re.sub(r'@[\S]+', 'USER_MENTION ', tweet)

4.1.3. Emotions

Users often use emojis in their tweets to express their emotions and feelings. It is not possible to completely match all these emojis used with their respective sentiments on social media as it keeps increasing day by day. However, we could able to handle a few of them which is mentioned in table 1 and table 2.

Table 1. Emojis and their sentiments

S.N.	EXPRESSIONS	EMOTIONS
1.	(:\s?\)|:-\)|\(\s?:|\(-:|:\’\))	SMILE
2.	(:\s?D|:-D|x-?D|X-?D	LAUGH
3.	(:\s?\(|:-\(|\)\s?:|\)-:)	WINK
4.	(<3|:*)	LOVE
5.	(:\s?\(|:-\(|\)\s?:|\)-:)	SAD
6.	(:,\(|:\’\(|:"\()	CRY

Table 2. Handling emojis

S.N.	EXPRESSIONS	REPLACEMENT
1.	(:\s?\)|:-\)|\(\s?:|\(-:|:\’\))	EMO_POS
2.	(:\s?D|:-D|x-?D|X-?D	EMO_POS

ISSN: 2005-4297 IJCA

3.	(:\s?\(|:-\(|\)\s?:|\)-:)	EMO_POS
4.	(<3|:*)	EMO_POS
5.	(:\s?\(|:-\(|\)\s?:|\)-:)	EMO_NEG
6.	(:,\(|:\’\(|:"\()	EMO_NEG

4.1.4. Retweet

Retweets are those tweets, which are already tweeted and shared by others. We simply removed these tweets, because they do not help us to classify the text. These retweets start with the character 'RT'. The expression used to handle retweet is tweet = re.sub(r'\brt\b', '', tweet).

4.1.5. Removing Special characters

After applying previous processing on each tweet, we processed each word of the tweets as follow:

- Removing any punctuation ['"?!,. ():;] from the word.
- Converted repeated characters into 2 characters. Some people post tweets like It’s soooo yummyyy, adding more characters to emphasize certain words.
- Remove – and '. This is done by treating words like a t-shirt and converting them into a standard word i.e. tshirt form.
- Accept only valid string whose character starts with an alphabet or numbers

In stemming we remove the affixes (circumfixes, infixes, suffixes, prefixes) from a word so that we can get the word stem, e.g in running, removing the suffix ‘ing’ we can get the stem word run.

4.1.7. Lemmatization

Lemmatization is a process of grouping together different instances of a word as a single word. e.g better as good both have the same instance, we can use any one of them, instead of using two different words.

4.2 Featurization

After preprocessing the tweets, we need to convert the string into a vector form and this is also called word to a vector representation .

4.2.1. Bag of Words

A bag of word [4] is a representation of a document (each tweet) as a vector that describes the count of occurrence of a word in a document. It involves the following processing:

- Obtain the most frequent word from the text corpus tweets and make a vocabulary.
- Now in a word document, we count the number of times a word occurs and gave them a number which acts as a feature representation of each tweet
- Similarly, we find the words to the vector representation of each word.

4.2.2. Tf Idf

Tf Idf [5] is calculated as the product of tf and idf.

a. Term Frequency

The term frequency (tf) of a term (word) is a measure that defines how frequently a term appears in a document. It is defined as the count frequency of a term in a document divided by the total number of the term (word) in the document.

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{i,j}}$$

where, $n_{i,j}$ represents count frequency of term t_i in a document d_j and $\sum_k n_{i,j}$ represents the total number of terms in document d_j.

b. Idf

The idf shows the importance of a term in document. While computing term frequency, all terms are given equal weightage while in idf it is not so. Here in idf we give higher weightage to less frequent word while low weightage to the high frequent word. It is calculated as per given formula:

$$idf(w) = log(\frac{N}{df_t})$$

N = total number of document

d_{ft} = number of the document in which the term t_i is present

TFIDF = td × idf

4.3 Model Implementation

4.3.1. Naïve Bayes

Naïve Bayes classifier [6] is a supervised learning technique that classifies a text/sentence into a particular group. This approach is one of the best techniques used for text classification. The Naïve Bayes (NB) classifier is based on Baye's theorem, a practical Bayesian learning model that is easy to understand and implement. It is the probabilistic approach, used for text classification and it can learn patterns from a set of existing labeled documents.

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

Bayes's theorem can be used to find the probability of an event A to be positive or negative**,** given that event B has already occurred in previous. Since event B has already occurred, it is called evidence and event A has not occurred, so P(A) is termed as the prior probability . P(A|B) is the posterior probability of B i.e probability of event after event is seen. Since each word in a statement is treated as a feature, so the probability of each feature is multiplied to get the overall sentiment of the statement. If the overall probability is greater than 0.5 it is considered as positive and if it is less than 0.5, it is considered a negative sentiment. The assumption made here is that each predictor/features are independent of each other and the occurrence of one feature does not affect the occurrence of other feature. That's why it is called Naïve.

4.3.2. Support Vector Machine (SVM)

SVM is a supervised machine learning algorithm used for classification problems. It takes positive and negative data points and finds a hyperplane that can separate the positive and negative data points. The hyperplane that divides the data points is class decision boundary and the points that lie on decision boundary are called *support vectors.*

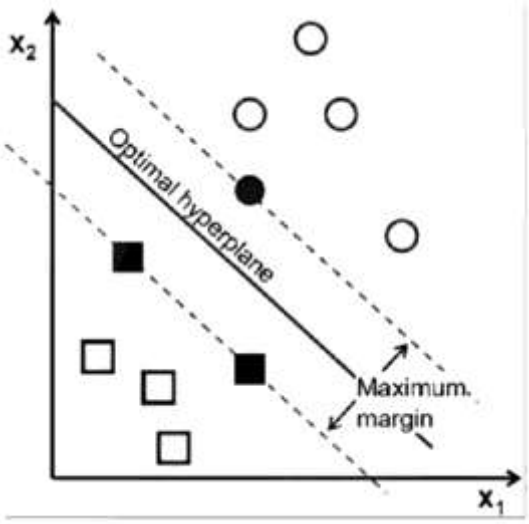

Figure 1. By Dr. Lance Eliot, the AI Trends Insider.

The main objective of SVM is to find a hyperplane in an N-dimensional hyperplane (N number of feature vectors) which can maximize the margin between positive and negative points, because larger the margin, lower is the generalization error.

5. Flow Chart

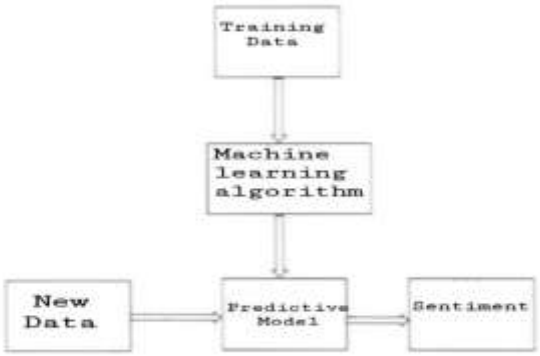

Figure 2. Flow chart of sentiment analysis.

6. Result

Similar to other text classification problems, the dataset was divided into two parts with 70: 30 ratio i.e. 70% of dataset to train the model and 30% of dataset to test the accuracy of the model. The dataset used in testing was not earlier used anywhere to train the model. The comparative study shows the different result on different algorithms used with different featurization techniques. Two different featurization techniques bow and tfidf of bigram were used to train the model.

From table 1 and table 2, we can easily say that tfidf featurization technique were recorded higher accuracy than bow. These accuracies were achieved with a deep hyperparameter tuning of parameter *alpha* based on auc score. Accuracy was used to measure the performance of model. From table 3 we can see that the accuracy of the Naïve Bayes model is 76.55% with bow representation while it is increased to 78% with tfidf representation. Table 4 also shows a similar result to table 3 on bow and tfidf representation but it is slightly better than one in table 3. So, we will use SVM model with tfidf featurization for our sentiment analysis work. Even we can stack both model on top of one another to form an ensemble model which could get us a slightly higher accuracy, but it will have a high time complexity.

The overall result shows that it will be better to use tfidf featurization technique on SVM model.

Table 3. Classification accuracy of Naïve Bayes Approach (Using sklearn ML toolbox)

S.N.	Featurization	Best Alpha	Test Accuracy
1	BOW	1000000	76.56
2	TF-IDF	100000	78.01

Table 4. Classification accuracy of SVM model (Using sklearn ML toolbox)

S.N.	Featurization	Best C	Test Accuracy
1	BOW	1	76.44
2	TF-IDF	0.1	78.37

7. Conclusion

In our research for sentiment analysis, a twitter dataset was used to train the model to classify the sentiment of the tweets. Its main objective is to classify the sentiments of the tweets by feeding the data into our machine learning model. We can also use this model for future work of sentiment analysis in different domains according to our needs. It is comprised of many steps like data collection, data pre-processing, analyzing the dataset, training, and testing of the model. However, this kind of model has some application issues too. It may not perform well on increasing the number of classes and also, it is not tested on any specific domain, so the accuracy may decrease in testing the model on a different domain. However, these kinds of problems may be solved very easily using different types of datasets. Sentiment analysis has a very wide range of applications and scope of development in the future. This paper provides two classification algorithms, anyone can be used to perform sentiment analysis on twitter. We can even combine both the model to achieve higher accuracy but it will have higher time complexity.

References

[1] Seema Sharma, Jitendra Agrawal, Shikha Agarwal, Sanjeev Sharma School of Information Technology,UTD, RGPV, Bhopal, M.P. India. DOI: 10.1109/ICCIC.2013.6724149. In 2013 IEEE International Conference on Computational Intelligence and Computing Research
[2] Bermingham, A., Smeaton, A.F.: "On using twitter to monitor political sentiment and predict election results (2011)"
[3] medium.com, blog by WARRIE USENOBONG, "Sampling Application in the field of Data Science and Machine Learning",Nov 15,2019 . In https://medium.com/@warrie.warrieus/sampling-application-in-the-field-of-data-science-and-machine-learning-614a770dce86. Accessed on Apr 13 2020
[4] machinelearningmastery.com, blog post by Jason Brownlee, "A gentle introduction to bag-of-words model", August 17,2019. In https://machinelearningmastery.com/gentle-introduction-bag-words-model/ Accessed on Apr 13 2020.
[5] Bafna P, Pramod D, Vaidya A (2016) Document clustering: TF-IDF approach. In: IEEE int. conf. on electrical, electronics, and optimization techniques (ICEEOT). pp 61–66
[6] Kavya Suppala, Narasinga Rao. "Sentiment Analysis Using Naïve Bayes Classifier". In International Journal of Innovative Technology and Exploring Engineering (IJITEE). ISSN: 2278-3075, Volume-8 Issue-8 June, 2019.

[7] Apoorv Agarwal Boyi Xie Ilia Vovsha Owen Rambow Rebecca Passonneau. “Sentiment Analysis of Twitter Data”. Department of Computer Science,Columbia University,New York, NY 10027 USA
[8] Rushdi-Saleh, M., Martín-Valdivia, M., Ureña-López, L., and Perea-Ortega, J. 2011. OCA: Opinion corpus for Arabic. Jour-nal of the American Society for Information Science and Technology .62, 10, 2045-2054
[9] Martin Ringsquandl and Dušan Petković. "Analyzing Political Sentiment on Twitter".University of Applied Sciences Rosenheim. Papers from the 2013 AAAI Spring Symposium
[10] Lampos, V., De Bie, T., Cristianini, N.: Flu detector-tracking epidemics on twitter.In: Machine Learning and KnowledgeDiscovery in Databases, pp. 599–602. Springer (2010)
[11] G.Vinodhini and RM.Chandrasekaran, “Sentiment Analysis and Opinion Mining: A Survey”. In Volume 2, Issue 6, June 2012 ISSN: 2277 128X International Journal of Advanced Research in Computer Science and Software Engineering
[12] omnicoregency.com, blog post, “Twitter by the Numbers: Stats, Demographics & Fun Facts” Feb 10, 2020. https://www.omnicoreagency.com/twitter-statistics/. Accessed Apr 4, 2020
[13] Pang, B., Lee, L., Vaithyanathan, S.: Thumbs up?Sentiment classification using machine learning techniques. In Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP). (2002) 79-86
[14] Gann W-JK, Day J, Zhou S (2014) Twitter analytics for insider trading fraud detection system In: Proceedings of the second ASE international conference on Big Data.. ASE.