Parallel & Distributing Computing

Project Report



# Dynamic Developers: Faculty:

## Muhammad Ghous Sarwar Dr. Ayaz

## Syed Fasih Ali

## Tehreem feroz

Contents

[Sentiment Analysis 2](#_Toc532238174)

[**Collecting the data set of the tweets** 2](#_Toc532238175)

[Setting Up twitter app. (developer account) 2](#_Toc532238176)

[Twitter Scraper 3](#_Toc532238177)

[Usage 3](#_Toc532238178)

[Installation 4](#_Toc532238179)

[Application of Sentiwordnet: 4](#_Toc532238180)

[lemmatizer: 4](#_Toc532238181)

[SynsNet 4](#_Toc532238182)

[Scoring 4](#_Toc532238183)

[**Application of MPI:** 5](#_Toc532238184)

[Getting username from CSV file: 5](#_Toc532238185)

[Setting up MPI4PY: 5](#_Toc532238186)

# Sentiment Analysis

In Natural Language Processing there is a concept known as Sentiment Analysis.

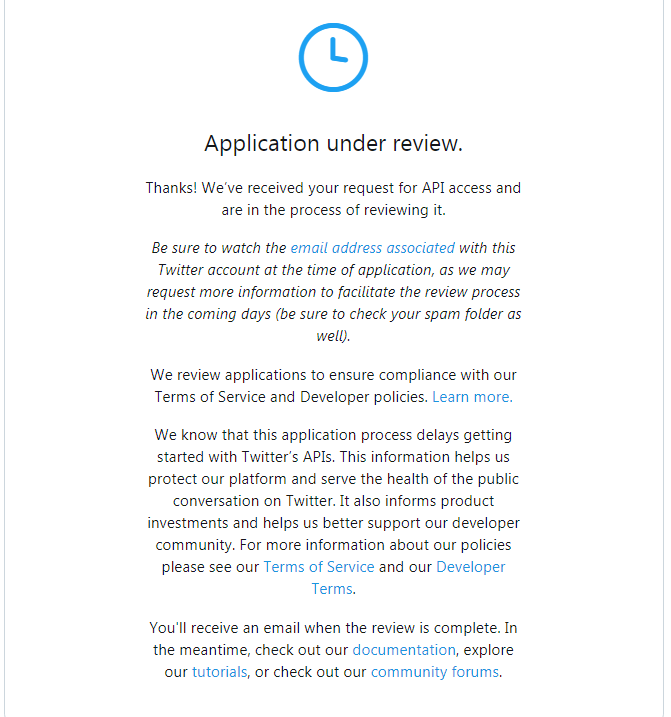
Given a movie review or a tweet, it can be automatically classified in categories.  
These categories can be user defined (positive, negative) or whichever classes you want.

**Collecting the data set of the tweets.** [by S fasih ali 60838 ]

# Setting Up twitter app. (developer account)

In order to extract tweets for a posterior analysis, we need to access to our Twitter account and create an app. The website to do this is <https://apps.twitter.com/>.

After applying for developers account:



As it was taking long to approved. We found another way or solution to get data from twitter by scrapping.

# Twitter Scraper [by S fasih ali 60838 ]

Twitter's API is annoying to work with, and has lots of limitations — luckily their frontend (JavaScript) has its own API, which I reverse–engineered. No API rate limits. No restrictions. Extremely fast.

You can use this library to get the text of any user's Tweets trivially.

## Usage

>>> from twitter\_scraper import get\_tweets

>>> for tweet in get\_tweets('kennethreitz', pages=1):

>>> print(tweet['text'])

It appears you can ask for up to 25 pages of tweets reliably (~486 tweets).

Reference: [ <https://github.com/kennethreitz/twitter-scraper>].

So we are going to use twitter\_Scrapper for data collection.

## Installation

pip install twitter-scraper

# Application of Sentiwordnet: [By Muhammad Ghous, 59455]

We did sentimental analysis by Python NLTK. SentiWordNet is a lexical resource for opinion mining. SentiWordNet assigns to each synset of WordNet three sentiment scores: positivity, negativity, objectivity.

The usage of sentiwordnet 3.0 for sentiment analysis is the main objective. The approach is simple:

* Each word is tagged using POS Tagger
* The lemma of each tagged word is found using wnl lemmatizer
* Scores using synsets for each lemmatized word are computed and added.
* This process is iterated over for each sentence/phrase to find the sentence score

## lemmatizer:

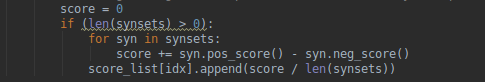
Lemmatisation (or lemmatization) in linguistics, is the process of grouping together the different inflected forms of a word so they can be analysed as a single item.

# SynsNet:

It groups English words into sets of synonyms, provides short definitions and usage examples, and records a number of relations among these synonym sets or their members.

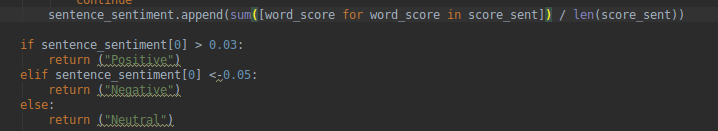
Scoring**:**

syn.pos\_score() and syn.neg\_score() are use for scoring negative and positive sentences than after that we score for each word in sentence. Finally we use formula



formula: sumof(score\_of\_word) / score of sentence

if we get less than -0.05 then tweet is Negative and more than 0.03 then it is positive else neutral.



# **Application of MPI:**

## Getting username from CSV file: [By Muhammad Ghous, 59455]

## Setting up MPI4PY: [by S fasih ali ]

Downloading**:**

pip3 install MPI4PY

Initializing**:**

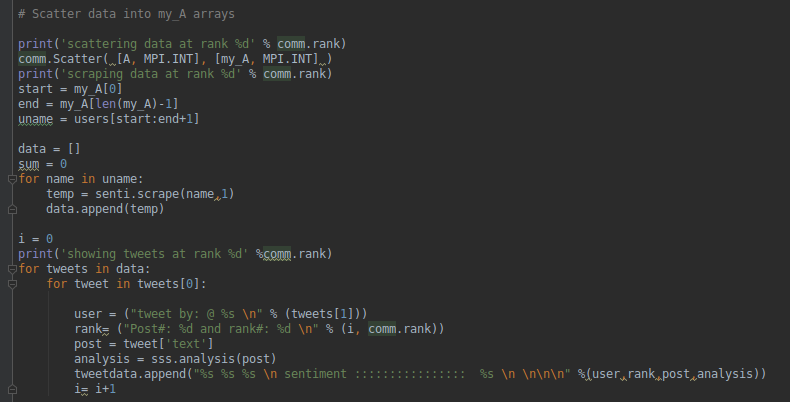
comm = MPI.COMM\_WORLD

size = comm.size

rank = comm.rank

Scatteringdata**:**

MPI\_scatter divides username according to given size (processes/system)

all nodes will scrape tweets and do analysis save them in array with sentiments

# Sequential vs Distributed Processing (Classification) [By Muhammad Ghous, 59455]

160 Tweets

Processor 1 with 1 core

3 GB Ram

|  |  |  |
| --- | --- | --- |
| Reading | Sequential Processing Time | Distributed Processing Time  2 processes |
| 1 | 16.609990119934082 seconds | 14.371477127075195 seconds |
| 2 | 16.30209755897522 seconds | 13.930231094360352 seconds |
| 3 | 17.63021206855774 seconds | 13.847803354263306 seconds |

Above analysis is performed on a single machine with above mentioned specifications, so we can say that if we have multiple machine having at least above mentioned specification our processing power will increase rapidly.

**Screenshot Attached**

