Sentimental Analysis on Apple Tweets with Machine Learning Technique

Dupinder Kaur
Assistant Professor
Department of Computer Science and Engineering
JCDM College of Engineering, Sirsa(HR)

Abstract: With the rapid growth of the internet, millions of people are sharing their views and opinions on a variety of topics on micro blogging sites. On these websites user makes real time short and frequent posts about everything. These posts also include Sentiments which refers to emotions, feelings, attitude or opinion. Sentiment analysis is basically study of emotions and opinions from text. The basic idea is to analyze the results and predict outcomes that are based on customer feedback or opinions. It is helpful for consumers who want to find out the sentiment of products before purchase, or companies that want to monitor the public sentiment of their brands. Twitter sentiment analysis is tricky as compared to broad sentiment analysis because it contains slang words, misspellings and repeated characters. This research paper present the results of machine learning algorithms by classifying the sentiment of Twitter messages using distant supervision with the help of preprocessing steps needed in order to achieve high accuracy. The conclusion of this paper is presented by ten different sentiments from data taken.

Keywords: Sentiments, Naive Bayes Classifier. Twitter, Machine learning algorithm.

I. Introduction

With the proliferation of World Wide Web, Individuals tends to do everything on-line which include discussions on social media like twitter and Facebook, expressing views by writing blogs and ratings and reviews of movie or any item. The textual data over internet has grown to more than 20 billion pages. Due to this, companies feel the need to analyze this text and calculate the insights for business. Business owners and advertising companies often employ sentiment analysis to discover new business strategies and advertising campaign. Sentiment analysis is mainly concerned with the identification and classification of opinions or emotions of each tweet.

Twitter Sentiment analysis on is the next step in the field of sentiment analysis, as tweets give us more varied resource of opinions and sentiments that can be about anything from the latest phone they bought, movie they watched, political issues, religious views or the individuals state of mind. In general, various symbolic techniques and machine learning techniques are used to analyze the sentiment from data. This research is about sentimental analysis on apple tweets.

The objective of this paper is to study the customer reviews on apple tweets and categorize it into different sentiments. In the first step pre-processing is done by cleaning the data which includes: removing the stop words, white spaces, repeating words, emoticons and #hash tags. In order to correctly classifying these tweets, machine learning technique are used and this technique does not require the database of words like used in knowledge-based approach. Several methods are used to extract the feature from the source text. Feature extraction is done in two phases: In the first phase extraction of data related to twitter is done i.e. twitters specific data is extracted. Now by doing this, the tweet is transformed into normal text. In the next phase, more features are extracted and added to feature vector. Each tweet in the training data is associated with a particular class label. This training data is passed to different classifiers and classifiers are then trained. After this test tweets are given to the model and classification is done with the help of these trained classifiers. So finally we get the tweets which are classified into n different categories.

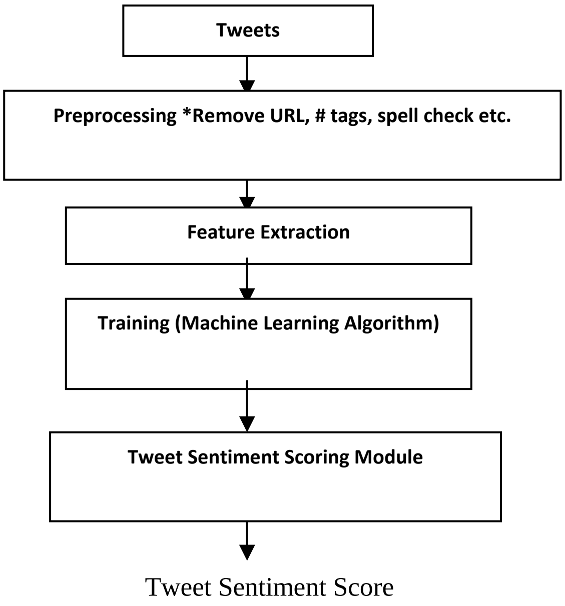
II. Objectives of the Study

As Teaching a machine to analyze the various grammatical rules, cultural variations, slang and misspellings that occur in online mentions is a difficult process. But by applying contextual understanding with the help of machine learning algorithm one can easily identify the sentiment of a sentence. Thus the objectives of this paper are:

- To provide training to the machine so that it can test the sentiments of a person.
- To study the frequencies of word which are used in tweets.
- To analyze the sentiments score of ten different categories to check the highest one.

III. Steps followed for Sentimental Analysis

Sentimental analysis in this paper is done with Naïve bayes classifier to train the machine. Algorithm followed is explained below:


Tweets
Preprocessing *Remove URL, # tags, spell check etc.
Feature Extraction
Training (Machine Learning Algorithm)
Tweet Sentiment Scoring Module
Tweet Sentiment Score

The following sub-sections expound the details of the proposed system:

1. **Dataset:** The training dataset contains the 16000 tweets and stored in the CSV file. Out of these 1200 tweets are used for training the classifiers. These datasets are collected from various sources and class labels are manually annotated whenever class labels are missing.

2. **Preprocessing of data**: This includes cleaning of data as:
   - Remove all URLs (e.g. www.example.com), hash tags (e.g. #topic), targets (@username), and special Twitter words ("e.g. RT").
   - Convert all data into lower case.
   - Correct spellings: A sequence of repeated characters is tagged by a weight.
   - Replace all the emoticons with their sentiment polarity.
   - Remove all punctuations after counting the number of exclamation marks.

3. **Classification of Tweets:** In the first step tweets and labels are passed to the classifier and feature extraction is done. Now, both these extracted features and tweets are passed to the Naïve Bayesian classifier. Then training is given to classifier with this training data. Then the classifier dump file opened in write back mode and feature words are stored in it along with a classifier. After that the file is close.

   Naïve Bayesian classifier is a probabilistic classifier that uses the properties of Bayes theorem assuming the strong independence between the features. One of the advantages of this classifier is that it demands very little measure of training data to calculate the parameters for prediction. For a given textual review d and for a class c the conditional probability for each class given a review is P(c|d). According to Bayes theorem this quantity can be computed by equation

$$P(c|d) = \frac{P(d|c) * P(c)}{P(d)}$$

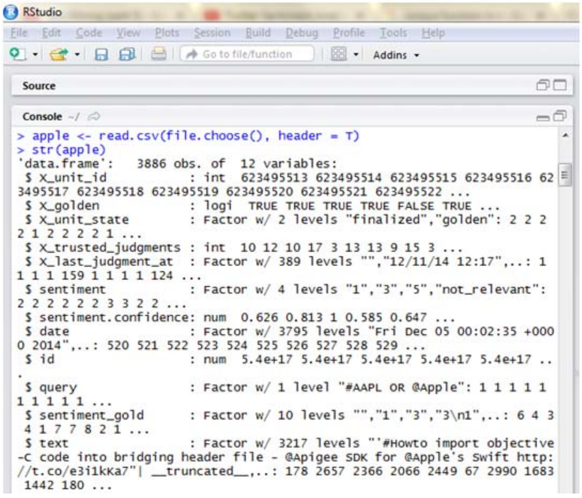
4. **Retrieving tweets for a particular topic:** After applying this procedure frequencies and probabilities of various words in tweets can be calculated. By using this data one can retrieve the sentiments of a tweet. The performance of the system depends on training datasets and also content (i.e. Tweets) in these data sets. Thus, this is very simple and effective approach to analyze the sentiment form text.

IV. RESULTS

Textual classification using machine learning is a well-studied field to measure different sentiments related to a product, brand or movie reviews. To clearly illustrate the effectiveness of the proposed method, experimental results are presented with a sample tweet. In this research apple tweets are taken as a sample data.

Now to analyze the tweets, programming is done in R which is an open source programming language and software environment for statistical computing and graphics that is supported by the R Foundation for Statistical Computing. The R language is widely used among statisticians and data miners for developing statistical software and data analysis.

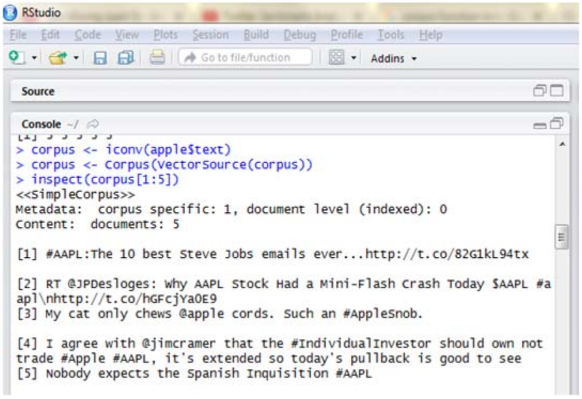
Figure 1 show the screenshot of program when apple CSV (Comma separated Value) file is attached for evaluation. The basic aim is here to study the sentiments of persons about tweets.



```
> apple <- read.csv(file.choose(), header = T)
> str(apple)
'data.frame':   3886 obs. of  12 variables:
 $ X_unit_id            : int  623495513 623495514 623495515 623495516 62
3495517 623495518 623495519 623495520 623495521 623495522 ...
 $ X_golden             : logi  TRUE TRUE TRUE TRUE FALSE TRUE ...
 $ X_unit_state         : Factor w/ 2 levels "finalized","golden": 2 2 2
2 1 2 2 2 2 1 ...
 $ X_trusted_judgments : int  10 12 10 17 3 13 13 9 15 3 ...
 $ X_last_judgment_at  : Factor w/ 389 levels "","12/11/14 12:17",..: 1
1 1 1 159 1 1 1 1 124 ...
 $ sentiment            : Factor w/ 4 levels "1","3","5","not_relevant":
2 2 2 2 2 2 3 3 2 2 ...
 $ sentiment.confidence: num  0.626 0.813 1 0.585 0.647 ...
 $ date                 : Factor w/ 3795 levels "Fri Dec 05 00:02:35 +000
0 2014",..: 520 521 522 523 524 525 526 527 528 529 ...
 $ id                   : num  5.4e+17 5.4e+17 5.4e+17 5.4e+17 5.4e+17 ..
.
 $ query                : Factor w/ 1 level "#AAPL OR @Apple": 1 1 1 1 1
1 1 1 1 1 ...
 $ sentiment_gold       : Factor w/ 10 levels "","1","3","3\n1",..: 6 4 3
 4 1 7 7 8 2 1 ...
 $ text                 : Factor w/ 3217 levels "'#Howto import objective
-C code into bridging header file - @Apigee SDK for @Apple's Swift http:
//t.co/e3i1kKa7"| __truncated__,..: 178 2657 2366 2066 2449 67 2990 1683
 1442 180 ...
```

Figure 1: Data content of apple CSV file

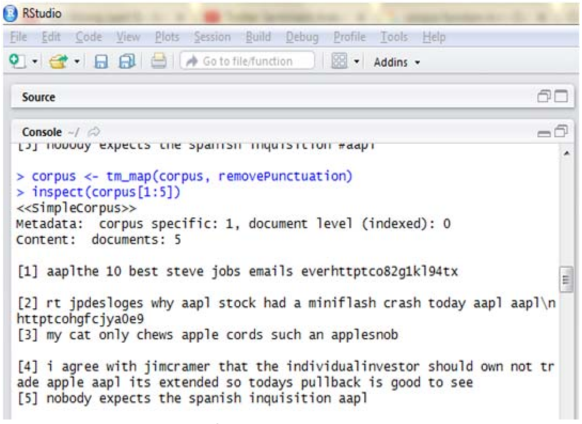
Next step is to make a corpus by using Vector source. Figure 2 shows 1 to 5 lines of the selected field.



```
[1] 5 5 5 5 5
> corpus <- iconv(apple$text)
> corpus <- Corpus(VectorSource(corpus))
> inspect(corpus[1:5])
<<SimpleCorpus>>
Metadata:  corpus specific: 1, document level (indexed): 0
Content:  documents: 5

[1] #AAPL:The 10 best Steve Jobs emails ever...http://t.co/82G1kL94tx

[2] RT @JPDesloges: Why AAPL Stock Had a Mini-Flash Crash Today $AAPL #a
apl\nhttp://t.co/hGFcjYa0E9
[3] My cat only chews @apple cords. Such an #AppleSnob.

[4] I agree with @jimcramer that the #IndividualInvestor should own not
trade #Apple #AAPL, it's extended so today's pullback is good to see
[5] Nobody expects the Spanish Inquisition #AAPL
```

Figure 2: Formation of a Vector source corpus field.

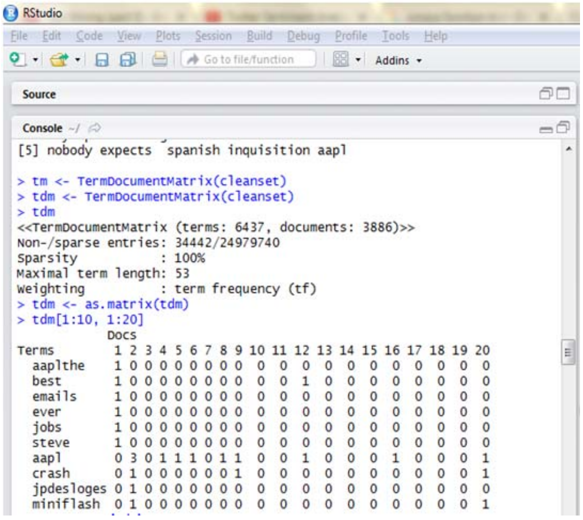
After this, preprocessing is done on input data by removing punctuation, URLs, Spell checking etc.



```
[5] nobody expects the spanish inquisition #aapl

> corpus <- tm_map(corpus, removePunctuation)
> inspect(corpus[1:5])
<<SimpleCorpus>>
Metadata:  corpus specific: 1, document level (indexed): 0
Content:  documents: 5

[1] aaplthe 10 best steve jobs emails everhttptco82g1kl94tx

[2] rt jpdesloges why aapl stock had a miniflash crash today aapl aapl\n
httptcohgfcjya0e9
[3] my cat only chews apple cords such an applesnob

[4] i agree with jimcramer that the individualinvestor should own not tr
ade apple aapl its extended so todays pullback is good to see
[5] nobody expects the spanish inquisition aapl
```

Figure 3: Removal of Punctuation: A preprocessing step

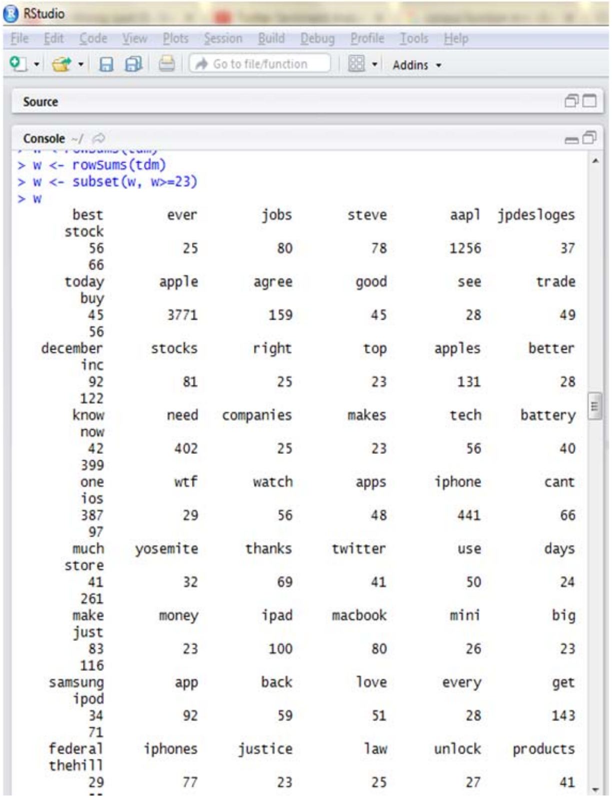
Now a term document matrix is prepared from cleaned data. In Figure 4 screenshot represent there are 6437 terms and 3886 documents are used in data file. A matrix is prepared which shows the frequency that how many times a word is used in a document. In this proposed method each row is considered as a document.


RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
Source
Console ~/

```
[5] nobody expects  spanish inquisition aapl

> tm <- TermDocumentMatrix(cleanset)
> tdm <- TermDocumentMatrix(cleanset)
> tdm
<<TermDocumentMatrix (terms: 6437, documents: 3886)>>
Non-/sparse entries: 34442/24979740
Sparsity           : 100%
Maximal term length: 53
Weighting          : term frequency (tf)
> tdm <- as.matrix(tdm)
> tdm[1:10, 1:20]
            Docs
Terms        1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
  aaplthe    1 0 0 0 0 0 0 0 0  0  0  0  0  0  0  0  0  0  0  0
  best       1 0 0 0 0 0 0 0 0  0  0  1  0  0  0  0  0  0  0  0
  emails     1 0 0 0 0 0 0 0 0  0  0  0  0  0  0  0  0  0  0  0
  ever       1 0 0 0 0 0 0 0 0  0  0  0  0  0  0  0  0  0  0  0
  jobs       1 0 0 0 0 0 0 0 0  0  0  0  0  0  0  0  0  0  0  0
  steve      1 0 0 0 0 0 0 0 0  0  0  0  0  0  0  0  0  0  0  0
  aapl       0 3 0 1 1 1 0 1 1  0  0  1  0  0  0  1  0  0  0  1
  crash      0 1 0 0 0 0 0 0 1  0  0  0  0  0  0  0  0  0  0  1
  jpdesloges 0 1 0 0 0 0 0 0 0  0  0  0  0  0  0  0  0  0  0  0
  miniflash  0 1 0 0 0 0 0 0 0  0  0  0  0  0  0  0  0  0  0  1
```

Figure 4: Formation of matrix for analysis

To calculate the frequency of individual word in the entire file row sum function is applied on the above matrix.


RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
Source
Console ~/

```
> w <- rowSums(tdm)
> w <- subset(w, w>=23)
> w
        best        ever        jobs       steve        aapl  jpdesloges
       stock
          56          25          80          78        1256          37
          66
       today       apple       agree        good         see       trade
         buy
          45        3771         159          45          28          49
          56
    december      stocks       right         top      apples      better
         inc
          92          81          25          23         131          28
         122
        know        need   companies       makes        tech     battery
         now
          42         402          25          23          56          40
         399
         one         wtf       watch        apps      iphone        cant
         ios
         387          29          56          48         441          66
          97
        much    yosemite      thanks     twitter         use        days
       store
          41          32          69          41          50          24
         261
        make       money        ipad     macbook        mini         big
        just
          83          23         100          80          26          23
         116
     samsung         app        back        love       every         get
        ipod
          34          92          59          51          28         143
          71
     federal     iphones     justice         law      unlock    products
     thehill
          29          77          23          25          27          41
```

Figure 5: Frequencies of words in apple file.

Figure 5 show that word apple is used 3771 times in the entire file. In this, only those words whose frequency equal to or greater than 23 are taken.

V. CONCLUSION

The proliferation of micro blogging sites like Twitter offers an unprecedented opportunity to create and employ theories & technologies that search and mine for sentiments. The work presented in this paper specifies a novel approach for sentiment analysis on Twitter data. Machine learning algorithms (Naive Bayes classifier) can achieve high accuracy for classifying sentiment when using this method. A study on apple tweets is done by using r programming language. By following above system model approach it is concluded that there are ten different sentiments among which positive shows the highest result. It means customer reviews are positive for these apple tweets.

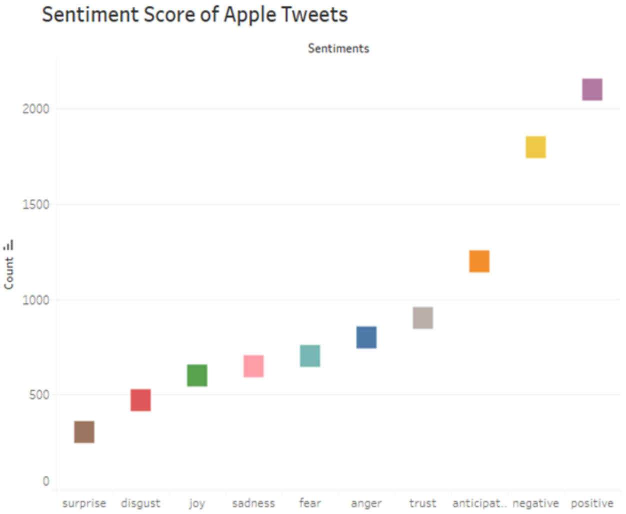


Figure 6: Graph for Sentiments Score

Thus this technique is very helpful for business owners and advertising companies to study the response of customer toward their product. Figure 6 show the output result of sentiments among which positive sentiments are larger one.

.

REFERENCES:

[1] S. Amir et al, “Exploiting unlabelled data for twitter sentiment analysis," pp. 673, 2014.

[2] R. Feldman, “Techniques and applications for sentiment analysis," in Communications of the ACM, vol. 56, pp. 82-89, 2013.

[3] Vivek Narayanan, et al., “Fast and accurate sentiment classification using an enhanced naive bayes model” in Intelligent Data Engineering and Automated Learning{IDEAL}, pp 194-201. Springer, 2013.

[4] K. Mouthami et al., “Sentiment analysis and classification based on textual reviews," in Information Communication and Embedded Systems (ICICES), pp. 271-276, IEEE, 2013.

[5] Z. Wang et al, “Semi-supervised learning for imbalanced sentiment classification," in IJCAI Proceedings-International Joint Conference on Artificial Intelligence, vol. 22, pp. 1826, 2011.

[6] Andrew L Maas et al., “Learning word vectors for sentiment analysis” in Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies. Vol 1, pp:142-150., 2011.

[7] A. Pak and P. Paroubek. “Twitter as a Corpus for Sentiment Analysis and Opinion Mining”. in Proceedings of the Seventh Conference on International Language Resources and Evaluation, pp.1320–1326, 2010.

[8] A. Bifet and E. Frank, ”Sentiment Knowledge Discovery in Twitter Streaming Data”, in Proceedings of the 13th International Conference on Discovery Science, Berlin, Germany: Springer , pp. 1–15, 2010.