IJDACR
ISSN: 2319-4863

IJDACR
International Journal Of Digital Application & Contemporary Research

**International Journal of Digital Application & Contemporary Research**
**Website: www.ijdacr.com (Volume 8, Issue 02, September 2019)**

# Sentiment Analysis on Twitter Data using SVM Classifier

Kanak Jagwani Ishi Raghuvanshi Janhvi Sharma

*Abstract* – **Analyzing the large volumes of data generated in social networks on public opinion about different topics can result in valuable discoveries. These activities are expensive to perform manually, they require many human resources and time. Sentiment analysis systems and data mining algorithms have proved to be very useful in order to obtain a general perception of the topics of interest and the opinion on them. In this paper we propose to analyze a set of data using a sentiment classifier to label publications made by users of social networks in conjunction with clustering algorithms to be able to detect which are the topics on which opinions are expressed. We propose to use a base of 2000 reviews of films labeled as positive and negative and then train support vector machine (SVM) classifier of sentiments. We performed our experiments using one thousand tweets. Experimental evaluations show that our proposed technique is more efficient and has higher accuracy compared to previously proposed methods.**

*Keywords* – **Machine Learning, Sentiment Analysis, SVM, Web 2.0.**

## I. INTRODUCTION

With the emergence of Web 2.0, users have the possibility to generate their own content and share it more easily. In this boom, social networks have gained great popularity, particularly the Twitter microblogging platform that allows its users to share text messages in 140 characters with their family, friends and followers. Daily more than 500 million messages are published, commonly called tweets. Because the main reason for these publications is to express the point of view and opinion of the users, they turn out to be of great interest to be analyzed.

For the exploitation of all these data that circulate publicly on the web we can perform various tasks that allow us to extract useful information from opinions. This work area, referred to as opinion mining, focuses on the automatic treatment of texts in which the opinions, sentiments, emotions and attitudes of people towards certain issues and its aspects are reflected.

Knowing what your users think may have different applications, such as recommending products or determining which political candidate will be voted in the next elections.

An especially interesting objective from the point of view of information extraction and knowledge representation is to classify positively or negatively, according to the opinion of different users, the different aspects of an entity and know what motivates such opinions. Being able to generate a solution to this complex problem requires great knowledge of the domain and can be transferred to the proposed software solution, with a great development effort.

However, in order to carry out this type of analysis we find natural language features that make it a striking area of research. They present with problems of ambiguity or semantic vagueness that are reflected, for example, at the time of detecting sarcasm and the intentions of the author of a sentence. [1]

Also, among the most common algorithms are those that use a set of language words, called lexicon [2], composed of positive words (good, great, excellent) and negative words (ugly, unpleasant , horrible) to be able to label the sentiment of a prayer. Depending on the context in which these words are used, they may have different meanings, they may not even express a particular sentiment. This makes it necessary to use common sense to be able to recognize the sentiment that the sentence conveys.

Sometimes, not only words but also the same expressions may have associated sentiments in different contexts. For example, the comment "Does not emit any sound" could be considered positive when talking about a new car, but it could also indicate negative if we were talking about a music equipment. This makes it necessary to know what we are analyzing and the particular implications of the subject [3, 4].

The same applies to sentences that only present facts: the specific knowledge of the domain allows the extraction of opinions even though a positive or negative sentiment is not transmitted [5].

In order to obtain a quick estimate of how public opinion is about a topic, we can group similar opinions into clusters that speak of the same aspect. Although it is a less expensive solution to approach the analysis of sentiments based on aspects, this solution will not be so adequate compared to a

IJDACR
International Journal Of Digital Application & Contemporary Research

**International Journal of Digital Application & Contemporary Research**
**Website: www.ijdacr.com (Volume 8, Issue 02, September 2019)**

system where large amounts of knowledge about the domain have been injected, but it is an alternative to make a first exploratory approach to the data. Finally, monitoring large volumes of data and identifying the most relevant portions to extract discoveries in a summary way is an expensive procedure to be performed manually. For this it is common to perform an exploratory analysis of the data through visualizations.

Visualizations allow us to summarize large volumes of data in graphic representations. Subsequently, an expert can quickly interpret them and draw conclusions that encompass all the data. Then, you can make a decision based on the information collected.

Our training data consists of generic (not topic-specific) Twitter messages with emoticons, which are used as noisy labels. We show that the accuracy obtained on a training dataset comprising 100K tweets, and a test dataset of 5000 tweets gives an accuracy of around 80% on the following classifiers: Naive Bayes, RBF-kernel Support Vector Machine, and Logistic Regression. Our model takes roughly half the time to train and achieves higher accuracy (than the baseline model) on all the classifiers. Because the amount of training time is expected to increase exponentially as the training data increases, we expect our model to outperform (in terms of higher accuracy) the baseline model at a speed which is at least twofold the speed of the baseline model on larger datasets.

## II. Sentiment Analysis

To identify opinions on the Internet, it is necessary to perform a sentiment analysis, a technique that uses language processing, text analysis and computational tools to classify subjective comments of different users, whether they are such sentiments or opinions on various topics. The methods used for this type of analysis have about 15 years of application, which have been used to classify mails, customer reviews, digital publications, etc.

When you want to design a system that analyzes and classifies sentiments or opinions, you must first be clear about the challenges that must be overcome, which are described in the literature [6].

- In the first place it is necessary to determine if there is an opinion in the tweet or not, since this does not always happen, being an objective comment, a response to another user, etc.
- Determine the topic on which you are talking in order to know if it is useful information, since you may be seeking opinions about a particular company and if the tweet is about politics, it does not provide relevant information about what you are looking for.
- Recognize typical abbreviations and idioms. With Twitter being an informal character, the language used is not always correct, since normally no tildes are used and popular words are used that do not appear in the dictionary (Ex. Occupy “bn” instead of “good”, “x” in instead of "by", the use of scribbles, using expressions like "po", "malooooo", etc.).
- Determine the polarity of a sentence and can have positive and negative words in the same sentence (Ex. "I'm glad it's over, the show is bad", "The movie was not good at all").

The tweets to evaluate are all those who give an opinion, an evaluation or express emotion on a topic of interest, leaving aside objective or informative messages. This is how there are several methods that can be applied to perform a sentiment analysis on Twitter. In general, this type of problem is solved by cataloging an opinion in polarities, determining whether it is positive or negative regarding a specific issue. However, this is not a simple matter to solve, since depending on the context there are words that can express both a positive and negative opinion.

In the case of having 2 polarities, which is more used in the literature, each message can be classified as positive or negative. This method includes studies on different topics, such as the case of extracting opinions in movie or book reviews ("good" or "bad"), in the opinion of products ("I like" or "I don't like") or in political elections ("will win" , "It will not win").

In addition, 6 universal emotions have been commonly considered [7] [8] [9] [10]: anger, disgust, fear, joy, sadness and surprise. In this way it is possible to catalog the tweets according to these emotions in order to determine their polarity and the degree of this, which can be very useful to differentiate messages in a greater number of categories and not only positive-negative. In the same way, it is possible to make a gradual scale between positive-negative, being able to have 7 degrees (3 positive, 1 neutral and 3 negative, varying from very negative to very positive) or 11 degrees (-5 to 5, being -5 very negative, 0 neutral and +5 very positive).

To obtain the polarity of a tweet there are 2 most used methods. The first is to use machine-learning approaches and the second is to use lexical dictionaries. This paper presents a machine learning based approach for sentiments analysis of Twitter data.

IJDACR
International Journal Of Digital Application & Contemporary Research

**International Journal of Digital Application & Contemporary Research**
**Website: www.ijdacr.com (Volume 8, Issue 02, September 2019)**

### *A. Machine Learning*

Machine learning seeks to analyze the information automatically in a supervised way, based on training sets, which will be used to catalog the rest of the opinions found on the web, conducting tests and then validating them. The main techniques of this method are: Support Vector Machines (SVMs), Naive Bayes, and maximum entropy classifiers. In this way, the grammatical category of words, the presence and frequency of some terms and their semantic composition [11] are used.

Most of these methods, however, are accompanied by a dictionary that provides information a priori of the terms to obtain the respective polarities. In some cases, these dictionaries are made by people [12] and in others a semi-automatic system [13] is used.

## III. Proposed Methodology

In this section, we explain the various pre-processing techniques used for feature reduction, and also the additional step of filtering the training dataset using the subjectivity score of tweets. We further describe our approach of using different machine learning classifiers and feature extractors. We also propose an additional heuristic for sentiment classification which can be used as a tag-along with the learning heuristics [14].

### *A. Corpus*

Our training dataset1 has 1.6 million tweets, and 5000 tweets in the test dataset. Since the test dataset provided comprised only 500 tweets, we have taken part of the training data (exactly 5000 tweets, distinct from the training dataset) as the test dataset.

### *B. Subjectivity Filtering*

This is a new step we propose to achieve higher accuracy on a smaller training dataset. We use TextBlob to classify each tweet as subjective or objective. We then remove all tweets which have a subjectivity level/score (score lies between 0 and 1) below a specified threshold. The remaining tweets are used for training purposes. We observe that a considerable number of tweets are removed as the subjectivity threshold increases. We show the effect of doing this procedure on the overall accuracy in the evaluation section of the paper [14].

### *C. Preprocessing*

The Twitter language model has many unique properties. We take advantage of the following properties to reduce the feature space. Most of the preprocessing steps are common to most of the previous works in the field. However, we have added some more steps to this stage of our model [14].

1. Basic steps: We first strip off the emoticons from the data. Users often include Twitter usernames in their tweets in order to direct their messages. We also strip off usernames (e.g. @Chinmay) and URLs present in tweets because they do not help us in sentiment classification. Apart from full stops, which are dealt in the next point, other punctuations and special symbols are also removed. Repeated whitespaces are replaced with a single space. We also perform stemming to reduce the size of the feature space.
2. Full Stops: In the previous works, full stops are just usually replaced by a space. However, we have observed that casual language in tweets is often seen in form of repeated punctuations. For example, "this is so cool...wow". We take into consideration this format, and replace two or more occurrences of "." and "-" with a space. Also, full stops are also quite different in usage. Sometimes, there isn't any space in between sentences. For example, "It's raining. Feeling awesome". We replace a single occurrence of a full stop with a space to ensure correct feature incorporation.
3. Parsing Hashtags: In the case of hashtags, most of the previous works just consider the case of hashtags followed by a single word; they just remove the hashtag and add the word to the feature vector. However, sometimes, there are multiple words after a hashtag, and more often than not, these words form an important, conclusive part of the Tweet. For example, #ThisSucks, or #BestMomentEver. These hashtags need to be dealt with in a proper fashion. We split the text after hashtags after before each capital letter, and add these as tokens to the feature vector. For hashtags followed by a single word, we just replace the pattern #word with the word, as conventional models do. The intuition behind this step is that quite often, the sentiment of a tweet is expressed in form of a hashtag. For example, #happy or #disappointed are frequently used hashtags, and we don't want to lose this information during sentiment classification.
4. Repeated letters: Tweets contain very casual language as mentioned earlier. For example, if we search "wow" with an arbitrary number of o's in the middle (e.g. wooow, woooow) on Twitter, there will most likely be a non-empty result set. We use preprocessing so that any letter occurring more than two times in a row is replaced with two occurrences. In the samples above, these words would be

IJDACR
International Journal Of Digital Application & Contemporary Research

**International Journal of Digital Application & Contemporary Research**
**Website: www.ijdacr.com (Volume 8, Issue 02, September 2019)**

converted into the token "woow". After all the above modifications, tweets are converted into lowercase to avoid confusion between features having same content, but are different in capitalization.

5. Stopwords, Acronyms and Negations: We gather a list of 400 stopwords. These words, if present in the tweets, are not considered in the feature vector.
   We store an acronym dictionary which has over 5000, frequently-used acronyms and their abbreviations. We replace such acronyms in tweets with their abbreviation, since these can be of great use while sentiment classification.
   All negative words like 'cannot', 'can't', 'won't', 'don't' are replaced by 'not', which effectively keeps the sentiment stable. It is observed that doing this makes the training faster, since the model has to deal with a smaller feature vector.

### *D. Baseline Model*

The baseline model for our experiments is explained in the paper by Alec Go [15]. The model uses the Naive Bayes, SVM, and the Maximum Entropy classifiers for their experiment. Their feature vector is either composed of Unigrams, Bigrams, Unigrams + Bigrams, or Unigrams + POS tags.
This work achieved the following maximum accuracies:

- 82.2 for the Unigram feature vector, using the SVM classifier,
- 83.0 for the Unigram + Bigram feature vector, using the MaxEnt classifier, and 82.7 using the Naive Bayes classifier.
- 81.9 for the Unigram + POS feature vector, using the SVM classifier.

These baseline accuracies were on a training dataset of 1.6 million tweets, and a test dataset of 500 tweets. We are using the same training dataset for our experiments. We later present the baseline accuracies on a training set of 1K tweets, and a test dataset of 5000 tweets; we compare our model's accuracy with these baseline accuracy values on the same test data of 5000 tweets.

### *E. Classifications using SVM*

Support vector machine (SVM) is highly used in the classification and detection of sentiments. SVM is based on kernel methods, which take the data and put it into an appropriate feature space. In this way they use linear algorithms to determine non-linear patterns. The method is based mainly on vectors where, using computational learning, it manages to make boundary decisions between two categories, separating them as much as possible [16], as seen in Figure 1.

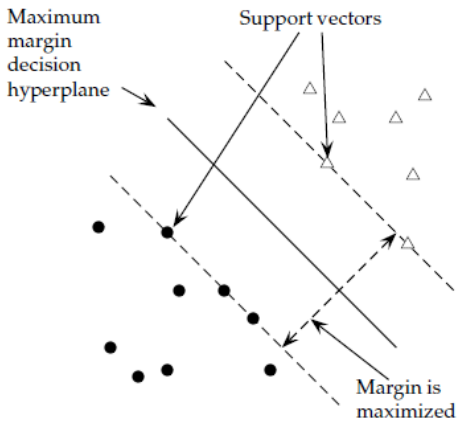


Figure 1: Vectors are separated by the hyperplane maximizing the separation between classes [17]

SVM sets the criterion of separation between classes that is as far as possible from any data. This distance, from the decision point, to the nearest point is the margin of the classifier. Thus, as the method is defined by a decision function that involves a subset of features or data (support vectors) that will define the position of the separator [16]. In this way, the decision of the limit or margin is quite important since the data that remain around it will have a lower probability of being cataloged correctly.
Algebraically, one can define a vector perpendicular to the hyperplane $\vec{\omega}$ which is known as the weight vector. To determine a single hyperplane, an intersection term $b$ is specified. Thus, all terms of the hyperplane $\vec{\omega}$ satisfy $\vec{\omega}^T\vec{x} = -b$, since the hyperplane is perpendicular to the normal vector $\vec{\omega}$ [16].
Then, to make the decisions between both classes, generally the classes can be defined with +1 and -1, $\vec{\omega}^T\vec{x}$ is calculated and compared with $b$ to determine which side of the hyperplane is $\vec{x}$, so that $f(\vec{x}) = sign\,(\vec{\omega}^T\vec{x} + b)$, gives us the expected classification (+1 or -1) [16]. On the other hand, if the new data $\vec{x}$ is very close to the hyper-plane of separation, usually neither of the two categories is assigned, which is done by setting a distance limit. Finally $f(\vec{x})$ can be transformed into a probability of classification in order to make decisions between classes.
This method was updated and used for text classification by Joachims in 1999 [18]. In this way, there is a training set where each sample has a weight and an associated vector that separates as

IJDACR
International Journal Of Digital Application & Contemporary Research

**International Journal of Digital Application & Contemporary Research**
**Website: www.ijdacr.com (Volume 8, Issue 02, September 2019)**

much as possible the positive cases from the negative ones. Generally, the data used are words (unigrams) to which a weight is assigned during the learning phase with the value $\delta \geq 0$. Each word labeled that meets its weight $\delta > 0$ is called support vector. In this way, the support vectors separate the hyperplane between the positive and negative classification. Thus, words that have not yet been trained are assigned to the nearest support vectors according to an equation that includes the appropriate kernel function.

To select the features to occupy in SVM correctly there are several methods. Usually single words are used that are used a certain number of times in the text to be analyzed. It is also possible to select bi-grams (two words together), tri-grams (3 words together), the grammatical category of the word, etc. This method is widely used in the classification of sentiments, which has had excellent results on both Twitter and other platforms on the web [11], achieving success in more than 70% of cases.

## IV. Simulation And Results

The performance of proposed algorithms has been studied by means of MATLAB simulation.

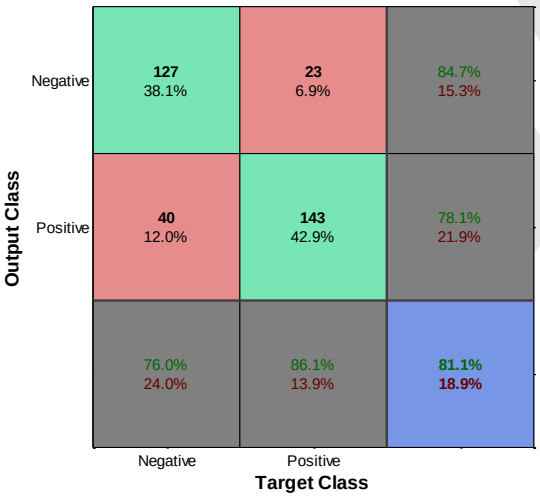


Figure 2: Confusion matrix plot for proposed sentiment analysis using SVM classifier

Here, TP=127, TN=143, FP=23 and FN=40

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} = \frac{127+143}{127+143+23+40} = 81.1\%$$
$$Precision = \frac{TP}{TP+FP} = \frac{127}{127+23} = 84.7\%$$
$$Sensitivity = \frac{TP}{TP+FN} = \frac{127}{127+40} = 76\%$$

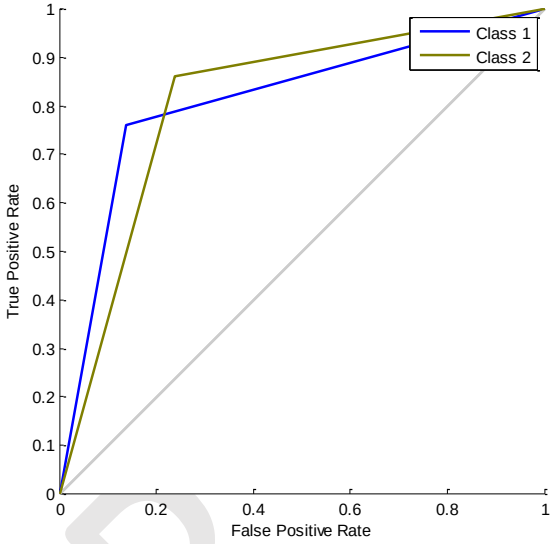


Figure 3: ROC curve

## V. Conclusion

In this work, we have studied a complex problem such as the mining of opinions on Twitter and the challenges associated with the exploratory analysis of the results of this process. Although Twitter has certain limitations when it comes to providing information, it is more than enough to carry out an exploratory analysis in order to better understand a market or an event.

We show that a higher accuracy can be obtained in sentiment classification of Twitter messages training on a smaller dataset and with a much faster computation time, and hence the issue of constraint on computation power is resolved to a certain extent. As Twitter data is abundant, our subjectivity filtering process can achieve a better generalised model for sentiment classification.

## Reference

[1] González-Ibánez, Roberto, Smaranda Muresan, and Nina Wacholder. "Identifying sarcasm in Twitter: a closer look." In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Short Papers-Volume 2*, pp. 581-586. Association for Computational Linguistics, 2011.

[2] Yang, Changhua, Kevin Hsin-Yih Lin, and Hsin-Hsi Chen. "Building emotion lexicon from weblog corpora." In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pp. 133-136. 2007.

[3] Gamon, Michael, Anthony Aue, Simon Corston-Oliver, and Eric Ringger. "Pulse: Mining customer opinions from free text." In *international symposium on intelligent data analysis*, pp. 121-132. Springer, Berlin, Heidelberg, 2005.

[4] Blitzer, John, Mark Dredze, and Fernando Pereira. "Biographies, bollywood, boom-boxes and blenders:

IJDACR
International Journal Of Digital Application & Contemporary Research

**International Journal of Digital Application & Contemporary Research**
**Website: www.ijdacr.com (Volume 8, Issue 02, September 2019)**

Domain adaptation for sentiment classification." In *Proceedings of the 45th annual meeting of the association of computational linguistics*, pp. 440-447. 2007.

[5] Zhang, Lei, and Bing Liu. "Identifying noun product features that imply opinions." In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*, pp. 575-580. Association for Computational Linguistics, 2011.

[6] Pang, Bo, and Lillian Lee. "Opinion mining and sentiment analysis." *Foundations and Trends® in Information Retrieval* 2, no. 1–2 (2008): 1-135.

[7] Ekman, Paul. "Emotion in the human face: Studies in emotion and social interaction." (1982).

[8] Alm, Cecilia Ovesdotter, Dan Roth, and Richard Sproat. "Emotions from text: machine learning for text-based emotion prediction." In *Proceedings of the conference on human language technology and empirical methods in natural language processing*, pp. 579-586. Association for Computational Linguistics, 2005.

[9] Subasic, Pero, and Alison Huettner. "Affect analysis of text using fuzzy semantic typing." *IEEE Transactions on Fuzzy systems* 9, no. 4 (2001): 483-496.

[10] Liu, Bing. "Sentiment analysis and subjectivity." *Handbook of natural language processing* 2, no. 2010 (2010): 627-666.

[11] Paltoglou, Georgios, and Mike Thelwall. "Twitter, MySpace, Digg: Unsupervised sentiment analysis in social media." *ACM Transactions on Intelligent Systems and Technology (TIST)* 3, no. 4 (2012): 66.

[12] Zaidan, Omar, Jason Eisner, and Christine Piatko. "Using "annotator rationales" to improve machine learning for text categorization." In *Human language technologies 2007: The conference of the North American chapter of the association for computational linguistics; proceedings of the main conference*, pp. 260-267. 2007.

[13] Whitelaw, Casey, Navendu Garg, and Shlomo Argamon. "Using appraisal groups for sentiment analysis." In *Proceedings of the 14th ACM international conference on Information and knowledge management*, pp. 625-631. ACM, 2005.

[14] Sahni, Tapan, Chinmay Chandak, Naveen Reddy Chedeti, and Manish Singh. "Efficient Twitter sentiment classification using subjective distant supervision." In *2017 9th International Conference on Communication Systems and Networks (COMSNETS)*, pp. 548-553. IEEE, 2017.

[15] Go, Alec, Richa Bhayani, and Lei Huang. "Twitter sentiment classification using distant supervision." *CS224N Project Report, Stanford* 1, no. 12 (2009): 2009.

[16] Manning, Christopher D., Prabhakar Raghavan, and Hinrich Schutze. "Introduction to Information Retrieval? Cambridge University Press 2008." *Ch* 20: 405-416.

[17] Guenther, Nick, and Matthias Schonlau. "Support vector machines." *The Stata Journal* 16, no. 4 (2016): 917-937.

[18] Kennedy, Alistair, and Diana Inkpen. "Sentiment classification of movie reviews using contextual valence shifters." *Computational intelligence* 22, no. 2 (2006): 110-125.