2020

Twitter Sentiment Classification using Emoticons

732A92

PRIYA KURIAN PULLOLICKAL(PRIKU577)

Abstract

The project aims at collecting the tweets from the twitter, automatically labelling them as positive or negative tweets, pre-processing the tweets and finding the classification accuracy. The tweets were scraped from twitter using emoticons and English language as query terms. The tweets were then classified as positive or negative based on the number of different emoticons it had. Pre-processing was the next step. The tweets were then tokenised and stemmed. The Term Frequency -Inverse Document Frequency was used to find the count and weight of the words. Later machine learning algorithms like Support Vector Machines, Naive Bayes, Max Entropy and XGBoost were used to find the classification accuracy and compared the accuracy with [1].

Introduction

I started with Swedbank AB last winter. One of the initial assignments I got as an intern was to do the sentiment classification using twitter data. I took the text mining course also at the same time. Then I was trying to find something that suits both the project and the work. I looked for research papers to get more idea of the sentiment classification and I came across [1].In that, the tweets were collected using emoticons and classified as positive or negative tweets depending on the number of positive or negative terms each tweet had. Later the tweet was pre-processed, and classification accuracy was found. I could understand it well and thought to work more on it. The first step in the project was to collect the data. The data for this project was collected from twitter. The tweets were collected in regular intervals for almost a week and the queries ran till around 180000 tweets were collected. The query terms were emoticons and hence the data was abundantly available in twitter. Tweets with language as English was considered for this project. Pre-processing and tokenisation was done to the tweets. The Term Frequency Inverse Document frequency method was used to find the count and weight of the words. Later Machine Learning algorithms like Support Vector Machine, Maximum Entropy Naive Bayes and XGBoost were used to find the classification accuracy and compared the values with [1].

Theory

Stemming:

Every word will have a root word from which other words can be derived from. Stemming is the process of reducing all these inflected words to the word stem. The suffix stripping algorithm removes all the suffixes like ing, ed, ly from a word so that we get the root word. For example, the words working, worked, works are all derived from the word work. Hence after doing the stemming all these different tenses of words become the same word and this would increase the frequency of the words when doing the word count.

Term Frequency-Inverse Document frequency

Term frequency shows how much times the word appeared in the document. The chances for appearing a word increases with the length of the document. Normalisation of this can be done by dividing the frequency with the length of the word. Document frequency is the count of occurrence of the term in the document set. Inverse document frequency is the inverse of term Frequency.

$$\mathrm{Wi, j} = \mathrm{tfi, j}\ \times \log(\frac{N}{df_{\mathrm{i}}})$$

$\mathrm{tf}_{\mathrm{i,\,j}}$ is the number of occurrences of i in j ,

N is the total number of documents and df_{i} is the number of documents containing i.

Support Vector Machine

Support Vector Machine is a supervised learning algorithm used for classification. When the model gets trained using labelled data, it is called supervised learning and labelling is the way of grouping the data. In SVM, the data is classified into different groups and when the new data comes, the data will be classified to one of the different groups. If a dataset has a labelled data of two classes, in two-dimensional space the two classes will be in two sides and hence obviously a line can divide the two groups. In a unigram feature extractor, each feature is a single word found in the tweet. If the feature is found, it is given value 1 and if it is not there, it is given a value 0. Stochastic Gradient Descent can be used in linear SVM for classification. This classifiers as the name suggest is stochastic, it does not take the entire training set always. It chooses some training point in some random fashion and hence each time we run it, we get different cost function and different local optimum.

Multinomial Naive Bayes

Naive Bayes classifier is based on the Bayes theorem which shows

$$Posterior = (Prior * likelyhood)/Evidence$$

The naive assumption that each of the feature is independent is the reason for the name Naive Bayes. Multinomial Naive Bayes algorithm is a type of Naive Bayes algorithm which uses multinomial distribution or more precisely when the naive bayes just tells whether a word is present or not, multinomial Naive Bayes finds the word count.

Maximum Entropy

The Maximum Entropy models are models based on features. They work the same way as logistic regression when only two classes are there. It works quite opposite to that of naive Bayes. Maximum entropy makes no independent assumptions like Naive Bayes.

Extreme Gradient Boost

The Extreme Gradient Boost uses the principle of the Gradient Boost. Gradient boosting algorithm build models in steps and uses loss function of base model as a proxy for minimising the error of the overall model. The new models created will predict the errors of the prior models and then they are added together to make the final prediction. Gradient boost uses gradient descent algorithm to minimize the loss when adding new models. The Xgboost uses a more regularised model when compared to gradient boost and hence overfitting can be controlled.

Data

The data was collected from the twitter. The Query term, language and count was given as input while collecting the tweets. The Query term was the emoticon, language was set to English and the tweets were collected in regular intervals until it reached almost 180000 tweets. It is possible that a single tweet might contain both positive and negative emoticon and then it is difficult to determine the emotion describing the tweet. Therefore, each of the tweets were compared with a set of emoticons and then the number of positive and negative emoticons in each tweet were calculated. If the number of positive emoticons is greater than the negative emotions, then the tweet was given a value 4 and it was classified as a positive tweet. If the number of negative emotions were greater in a tweet, then the tweet was given a value 0 and it was classified as a negative tweet. The tweets were collected for nearly two weeks.

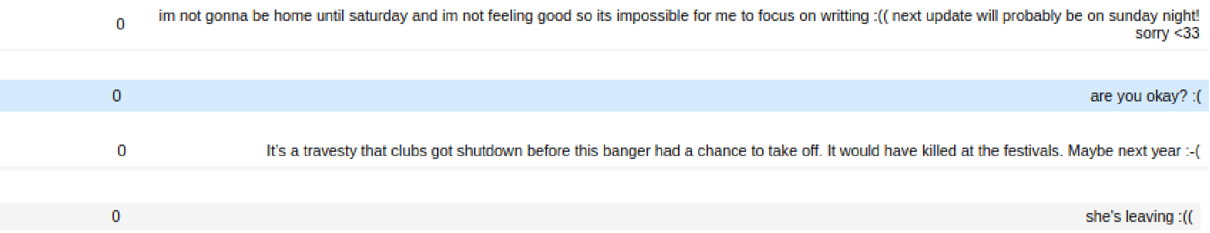
Query terms used:

Positive Emoticons	:)
Negative Emoticons	:(

Set of emoticons used to compare each tweet:

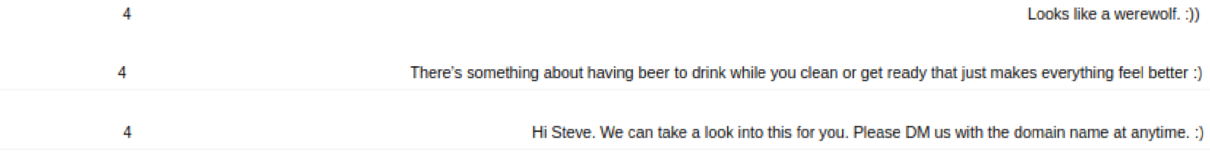
Positive Emoticons	:), :-), :), :D, =), :p, ;)
Negative Emoticons	:(, :-(, : (

Examples of data collected for negative sentiment:



0	im not gonna be home until saturday and im not feeling good so its impossible for me to focus on writting :((next update will probably be on sunday night! sorry <33
0	are you okay? :(
0	It's a travesty that clubs got shutdown before this banger had a chance to take off. It would have killed at the festivals. Maybe next year :-(
0	she's leaving :((

Examples of data collected for positive sentiment:



4	Looks like a werewolf. :))
4	There's something about having beer to drink while you clean or get ready that just makes everything feel better :)
4	Hi Steve. We can take a look into this for you. Please DM us with the domain name at anytime. :)

Methods

Cleaning the data marks as the first step for any machine learning technique. The cleaning methods used in the project are

1. Remove URLS

2. Remove user details
3. Remove all words less than three characters
4. Remove all the digits
5. Remove non-English characters
6. Remove punctuations
7. Remove stop words
8. Convert all the words to lower cases

The data obtained was then tokenised and stemmed. TF-IF method is used to find the weight of each word. 80% of the data was used as training data and 20% as test data. The sentiment score labelled for each tweet was compared with TF-IF using the machine learning algorithms SVM, Naive Bayes, Maximum Entropy, XGBoost and the accuracy were recorded. The built in libraries were used for classifiers in this project as opposed to the Stanford classifiers in [1].The accuracy of the classifiers was then compared with the values in [1].

Results

Technique	SVM	Naive Bayes	Max Entropy	XGBoost
TF -IF	75	77	79	72

Table 1

As seen in the table 1, Term Frequency-Inverse Document Frequency gave 75%,77%,79%,72% accuracy for the SVM, Naive Bayes, Max Entropy and XGBoost respectively.

Discussion

The result shows Max Entropy gives the best performance when using the TF-IF method.

The results in [1] are

Technique	SVM	Naive Bayes	Max Entropy	XGBoost
Bigram	78	81	79	-

The data was collected from twitter for this project. The methods of classification of tweet as negative and positive was different from [1]. The cleaning methods were different from the methods described in the paper [1]. Stemming, tokenisation, Term Frequency-Inverse Document frequency methods were all used to get the result in the project. The paper [1] just mention they have used a bigram extractor to get the result. Paper [1] has used more tweets than our project because the data was collected for more than a month. The results show that the accuracy got in the project is less than when compared to [1].

Conclusion

Sentiment classification of twitter data using Max Entropy gave 79% accuracy using this project. If we use better pre-processing steps, we might be able to get better results. If we need to find the sentiments regarding a product or a hashtag term the methods in this project can be used. The news we read are mostly biased. Twitter is a commonly used social media platform. So, to get the real sentiments regarding a topic, projects like this would be the best way.

[3]

References

[1] A. Go, R. Bhayani and L. HUang, "Twitter Sentiment Classification using Distant Supervision," in *ieee xoexoe*, Atlanta, 2012.

[2] "Sentiment 140," [Online]. Available: http://help.sentiment140.com .

[3] "https://machinelearningmastery.com/gentle-introduction-xgboost-applied-machine-learning/," [Online].

[4] •. https://medium.com/@sangha_deb/naive-bayes-vs-logistic-regression-a319b07a5d4c, "Naive Bayes and Logistic Regression algorithms comparison".

[5] "https://en.wikipedia.org/wiki/Naive_Bayes_classifier," [Online].

[6] "https://machinelearningmastery.com/gentle-introduction-gradient-boosting-algorithm-machine-learning/," [Online].