ISSN 2395-1621

Twitter Sentiment Analysis Using Machine Learning Techniques

Tejaswini Zope, Dr. K. Rajeswari

tejaswini.zope20@pccoepune.org
kannan.rajeswari@pccoepune.org

Department of Computer Engineering
Pimpri Chinchwad College of Engineering, Akurdi,
Pune, India.

ABSTRACT

Sentiment analysis is use to identifying and classifying opinions or sentiments expressed in origin text. Social media is generating a huge amount of sentiment analysis data in the form of tweets, status updates, blog posts etc. Sentiment analysis of this user generated data is very useful in knowing the opinion of the public. Twitter sentiment analysis is difficult compared to general sentiment analysis due to the presence of slang words and misspellings. The maximum limit of characters that are allowed in Twitter is 140. Knowledge base approach and Machine learning approach are the two strategies used for analyzing sentiments from the text. In this paper, we try to analyze the twitter tweets using Machine Learning Algorithm. By doing sentiment analysis in a specific domain, it is possible to identify the effect of domain information in sentiment classification.

Keywords: Twitter, Machine learning, Logistic Regression, Random Forest, Fuzzy Rule.

ARTICLE INFO

Article History

Received: 6th July 2022

Received in revised form :

6th July 2022

Accepted : 9th July 2022

Published online :

9th July 2022

I. INTRODUCTION

Opinion mining is one of the most important tasks of natural language processing, which is also known as sentiment analysis, used to identify about what people have an impression about their Tweets. Twitter is a social networking web site where members can post messages in the form of "tweets". This is a platform where individuals can share ideas or sentiments on diverse subjects, fields or themes. It is a collection of user thoughts and sentiments spanning across various topics including standard net articles and net blogs. The quantity of pertinent data is bigger for twitter, when contrasted with former social media and blogging platforms.

When compared to other blogging sites, the response rate on Twitter is much more quicker. Sentiment analysis is widely utilized by different parties such as shoppers or marketers to gain insights into merchandise or understand the market trends.

II. LITERATURE REVIEW

"Comparative Study of Machine Learning Algorithms for Twitter Sentiment Analysis",Yash Indulkar et al,.q Three different algorithms are used to find out the best accuracy.

Google word2vec is used to group with similar words in the nearest vector.

"SENTIMENT ANALYSIS USING DEEP LEARNING",Shilpa P C et al,. For the negative classification we obtained a training accuracy of 89.13% and testing accuracy of 87.46%. For the positive classification we obtained a training accuracy of 91.32% and testing accuracy of 90.75%.LSTM gives the high accuracy As compared to RNN.

"Real-time Sentiment Analysis On E-Commerce Application",Jahanzeb Jabbar et al,.Using classification technique SVM along with Recall, Precision, F1 and ROC AUC, it gives highest accuracy.

"An ANEW based Fuzzy Sentiment Analysis Model",Andres Montoro1 et al,. "Inferring Sentiments from Supervised Classification of Text and Speech cues using Fuzzy Rules", Srishti Vashishtha et al,.

III. RELATED WORK

In current times, the opportunity to apprehend people's opinions has embossed the expanding interest both within the scientific society for the new research challenges, and in the business world due to the notable benefits in market analysis, financial sector, market prediction, etc. The power of sentiment analysis has been realized during the past decade. Different modes of communication used by humans to express their sentiments, other than text, like speech is gaining popularity; hence demanding multimodal sentiment analysis. Our work is closely related to two research areas: text-based sentiment analysis, which has been studied extensively in the field of computational linguistics, and audio emotion recognition from the fields of speech processing. Some works have shown that concatenation of text and speech features into a single vector when fed into a classifier yields higher accuracy compared to only text or only speech feature vector. But their feature set is small in size around 100 samples only, while our work involves a larger feature set: 3079 text features and 6373 audio features. The results of sentiment classification of affective speech using multiple classifiers can be enhanced by integrating the acoustic-prosodic features of speech with textual sentiment labels . Acoustic feature extraction of speech can be done. Linguistic features like Bag-of-Words , Term Frequency-Inverse Document Frequency (TF-IDF) , word embeddings: word2vec ,extracted from textual data. One of the commonly used text features in sentiment analysis is TF-IDF and its variants have shown an increase in accuracy . In multimodal sentiment analysis, the textual features can be obtained using only Bag of Words or TF-IDF with Bag of Words or TF-IDF with word vectors. some authors use word2vec vectors. Feature Selection techniques for finding significant keywords for supervised classification are also popular .

A new set of speech cues was developed based on the randomness in the values of pitch, energy and MFCC speech cues, using non extensive entropy, that are the core features for speech . Our proposed fuzzy rule-based system uses open SMILE tool for extracting speech features and TF-IDF for text features.

IV. OBJECTIVE

- To implement an algorithm for automatic classification of text into positive, negative or neutral.
- Sentiment Analysis to determine the attitude of the mass is positive, negative or neutral towards the subject of interest.
- Graphical representation of the sentiment in the form of Bar Graph.

V. DATASET

1. Understand the Problem Statement

The objective of this task is to detect hate speech in tweets. For the sake of simplicity, we say a tweet contains hate speech if it has a racist sentiment associated with it. So, the task is to classify racist tweets from other tweets.

2. Tweets Preprocessing and Cleaning -The preprocessing of the text data is an essential step as it makes the raw text ready for mining, i.e., it becomes easier to extract information from the text and apply machine learning algorithms to it. If we skip this step then there is a higher chance that you are working with noisy and inconsistent data. The objective of this step is to clean noise those are less relevant to find the sentiment of tweets such as punctuation, special characters, numbers, and terms which don’t carry much weightage in context to the text.

In one of the later stages, we will be extracting numeric features from our Twitter text data. This feature space is created using all the unique words present in the entire data. So, if we preprocess our data well, then we would be able to get a better quality feature space. Initial data cleaning requirements that we can think of after looking at the top 5 records:

· The Twitter handles are already masked as @user due to privacy concerns. So, these Twitter handles are hardly giving any information about the nature of the tweet.

· We can also think of getting rid of the punctuations, numbers and even special characters since they wouldn’t help in differentiating different kinds of tweets.

· Most of the smaller words do not add much value. For example, ‘pdx’, ‘his’, ‘all’. So, we will try to remove them as well from our data.

· Once we have executed the above three steps, we can split every tweet into individual words or tokens which is an essential step in any NLP task.

· In the fourth tweet, there is a word ‘love’. We might also have terms like loves, loving, lovable, etc. in the rest of the data. These terms are often used in the same context. If we can reduce them to their root word, which is ‘love’, then we can reduce the total number of unique words in our data without losing a significant amount of information.

A) Removing Twitter Handles (@user)

As mentioned above, the tweets contain lots of twitter handles (@user), that is how a Twitter user acknowledged on Twitter. We will remove all these twitter handles from the data as they don’t convey much information. For our convenience, let’s first combine train and test set. This saves the trouble of performing the same steps twice on test and train.

B) Removing Punctuations, Numbers, and Special Characters

Punctuations, numbers and special characters do not help much. It is better to remove them from the text just as we removed the twitter handles. Here we will replace everything except characters and hashtags with spaces.

C) Removing Short Words

We have to be a little careful here in selecting the length of the words which we want to remove. So, I have decided to remove all the words having length 3 or less. For example, terms like “hmm”, “oh” are of very little use. It is better to get rid of them.

D) Tokenization

Now we will tokenize all the cleaned tweets in our dataset. Tokens are individual terms or words, and tokenization is the process of splitting a string of text into tokens.

E) Stemming

Stemming is a rule-based process of stripping the suffixes (“ing”, “ly”, “es”, “s” etc) from a word. For example, For example – “play”, “player”, “played”, “plays” and “playing” are the different variations of the word – “play”.

3.Story Generation and Visualization from Tweets

Exploring and visualizing data, no matter whether its text or any other data, is an essential step in gaining insights. Do not limit yourself to only these methods told in this tutorial, feel free to explore the data as much as possible.

A) Understanding the common words used in the tweets:

Now I want to see how well the given sentiments are distributed across the train dataset. One way to accomplish this task is by understanding the common words by plotting word clouds. A word cloud is a visualization wherein the most frequent words appear in large size and the less frequent words appear in smaller sizes.

B) Words in non-racist tweets:

We can see most of the words are positive or neutral. With happy, smile, and love being the most frequent ones. Hence, most of the frequent words are compatible with the sentiment which is non-racist tweets.

C) Racist Tweets:

As we can clearly see, most of the words have negative connotations. So, seems we have a pretty good text data to work on. Next we will the hashtags/trends in our twitter data.

D) Understanding the impact of Hashtags on tweets sentiment

Hashtags in twitter are synonymous with the ongoing trends on twitter at any particular point in time. We should try to check whether these hashtags add any value to our sentiment analysis task, i.e., they help in distinguishing tweets into the different sentiments.

4. Extracting Features from Cleaned Tweets:

To analyze a preprocessed data, it needs to be converted into features. Depending upon the usage, text features can be constructed using assorted techniques – Bag-of-Words, TF-IDF, and Word Embeddings. In this article, we will be covering only Bag-of-Words and TF-IDF.

1) **Bag-of-Words Features:**

Bag-of-Words is a method to represent text into numerical features. Consider a corpus (a collection of texts) called C of D documents {d1,d2…..dD} and N unique tokens extracted out of the corpus C. The N tokens (words) will form a list, and the size of the bag-of-words matrix M will be given by D X N. Each row in the matrix M contains the frequency of tokens in document D(i).

Let us understand this using a simple example. Suppose we have only 2 document

D1: He is a lazy boy. She is also lazy.

D2: Smith is a lazy person.

The list created would consist of all the unique tokens in the corpus C.

= [‘He’,’She’,’lazy’,’boy’,’Smith’,’person’]

Here, D=2, N=6

The matrix M of size 2 X 6 will be represented as –

Now the columns in the above matrix can be used as features to build a classification model. Bag-of-Words features can be easily created using sklearn’s CountVectorizer function. We will set the parameter max_features = 1000 to select only top 1000 terms ordered by term frequency across the corpus.

2) **TF-IDF Features**

This is another method which is based on the frequency method but it is different to the bag-of-words approach in the sense that it takes into account, not just the occurrence of a word in a single document (or tweet) but in the entire corpus.

TF-IDF works by penalizing the common words by assigning them lower weights while giving importance to words which are rare in the entire corpus but appear in good numbers in few documents.

Let’s have a look at the important terms related to TF-IDF:

· TF = (Number of times term t appears in a document)/(Number of terms in the document)

· IDF = log(N/n), where, N is the number of documents and n is the number of documents a term t has appeared in.

· TF-IDF = TF*IDF

5. Model Building: Sentiment Analysis:

We are now done with all the pre-modeling stages required to get the data in the proper form and shape. Now we will be building predictive models on the dataset using the two feature set — Bag-of-Words and TF-IDF.

We will use logistic regression to build the models. It predicts the probability of occurrence of an event by fitting data to a logit function.

A) Building model using Bag-of-Words features

We trained the logistic regression model on the Bag-of-Words features and it gave us an F1-score of 0.53 for the validation set. Now we will use this model to predict for the test data.

The public leader board F1 score is 0.567. Now we will again train a logistic regression model but this time on the TF-IDF features. Let’s see how it performs.

B) Building model using TF-IDF features

The validation score is 0.544 and the public leaderboard F1 score is 0.564. So, by using the TF-IDF features, the validation score has improved and the public leaderboard score is more or less the same.

Classification

The attributes mentioned are provided as input to the different ML algorithms such as Random Forest, Decision Tree, Logistic Regression and Naive Bayes classification techniques. The input dataset is split into 80% of the training dataset and the remaining 20% into the test dataset. Training dataset is the dataset which is used to train a model. Testing dataset is used to check the performance of the trained model. For each of the algorithms the performance is computed and analysed based on different metrics used such as accuracy, precision, recall and F-measure scores as described further.

Random Forest

Random Forest algorithms are used for classification as well as regression. It creates a tree for the data and makes prediction based on that. Random Forest algorithm can be used on large datasets and can produce the same result even when large sets record values are missing. The generated samples from the decision tree can be saved so that it can be used on other data. In random forest there are two stages, firstly create a random forest then make a prediction using a random forest classifier created in the first stage.

VI. CONCLUSION

The regression algorithm used for binary classification is Logistic Regression & the classification algorithms used are support vector machine & Random Forest. It can be observed that from the three algorithms used, the best accuracy was generated from Random Forest for both the respective datasets. The Random Forest gave better accuracy because it created multiple decision trees and then calculated a mean value from all the decision trees. novel text and speech based fuzzy rule-based system has been proposed for multimodal sentiment analysis of review videos posted on social media.

VII.REFERENCES

[1] Deonna, Julien, and Fabrice Teroni.(2012) The emotions: A philosophical introduction. Routledge.

[2] Soleymani, Mohammad, David Garcia, Brendan Jou, Björn Schuller, Shih-Fu Chang, and Maja Pantic. (2017) "A survey of multimodal sentiment analysis." Image and Vision Computing65: 3-14.

[3] Morency, Louis-Philippe, Rada Mihalcea, and Payal Doshi. (2011) "Towards multimodal sentiment analysis: Harvesting opinions from the web." Proceedings of the 13th international conference on multimodal interfaces. ACM.

[4] Poria, Soujanya, Erik Cambria, Rajiv Bajpai, and Amir Hussain. (2017) "A review of affective computing: From unimodal analysis to multimodal fusion." Information Fusion 37: 98-125.

[5] Zadeh, Amir, Rowan Zellers, Eli Pincus, and Louis-Philippe Morency. (2016) "MOSI: multimodal corpus of sentiment intensity and subjectivity analysis in online opinion videos." arXiv preprint arXiv:1606.06259.

[6] Zadeh, Lotfi A. "Fuzzy logic—a personal perspective. (2015)" Fuzzy sets and systems 281: 4-20.

[7] Ross, Timothy J. (2004) Fuzzy logic with engineering applications. Vol. 2. New York: Wiley.

[8] "Quickstart – Flask" 2019 [Online]. Available: http://flask.pocoo.org/ docs/1.0/quickstart/#routing.

[9] R. He and J. McAuley, "Ups and downs: Modeling the visual evolution of fashion trends with oneclass collaborative filtering," Proceedings of the 25th International Conference on World Wide Web, International World Wide Web Conferences Steering Committee, 2016. doi: 10.1145/ 2872427.2883037.

[10] "Aamzon review data" Julian McAuley, UCSD, 2019. [Online] Available: http://jmcauley.ucsd.edu/data/ amazon/.

[11] G. Murray, E. Hoque, G. Carenini, Chapter 11 – Opinion Summarization and Visualization, Editor(s): Federico Alberto Pozzi, Elisabetta Fersini, Enza Messina, Bing Liu, Sentiment Analysis in Social Networks, Morgan Kaufmann, 2017, Pages 171-187, ISBN 9780128044124.

[12] Giatsoglou, Maria, et al. "Sentiment analysis leveraging emotions and word embeddings." Expert Systems with Applications 69 (2017): 214-224.

[13] Sarlan, Aliza, Chayanit Nadam, and Shuib Basri. "Twitter sentiment analysis." Proceedings of the 6th International conference on Information Technology and Multimedia. IEEE, 2014.