

DATA ANALYSIS USING TWITTER API

BY:

Eddie Aguilar
Pranik Chainani
Zachariah Dohogne
Mohamed Gour
Shawn Manalel

GitHub Repo:



<https://github.com/duardo56/BluJMine.git>

Abstract

Globally, Millions of tweets are being generated by the hour. Smartphone users are clicks and taps away from updating and sharing their thoughts and expressions about anything and everything out there. Every word, image, hashtag, even emojis that are being uploaded on social media are not confined within the realms of words, sentences, expressions or sentiments. This massive stream of constantly updated information is being monitored and studied by various firms to come up with prediction algorithms for marketing and other research. The primary goal of this research is to demonstrates the power of Twitter API and the how the data is streamed, organized and displayed for the various purposes suitable to a given entity that chooses to mine the data of their choice and preference.

Table of Contents

Requirements.....	2
Introduction.....	2
Twitter Application Interface.....	3
Tweepy.....	5
Pandas.....	6
Matplotlib.....	6
Extracting Data.....	6
Conclusion.....	19
Sources.....	19

Requirements

To complete this guide you will need:

- Computer
- Internet Connection
- Twitter account
- A Python Compiler
- Terminal/Linux/CMD
- Text editor

Introduction

What is Twitter? Twitter is an online news and social networking service created in 2006, where users post and interact with messages (“tweets”) restricted to 140 characters. As of 2016, twitter has 319 million monthly active users who post 500 million tweets per day. It’s users include political figures, religious leaders, corporations, celebrities, and students among the few broad topics. Typically, when a twitter user posts a tweet, they incorporate hashtags. What are **hashtags**? Hashtags are words or phrases prefixed with the “#” symbol that are used to group tweets together by topic or type (See Tweet Example).



With millions of tweets being posted each day, Twitter provides a rich source of information that is available for consumption with the right tools. The data can be extracted and used to determine the popularity of presidential candidates, gather customer sentiment, or discover trends based on keywords (hashtags).

Twitter Application Program Interface

Twitter has two kinds of Application Program Interfaces available to developers that extract data from their services. The REST API and the Streaming API are the two means of data extraction from twitter's website for the purpose of data mining. The REST API is used by developers who need to go through archived tweets to learn trends from the past in order to make stronger future predictions. The Streaming API is used whenever the developer needs to go through the tweets that are being tweeted live for instant pattern recognitions and spontaneous predictions. In Streaming API, all upcoming tweets are retrieved, printed out or saved to a file. This paper focuses on Streaming API in order to gather current data on three popular sports "football", "basketball", and "soccer" for the purpose of comparing popularity and longevity of the two games. The data gathered for this project was derived by filtering tweets using keywords which returned live tweet data relevant to the three keywords or hashtags.

Tweepy

Tweepy is a python library for accessing the Twitter API. The twitter API can handle all the data streaming that goes through twitter every day. The data that flows through twitter are defined in tweets which are in JSON format (JavaScript Object Notation). This format is used to define tweets in a more organized manner. Tweepy extracts the tweets and encapsulates them within brackets shown below.

```
1 {"created_at":"Sun Apr 02 05:40:49 +0000 2017","id":848409865500319744,"id_str":"848409865500319744",
  "text":"\u3010\u30dd\u30b1\u30e2\u30f3\u3011\u304a\u308c\u3001\u30dd\u30b1\u30e2\u30f3\u304c\u306e\u5c71\u3075\u3052\u30dd\u30d
f1\u3076\u3077\u3059\u308f!!\u3000 \u3073\u307d\u3012https://t.co/7JW5aFQAK\u3000 pokemongo","source":"\u003ca
href=\"http://t.co/7JW5aFQAK\" rel=\"nofollow\" \u003etwitter\u003c/a\u003e","truncated":false,
  "in_reply_to_status_id":null,"in_reply_to_status_id_str":null,"in_reply_to_user_id":null,"in_reply_to_user_id_str":null,
  "in_reply_to_screen_name":null,"user":{"id":557552289,"id_str":"557552289","name":"\u5c0f\u60aa\u9b54",
  "screen_name":"koakuma2011","location":"\u65e5\u672c \u5927\u962a","url":null,"description":"#pokemongo
#\u30dd\u30b1\u30e2\u30f3\u3076","protected":false,"verified":false,"followers_count":584,"friends_count":293,"listed_count":9,
  "favourites_count":0,"statuses_count":19624,"created_at":"Thu Apr 19 06:15:33 +0000 2012","utc_offset":null,
  "time_zone":null,"geo_enabled":false,"lang":"ja","contributors_enabled":false,"is_translator":false,
  "profile_background_color":"DBE9ED",
  "profile_background_image_url":"http://abs.twimg.com/images/themes/theme17/bg.gif",
  "profile_background_image_url_https":"https://abs.twimg.com/images/themes/theme17/bg.gif",
  "profile_banner_url":"https://pbs.twimg.com/profile_banners/557552289/1469414905","default_profile":false,
  "profile_link_color":"CC3366","profile_sidebar_border_color":"DBE9ED",
  "profile_sidebar_fill_color":"E6F6F9","profile_text_color":"333333","profile_use_background_image":true,
  "profile_image_url":"http://pbs.twimg.com/profile_images/2575455152/0zy0vvm32wduo1n7q5n4_normal.jpeg",
  "profile_image_url_https":"https://pbs.twimg.com/profile_images/2575455152/0zy0vvm32wduo1n7q5n4_normal.jpeg",
  "default_profile_image":false,"following":null,"follow_request_sent":null,"notifications":null},"geo":null,
  "coordinates":null,"place":null,"contributors":null,"is_quote_status":false,"retweet_count":0,"favorite_count":0,"entities":
  {"hashtags":[],"urls":[{"url":"https://t.co/7JW5aFQAK",
  "expanded_url":"http://game2019.blog.jp/archives/5018863.html","display_url":"game2019.blog.jp/archives/50188\u2026",
  "indices":[34,57]}],"user_mentions":[],"symbols":[]},"favorited":false,"retweeted":false,"possibly_sensitive":false,
  "filter_level":"low","lang":"ja","timestamp_ms":"149111649285"}
```

This picture shows the data for only one tweet including all the attributes: name, location, time posted, friend count, etc. These attributes become important when we need to define a data set. They will also help define a dependable class label that can be used to predict values.

Pandas

Pandas is a Python library needed to manipulate the data set which is not formatted. In this application, the data that was obtained was approximately 15,000 tweets with each tweet being generated and saved in a .txt format which is not readable by python or by an ordinary person.

Pandas framework is made to load the massive data set and to make sure that there is no inconsistencies. This data is later formatted by the JSON library which facilitates python to parse the data.

Matplotlib

Matplotlib is a python package that turns data into informative graphs and visualizations including scatter plots, bar graphs, and pie charts. This project utilized bar graphs to demonstrate the popularity of the three sports football, basketball, and soccer. The results were quite astonishing to our discovery.

Extracting the Data

To extract data from twitter, a developer needs to obtain access to the Twitter API which will enable the developer to obtain the needed information to get all the tweets. The following steps will show how to extract the data from Twitter.

1. Make a Twitter Account
2. Create a Twitter Application
3. Install Packages/Libraries
4. Data Extraction File
5. Plotting the Data

Step 1: Make a Twitter account

Join Twitter today.

<input type="text" value="ShoutMarks"/>	✓ Name looks great.
<input type="text" value="admin@shoutmarks.com"/>	✓ We will email you a confirmation.
<input type="password" value="....."/>	✓ Password is okay.
<input type="text" value="ShoutMarks"/>	✓ Username is available. You can change it later.
Suggestions: MarksShout · marks_shout · shout_marks	
<input checked="" type="checkbox"/> Keep me signed-in on this computer.	
<input checked="" type="checkbox"/> Tailor Twitter based on my recent website visits. Learn more.	
By clicking the button, you agree to the terms below: <small>These Terms of Service ("Terms") govern your access to and use of the services, including our various websites, SMS, APIs, email notifications,</small>	
Privacy Policy · Terms of Service	
<input type="button" value="Create my account"/>	



A Twitter account (Fig. 1) is needed to have access to the twitter API, so developers and users alike need to have this account for Twitter to authorize access.

Step 2: Make a Twitter Application

Once the account has been successfully made, the next step is to communicate with the Twitter's API. This is done by creating a Twitter Application. Follow the link below to do so: <https://apps.twitter.com/>.



Once the user has clicked on the link, they're brought to the Application Manager. Then they click on "Create New App." The "Create an Application" window (Fig. 5) is brought up where they're required to fill out all of the necessary information to register for a twitter application.

 Application Management 

Create an application

Application Details

Name *

Your application name. This is used to attribute the source of a tweet and in user-facing authorization screens. 32 characters max.

Description *

Your application description, which will be shown in user-facing authorization screens. Between 10 and 200 characters max.

Website *

Your application's publicly accessible home page, where users can go to download, make use of, or find out more information about your application. This fully-qualified URL is used in the source attribution for tweets created by your application and will be shown in user-facing authorization screens. (If you don't have a URL yet, just put a placeholder here but remember to change it later.)

Callback URL

Where should we return after successfully authenticating? [OAuth 1.0a](#) applications should explicitly specify their oauth_callback URL on the request token step, regardless of the value given here. To restrict your application from using callbacks, leave this field blank.

Developer Agreement

☐ Yes, I have read and agree to the [Twitter Developer Agreement](#).

Create your Twitter application

Once the information has been filled out, the user clicks “Create your Twitter application” button. Now the application has been created, it will give the following information that is needed to enable access to Twitter’s API.

BluJ Mine

[Test OAuth](#)[Details](#)[Settings](#)[Keys and Access Tokens](#)[Permissions](#)

Application Settings

Keep the "Consumer Secret" a secret. This key should never be human-readable in your application.

Consumer Key (API Key) HZ4nQYKP34QpvtXqBns5XLzrQ

Consumer Secret (API Secret) ChtsigD3vXsXDVbGjNFYLeRoTfNEbnmRAXennoX009mZsZFFN

Access Level Read, write, and direct messages ([modify app permissions](#))

Owner Heavy_linux_guy

Owner ID 838993675552239616

Application Actions

[Regenerate Consumer Key and Secret](#)[Change App Permissions](#)

Your Access Token

This access token can be used to make API requests on your own account's behalf. Do not share your access token secret with anyone.

Access Token 838993675552239616-
H0oj90kNPkcErxmLZ7E5Q4xQT0Up2kj

Access Token Secret 7tgeJC12BUzs6UyxpuEg8y9cWkegKJEcoWLg40NEnl1UH

Access Level Read, write, and direct messages

Owner Heavy_linux_guy

Owner ID 838993675552239616

Token Actions

[Regenerate My Access Token and Token Secret](#)[Revoke Token Access](#)

The consumer keys and access tokens are used to establish authentication to the Twitter API. “Your Access Token” (Fig. 7) contains the primary access token and the secret access token. The tokens are used to connect to the API on the user’s behalf and will define the privileges of the user. Meaning what the user can and cannot access. Every time the user wants access another user’s data, the secret access token is sent to the a server with the access token as a password. Since the tokens are associated with the user’s account, they should not be shared with anyone! The “Application Settings” (Fig. 6) will display the consumer key and the consumer secret key. The consumer key is associated with the application, in the case Twitter. This key is what identifies the application with the servers. The consumer secret key is the password for the application and used to authenticate with the authentication server. These keys are needed to make tweet extraction possible. For more information about the keys, visit <https://tools.ietf.org/html/rfc6749#section-1.4>.

Step 3: Install Packages /Libraries

The next step is connecting to the streaming Twitter API and downloading tweet information through a python program. The following code and tutorial is done in Python 2.7, this is important as syntax can change when using 3.0+ versions. Note to Mac and Linux users: Python is preinstalled on your OS (keep this in mind if you download a python IDE.)

Helpful Terminal Commands: ls, cd, wd, sudo, install, easy_install, source

*The commands: install, easy_install, sudo install can be used interchangeably.

****Warning!** The ‘sudo’ command will install as a root user, and can potentially install packages where it shouldn’t. Sudo should be used with care.

- 1) To install Tweepy, 'pip' ("Pip Installs Packages" or "Pip Installs Python") must be installed first. After installing the 'pip' package, other python packages can be installed more easily.

Alternative Commands: `install pip`, `easy_install pip`, `sudo easy_install pip`, `curl https://bootstrap.pypa.io/get-pip.py|python`

- 2) Installing Tweepy will install the latest version, and contains all of the need components to mine the tweets from Twitter. To install, simply input the following command into the python terminal(Windows) or the native terminal(Mac and Linux):

```
pip install tweepy
```

- 3) The following libraries are also needed to complete this tutorial:

Pandas: `pip install pandas`

Matplotlib: `pip install matplotlib`

Step 4: Data Extraction File

Now it's time to create a new python file that will extract the tweets we want. Creating a new python file can vary and depends on the environment you are using.

The code below can be download at: <https://github.com/duardo56/BluJMine.git>

```

1  from tweepy import Stream
2  from tweepy import OAuthHandler
3  from tweepy.streaming import StreamListener
4  import time
5
6  ckey = 'Enter Your API Key'
7  csecret = 'Enter Your API Secret'
8  atoken = 'Enter Your Access Token'
9  asecret = 'Enter Your Access Toekn Secret'
10
11 #This is a basic listener that just prints received tweets to stdout.
12 class StdOutListener(StreamListener):
13
14     def on_data(self, data):
15
16         try:
17             print (data)
18             saveFile = open('ENTER-NAME-OF-YOUR-TEXT-FILE.txt', 'a')
19             saveFile.write(data)
20             saveFile.write('\n')
21             saveFile.close()
22             return True
23         except BaseException:
24             time.sleep(5)
25
26     def on_error(self, status):
27         print (status)
28
29 if __name__ == '__main__':
30
31     #This handles Twitter authentication and the connection to Twitter Streaming API
32     auth = OAuthHandler(ckey, csecret)
33     auth.set_access_token(atoken, asecret)
34     twitterStream = Stream(auth, StdOutListener())
35
36     #This line filter Twitter Streams to capture data by the keywords: 'pokemonGO', 'gameofThornes'
37     twitterStream.filter(track=["pokemonGo", "gameofThrones"])

```

After the code is downloaded and saved into a file, it's time to change the following (Fig. 6):

- 1.) Input the access tokens and consumer keys from the Twitter Application you obtained in lines 6 through 9.
- 2.) Enter the name of the text file where the streaming twitter data will be saved, line 18.

3.) Input the keywords (line 37) that will be used to filter through the twitter stream and capture the tweets related to these keywords here.

After the above changes are made, it's time to run the program. The following command can be type-in into the terminal: `python twitter_data.py`

When the code is executed it will use the access token, access secret token, consumer key, and the consumer secret key to connect and gain authorization from the Twitter API.

Afterwards, the keywords will be used to find streaming tweets with the words "football", "basketball", and "soccer". Once found, the tweets will be written into a text file. Since we are finding streaming tweets, the program needs to stay connect to run. The tweets will be continuously written into the text file as the tweets are tweeted in order to be later processed through the pandas framework. The tweets are stored in JSON format, making it easier to extract data from it. Ending the program will stop gathering tweets, which can be done by entering: Control-C

Step 5: Plotting the Data

In this step, we will extract predefined attributes from the data we gathered and plot it. First, we will need to make a new python file, and the code for it is can be downloaded at:

<https://github.com/duardo56/BluJMine.git>. Once done, we can begin reading the tweets in the text file created in step 4. The data extracted in that text file is in the form of a JSON file. We will need to parse the data into a readable form.

```

# Reading Tweets
print('Reading Tweets\n')
# File Name be sure to change!!!!
tweets_data_path = 'twitDB.txt'

tweets_data = []
tweets_file = open(tweets_data_path, "r")
for line in tweets_file:
    try:
        tweet = json.loads(line)
        tweets_data.append(tweet)
    except:
        continue

```

To do so, an array is created “tweets_data” to hold each line from the text file. While reading, it restructures the tweet into a readable form.

Once the file has been reformatted, it's time to read the attributes. The attributes can be found at <https://dev.twitter.com/overview/api/tweets>. Create a pandas data frame to hold the selected attributes. Below is a sample of some attributes that were collected.

```

# Structuring Tweets
print('Structuring Tweets\n')

tweets = pd.DataFrame()
tweets['id'] = list(map(lambda tweet: tweet.get('id', None), tweets_data))
tweets['country'] = list(map(lambda tweet: tweet['place']['country'] if tweet['place'] != None else None, tweets_data))
tweets['full_name'] = list(map(lambda tweet: tweet['place']['full_name'] if tweet['place'] != None else None, tweets_data))
tweets['lang'] = list(map(lambda tweet: tweet['lang'], tweets_data))
tweets['favorite_count'] = list(map(lambda tweet: tweet.get('favorite_count', None), tweets_data))
tweets['favourites_count'] = list(
    map(lambda tweet: tweet['user']['favourites_count'] if tweet['user'] != None else None, tweets_data))
tweets['followers_count'] = list(
    map(lambda tweet: tweet['user']['followers_count'] if tweet['user'] != None else None, tweets_data))
tweets['friends_count'] = list(
    map(lambda tweet: tweet['user']['friends_count'] if tweet['user'] != None else None, tweets_data))
tweets['statuses_count'] = list(
    map(lambda tweet: tweet['user']['statuses_count'] if tweet['user'] != None else None, tweets_data))
tweets['hashtags'] = list(
    map(lambda tweet: tweet['entities']['hashtags'] if tweet['entities'] != None else None, tweets_data))
tweets['entities'] = list(map(lambda tweet: tweet.get('entities', None), tweets_data))
tweets['retweet_count'] = list(map(lambda tweet: tweet.get('retweet_count', None), tweets_data))
tweets['text'] = list(map(lambda tweet: tweet.get('text', None), tweets_data))

```

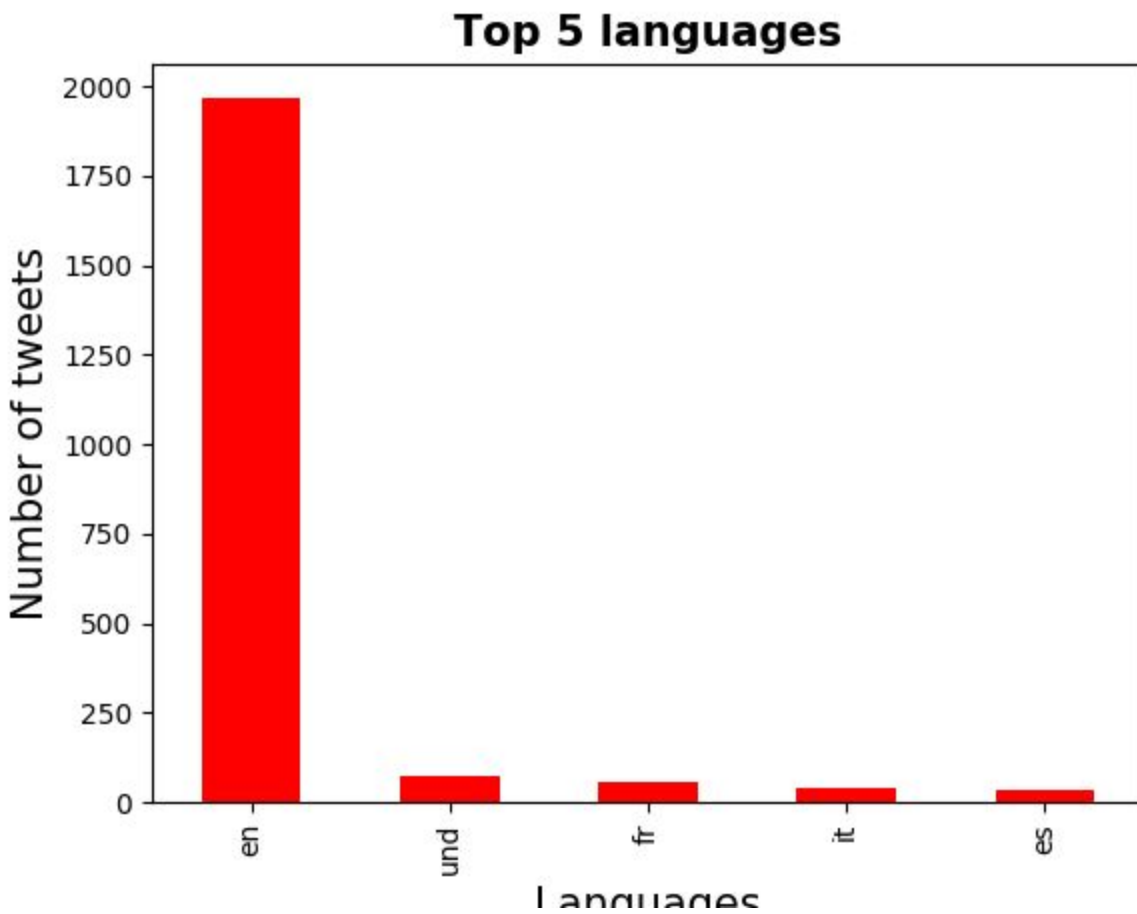

Although not required, writing the attributes to an excel file is a great way to organize and analyze the data gathered. The code below will also sort the columns by attribute. The following packages will need to be installed to write to an excel file:

```
pip install xlwt
```

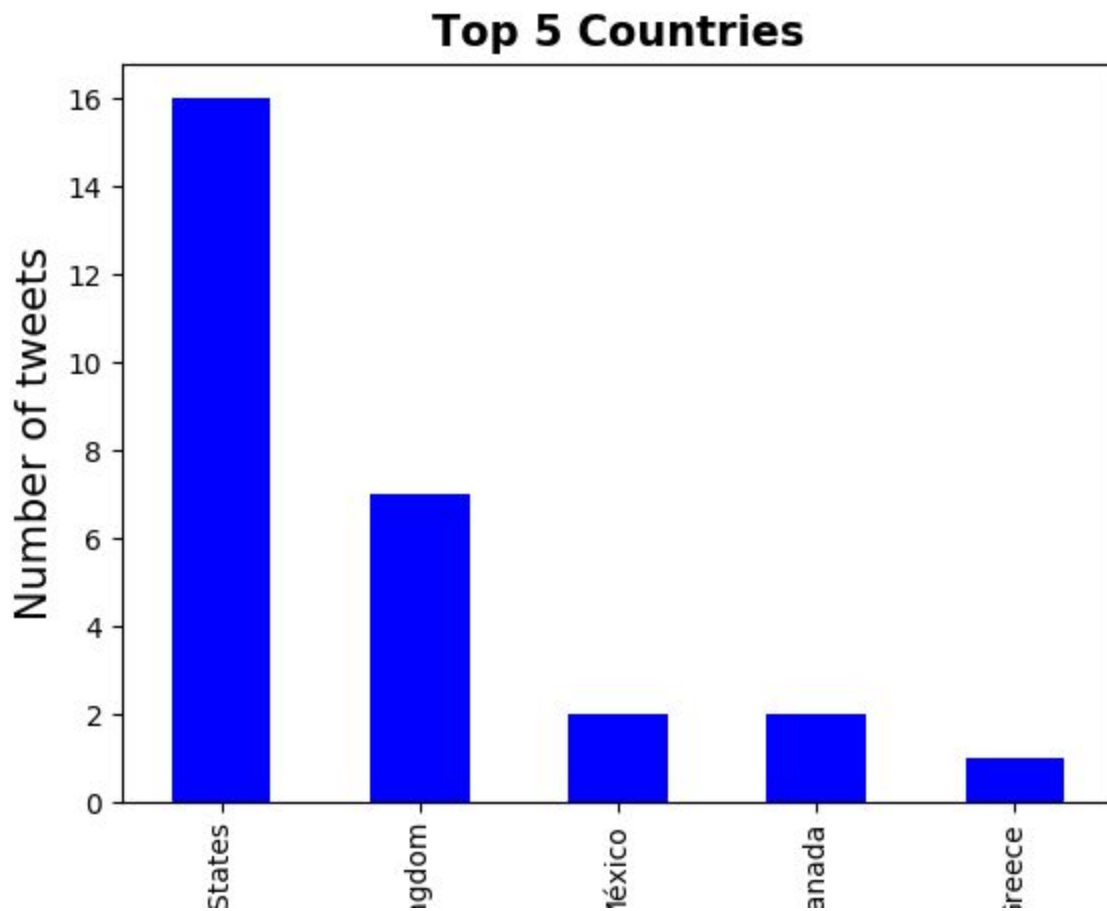
```
pip install xlswriter
```

```
# Bonus!!! Write to Excel
# Create a Pandas Excel writer using XlsxWriter as the engine.
writer = pd.ExcelWriter('twitData.xlsx', engine='xlsxwriter')
# Convert the dataframe to an XlsxWriter Excel object.
tweets.to_excel(writer, sheet_name='Sheet1')
# Close the Pandas Excel writer and output the Excel file.
writer.save() # tweets_by_languages = tweets['lang'].value_counts()
```

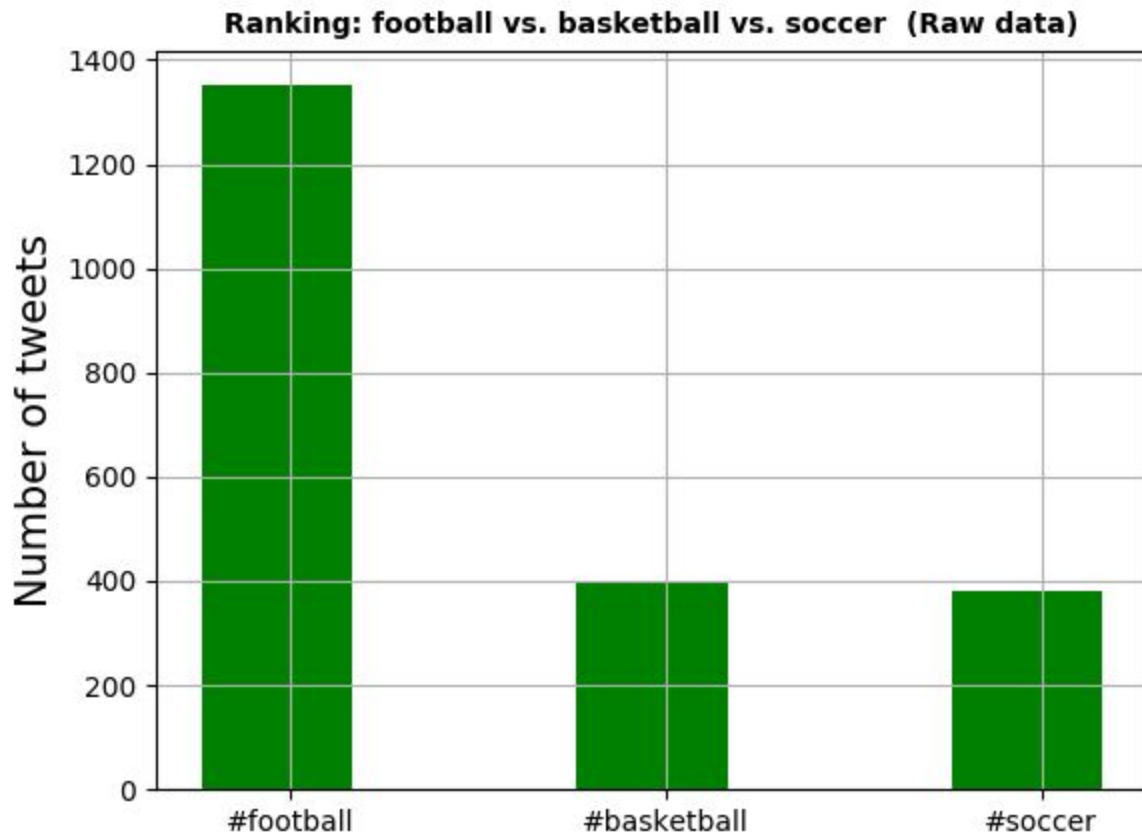
Next, the package matplotlib is needed to make sure the data is being graphed correctly and that the selected attributes of the tweets can be graphed properly. With the given data that was retrieved from the three twitter tags the first figure shows all the tweets organized by country and the second figure shows the tweets by language.



From all the mined tweets, these are the top 5 countries that showed up the most in all the tweets obtained. The dominating country in this situation is the United States followed by Canada, South Africa, United Kingdom and Spain. This graph was achieved by using the Matplotlib library, which was downloaded and later imported to generate the graphs. The various axis was specified and labeled in the line of codes. Attributes such as color of the bars, font size, and axis label were specified in the respective lines of code in order to generate a clean and concise graph of our derived results.



In the graph above (Fig. 10) shows the top 5 languages of all the tweets mined from hashtags “football”, “basketball”, and “soccer”. Tweets contain an attribute that define the language of each tweet. From figure 9, it seems that the maximum number of tweets were generated from the United States, however the maximum tweets that were generated were in the Japanese language, followed by English, Spanish, Unified Mandarin and last but not the least French amongst the many others.



A python file by the name of tweet_mining.py was created to mine the useful data that was extracted from twitter. Data was parsed using json. A new library was introduced called “re” which stands for “regular expression”. The essence of this library is to take keywords out such as “football”, “basketball”, and “soccer”. The tweets are then formulated accordingly and a new graph is generated, this time with the motives of comparing the popularity of the three sports. The popularity of soccer has been assumed to be far more superior than football. But we can see from the resulting data that the tables have turned and football has far surpassed the popularity of soccer.

Conclusion

There are many things that can be taken away from this experiment including how data is extracted and formatted in a variety of ways for a range of predictions. This is one of many aspects of data mining that was used to compare two random but relevant tweets to compare the popularity of three sports that are ruling the sports. Python has aided developers with a multitude of libraries that serve different purposes. For this project, the intentions were to demonstrate the locations and countries where the tweets were being generated from, what languages the tweets were being posted in, and ultimately the comparison of the popularity of the three sports.

Sources

"An Introduction to Text Mining Using Twitter Streaming API and Python." *An Introduction to Text Mining Using Twitter Streaming API and Python* // Adil Moujahid // *Data Analytics and More*. N.p., n.d. Web. 06 Apr. 2017.

Bonzanini, Marco. "Chapter 2. #MiningTwitter – Hashtags, Topics, and Time Series." *Mastering Social Media Mining with Python: Acquire and Analyze Data from All Corners of the Social Web with Python*. Birmingham: Packt, 2016. N. pag. Print.

<https://tools.ietf.org/html/rfc6749#section-1.4>

<http://tweepy.readthedocs.io/en/v3.5.0/>