Tweet Sentiment Analysis and Study of Preprocessing Methods and Various Algorithms for Prediction

Aditya Kulkarni
Pune, India
adityakulkarni9@acm.org

Shubham Mhaske
Pune, India
mhaskes@acm.org

***Abstract*—Starting from March 2006, Twitter has been a major face of social media. As Twitter provides a swift and efficient way for users to share data in textual, pictorial and video form, a massive volume of data is generated by the platform. Users share multiple aspects of personal and public opinions on various events happening around them. This social interaction has both positive and negative aspects associated with it. This article explains, in detail, the processes used to draw conclusions from the data (in the form of tweets) and classify them as positive, negative and neutral. This can help in reducing the rate of online harassment and hate speech.Unfortunately, hate speech has been a major issue and as a consequence one of the major drawback of social media platforms. Despite numerous attempts, a perfect is hard to develop due to the vague definition of hate speech and the intent of the writer is not always accurately reflected in the tweet. This project was developed to compare and evaluate various methods of preprocessing of data, feature selection, and predictive algorithms.**

***Index Terms*—Tweet sentiment Analysis, preprocessing , NLP,stopwords,data cleansing,word-count,wordcloud,nltk**

I. Introduction

Millions of people are using social network sites to express their emotions, opinions and disclose various aspects of their daily lives. By analyzing these tweets various conclusions can be drawn about the current state of person and society regarding the issue that is addressed in the tweet. This research explores the methodology of tweet analysis to get the sentiment from the tweet and apply this knowledge to predict the sentiment from future tweets.
This study explains the logic and process behind building a model that classifies tweets as positive, negative or neutral. The main goal of this project is to define and explore the impact of various methods of preprocessing and various machine learning algorithms that can be used and their respective impact on accuracy and performance. A model sample tweet dataset is used for the same.

A. Objective

The objective of this research is to detect hate speech in tweets. For the sake of simplicity, we say a tweet contains hate speech if it has a racist or sexist sentiment associated with it. So, the task is to classify racist or sexist tweets from other tweets. Hate speech is an unfortunately common occurrence on the Internet. Often social media sites like Facebook and Twitter face the problem of identifying and censoring problematic posts while weighing the right to freedom of speech. The importance of detecting and moderating hate speech is evident from the strong connection between hate speech and actual hate crimes. Early identification of users promoting hate speech could enable outreach programs that attempt to prevent an escalation from speech to action. Sites such as Twitter and Facebook have been seeking to actively combat hate speech. In spite of these reasons, NLP research on hate speech has been very limited, primarily due to the lack of a general definition of hate speech, an analysis of its demographic influences, and an investigation of the most effective features. In this article, various approaches to classify text as hate speech will be evaluated.

B. Undertstanding the dataset

The dataset used in this project is a labeled dataset of 31,962 tweets. The dataset is provided in the form of a CSV file with each line storing a tweet id, its label, and the tweet. In the given dataset, label '1' denotes the tweet is racist/sexist and label '0' denotes the tweet is not racist/sexist. The tweet can be downloaded from the tweet sentiment analysis competition hosted on AnalyticsVidhya web-portal. [1]

II. Methodolgy

As majority of this project deals with data in textual form, a large use of NLP concepts is done. The project is broadly divided in 5 steps. The steps are listed below:

1) Data Preprocessing and Cleaning: This step involves making data compatible with the input data format of the models.
2) Extracting Features from Cleaned Tweets: After preprocessing data, important features are identified and weighed to create a more efficient model.
3) Model Building: Sentiment Analysis: Based on the nature of data and compatibility, a machine learning algorithm is selected and a model is created.
4) Applying Ensemble Learning Methods: Various ensemble methods like bagging and boosting are applied and variation in performance is observed.

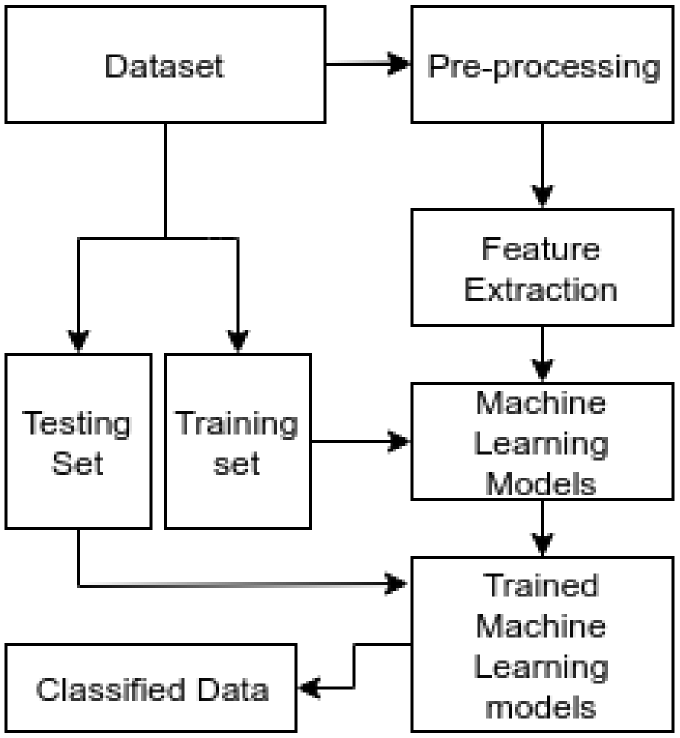


Fig. 1. Sentiment analysis methodology

III. Preprocessing of data

The data present on twitter is mostly in textual form. The difficulty with analyzing this data is that it contains multiple unwanted features or elements, like twitter handles, emoticons and multiple exceptional characters which may or may not have any effect on the meaning of the tweet.
Thus, preprocessing step has high importance in tweet sentiment analysis.Preprocessing the data requires the use of Regular Expressions, NLP libraries to remove stop-words and stemming or lemmatizing algorithms. The major tasks in preprocessing are listed below:

A. Removing Twitter user handles(IDs), Punctuations, Numbers, and Special Characters

In the dataset, the Twitter handles are masked as @user due to privacy concerns. So, these Twitter handles are hardly giving any information about the nature of the tweet, thus there is a need for removing them. The tweets also consist of a large number of symbols like punctuation symbols and numbers. Both these text elements provide very less relevance to the definition of the tweet so it is safe and beneficial to remove these elements. To remove these elements, Regular expressions will be used.

- Regular Expressions:
  A regular expression(RegEx) is a pattern describing a certain amount of text, called a search pattern. Regular expressions can be used to check whether a text block has the search pattern or not. Regular Expressions are defined using various symbols that vary depending on the regex engine. In this project, the python library 're' is used to define regular expressions.

B. Tokenization

Text is a linear sequence of symbols. The first step in the actual processing of text is segmentation of the block into linguistic units such as words, punctuation, numbers, alphanumerics, etc. This process is called tokenization. Tokenization is a kind of pre-processing in a sense; an identification of basic units to be processed. It is conventional to concentrate on pure analysis or generation while considering basic units. The decision of the basic unit must be taken depending on the need of the application. Most of the applications require tokens in the form of singular words. In tweet analysis, the use of words is an important factor in the determination of the overall sentiment of a tweet. There is no specific function dedicated to tokeniztion, it must be defined as per the need of the application. String and List manipulation functions and slicing possess an important role in writing a tokenization function.

C. Removing Short Words and Stop-words

In most of the tokenized text blocks, the words with length less than or equal to 3 pose no importance. For example, the, oh, lol, etc. So removing these words will reduce additional unnecessary computations.
Along with short words, stop words are also removed from the text. Stop words are the words that are the most commonly used in natural languages; there is no single universal list of stop words used. Any set of words can be chosen as the stop words for a given purpose. In the context of this project, most auxiliary verbs, pronouns and other words with little lexical content like: too, also, just, etc. are considered as stopwords. To remove stopwords, we use the colection of stopwords present in the nltk.corpus package. This package has collection of stopwords from all prominent languages.

D. Stemming and Lemmatizing

- Stemming:
  The process of converting words to their word stems is called stemming. A word stem may not be the same as a morphological root, it is just equal or smaller form of the word. Stemming algorithms are rule-based. It can be viewed as a heuristic process that transforms the words in its smallest possible form i.e. stem. A set of conditional statements are applied to a word and its stem is determined by matching it through multiple conditions.Some of the commonly used stemming algorithms are Porter stemming algorithm and Snowball stemming algorithm.
  Porter stemming algorithm: This stemming algorithm is an older one. It's from the 1980s and its main concern is removing the common endings to words so that they can be resolved to the stem form. It is a very simple algorithm with very little computational complexity. But it should not be used in complex applications as it may provide inaccurate results as it only trims the words.

Snowball stemming algorithm: This algorithm is also known as the Porter2 stemming algorithm. It is almost universally accepted as better than the Porter stemmer. That being said, it is also more aggressive than the Porter stemmer. A lot of the things added to the Snowball stemmer were because of issues noticed with the Porter stemmer. There is about a 5% difference in the way that Snowball stems versus Porter.

- Lemmatizing:
  Lemma is a term used to represent all the other possible forms of the word. The lemma ”build,” for example, reflects ”builds”, ”building”, ”built”, etc.Lemmatization is the process of converting words into their lemma. Lemmatization uses word vocabulary and morphological analysis to reduce the different forms of the word to it’s dictionary form.To solve a word to his lemma, a lemmatizer needs to know much more about a language’s structure and therefore requires extra computational linguistic power.Algorithms for lemmatization can be either simple dictionary-based or rule-based . Rules can either be handcrafted or learned automatically from an annotated corpus for the rule-based lemmatizers.
- Comparison of Stemming and Lemmatizing:
  Lemmatization uses a language dictionary to perform an accurate reduction to root words. Stemming uses simple pattern matching to simply strip suffixes of tokens (e.g. remove “s”, remove “ing”, etc.).Lemmatization is strongly preferred to stemming if available.

IV. Feature Extraction from data Using TF/IDF

Term frequency-Inverse document frequency uses all the tokens in the dataset as vocabulary.Frequency of occurrence of a token from vocabulary in each document consists of the term frequency and number of documents in which token occurs determines the Inverse document frequency.What this ensures is that,if a token occurs frequently in a document that token will have high TF but if that token occurs frequently in majority of documents then it reduces the IDF ,so stop words like an,the,i which occur frequently are penalized and important words which contain the essence of document get a boost.Both these TF and IDF matrices for a particular document are multiplied and normalized to form TF-IDF of a document.

$$w_{i,j} = tf_{i,j}.log\left(\frac{N}{tf_{i,j}}\right)$$

V. Model Building: Sentiment Analysis

A. Logistic Regression

Logistic regression is a statistical model used to model a binary dependent variable using a logistic function and it is widely used for classification problems.A linear equation with independent predictors is used by the logistic regression algorithm to predict a value that is a real number, which is squashed to class value (0 or 1) using the Sigmoid function [2].

$$\sigma(z) = \frac{1}{1+e^{-z}}$$

Here z is the value we want to transform to the class value.Sigmoid function for the Logistic regression is given below in which z is linear function in a univariate regression model and y is the output variable.

$$y = \frac{1}{1+e^{-(\beta 0+\beta 1x)}}$$

B. Decision Tree

C. k-Nearest Neighbours Classifier

The k-Nearest-Neighbours (kNN) is a simple non-parametric classification method. For a data record to be classified, its ’k’ nearest neighbours are retrieved, and this forms a neighbourhood for the particular data record.The classification of the data item is done based on the voting among the records in the neighbourhood. The classification can be done with or without consideration of distance-based weighting [3]. k-NN is a type of instance-based learning, or lazy learning, where the function is only approximated locally and all computation is deferred until classification. This signifies that the model does not actually ”learn” anything but computes all the required metrics for every instance.
However, to apply kNN we need to choose an appropriate value for k, and the success of classification is very much dependent on this value. In a sense, the kNN method is biased by k. There are many ways of choosing the k value, but a simple one is to run the algorithm many times with different k values and choose the one with the best performance.

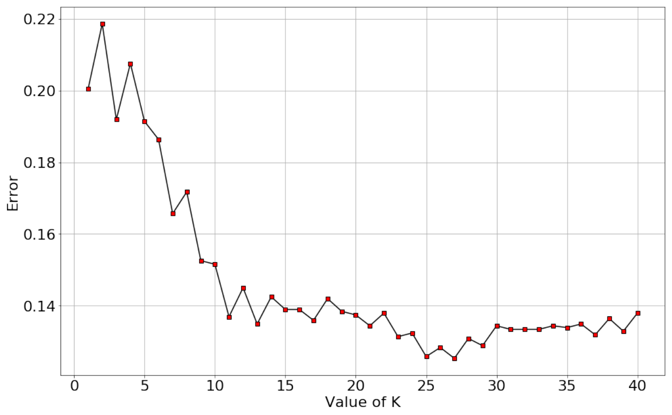


Fig. 2. The variation in value of error with diffrent values for ’k’.This is a result of application of the algorithm on a small block of data from training set with different k values.

D. Support Vector Machine Classifier

A Support Vector Machine (SVM) is a discriminative classifier. For a given supervised learning instance the SVM gives an optimal hyperplane as output which is used to categorize new data records. In two dimentional space this hyperplane is a line dividing a plane in two parts where in each class lay in either side.
The SVM algorithms are implemented using a kernel. The kernel is responsible to transform the input data into the required form. The SVM kernel takes a low dimensional input

data and transforms it into higher dimension data space, thus improving the seprability of data. This technique is called as kernel trick. The most common kernels are listed below:

- Linear Kernel: A linear kernel can be used as normal dot product any two given observations.The product between two vectors is the sum of the multiplication of each pair of input values. The linear kernel can be defined by the following equation:

  $$K(x, x_i) = \sum(x.x_i)$$

  Linear SVM kernel is used mostly to handle the textual data as it contains multiple features and is mostly linearly separable.
- Polynomial Kernel: Polynomial kernel is a more generalized form of linear kernel. In the essence, a polynomial kernel with degree one can be treated as linear kernel.The polynomial kernel can distinguish curved or nonlinear input space.The homogenous polynomial kernel can be defined by the following equation:

  $$K(x, x_i) = \sum(x.x_i)^d$$

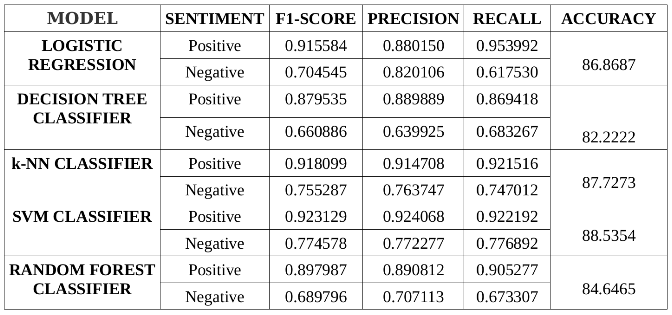
  Here , d is called as degree of kernel .
  The polynomial kernel is used in natural language processing (NLP). The most common degree is d = 2 (quadratic), since larger degrees tend to overfit on NLP problems.
- Radial Basis Function(RBF) Kernel: The Radial basis function kernel is a popular kernel function commonly used in support vector machine classification. RBF can map an input space in infinite dimensional space.

  $$K(x, x_i) = e^{-\gamma \cdot \sum(x - x_i^2)}$$

  Here gamma is a parameter, which ranges from 0 to 1. A higher value of gamma will perfectly fit the training dataset, which causes over-fitting.

E. Random Forest

VI. Results



MODEL	SENTIMENT	F1-SCORE	PRECISION	RECALL	ACCURACY
LOGISTIC REGRESSION	Positive	0.915584	0.880150	0.953992	86.8687
	Negative	0.704545	0.820106	0.617530	
DECISION TREE CLASSIFIER	Positive	0.879535	0.889889	0.869418	
	Negative	0.660886	0.639925	0.683267	82.2222
k-NN CLASSIFIER	Positive	0.918099	0.914708	0.921516	87.7273
	Negative	0.755287	0.763747	0.747012	
SVM CLASSIFIER	Positive	0.923129	0.924068	0.922192	88.5354
	Negative	0.774578	0.772277	0.776892	
RANDOM FOREST CLASSIFIER	Positive	0.897987	0.890812	0.905277	84.6465
	Negative	0.689796	0.707113	0.673307	

Fig. 3.

VII. Conclusion

VIII. Acknowledgement

References

[1] P. Joshi, "Twitter-sentiment-analysis-supervised-learning," 2018. [Online]. Available: https://datahack.analyticsvidhya.com/contest/practice-problem-twitter-sentiment-analysis/,https://www.analyticsvidhya.com/blog/2018/07/hands-on-sentiment-analysis-dataset-python/

[2] R. Gandhi, "Introduction to machine learning algorithms: Logistic regression." [Online]. Available: https://hackernoon.com/introduction-to-machine-learning-algorithms-logistic-regression-cbdd82d81a36

[3] G. Guo, H. Wang, D. Bell, and Y. Bi, "Knn model-based approach in classification," 08 2004.

[4] A. Sarlan, C. Nadam, and S. Basri, "Twitter sentiment analysis," 11 2014, pp. 212–216.

[5] T. W. Gruen, T. Osmonbekov, and A. J. Czaplewski, "eWOM: The impact of customer-to-customer online know-how exchange on customer value and loyalty," *Journal of Business Research*, vol. 59, no. 4, pp. 449–456, April 2006. [Online]. Available: https://ideas.repec.org/a/eee/jbrese/v59y2006i4p449-456.html

[6] A. Krouska, C. Troussas, and M. Virvou, "The effect of preprocessing techniques on twitter sentiment analysis," *2016 7th International Conference on Information, Intelligence, Systems Applications (IISA)*, 2016.

[7] J. Goyvaerts, *Regular expressions: the complete tutorial*. Lightning Source, 2006. [Online]. Available: https://www.princeton.edu/ mlovett/reference/Regular-Expressions.pdf

[8] P. Wang, G. R. Bai, and K. T. Stolee, "Exploring regular expression evolution," *2019 IEEE 26th International Conference on Software Analysis, Evolution and Reengineering (SANER)*, 2019.

[9] "Accessing text corpora and lexical resources ." [Online]. Available: https://www.nltk.org/book/ch02.html

[10] V. S and J. R, "Text mining: open source tokenization tools – an analysis," *Advanced Computational Intelligence: An International Journal (ACII)*, vol. 3, no. 1, 2016.

[11] C. Trim, "The art of tokenization," Jan 2013. [Online]. Available: https://www.ibm.com/developerworks/community/blogs/nlp/entry/tokenization ?lang=en

[12] H. Heidenreich, "Stemming? lemmatization? what?" Dec 2018. [Online]. Available: https://towardsdatascience.com/stemming-lemmatization-what-ba782b7c0bd8

[13] T. Evgeniou and M. Pontil, "Support vector machines: Theory and applications," vol. 2049, 01 2001, pp. 249–257.