Twitter Sentiment Analysis

Bhagirathsinh Sarvaiya(B19CSE021)
Bharat Biradar(B19CSE022)
Yashasvi Chauhan(B19CSE100)

Abstract

Many sectors and also individuals use sentiment analysis of tweets to determine the public's emotion towards them or their competitors. The objective of this project is to develop an algorithm using three different classifiers that can correctly classify tweets as either negative or positive, concerning a query term. We hypothesize that we can obtain high precision by classifying tweets using machine learning techniques. We are going to be using three machine learning techniques and compare their performance using a fixed set of parameters to see which one gives the best results.

I. Introduction

Twitter is a popular microblogging service used to send and receive short posts called tweets. It is widely used to express opinions on various matters. Generally, this type of sentiment analysis is beneficial for marketers examining the public view of their firm or consumers who are trying to examine a product or service. Sentiment can broadly be classified as either positive or negative. Sentiment analysis is a natural language processing technique to quantify an expressed opinion or sentiment within a selection of tweets. Sentiment analysis refers to the general method to extract polarity and subjectivity from semantic orientation which refers to the strength of words and polarity text or phrases.

II. Data

This is the sentiment140 dataset. It contains 1,600,000 tweets extracted using the Twitter API. It contains the following 6 fields:

1) target: the polarity of the tweet (it is binary i.e. either positive or negative)
2) ids: The id of the tweet (can be any number)
3) date: the date of the tweet (Example to describe format is: Sun June 21 13:28:41 UTC 2021)
4) flag: The query (If there is no query, then this value is NOQUERY)
5) user: the username of the person that tweeted
6) text: the text of the tweet

III. Approach

We start off by importing the dataset and extracting all the useful columns from the dataset. Now we plot the distribution based on the attribute target which gives us insights into the number of positive and negative values. Now we store the columns of 'text' and 'target' in two different arrays. We will preprocess our data over here and then train the final dataset. After the transformation of our dataset, we will be using different classifiers to find the accuracy, precision, and f1 score of the data for each of them.

IV. Data Preprocessing

There is a lot of data in the dataset that needs to be removed or modified as it is not significant enough. So we remove the URLs, convert all the words to the same case(over here, lowercase), remove stopwords, emojis, emoticons. After the completion of this, we convert the remaining words into vectors. We do the training and testing of our data set with the test_size = 0.15 i.e. 15% goes into testing the dataset and the remaining 85% into training.

V. Feature Selection

Feature selection is the process of reducing the number of input variables when developing a predictive model. We automatically or manually select those features which contribute most to your prediction variable or output in which we are interested.
It follows a greedy search approach by evaluating all the possible combinations of features against the evaluation criterion. We have used the Term Frequency — Inverse Document Frequency tool to convert words into vectors. What this does is that it changes the weightedness of words based on their frequency of occurrence. It is similar to the tokenization approach that is applied in creating plagiarism detection machines.

VI. Building Models:

Now that the preprocessing part was complete, we started creating models. What we were thinking is that we will use different types of models ranging from simple models to complex models. This was done to see if the performance is affected by the complexity of the model. We have evaluated the performance of the models based on their accuracy, precision, F1 score, and confusion matrix.

A. Multinomial Naive Bayes Algorithm

These were the results of this model:

- Accuracy : 0.806425
- F1 score : 0.804135
- Average precision: 0.737867

Confusion matrix:

```
array([[98174, 24582],
       [21876, 95368]], dtype=int64)
```

B. Random Forest Classifier

These were the results of this model:

- Accuracy : 0.81501
- F1 score : 0.81626
- Average precision: 0.76242

Confusion matrix:

```
array([[96985, 21332],
       [23065, 98618]], dtype=int64)
```

C. Logistic Regression Classifier

These were the results of this model:

- Accuracy : 0.82557
- F1 score : 0.82692
- Average precision: 0.77517

Confusion matrix:

```
array([[ 98134,  19945],
       [ 21916, 100005]], dtype=int64)
```

VII. Conclusion

Based on the scores that we received for the different parameters that we had set, we can say logistic regression is the best model followed by random forest and last but not the least, multinomial naive Bayes algorithm. One of the interesting things about this result is that logistic regression being the simplest model turned out to have the best performance.
In this work, we presented an analysis and overview of the most prominent methods for sentiment analysis in Twitter. The innovation of this work is concentrated in the meticulous evaluation of the efficiency of various sentiment analysis mechanisms using manually annotated datasets, as well as in the demonstration of the possibility to combine methods, creating new techniques for enhancing the quality of the outcome

Acknowledgment

We would like to express our special thanks of gratitude to our instructor Dr.Richa Singh for their great efforts in clearing our concepts and helping us in exploring new things. We would also like to thank our TA's who helped in clearing our doubts regarding the project and cleared our queries throughout the semester on time.

References

- Sanket Sahu Mining Engineering (B. Tech)IITKharagpur, Midnapore West Bengal, India, Suraj Kumar Rout Instrumentation Engineering (B.Tech)College of Engineering Technology, Debasmit Mohanty CEO StratLytics Consulting Private Limited," Twitter Sentiment AnalysisA more enhanced way of classification and scoring",2015.
- Agarwal, A. et al. 2011. Sentiment Analysis of Twitter Data. Proceedings of the Workshop on Languages in Social Media (Stroudsburg, PA, USA, 2011), 30–38.
- A. Go, R. Bhayani, and L. Huang. Twitter sentiment classification using distant supervision. In CS224N Project Report, Stanford, 2009.
- Y. Kim. Convolutional neural networks for sentence classification. In EMNLP, 2014.

Members' Contribution

- Bhagirathsinh Sarvaiya - model making, performance analysis, and report making
- Bharat Biradar - data preprocessing,readme and report making
- Yashasvi Chauhan - background research, model analysis, and report making