SENTIMENT ANALYSIS

Submitted in partial fulfillment of the requirements of the

PYTHON LAB

By

**SIDDHARTH SINHA**

**SANTOSH SAVALIYA**

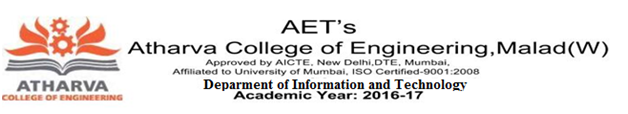
**DEVENDRA PATIL**

**AKASH VISHWAKARMA**

**VIRENDRA RATHORD**

Supervisor:

**Prof. Samira Nigrel**



# Department of Information Technology

### Atharva College of Engineering

*Year: 2017-18*

Introduction

Twitter users around the world post around 350,000 new Tweets every minute, creating 6,000 140-character long pieces of information every second. Twitter is now a hugely valuable resource from which you can extract insights by using text mining tools like sentiment analysis.

Within the social chatter being generated every second, there are vast amounts of hugely valuable insights waiting to be extracted. With sentiment analysis, we can generate insights about consumers’ [reactions to announcements](http://blog.aylien.com/ryanair-handles-pr-crisis-text-analysis-case-study/), opinions on products or brands, and even track opinion about [events as they unfold](http://blog.aylien.com/using-nlp-to-understand-how-twitter-and-the-media-reacted-to-the-super-bowl-51-ads-battle/). For this reason, you’ll often hear sentiment analysis referred to as “opinion mining”.

With this in mind, we decided to put together a useful tool built on a single Python script to help you get started mining public opinion on Twitter.

**What is sentiment analysis?**Sentiment Analysis is the process of ‘computationally’ determining whether a piece of writing is positive, negative or neutral. It’s also known as **opinion mining**, deriving the opinion or attitude of a speaker.

**Why sentiment analysis?**

* **Business:** In marketing field companies use it to develop their strategies, to understand customers’ feelings towards products or brand, how people respond to their campaigns or product launches and why consumers don’t buy some  
  products.
* **Politics:** In political field, it is used to keep track of political view, to detect consistency and inconsistency between statements and actions at the government level. It can be used to predict election results as well!
* **Public Actions:** Sentiment analysis also is used to monitor and analyse social phenomena, for the spotting of potentially dangerous situations and determining the general mood of the blogosphere.

**Installation:**

* **Tweepy:** [tweepy](http://docs.tweepy.org/en/v3.5.0/) is the python client for the official [Twitter API](https://dev.twitter.com/rest/public).  
  Install it using following pip command:

pip install tweepy

* **TextBlob:** [textblob](http://textblob.readthedocs.io/en/dev/) is the python library for processing textual data.  
  Install it using following pip command:

pip install textblob

Also, we need to install some NLTK corpora using following command:

python -m textblob.download\_corpora

(Corpora is nothing but a large and structured set of texts.)

**Authentication:**In order to fetch tweets through Twitter API, one needs to register an App through their twitter account. Follow these steps for the same:

* Open *apps.twitter.com* and click the button: ‘Create New App’
* Fill the application details. You can leave the callback url field empty.
* Once the app is created, you will be redirected to the app page.
* Open the ‘Keys and Access Tokens’ tab.
* Copy ‘Consumer Key’, ‘Consumer Secret’, ‘Access token’ and ‘Access Token Secret’.

We follow these 3 major steps in our program:

* Authorize twitter API client.
* Make a GET request to Twitter API to fetch tweets for a particular query.
* Parse the tweets. Classify each tweet as positive, negative or neutral.

Now, let us try to understand the above piece of code:

* First of all, we create a **TwitterClient** class. This class contains all the methods to interact with Twitter API and parsing tweets. We use **\_\_init\_\_** function to handle the authentication of API client.
* In **get\_tweets** function, we use:

fetched\_tweets = self.api.search(q = query, count = count)

to call the Twitter API to fetch tweets.

* In **get\_tweet\_sentiment** we use textblob module.

analysis = TextBlob(self.clean\_tweet(tweet))

TextBlob is actually a high level library built over top of [NLTK](http://www.nltk.org/) library. First we call **clean\_tweet** method to remove links, special characters, etc. from the tweet using some simple regex.  
Then, as we pass **tweet** to create a **TextBlob** object, following processing is done over text by textblob library:

* + Tokenize the tweet ,i.e split words from body of text.
  + Remove stopwords from the tokens.(stopwords are the commonly used words which are irrelevant in text analysis like I, am, you, are, etc.)
  + Do POS( part of speech) tagging of the tokens and select only significant features/tokens like adjectives, adverbs, etc.
  + Pass the tokens to a **sentiment classifier** which classifies the tweet sentiment as positive, negative or neutral by assigning it a polarity between -1.0 to 1.0 .

Here is how **sentiment classifier** is created:

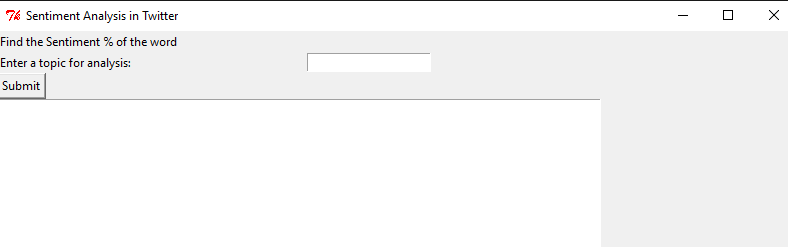
* + **TextBlob** uses a Movies Reviews dataset in which reviews have already been labelled as positive or negative.
  + Positive and negative features are extracted from each positive and negative review respectively.
  + Training data now consists of labelled positive and negative features. This data is trained on a [Naive Bayes Classifier](https://en.wikipedia.org/wiki/Naive_Bayes_classifier).

Then, we use **sentiment.polarity** method of **TextBlob** class to get the polarity of tweet between -1 to 1.  
Then, we classify polarity as:

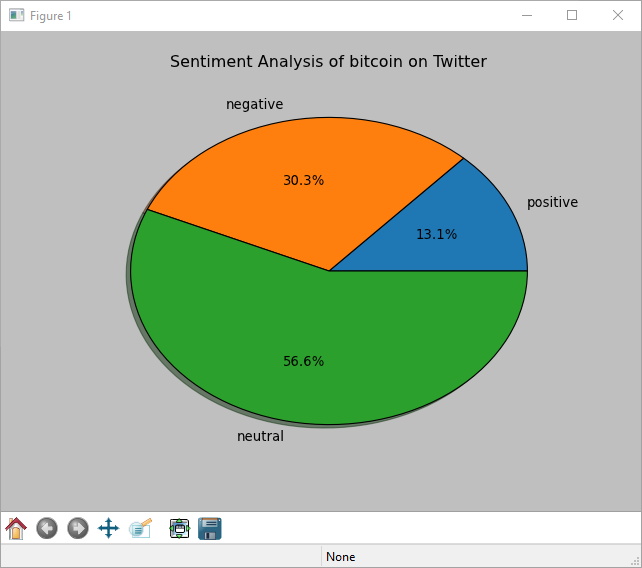
* Finally, parsed tweets are returned. Then, we can do various type of statistical analysis on the tweets. For example, in above program, we tried to find the percentage of positive, negative and neutral tweets about a query.

Screenshots:

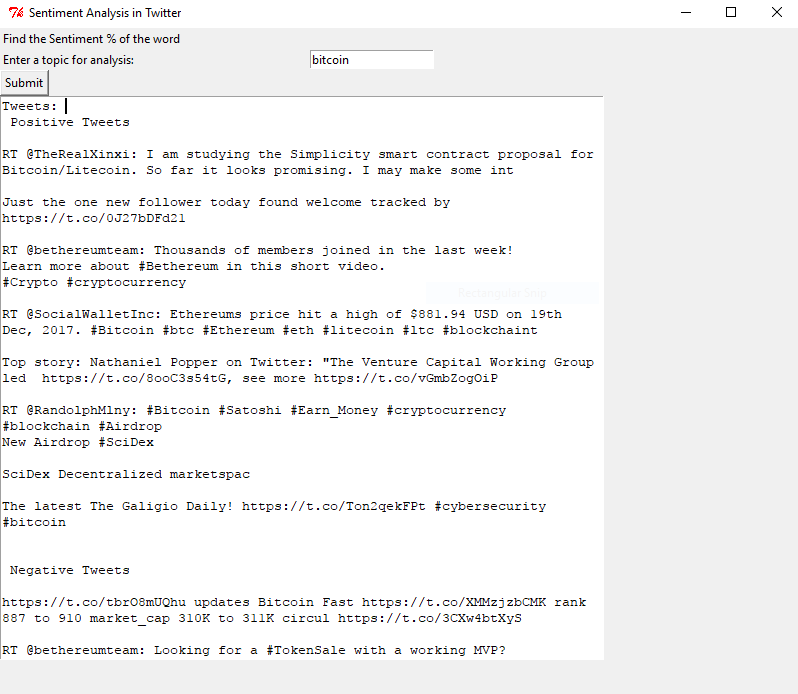
1. Run the project. A GUI window pops up. Enter the topic to be analyzed in the text box provided & click on ‘Submit’ button to submit.



1. After submitting the topic, a window with pie chart pops up showing the percentages of positive, negative & neutral sentiments of the tweets. The chart can be panned, zoomed & saved offline.



1. On closing the pie chart window, some positive & negative tweets shows up on the previous window along with the percentages of positive, negative & neutral sentiments of the tweets.



**References:**

<http://www.ijcaonline.org/research/volume125/number3/dandrea-2015-ijca-905866.pdf>

<https://textblob.readthedocs.io/en/dev/quickstart.html#sentiment-analysis>

[textblob.readthedocs.io/en/dev/\_modules/textblob/en/sentiments.html](http://textblob.readthedocs.io/en/dev/_modules/textblob/en/sentiments.html)