Sentiment Analysis for Tweets

Girija Godbole,
University of Texas at Dallas,
Dallas, TX
gsg160130@utdallas.edu

Abstract - ***User feedback and reviews are one of the major factors for a lot of companies to make necessary modifications in their products to gain better profits. Sentiment Analysis, which is also known as opinion mining, is a process of finding the emotional tone behind a user's statement or review. This paper, attempts to build a simple sentiment analyzer which tags the Tweets related to 'iPhoneX' as either 'positive' or 'negative'. Since iPhoneX has been recently launched by Apple Inc., we are interested to know whether the customers are liking this new product or not. The tweets that we attempt to classify are extracted via the Twitter Search API. Our model is trained on the Naive Bayes Classifier using the TextBlob library of Python.***

Keywords - sentiment analysis, textblob, Twitter, Twitter API, Naive Bayes Classifier, natural language processing, machine learning.

I. INTRODUCTION

'Sentiment Analysis' or 'Opinion mining' is a field of Natural Language Processing which builds systems that try to identify and extract opinions within texts. It is the contextual mining of text which identifies and extracts informations in source material, that in turn helps a business to understand the social sentiment of their brand, product or service by monitoring online reviews and conversations by people.

iPhoneX, which is a new cell phone in the iPhone series, has been recently launched (2018) by Apple Inc. People talk about their experiences with this new product widely on Twitter. Twitter is a social networking platform where people communicate by posting short messages called 'tweets'. It is very helpful for a business like Apple Inc. to know if the users are really liking their newly launched product or if they need to make any changes for making sure that maximum iPhoneXs' are sold. They can do this by analyzing the sentiments in the tweets posted by people on Twitter. Twitter provides a developer platform with a Search API that can be used by almost anyone having a twitter account. This API can be used to get tweets related to a particular topic.

In this paper, we explain a simple approach for building a sentiment analyzer and use it for such iPhoneX tweets on Twitter and try to look at users' opinions about the new launch.

II. PREVIOUS WORK

The roots of sentiment analysis are in the studies on public opinion analysis at the beginning of 20th century and in the text subjectivity analysis performed by the computational linguistics community in 1990's. However, the outbreak of computer-based sentiment analysis only occurred with the availability of subjective texts on the Web. Consequently, 99% of the papers have been published after 2004. Sentiment analysis papers are scattered to multiple publication venues, and the combined number of papers in the top-15 venues only represent ca. 30% of the papers in total.[1]

There have been many different approaches for sentiment analysis. The large user generated content requires the use of automated techniques for mining and analyzing since crowd sourced mining and analysis are difficult. Today, traditional news outlets have an online version of their news.[2]

Opinion mining (sentiment extraction) is employed on Twitter posts by means of following techniques

1. Lexical analysis:

This technique is governed by the use of a dictionary consisting pre-tagged lexicons. The input text is converted to tokens by the Tokenizer. Every new token encountered is then matched for the lexicon in the dictionary. If there is a positive match, the score is added to the total pool of score for the input text. The classification of a text depends on the total score it achieves. [2]

2. Machine learning based analysis:

Machine learning is one of the most prominent techniques gaining interest of researchers due to its adaptability and accuracy. In sentiment analysis, mostly the supervised learning variants of this technique are employed. It comprises of three stages: Data collection, Pre-processing, Training data, Classification and plotting results. In the training data, a collection of tagged corpora is provided. The Classifier is presented a series of feature vectors from the previous data. A model is created based on the training data set which is employed over the new/unseen text for classification purpose. In machine learning technique, the key to accuracy of a classifier is the selection of appropriate features. Generally, unigrams (single word phrases), bi-grams (two consecutive phrases), tri-grams (three consecutive phrases) are selected as feature vectors. There are a variety of proposed features namely number of positive words, number of negative words, length of the document, Support Vector Machines (SVM) and Naïve Bayes algorithm [2]

3. Hybrid/Combined analysis:

Another approach derived a "unified framework" using background lexical information as word class associations. Authors renewed information for particular areas using the available datasets or training examples and proposed a classifier called as Polling Multinomial Classifier (PMC) (also known as the multi - nomial naïve bayes). Manually labeled data was incorporated for training purpose. They claimed that making use of lexical knowledge improved performance.[2]

III. DATASET

We use two types of dataset for our model. One is the training data and the other is the test data:-

1. Training Dataset:-

For our model to learn the patterns for classifying new sentences, we need to train it on a similar kind of dataset. For this training purpose, we have used the "Sentiment Labelled Sentences Dataset" by the University of California Irvine.

This dataset is publicly available and has been created for paper [3].

It contains sentences labelled with ‘positive’ or ‘negative’ sentiment, extracted from reviews of products (amazon.com) , movies (imdb.com), and restaurants(yelp.com). As we are trying to classify sentences related to a product, we have used the amazon.com reviews data to train our model. This helps our model learn very similar patterns and thereby increases the accuracy of classification.

The format of this dataset is as follows: -

sentence \t score \n

The score is either ‘1’ (for positive) or ‘0’ (for negative)

For each of the three websites (amazon.com, yelp.com and imdb.com), , there exist 500 positive and 500 negative sentences. Therefore, our model uses 500 positive and 500 negative training examples from amazon.com.
The examples were selected randomly for larger datasets of reviews. They have attempted to select sentences that have a clearly positive or negative sentiment. This discards the notion of ‘neutral’ sentences and thus helps us as we are doing a similar two-class classification.

2. Test Dataset: -

The test data for our model is the tweets we classify related to iPhoneX. To get these tweets we query the Twitter search API. The query specifies the details about the retrieval of tweets such as the keywords, count, languages, etc.

The format of this dataset is: -
Sentence\n

IV. APPROACH

The steps and processes involved in building the sentiment analysis system include the following: -

1. The Twitter API Client: -

Twitter provides a developer platform that includes a number of API endpoints and tools that help users build apps and solutions. We have used the Twitter Search API to retrieve the tweets that we need. We have used the tweepy library. Tweepy is a python library for accessing the Twitter API.
The first step that I had to perform was to gain access to build apps on the Twitter developer platform. Twitter has made rules since Aug 2018, regarding their developer center and we need to request an access to create apps. For this, I had to submit a request form stating clearing what app I want to build and how would I make use of the tweets I retrieve. I could do all this only after logging in to my twitter account. Once my request got approved, I was able to create an app in the development platform for my sentiment analyzer. This is when Twitter provides us with the consumer key, the consumer secret key, the access key and the access secret key. These are authentication keys that we need to use while we try to access Twitter API.

We can search for tweets related to a particular topic by specifying it in the query. We can filter our search by various parameters like the count of tweets retrieved and the languages.

2. Our Model: -

The textblob library of Python provides us with a Naive Bayes Classifier module that takes in training data as the input. We have trained our model on the amazon.com dataset by UCI as mentioned above. Once our model is

trained, we can use it to classify further test sentences. In our case, these test sentences are the Tweets we are retrieving using the Twitter API Client.

The classifier classifies every tweet into two classes: positive (denoted by 1) or negative (denoted by 0). The classifier gives a fair accuracy and mostly classifies many test examples correctly. For us to know what it is exactly classifying, we put a 'English' language filter on the query. This also increases performance because we have trained our model on English sentences.

To know how many of the tweets we retrieve have a positive or a negative sentiment, we simply calculate the percentage of the tweets in that particular class.

V. EXPERIMENTS

As an experiment, we can change the parameters to our model. These parameters include the training dataset, the count of the retrieved tweets and the language filters. The following tables show the results of these experiments.

VI. RESULTS

1. When the language is set to 'English' :-

Count of Tweets Retrieved	Classified as "Positive"	Classified as "Negative"
10	75%	25%
50	61%	38%
100	55%	44%
250	56%	43%
500	55%	44%

2. When no language is set (all languages considered): -

Count of Tweets Retrieved	Classified as "Positive"	Classified as "Negative"
10	90%	10%
50	76%	23%
100	79%	20%
250	79%	20%
500	78%	21%

From the above results, the ones with English language are more accurate because our model is trained on English training examples.

VII. COMMENTS / CONCLUSION

This project helped me understand how sentiment analysis helps businesses improve their products and adjust according to customer needs and likings. This was good learning experience for me as a student as it introduced me to various approaches of sentiment analysis coupled with machine learning. It made me realise that natural language processing along with machine learning can help create very useful and powerful systems.

REFERENCES

[1] Mika V. Mäntylä, Daniel Graziotin, Miikka Kuutila, "The evolution of sentiment analysis—A review of research topics, venues, and top cited papers", Computer Science Review, Volume 27, February 2018, Pages 16-32, ISSN 1574-0137.

[2] Harsh Thakkar and Dhiren Patel, "Approaches for Sentiment Analysis on Twitter: A State-of-Art study"Department of Computer Engineering, National Institute of Technology, Surat-395007

[3] Kotzias et. al,.'From Group to Individual Labels using Deep Features', KDD 2015.