Al-Nahrain Journal of Science
ANJS, Vol.24 (3), September, 2021, pp. 50-54

Twitter Sentiment Analysis Using Different Machine Learning and Feature Extraction Techniques

Mohammad W. Habib* and Zainab N. Sultani

Computer Science Department, College of Science, Al-Nahrain University, Baghdad-Iraq

Article's Information

Received:
24.04.2021
Accepted:
05.06.2021
Published:
30.09.2021

Keywords:
Sentiment analysis
Natural language processing
Machine learning
Twitter data

Abstract

Twitter is considered a significant source of exchanging information and opinion in today's business. Analysis of this data is critical and complex due to the size of the dataset. Sentiment Analysis is adopted to understand and analyze the sentiment of such data. In this paper, a Machine learning approach is employed for analyzing the data into positive or negative sentiment (opinion). Different arrangements of preprocessing techniques are applied to clean the tweets, and various feature extraction methods are used to extract and reduce the dimension of the tweets' feature vector. Sentiment140 dataset is used, and it consists of sentiment labels and tweets, so supervised machine learning models are used, specifically Logistic Regression, Naive Bayes, and Support Vector Machine. According to the experimental results, Logistic Regression was the best amongst other models with all feature extraction techniques.

DOI: 10.22401/ANJS.24.3.08

*Corresponding author: st.mohammed.wahab@ced.nahrainuniv.edu.iq

1. Introduction

Micro-blogging websites such as Twitter and Facebook are not strictly for social communication but also act as valuable information sources. Twitter is the most favorite site for millions of users to interact through tweets [1] daily. Tweets discover the public feeling/sentiment on various topics such as product, event, political issues, etc. Analyzing those tweets gives helpful feedback, especially for private and public businesses. Sentiment Analysis is selected as a technique to analyze large data since manually examining millions of tweets is not practicable [2]. Sentiment Analysis is considered one of the Natural Language Processing (NLP) applications that specify the categorization of text as representing a positive or negative opinion/feeling [1], [2].

Sentiment Analysis is implemented in three approaches: Machine learning-based, Sentiment lexicon-based, and Hybrid. Classifying a tweet into either a positive or a negative is a natural process for humans. Still, this manual method is not adequate to deal with vast amounts of data over the Internet. Machine learning algorithms are developed to solve this issue. The Machine learning approach (ML) applies machine learning classifiers (models) in categorizing the text data. In the Sentiment lexicon approach, lexicons (vocabulary) are used to analyze the text data. If the text includes more positive words, then the text sentiment is considered a positive; however, a negative label is given if more negative words are given. The hybrid approach employs both Machine learning and sentiment lexicon.

The most common tweet size is 140 characters in length, ranging from 13 to 15 words on an average [4]. These tweets require preprocessing since they contain misspellings, informal language, and symbolic words; therefore facing many challenges in processing and analyzing the data [2].

In this paper a machine learning approach for sentiment analysis is presented where different ML models are implemented using different feature representation techniques. Two Preprocessing phases are designed to explore their impact on the ML classifiers accuracy results.

The paper is organized as following, in section 2 related works are discussed then the proposed model is described in section 3. Section 4 focuses on the experimental results and discussion finally the conclusion is derived.

2. Related Work

Authors in [3] designed a hybrid sentiment analysis system using Hybrid Lexicon and Naive Bayesian Classifier on a Twitter dataset. The dataset was preprocessed by removing: stop words, URL, punctuations and replacing negation with "not". The hybrid system produces more reliable results than the two approaches. The hybrid system accuracy was 82%, while NB and Lexicon accuracy was 75% and 61%. In [4], researchers try to build a recommendation system depending on users' feelings. The dataset was a collection of tweets and re-tweets. The tweets were preprocessed by ignoring retweets, removing user mention, hashtag, non-alphabetic characters, and URL, then TF-IDF was used as a feature vector representation. More than one ML classifier is implemented; however, NB gave the best results where the accuracy was 66.86%. Hybrid Sentiment analysis is proposed in [5] Different ML algorithms were applied, such as SVM and NB, to analyze and classify political opinions. TF-IDF was implemented to represent the tweet as a feature vector after removing the URL and special characters. The hybrid system categorized 100,000 tweets, and the accuracy was 79%. In [6], the authors designed a topic sentiment analysis system to deal with social media websites inflation. The dataset contains 3731 tweets where they were cleaned by removing stop words, numbers, symbols, and URLs; then, it was represented in a feature vector form using TF-IDF. NB model was implemented where the accuracy was 81.4%. Rule-based with NB classifier was adopted in [7] for Twitter sentiment analysis. In the rule-based method, a set of rules based is presented on the occurrences to classify tweets' feelings, and emoticons are utilized for training the NB classifier. Experiments on standard emotion 140 datasets are conducted, and the text is represented in BOW feature vector representation. The proposed hybrid system achieves precision equal to 84.96%. In [8], sentiment analysis on labeled Twitter movie reviews was conducted. NB, SVM, and maximum entropy were implemented to classify Twitter data. The results show that SVM outperformed other classifiers for movie reviews with 84% accuracy.

3. Proposed Twitter Sentiment Analysis Model

The system aims to carry out sentiment analysis over tweets gathered from the Twitter dataset. Various algorithms have been utilized and tested against the Sentiment140 dataset, and the accuracy results were recorded. In this section, the main steps are discussed, including: preprocessing and feature extraction techniques using natural language processing (NLP), and machine learning classifiers. Figure 1 presents the framework of the proposed twitter sentiment analysis model.

Once the dataset has been preprocessed (cleaned) and split into training and testing datasets, it will be transformed into feature vector representation using the techniques specified below. Features will be extracted to minimize the dimension of the dataset. The next step is to apply different ML models to classify the tweets into positive and negative.

Text Preprocessing and Feature Extraction Techniques

Data preprocessing and feature extraction are considered a crucial step in ML process. In this step the data is transformed into an understandable and acceptable format. Tweets data consist of noise, many unwanted and undesirable features like: URLs, User mention, numbers, stop words, etc.; see Table 1. These features are not necessary while conducting sentiment analysis. Therefore tweet text data should be pre-processed using different NLP methods [9]. Some of these features are removed such as stop words, while the other features are replaced such as URL and User mention. Lemmatization is also applied where the words are transformed to their base form [10].

Table 1. Tweet Text from Sentiment140 Dataset.

Tweet	Label
@louloulou no we didn't meet	positive
Just now, I hate everything.	positive
watching titanic	negative
im boreed!	positive
goodnight.......	negative
And this is a bad thing?	negative

Feature vector representation allows the classifier to perform the classification process efficiently [11]. Extraction methods are divided into two categories. The first is known as bag-of-words strategies such as Bag of Words (BOW), Term Frequency–Inverse Document Frequency (TF-IDF), which represent a document based on the frequency of terms in the document without taking into account their order within the document. Since the features are distinct terms, even collections with a reasonable number of tweets can yield thousands of features. The semantic word embeddings are used in the second form of feature extraction process. Word vectors or semantic word embeddings are continuous dense vector representations of terms, for example Document to Vector (doc2vec) and Word to Vector (word2vec) [12].

- **(BOW):** Bag - Of - words model is one of the most basic and effective strategies for extracting feature vector from text documents. The core of this model is to translate text documents to vectors, with each document resulting in a vector that expresses the occurrence of all distinct terms present in the document vector space for that particular document [13].
- **(TF-IDF):** It is a widely used weighting technique. It is a mathematical tool for evaluating the value of a word to a text or a corpus. The value of a word increases as the number of occurrences in the file increases [14]. The term frequency (TF) in documents refers to the number of times a word appears in the text. The inverse document frequency (IDF) is an indicator of a word's overall value [15].
- **(word2vec):** Word2vec is a neural network that processes data using two layers. It is, however, not a deep neural network since deep neural networks have more layers than word2vec. The input data for word2vec is a text corpus, and the output data is a series of vectors. Word2vec converts general input corpus data into numerical form, making the data easier to interpret and evaluate. Word2vec has two primary models: the continuous bag-of-words (CBOW) model and the continuous skip gram model. The skip-gram predicts surrounding terms provided the current term, and the CBOW structure predicts the current word depending on the context (meaning) [16].

- **(doc2vec):** doc2vec is a word2vec extension for training text embeddings [17]. Inside doc2vec, there are two approaches: Distributed Memory version of Paragraph Vector (PV-DM) and Distributed Bag of Words version of Paragraph Vector (PV-DBOW). Word2Vec CBOW is similar to PV-DM. The doc-vectors are derived from training a neural network on the false predictions task for a center term based on an average of all background word-vectors, PV-DBOW is like a Word2Vec SG (skip-gram). It's also popular to mix PV-DBOW with testing of skip-gram, which predicts a single target word using both the doc-vector and nearby word-vectors, but just one at a period [18].

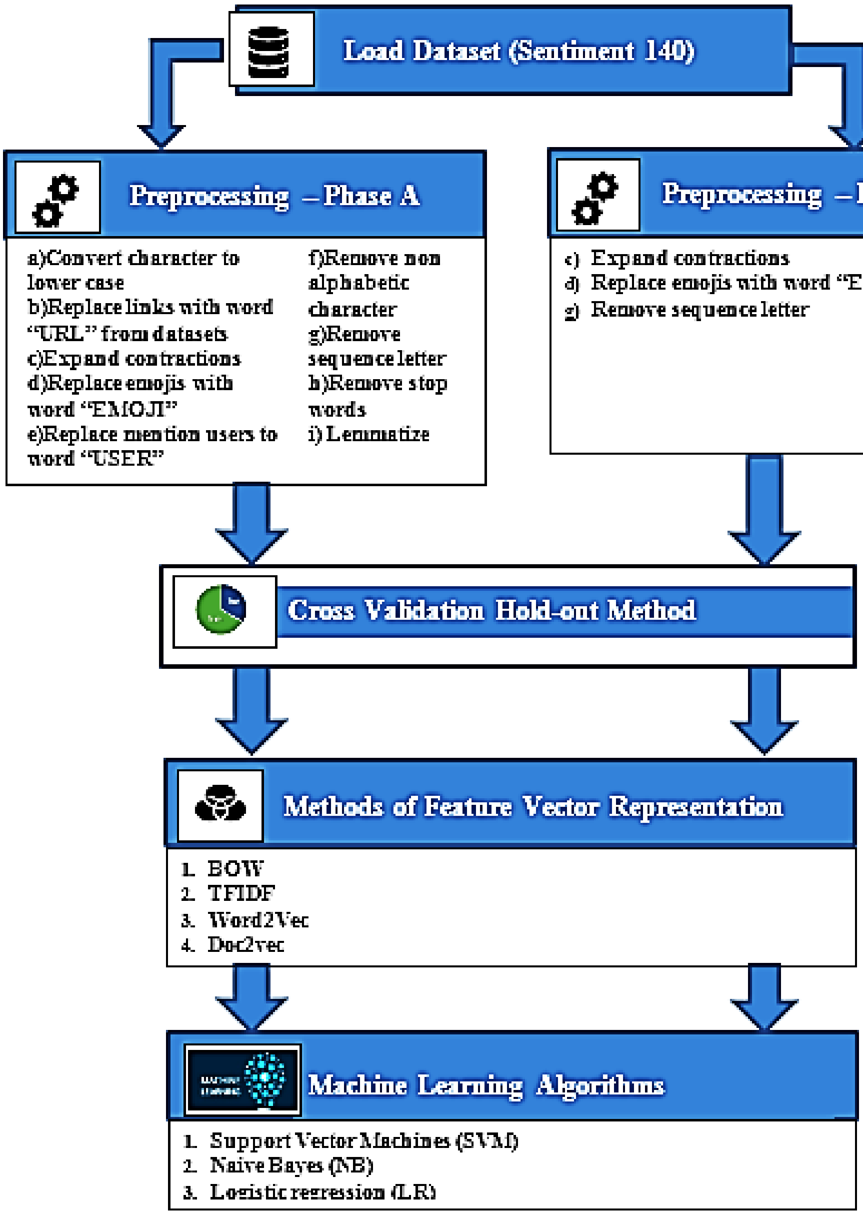
Load Dataset (Sentiment 140)
Preprocessing – Phase A
a)Convert character to lower case
b)Replace links with word "URL" from datasets
c)Expand contractions
d)Replace emojis with word "EMOJI"
e)Replace mention users to word "USER"
f)Remove non alphabetic character
g)Remove sequence letter
h)Remove stop words
i) Lemmatize
Preprocessing – Phase B
c) Expand contractions
d) Replace emojis with word "EMOJI"
g) Remove sequence letter
Cross Validation Hold-out Method
Methods of Feature Vector Representation
1. BOW
2. TFIDF
3. Word2Vec
4. Doc2vec
Machine Learning Algorithms
1. Support Vector Machines (SVM)
2. Naive Bayes (NB)
3. Logistic regression (LR)

Figure 1. The Proposed Framework for Twitter Sentiment Analysis using Machine Learning Approach.

Machine Learning Approach for Sentiment Analysis

A supervised machine learning classifier learns and trains using labeled training dataset with attention to the properties of text set, where the classifier's output is evaluated using the test dataset. For text classification, different types of machine learning algorithms like Support Vector Machines (SVM), Naive Bayes (NB) and Logistic regression (LR) are used in this paper.

- **SVM:** These are the most recent methods for supervised machine learning. Support Vector Machine (SVM) models have a lot in common with multilayer perceptron neural networks. The concept of a margin-either side of a hyperplane that divides two data classes-is central to SVMs. It has been demonstrated that increasing the margin and thereby establishing the maximum possible gap between the separating hyperplane and the instances on either side of it reduces the predicted generalization error which increase the accuracy [19].

- **NB:** A simple probabilistic classifier based on Bayes' theorem with strong independence assumptions. The most popular models are the Multivariate Bernoulli Occurrence Model and the multinomial model, which uses word instance rates. The multivariate Naive Bayes model is considered in this paper because it is typically superior to the binary individual model for text classification [20].
- **LR**: It is a classifier that learns what features from the input are the most valuable to distinguish between the various classes. Logistic Regression is a machine learning method that works by taking input and multiplying the input value with the weight value. Logistic Regression is a discriminative model that computes the probability of the class given set of features by discriminating among the different possible values of the class "y" based on the given input "x" [21].

4. Experimental Results and Discussion

Sentiment140 dataset [22] was used in this paper for training and testing the models, and it includes tweets on general topics. The dataset is one million and six hundred thousand tweets, which are divided equally into positive and negative sentiment, which means that it is a balanced dataset.

Hold-out cross validation is used to split the dataset into 80% training and 20% testing sets. Three different ML models are prepared and trained: NB, SVM, and LR. Four methods of feature vector representation are implemented: BOW, TFIDF, Word2vec and Doc2vec, with two mixes of preprocessing operations.

In Table 2 preprocess stage symbols are explained which refers to the initial treatment adopted in this paper.

Table 2. Preprocess symbols.

Preprocess	Symbols
Convert character to lower case	A
Replace links with word "URL"	B
Expand contractions	C
Replace emojis with word "EMOJI"	D
Replace mention with word "USER"	E
Remove non alphabetic character	F
Remove sequence letter	G
Remove stop words	H
Lemmatize	I

Parameters value in our experiment for each feature vector representation is as follows:

- BOW and TFIDF: Ngram = (1,2).
- Word2vec: Feature vector size = 300.
- Doc2vec: Feature vector size = 300.

Table 3 shows the results of applying Phase A, which includes all the preprocessing techniques (A-B-C-D-E-F-G-H-I) with different classifiers using different feature vector representation techniques.

Table 3. Classifiers' Accuracy Results after performing all the preprocessing techniques.

	SVM	NB	LR
TFIDF	0.81149102	0.79430152	0.81516617
BOW	0.82615355	0.80684844	**0.82760022**
Word2vec	0.75750532	0.72456358	0.75969560
Doc2vec	0.76472801	0.73843536	0.77640897

Using word2vec and doc2vec didn't enhance the performance of the classifiers; on the contrary, they give the worse results. The feature vector size can be the cause of these low results compared to other feature vector techniques. The size of the feature vector is set to a small number which is inadequate for the used dataset.

LR-BOW provides the best accuracy of 82.76%; since LR classifier works by taking a set of features for each tweet, and each feature is multiplied by the load or weight. LR can produce better results in recognizing and classifying text than any other ML methods like Naïve Bayes, Decision Tree, and others.

To demonstrate the effect of cleaning and preprocessing on the accuracy, only three techniques are applied in Phase B: removing sequence letters, replacing Emojis with the word Emoji, and expanding the contradiction. Table 4 shows the accuracy results after applying the three preprocessing techniques, specifically (C-D-G), on the Twitter dataset. It is clearly shown that the performance of all classifiers was improved regardless of the text feature representation. LR-BOW accuracy was increased from 82.76% to 83.95%. Preprocessing does affect the ML accuracy, and performing only three operations reduces the effect of dimensionality reduction. Table 5 shows the results of applying proposed method on real tweets from the same sentiment 140 dataset.

Table 4. Classifiers' Accuracy Results after performing CDG preprocessing techniques.

	SVM	NB	LR
TFIDF	0.82290208	0.80361875	0.82709583
BOW	0.83479245	0.81581346	**0.83954932**
Word2vec	0.77293127	0.76216842	0.77198372
Doc2vec	0.78005843	0.77160651	0.78716640

Table 5. LR-BOW classifier Output Results on a sample tweets from Sen140.

Tweet	Label
I hate twitter	Negative
May the Force be with you.	Positive
Mr. Stark, I don't feel so good	Negative
@ztnewetnorb he wanted to be my girlfriend	Negative
nice to see you	Positive
Exams are NOT good	Negative
this shall be fun	Positive
Sleeping in my bed at home	Positive
my head hurts really bad	Negative
My goldfish died!!!!	Negative

5. Conclusion

Due to the rapid and wide usage of social media, sentiment analysis has become an important topic to be studied and focused on. Twitter sentiment analysis aims to obtain opinions from tweets for business decision-making purposes; therefore, it has become a popular research topic. In this paper, the Machine learning approach was adopted on Sentiment140 dataset. NB, LR, and SVM models were implemented, and BOW, TF-IDF, doc2vec, and word2vec feature extraction methods are applied to extract and reduce features from the unstructured tweet text data. LR classifier produced a more accurate result since LR works best when the output is binary classification. The preprocessing step affects the quality of the generated feature vector, and this was concluded from Tables 3 and 4.

References

[1] Krommyda M.; Rigos A.; Bouklas K. and Amditis A.; "An Experimental Analysis of Data Annotation Methodologies for Emotion Detection in Short Text Posted on Social Media". Informatics, 8(1), 19, 2021.

[2] Manda K. R.; "Sentiment Analysis of Twitter Data Using Machine Learning and Deep Learning Methods". (June), 2019.

[3] Rodrigues A. P. and Chiplunkar N. N.; "A new big data approach for topic classification and sentiment analysis of Twitter data". Evol. Intell.; 2019.

[4] Sailunaz K. and Alhajj R.; "Emotion and sentiment analysis from Twitter text". J. Comput. Sci.; 2019.

[5] Hasan A.; Moin S.; Karim A. and S. Shamshirband, "Machine Learning-Based Sentiment Analysis for Twitter Accounts". Math. Comput. Appl.; 2018.

[6] Malik V. and Kumar A.; "Analysis of Twitter Data Using Deep Learning Approach: LSTM". Int. J. Recent Innov. Trends Comput. Commun.; 2018.

[7] Siddiqua U. A.; Ahsan T. and Chyy A. N.; "Combining a Rule-based Classifier with Weakly Supervised Learning for Twitter Sentiment Analysis". 2017.

[8] Joshi R. and Tekchandani R.; "Comparative analysis of twitter data using supervised classifiers". 2016.

[9] Jain A. P. and Dandannavar P.; "Application of machine learning techniques to sentiment analysis". 2017.

[10] Yadav N.; Kudale O.; Rao A.; Gupta S. and Shitole A.; "Twitter Sentiment Analysis Using Supervised Machine Learning". Lect. Notes Data Eng. Commun. Technol.; 57(March), 631-642, 2021.

[11] Zheng A. and Casari A.; Feature engineering for machine learning, (September), 2018.

[12] Uysal A. K. and Murphey Y. L.; "Sentiment Classification: Feature Selection Based Approaches Versus Deep Learning". 2017.

[13] Canedo E. D. and Mendes B. C.; "Software requirements classification using machine learning algorithms". Entropy, 2020.

[14] Zhang W.; Yoshida T. and Tang X.; "A comparative study of TF*IDF, LSI and multi-words for text classification". Expert Syst. Appl.; 2011.

[15] Yuan H.; Tang Y.; Sun W. and Liu L.; "A detection method for android application security based on TF-IDF and machine learning". PLoS One, 2020.

[16] Zhang D.; Xu H.; Su Z. and Xu Y.; "Chinese comments sentiment classification based on word2vec and SVMperf". Expert Syst. Appl.; 2015.

[17] Le Q. and Mikolov T.; "Distributed representations of sentences and documents". 2014.

[18] Lau J. H. and Baldwin T.; "An Empirical Evaluation of doc2vec with Practical Insights into Document Embedding Generation". 2016.

[19] Osisanwo F. Y.; Akinsola J. E. T.; Awodele O.; Hinmikaiye J. O.; Olakanmi O. and Akinjobi J.; "Supervised Machine Learning Algorithms: Classification and Comparison". Int. J. Comput. Trends Technol.; 2017.

[20] Qiang G.; "An effective algorithm for improving the performance of Naive Bayes for text classification". 2010.

[21] Indra S. T.; Wikarsa L. and Turang R.; "Using logistic regression method to classify tweets into the selected topics". 2016 Int. Conf. Adv. Comput. Sci. Inf. Syst. ICACSIS 2016, (October), 385-390, 2017.

[22] Go A.; Bhayani R. and Huang L.; "Twitter Sentiment Classification using Distant Supervision". CS224N project report, Stanford, 1(12), 1-6, 2009.