Twitter Sentiment Analysis

Paras Babu Tiwari

1 Introduction

Millions of people are using social media such as Facebook, Twitter and Blog to express their opinion and reaction. People are posting messages about companies' products in these medias. Companies need to listen to their customers voice raised in these medias. These medias provide valuable information to companies to understand the customers' feedback.

The volume of data produced per day by these medias is overwhelming. There are around 400 million tweets sent each day. Facebook has around 500 million users and almost 50% of them login in each day. So it's impossible to monitor the massive data produced in these platform manually. We need a tool that will monitor these medias and provide a nice presentation of the information presented in these medias.

Sentiment analysis is important for the companies as it helps to monitor the customers' feedback. Twitter is one of the popular social media and a number of companies, celebrities, and politicians are being mentioned in Twitter. The sentiment analysis of Twitter data classifies each tweet into one of three categories namely positive, negative and neutral. In this project work, we developed an application that can perform sentiment analysis of Twitter data for the specified search term. We developed following different modules for the project works:

1) Module to fetch the data from Twitter for the specified search terms
2) Classifier to classify the tweets into positive, negative and neutral class
3) A module to visualize the result using pie chart.

2 Method

We developed three different modules to build the sentiment analysis application.

2.1 Module to fetch data from Twitter

We used Twitter REST API [1] to get the tweets corresponding to the search term. User must be authenticated to request the data from Twitter REST API. Twitter uses OAuth authentication [2] mechanism. The OAuth mechanism requires the secret key and password to request the access token. After having secret key and password, we need an access token to fetch the data from the Twitter. The access token is generated for each user,

and user needs to authorize the application to get the access token. Finally, application can fetch the data using the access token.

We registered a new application in Twitter to get the secret key and password. We used the secret key and password to request the access token. During this process, Twitter redirects user to authentication page. We got the access token when user authorized our application. After that, we can use access token to fetch the data from Twitter. We used Twitter search API[3] to fetch the data from Twitter. The Twitter search API hides the low level detail and simplifies the data fetching process.

2.2 Classifier to classify the sentiment of Tweets

We used Naïve Bayes algorithm[4] to classify tweets' sentiment. The tweets sentiment could be positive, negative and neutral. First, we classified tweets into neutral and non-neutral, and then non-neutral tweets were classified into positive and negative.

2.2.1 Training Data

Twitter API returns the tweets based on smiley. We got positive and negative tweets when we query Twitter with the happy, and sad smiley respectively. Sentiment140 [5] collected the positive and negative training data using the Twitter API, and we used the data set to train the classifier. However, the data set did not have neutral tweets. Generally, the newspaper headlines has the neutral sentiment [6]. We fetched tweets of BBC, New York Times and CNN, performed post processing and labeled them as neutral tweets. In the post-processing step, we removed re-tweets from tweets so that we don't give any special weight to some group of words. We used 5000 instances of each type of sentiment, and have total of 15000 tweets to train the classifier.

2.2.2 Features

Each unique word of tweets is a feature for the classifier. The number of feature equals to the numbers of unique words in the training instance. The number of rows in the feature matrix equals to the number of training instance and the number of column equals the number of unique words in the training instance. We have 33073 features for the neutral and non-neutral classifier and 19788 features for the positive and negative classifier. So the feature matrix has 15000 rows and 33073 columns for the neutral and non-neutral classifier, and 15000 rows and 19788 columns for the positive and negative classifier. We used feature presence instead of feature count for the experiment as feature presence gives better accuracy than feature count for sentiment analysis [6]. Therefore, each entry of the feature matrix is a binary value with the true value indicating that the word is present in the tweet, and false value indicating that the word is not in the tweet.

2.3 Features Reduction

Not all features are informative for the sentiment analysis. Therefore, we applied following approach for the feature reduction

1) Converting each word to lower case
2) Removing Punctuation Marks
3) Removing stop words
4) Removing the user name from the tweet

Table 1 and 2 shows the percentage of feature reduction for neutral and non-neutral and positive and negative classifiers.

Method	**Number of features before applying the transformation (A)**	**Number of Features after applying the transformation (B)**	**Percentage Reduction (A-B)/A *100**
Converting each word to lower case	33615	33073	1.61
Removing Punctuation Marks	38947	33073	15.08
Removing stop words	34106	33073	3.02
Removing the user name from the tweet	38444	33073	13.97

Table 1: Percentage of feature reduction for the classifier that classifies tweets into neutral and non-neutral classes. The second column lists the number of features without applying the feature reduction method and third column lists the number of features after applying all feature reduction methods.

Method	**Number of features before applying the transformation (A)**	**Number of Features after applying the transformation (B)**	**Percentage Reduction (A-B)/A *100**
Converting each word to lower case	20256	19788	2.310
Removing Punctuation Marks	23620	19788	16.22
Removing stop words	20737	19788	4.57

Removing the user name from the tweet	24615	19788	19.60

Table 2: Percentage of feature reduction for the classifier that classifies tweets into positive and negative classes. The second column lists the number of features without applying the feature reduction method and third column lists the number of features after applying all feature reduction methods.

2.3.1 Converting each word to lower case

People use casual language in the tweet and the sentence may not be grammatically correct. Therefore, the same word may appear in lower and upper case. The case of a word does not tell any information about the sentiment of the tweets. Therefore, we converted all words in the tweets to the lower case so that the word like “Obama”, and “obama” is treated as one feature. We approximately reduce 1.6% - 2.3% of features using lower case representation.

2.3.2 Removing Punctuation Marks

The tweets have very poor punctuation because of the causality nature of it. So the same words might be appear with different punctuation symbols. For e.g. the word love might appear as “love!!!!”, “love….”, “love,” in different tweets. The extra punctuation symbols carry no information about the sentiment. Therefore, we removed punctuation symbols from the tweets. This method reduced the feature by 15%-16%.

2.3.3 Removing stop words

There are a lot of words that appears frequently in a sentence. Example of those words are article (a,an, the,) relative clauses, prepositions etc. These words do not carry any sentiment. We found a list of stop words [7] and removed these words from the tweets. This method reduced feature by 3% - 4.5%

2.3.4 Removing the user name from the tweet

Each tweet starts with the username. These are the identifier used to identify user uniquely in tweets. These words do not carry any sentiment. Therefore, we removed the username from tweets, thus, reducing the feature by 13-19%

2.4 Drawing Pie Chart

We used matplotlib [8] library to draw the pie chart of the result.

2.5 Test Data

We obtained the test data from sentiment140.com.

3 Result

We tested the classifier using test data. The classifier has accuracy of around 67% . Figure 1 shows the result of applying sentiment analysis in tweets obtained for the "Lady Gaga","Obama" and "Mcdonald". We used recent 100 tweets for each term and classify the tweets into positive, negative and neutral. The Lady Gaga search term returned a lot of positive tweets, and a fewer negative tweets. Similarly, Obama search term returned more number of neutral tweets than the positive and negative. Finally the Mcdonald search term returned almost equal number of positive and negative tweets.

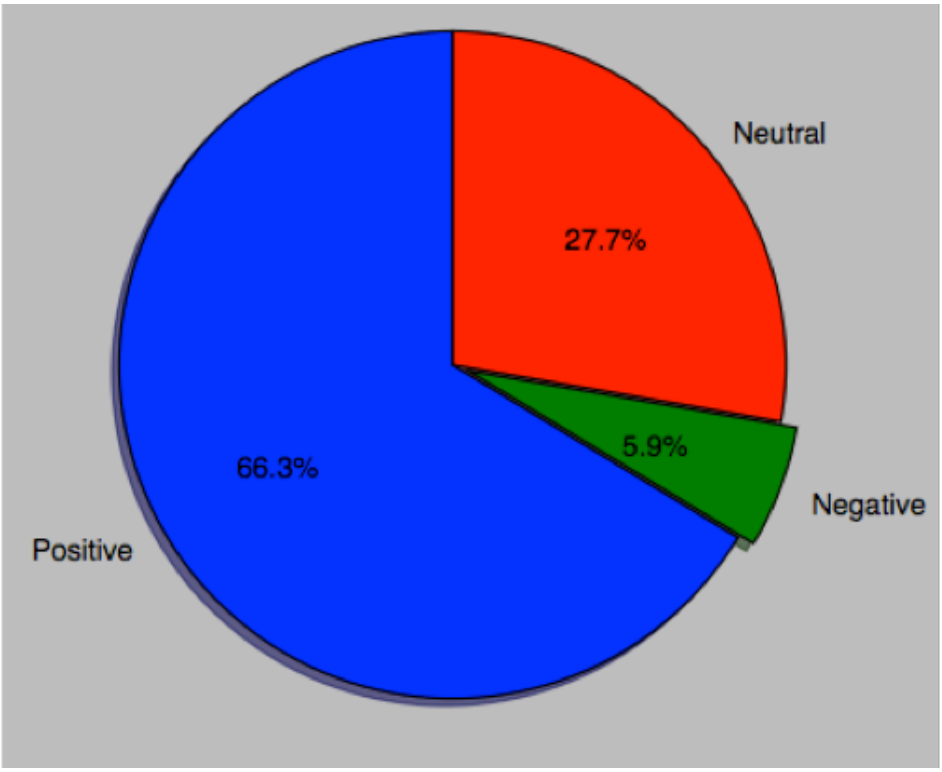

Figure 1a Sentiment Analysis of Lady Gaga's tweets.

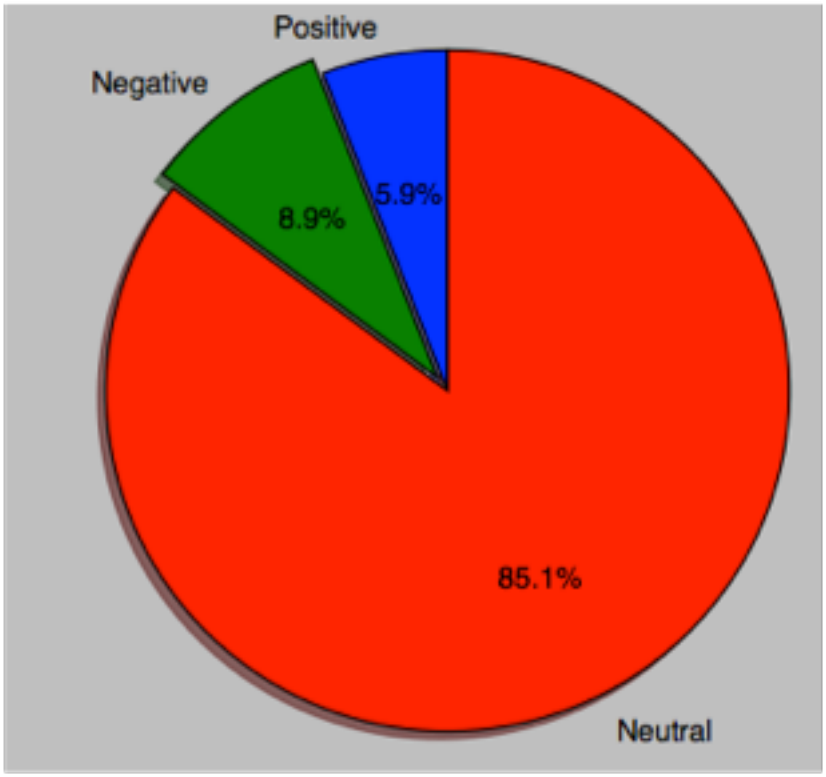

Figure 1b. Sentiment analysis of Obama's tweets.

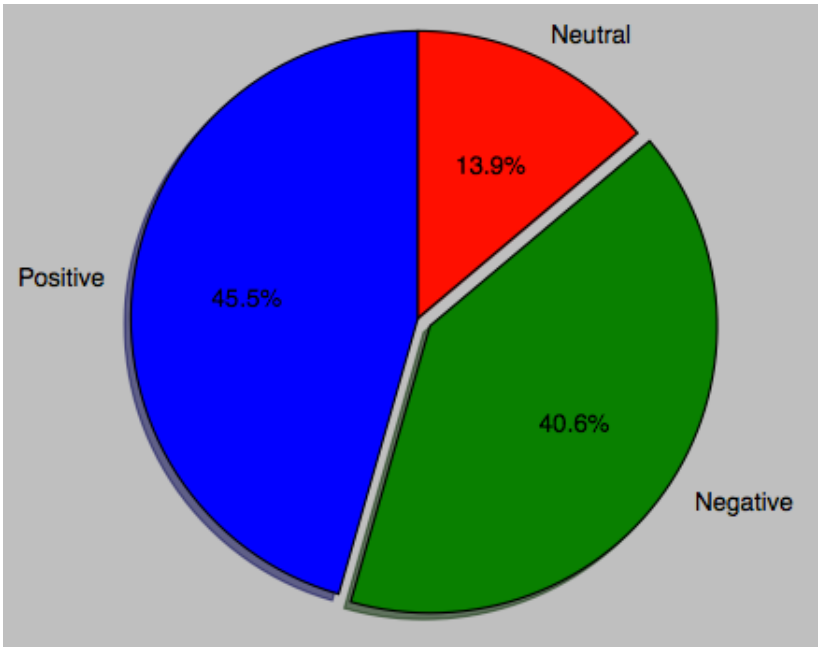

Figure 1c Sentiment Analysis of McDonald tweets.

Figure 1: Sentiment Analysis of the Twitter data using Naïve Bayes model. We used Lady Gaga, Obama, and McDonald search terms and the sentiment analysis of the tweets corresponding to those search terms are shown in Figure 1a, 1b and 1c. Tweets were categorized into positive, negative and neutral. We classified tweets into neutral and non-neutral, and then non-neutral tweets were classified into positive and negative tweets.

4 Discussion

We implemented the Naïve Bayes classifier for the sentiment analysis of Twitter data. We tested the application in real-word tweets for three different search terms. Lady Gaga is a celebrity and she gets positive tweets from her fan. Therefore, her tweet had more number of positive tweets than negative and neutral tweets. Barack Obama is the president of the USA and he is mentioned frequently in the newspaper article. At the time of generating the result, there was a lot of news about Obama travelling to South Africa to attend the funeral of Nelson Mandela. People have mix reaction about McDonald, and around 80% of tweets about McDonald had sentiment.

5 Conclusion and Future Work

We developed a Naïve Bayes classifier to classify tweets into positive, negative, and neutral sentiment. The classifier has the accuracy of 67%. The application can search and fetch data from Twitter for the specified search term.

We have used only 15000 training data instances, which is not enough to build a powerful classifier. We can improve the accuracy of the classifier by increasing the number of training data. Similarly, we can reduce feature count using other feature reduction method discussed in data mining.

6 References

[1] *Twitter REST API. https://dev.twitter.com/docs/api/1.1*. .
[2] *OAuth, available from http://oauth.net/documentation/getting-started/*. .
[3] *Twitter Search, available from https://pypi.python.org/pypi/TwitterSearch/*. .
[4] E. Loper and S. Bird, "NLTK: The Natural Language Toolkit," in *Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics - Volume 1*, Stroudsburg, PA, USA, 2002, pp. 63–70.
[5] *Sentiment140, available from http://www.sentiment140.com/*. .
[6] A. Agarwal, B. Xie, I. Vovsha, O. Rambow, and R. Passonneau, "Sentiment Analysis of Twitter Data," in *Proceedings of the Workshop on Languages in Social Media*, Stroudsburg, PA, USA, 2011, pp. 30–38.
[7] "List of stop words, available from http://jmlr.org/papers/volume5/lewis04a/a11-smart-stop-list/english.stop." .
[8] *Matplotlib available at http://matplotlib.org/*. .