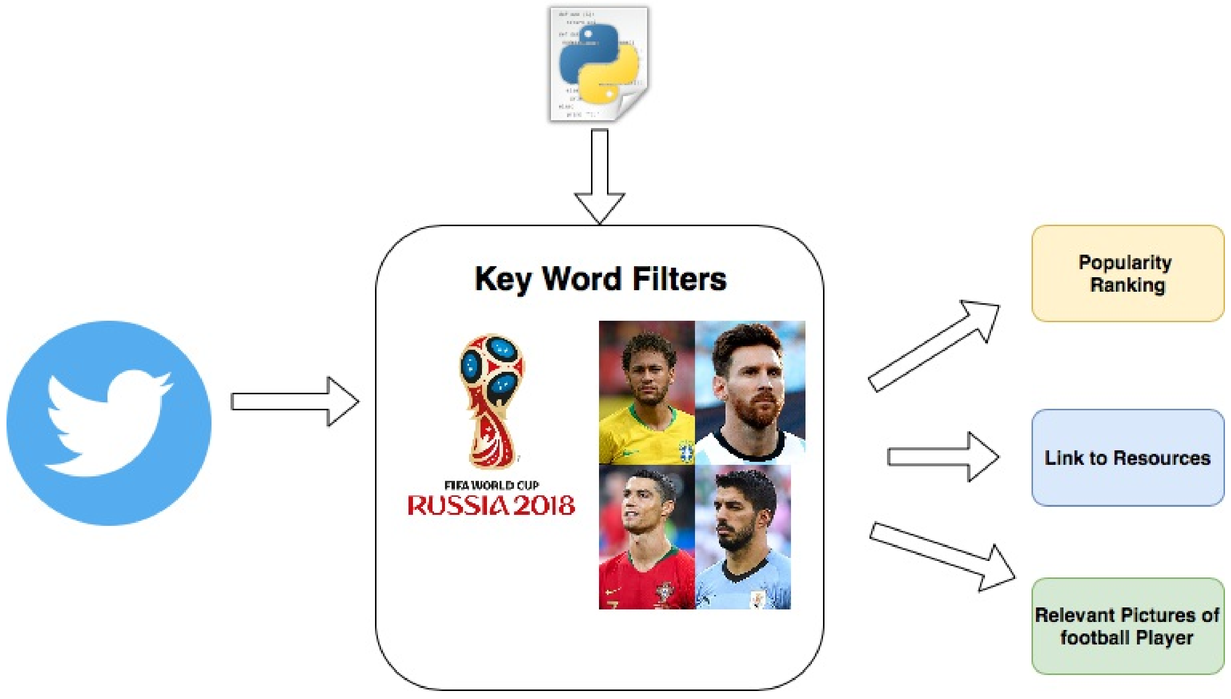
Text Mining and Analysis Using Twitter Streaming API


Key Word Filters
FIFA WORLD CUP RUSSIA 2018
Popularity Ranking
Link to Resources
Relevant Pictures of football Player

Introduction

Text mining is the application of natural language processing techniques and analytical methods to text data in order to derive relevant information. in this following works, It shows how to collecting data from Twitter with Twitter Streaming API that allow us to capture tweets real-time filter.

In this study case I will use #WorldCup data to compare the popularity of 4 most popularity during Fifa World Cup Russia 2018 and most football player: Cristiano Ronaldo, Neymar,Lionel Messi and Luis Suarez, and to retrieve links to the news resoruces such as tweet,website,video,youtube etc.. In the first Part, I will explaing how to connect to Twitter Streaming API and how to get the data. In the second Part, I will explain and show how to structure the data for analysis, and in the last paragraph, And Finally I will explain how to filter the data and extract links from tweets.

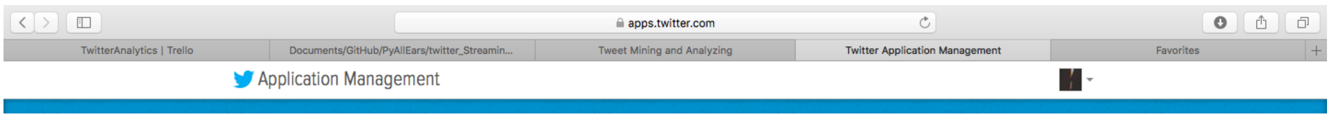
-- Part 1:Getting Start with Twitter API --

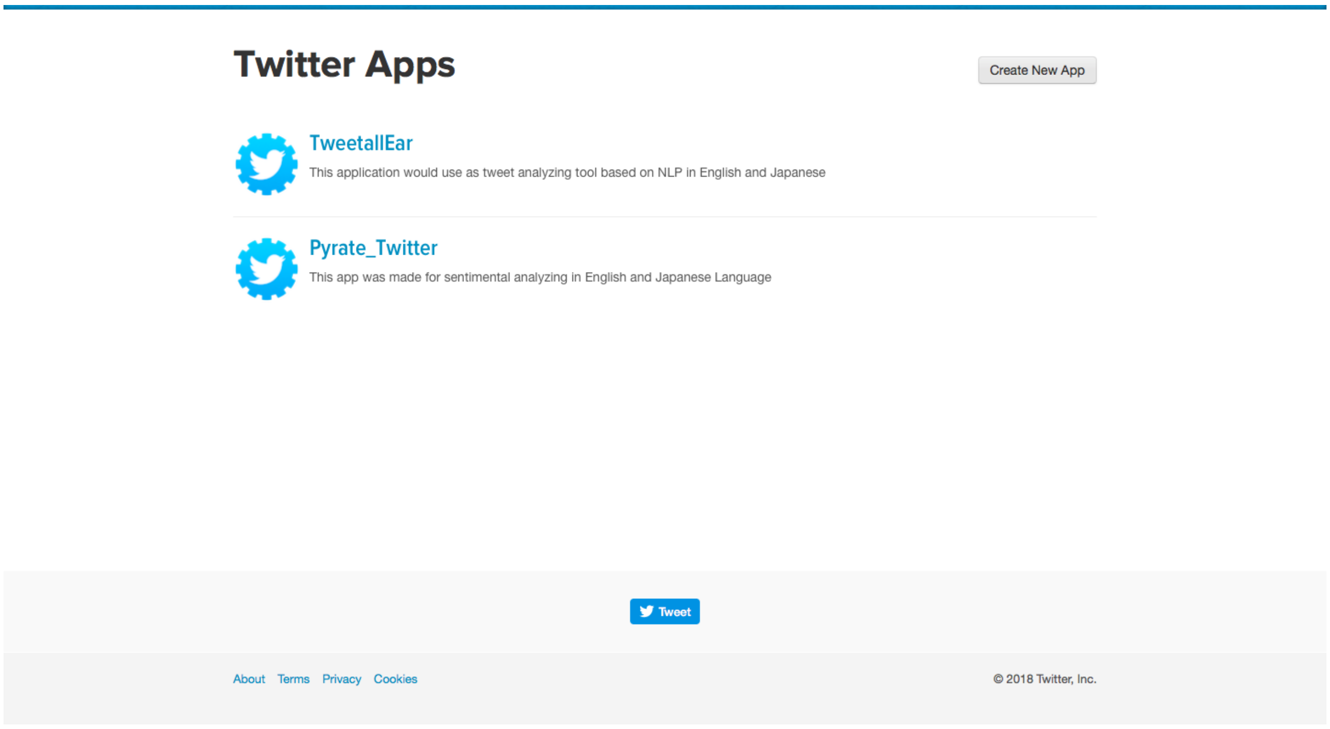
Understanding of Twitter API

API stands for Application Programming Interface. It is a tool that makes the interaction with computer programs and web services easy

Getting API

- Create a twitter account if you do not already have one.
- Go to https://apps.twitter.com/ and log in with your twitter credentials.
- Click "Create New App"
- Fill out the form, agree to the terms, and click "Create your Twitter application"
- In the next page, click on "API keys" tab, and copy your "API key" and "API secret".
- Scroll down and click "Create my access token", and copy your "Access token" and "Access token secret".


apps.twitter.com
TwitterAnalytics | Trello
Documents/GitHub/PyAllEars/twitter_Streamin...
Tweet Mining and Analyzing
Twitter Application Management
Favorites
Application Management

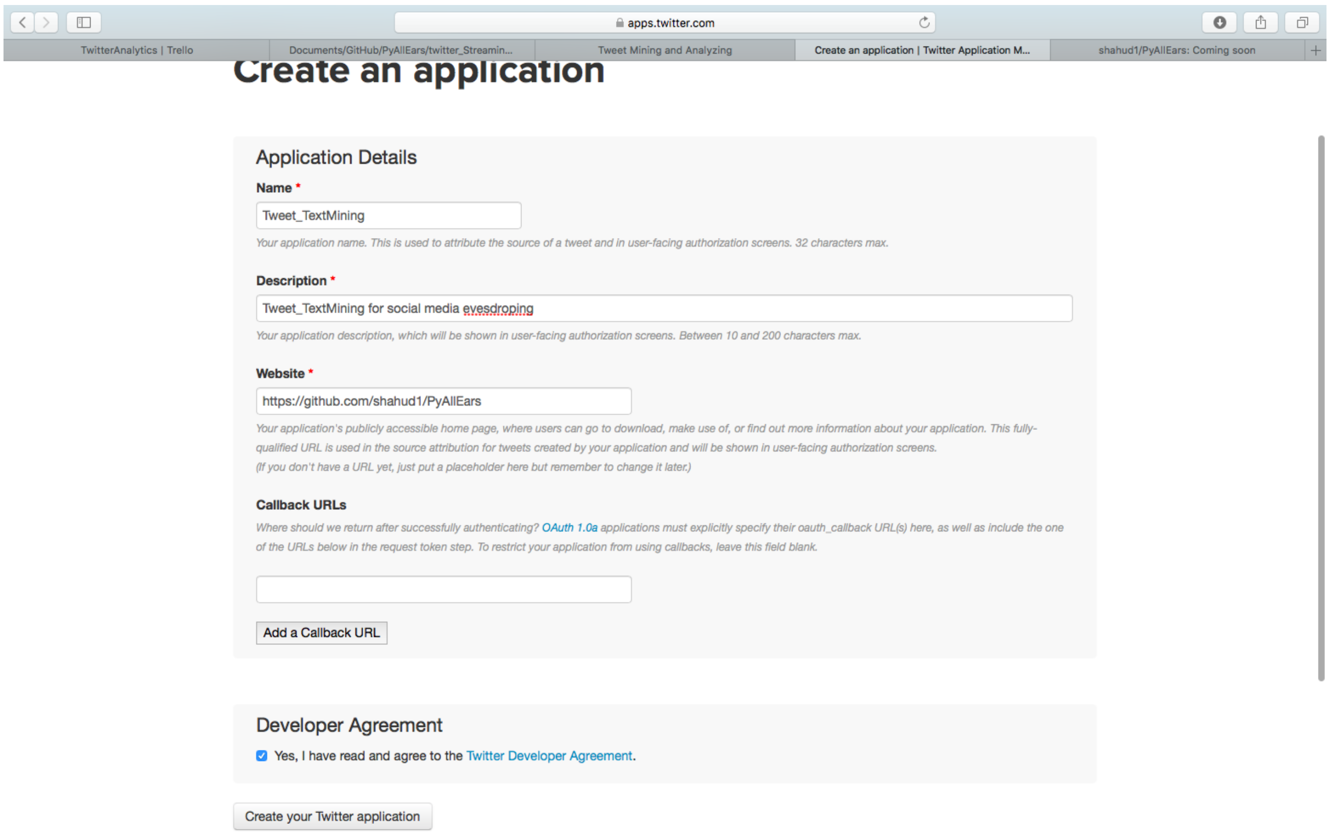


Twitter Apps

Create New App

TweetallEar
This application would use as tweet analyzing tool based on NLP in English and Japanese

Pyrate_Twitter
This app was made for sentimental analyzing in English and Japanese Language



apps.twitter.com
TwitterAnalytics | Trello
Documents/GitHub/PyAllEars/twitter_Streamin...
Tweet Mining and Analyzing
Create an application | Twitter Application M...
shahud1/PyAllEars: Coming soon

Create an application

Application Details

Name *

Tweet_TextMining

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

Description *

Tweet_TextMining for social media evesdroping

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

Website *

https://github.com/shahud1/PyAllEars

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 URLs

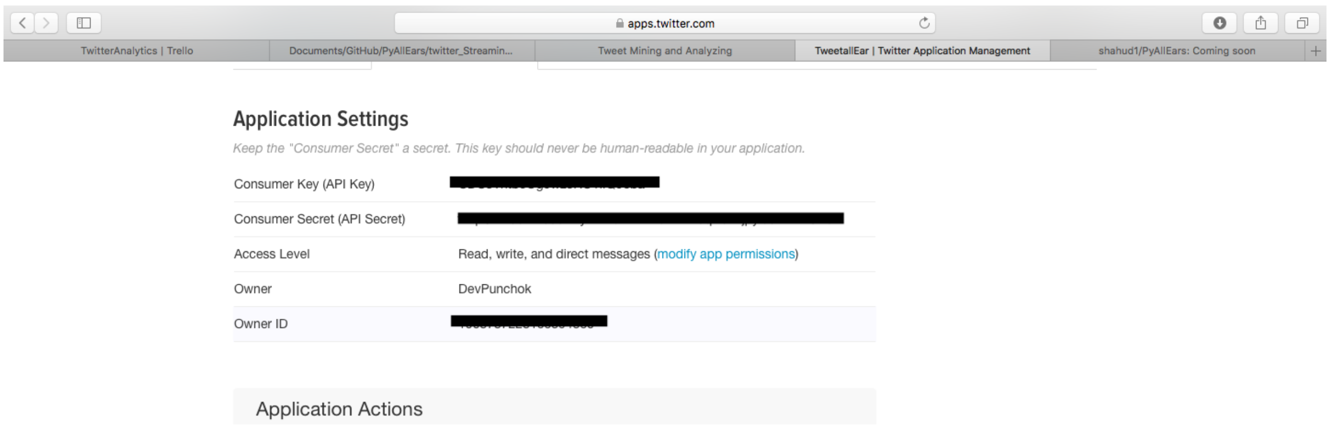
Where should we return after successfully authenticating? OAuth 1.0a applications must explicitly specify their oauth_callback URL(s) here, as well as include the one of the URLs below in the request token step. To restrict your application from using callbacks, leave this field blank.

Add a Callback URL

Developer Agreement

☑ Yes, I have read and agree to the Twitter Developer Agreement.

Create your Twitter application



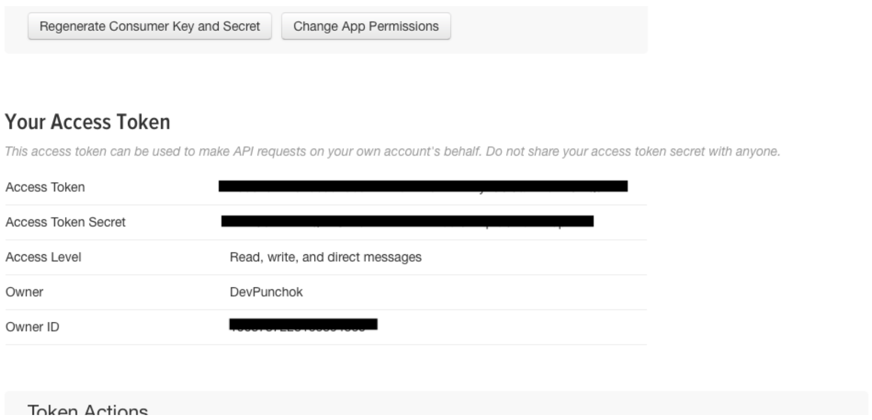
apps.twitter.com
TwitterAnalytics | Trello
Documents/GitHub/PyAllEars/twitter_Streamin...
Tweet Mining and Analyzing
TweetallEar | Twitter Application Management
shahud1/PyAllEars: Coming soon

Application Settings

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

Consumer Key (API Key)	[redacted]
Consumer Secret (API Secret)	[redacted]
Access Level	Read, write, and direct messages (modify app permissions)
Owner	DevPunchok
Owner ID	[redacted]

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.

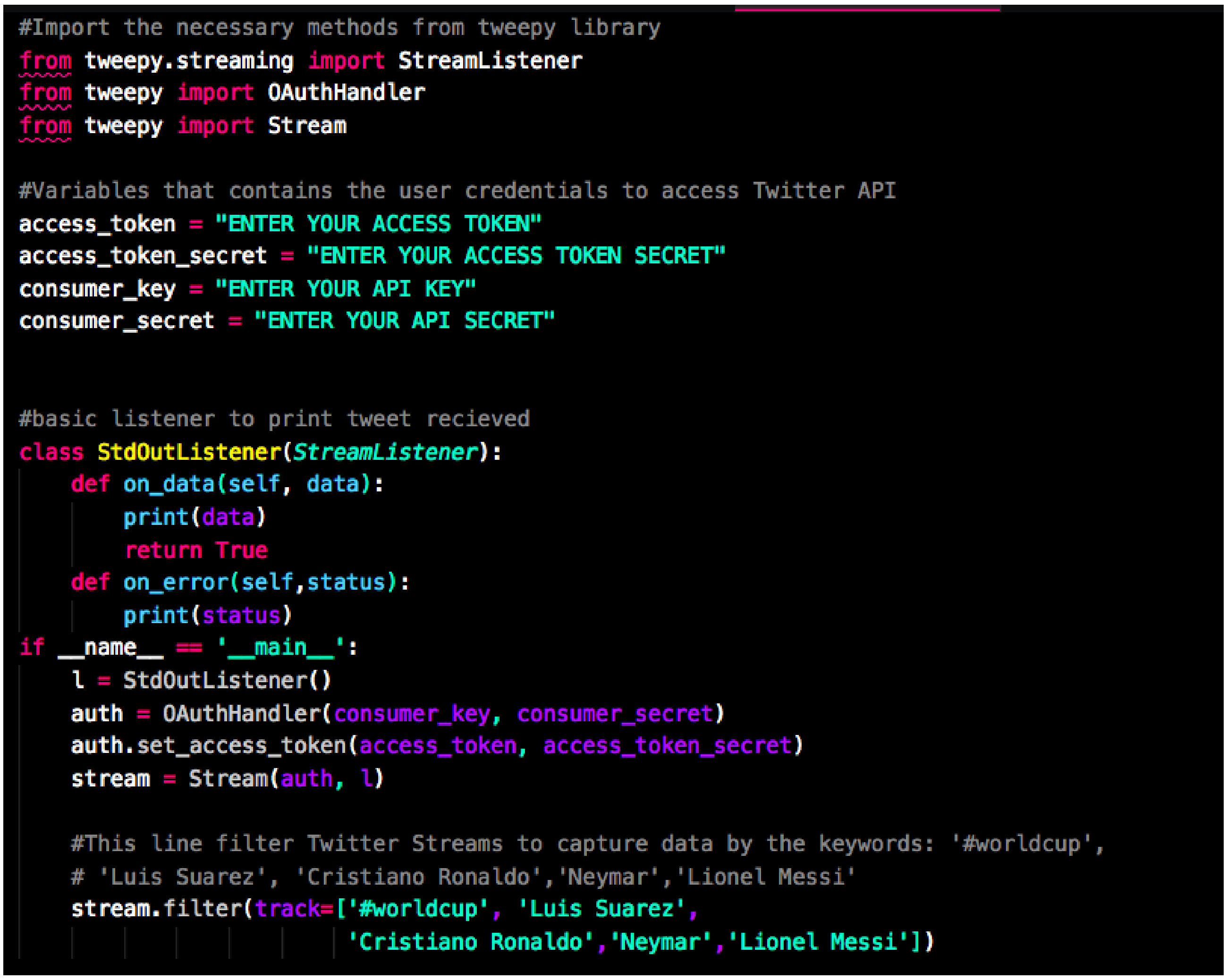
Access Token	[redacted]
Access Token Secret	[redacted]
Access Level	Read, write, and direct messages
Owner	DevPunchok
Owner ID	[redacted]

Token Actions

Create Twitter streaming API file to shows the result of realtime filter streaming

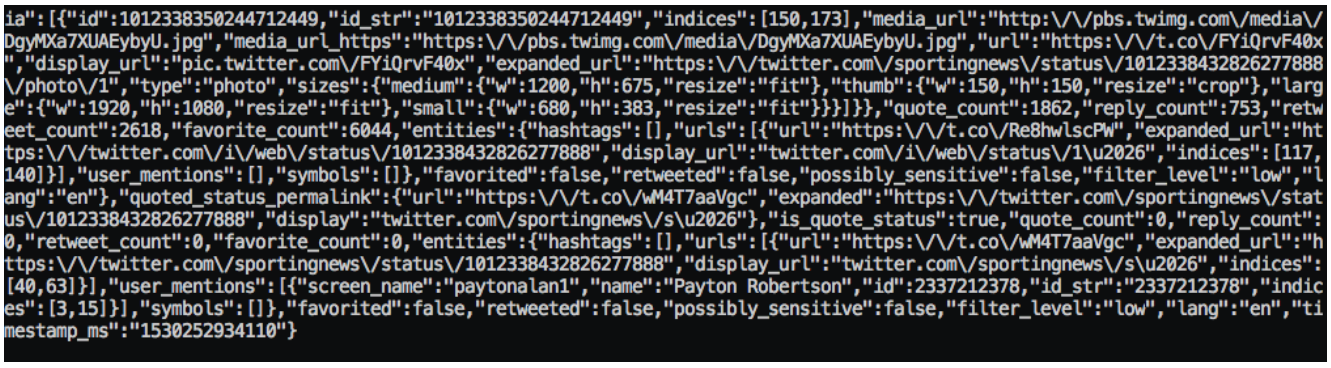
Create a file name Twitter_stream_api.py Using Tweepy library. We will be using a Python library called Tweepy to connect to Twitter Streaming API and downloading the data. If you don't have Tweepy installed in your machine, go to this link, and follow the installation instructions.

Next create, a file called twitter_streaming.py, and copy into it the code below. Make sure to enter your credentials into access_token, access_token_secret, consumer_key, and consumer_secret.


```
#Import the necessary methods from tweepy library
from tweepy.streaming import StreamListener
from tweepy import OAuthHandler
from tweepy import Stream

#Variables that contains the user credentials to access Twitter API
access_token = "ENTER YOUR ACCESS TOKEN"
access_token_secret = "ENTER YOUR ACCESS TOKEN SECRET"
consumer_key = "ENTER YOUR API KEY"
consumer_secret = "ENTER YOUR API SECRET"


#basic listener to print tweet recieved
class StdOutListener(StreamListener):
    def on_data(self, data):
        print(data)
        return True
    def on_error(self,status):
        print(status)
if __name__ == '__main__':
    l = StdOutListener()
    auth = OAuthHandler(consumer_key, consumer_secret)
    auth.set_access_token(access_token, access_token_secret)
    stream = Stream(auth, l)

    #This line filter Twitter Streams to capture data by the keywords: '#worldcup',
    # 'Luis Suarez', 'Cristiano Ronaldo','Neymar','Lionel Messi'
    stream.filter(track=['#worldcup', 'Luis Suarez',
                         'Cristiano Ronaldo','Neymar','Lionel Messi'])
```

There are the outputs when execute the instruction from above

the output returns the value in JSON from which contain more than 100 keys in 1 tweet, I've been streaming for 2 hours to collect data form Twitter

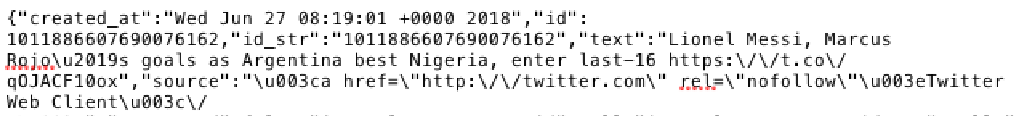


```
ia":[{"id":1012338350244712449,"id_str":"1012338350244712449","indices":[150,173],"media_url":"http:\/\/pbs.twimg.com\/media\/
DgyMXa7XUAEybyU.jpg","media_url_https":"https:\/\/pbs.twimg.com\/media\/DgyMXa7XUAEybyU.jpg","url":"https:\/\/t.co\/FYiQrvF40x
","display_url":"pic.twitter.com\/FYiQrvF40x","expanded_url":"https:\/\/twitter.com\/sportingnews\/status\/1012338432826277888
\/photo\/1","type":"photo","sizes":{"medium":{"w":1200,"h":675,"resize":"fit"},"thumb":{"w":150,"h":150,"resize":"crop"},"larg
e":{"w":1920,"h":1080,"resize":"fit"},"small":{"w":680,"h":383,"resize":"fit"}}}]}},"quote_count":1862,"reply_count":753,"retw
eet_count":2618,"favorite_count":6044,"entities":{"hashtags":[],"urls":[{"url":"https:\/\/t.co\/Re8hwlscPW","expanded_url":"ht
tps:\/\/twitter.com\/i\/web\/status\/1012338432826277888","display_url":"twitter.com\/i\/web\/status\/1\u2026","indices":[117,
140]}],"user_mentions":[],"symbols":[]},"favorited":false,"retweeted":false,"possibly_sensitive":false,"filter_level":"low","l
ang":"en"},"quoted_status_permalink":{"url":"https:\/\/t.co\/wM4T7aaVgc","expanded":"https:\/\/twitter.com\/sportingnews\/stat
us\/1012338432826277888","display":"twitter.com\/sportingnews\/s\u2026"},"is_quote_status":true,"quote_count":0,"reply_count":
0,"retweet_count":0,"favorite_count":0,"entities":{"hashtags":[],"urls":[{"url":"https:\/\/t.co\/wM4T7aaVgc","expanded_url":"h
ttps:\/\/twitter.com\/sportingnews\/status\/1012338432826277888","display_url":"twitter.com\/sportingnews\/s\u2026","indices":
[40,63]}],"user_mentions":[{"screen_name":"paytonalan1","name":"Payton Robertson","id":2337212378,"id_str":"2337212378","indic
es":[3,15]}],"symbols":[]},"favorited":false,"retweeted":false,"possibly_sensitive":false,"filter_level":"low","lang":"en","ti
mestamp_ms":"1530252934110"}
```

Capturing and Reading the Data

In order to capture the data for the analysis. I collect by following command to store data in txt file

python3 Twitter_stream_api.py > data/worldcup2018_twitter_data.txt



```
{"created_at":"Wed Jun 27 08:19:01 +0000 2018","id":
1011886607690076162,"id_str":"1011886607690076162","text":"Lionel Messi, Marcus
Rojo\u2019s goals as Argentina best Nigeria, enter last-16 https:\/\/t.co\/
qOJACF10ox","source":"\u003ca href=\"http:\/\/twitter.com\" rel=\"nofollow\"\u003eTwitter
Web Client\u003c\/
```

The data that we retrived is store in worldcup2018_twitter_data.txt which are JSON forfrt you can see that the tweet contain additional and more information example :

"text":"Lionel Messi, Marcus Rojo\u2019s goals as Argentina best Nigeria

---

-- Part 2: Structured data and analysis --

import necessary library which contain

- Json
- pandas
- matplotlib
- re
- matplotlib

In [1]:

```
%matplotlib inline
import re
import json
import pandas as pd
import matplotlib.pyplot as plt
```

Read captured data from Txt File

In [2]:

```
#path to tweets collected data
tweets_data_path = 'data/worldcup2018_twitter_data.txt'
#open the file
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
```

Show the total captured tweet data

using print and len ( ) function to read all count tweet data that has captured

In [3]:

```
print(len(tweets_data))
```

```
6008
```

In This data/worldcup2018_twitter_data.txt we've capture totally **6008** tweets from twitter

In [4]:

```
tweets = pd.DataFrame()
```

Mapping capture tweet from JSON format fileform text file into data frame

In [5]:

```
tweets['text'] = list(map(lambda tweet: tweet['text'], tweets_data))

tweets['lang'] = list(map(lambda tweet: tweet['lang'], tweets_data))

tweets['country'] = list(map(lambda tweet: tweet['place']['country'] if tweet['place'] != None else
None, tweets_data))
```

In [6]:

```
tweets_by_lang = tweets['lang'].value_counts()
```

In [7]:

```
tweets_by_lang.head()
```

Out[7]:

```
en    3894
pt     402
es     335
fr     270
ja     221
Name: lang, dtype: int64
```

this shows top 5 language that has been tweet the most tweets are in English(en) and second in Protugal(pt) and third in Spanish(es),French(fr) and Japanese(jp)

Drawing the Graph

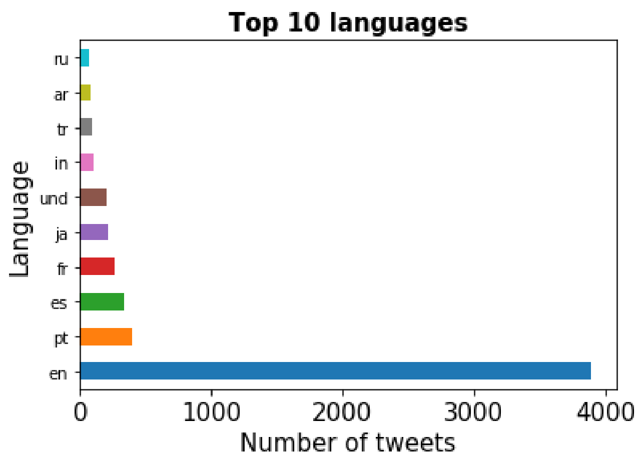
In order to impliment the graph we use Mathplotlib library to draw the grph which has many kind of graph.In a simple implimentation I use bar graph for showing counting result from above and finding top 10 Languages from 60008 tweets that's captured

In [8]:

```
fig, ax = plt.subplots()
ax.tick_params(axis='x', labelsize=15)
ax.tick_params(axis='y', labelsize=10)
ax.set_xlabel('Number of tweets', fontsize=15)
ax.set_ylabel('Language' , fontsize=15)
ax.set_title('Top 10 languages', fontsize=15, fontweight='bold')
tweets_by_lang[:10].plot(ax=ax, kind='barh')
```

Out[8]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x123ddbc18>
```



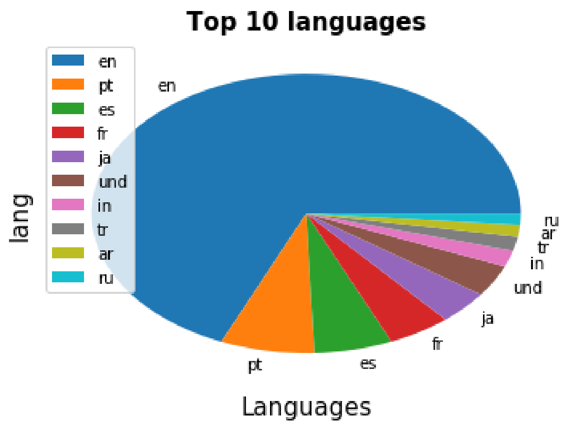
Showing different result in different kind of graph in **Pie Graph**

In [9]:

```
tweets_by_lang = tweets['lang'].value_counts()

fig, ax = plt.subplots()
ax.tick_params(axis='x', labelsize=15)
ax.tick_params(axis='y', labelsize=10)
ax.set_xlabel('Languages', fontsize=15)
ax.set_ylabel('Number of tweets' , fontsize=15)
ax.set_title('Top 10 languages', fontsize=15, fontweight='bold')
tweets_by_lang[:10].plot(ax=ax, kind='pie',legend=True)
```

Out[9]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x124938cc0>
```



Drawing a Graph for 10 countries that tweet about '#WorldCup2'

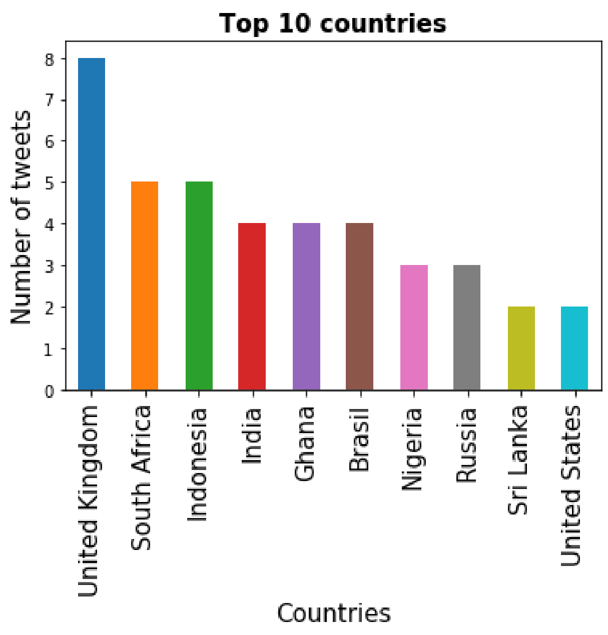
In [10]:

```
tweets_by_country = tweets['country'].value_counts()

fig, ax = plt.subplots()
ax.tick_params(axis='x', labelsize=15)
ax.tick_params(axis='y', labelsize=10)
ax.set_xlabel('Countries', fontsize=15)
ax.set_ylabel('Number of tweets' , fontsize=15)
ax.set_title('Top 10 countries', fontsize=15, fontweight='bold')
```

```
tweets_by_country[:10].plot(ax=ax, kind='bar')
```

Out[10]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x124a3f2e8>
```



-- Part 3:Text Mining and Extracting Link --

Our main goals in these text mining tasks are: compare the popularity of Cristiano Ronaldo, Luis Suarez Neymar programming languages and to retrieve programming tutorial links. We will do this in 3 steps:

- We will add tags to our tweets DataFrame in order to be able to manipualte the data easily.
- Target tweets that have "WorldCup" or "Fifa" keywords.
- Extract links from the relevants tweets

Defind the function to convert all text that contain Capital and mixing text to lower case also using search function to find word in column text

In [11]:

```
def word_in_text(word,text):
    word = word.lower()
    text = text.lower()
    match = re.search(word,text)
    if match :
        return True
    return False
```

In [12]:

```
tweets['Cristiano Ronaldo'] =list(tweets['text'].apply(lambda tweet: word_in_text('Cristiano Ronald
o', tweet)))
tweets['Luis Suarez'] =list(tweets['text'].apply(lambda tweet: word_in_text('Luis Suarez', tweet)))
tweets['Neymar'] =list(tweets['text'].apply(lambda tweet: word_in_text('Neymar', tweet)))
tweets['Lionel Messi'] =list(tweets['text'].apply(lambda tweet: word_in_text('Lionel Messi', tweet)
))
tweets['#WorldCup'] =list(tweets['text'].apply(lambda tweet: word_in_text('#WorldCup', tweet)))
```

In [13]:

```
#print all count False means not appear in Tweets and True means appear in Tweets
print(tweets['Cristiano Ronaldo'].value_counts())
print(tweets['Luis Suarez'].value_counts())
print(tweets['Neymar'].value_counts())
print(tweets['Lionel Messi'].value_counts())
print(tweets['#WorldCup'].value_counts())
```

```
False    5756
True      252
Name: Cristiano Ronaldo, dtype: int64
False    6000
True        8
Name: Luis Suarez, dtype: int64
False    5634
True      374
Name: Neymar, dtype: int64
False    5340
True      668
Name: Lionel Messi, dtype: int64
False    3569
True     2439
Name: #WorldCup, dtype: int64
```

In [14]:

```
CR = tweets['Cristiano Ronaldo'].value_counts()[True]
LS = tweets['Luis Suarez'].value_counts()[True]
NM = tweets['Neymar'].value_counts()[True]
LM = tweets['Lionel Messi'].value_counts()[True]
```

In [15]:

```
print("total tweets of Cristiano Ronaldo are \n",CR,"tweets") #print total count of Cristiano
Ronaldo
print("total tweets of Luis Suarezare \n",LS,"tweets"),#print total count of Luis Suarez
print("total tweets of Neymar \n",NM,"tweets")#print total count of Neymar
print("total tweets of Neymar Lionel Messi \n",LM,"tweets")#print total count of Lionel Messi
```

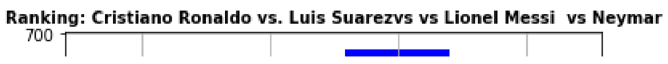
```
total tweets of Cristiano Ronaldo are
 252 tweets
total tweets of Luis Suarezare
 8 tweets
total tweets of Neymar
 374 tweets
total tweets of Neymar Lionel Messi
 668 tweets
```

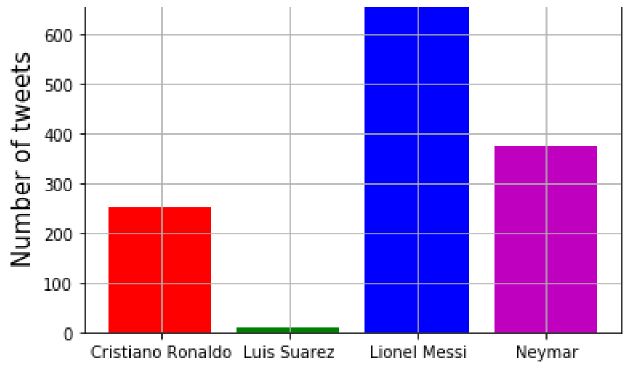
Ranking

to show all ranking of popularity on football player from above

In [16]:

```
Fb_player = ['Cristiano Ronaldo', 'Luis Suarez', 'Lionel Messi','Neymar']
tweets_by_Fb_player = [CR,LS,LM,NM]
x_pos = list(range(len(Fb_player)))
width = 0.8
fig, ax = plt.subplots()
#ploting graph of tweets_by_Fb_player
pack_color = 'rgbm'
plt.bar(x_pos, tweets_by_Fb_player, width, alpha=1,color=pack_color)
# Setting axis labels and ticks
ax.set_ylabel('Number of tweets', fontsize=15)
ax.set_title('Ranking: Cristiano Ronaldo vs. Luis Suarezvs vs Lionel Messi  vs Neymar', fontsize=1
0, fontweight='bold')
ax.set_xticks([p + 0.01 * width for p in x_pos])
ax.set_xticklabels(Fb_player)
plt.grid()
```





Specifying Relevant Tweet text

In this part I'll try to specifying the keyword in order to match football players who were mention during the WorldCup 2018 with keywords 'FiFa2018' or'World Cup'

Mapping keywords of Fifa2018 and Worldcup that appear in text relevant that take value True if the tweet has either "programming" or "tutorial" keyword, otherwise it takes value False.

In [17]:

```
#Mapping keywords of Fifa2018 and Worldcup that appear in text
tweets['FIFA2018'] =list(tweets['text'].apply(lambda tweet: word_in_text('FIFA2018', tweet)))
tweets['World Cup'] =list(tweets['text'].apply(lambda tweet: word_in_text('World Cup', tweet)))
```

Create Relevent to apply with words in text that appear on tweets

In [18]:

```
tweets['relevant'] = tweets['text'].apply(lambda tweet: word_in_text('FIFA2018', tweet) or
                                          word_in_text('World Cup', tweet))
```

In [19]:

```
#print keyword and Count values that appear in Captured Tweet
print(tweets['relevant'].value_counts())
print(tweets['FIFA2018'].value_counts())
print(tweets['World Cup'].value_counts())
```

```
False    5692
True      316
Name: relevant, dtype: int64
False    6005
True        3
Name: FIFA2018, dtype: int64
False    5695
True      313
Name: World Cup, dtype: int64
```

Showing Matching keyword and Football player name values that appear in Captured Tweet

In [20]:

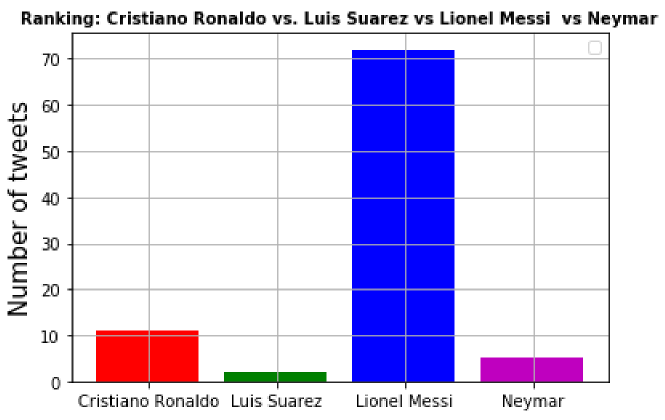
```
#print Matching keyword and Football player name values that appear in Captured Tweet
print(tweets[tweets['relevant'] == True]['Cristiano Ronaldo'].value_counts())
print(tweets[tweets['relevant'] == True]['Luis Suarez'].value_counts())
print(tweets[tweets['relevant'] == True]['Neymar'].value_counts())
print(tweets[tweets['relevant'] == True]['Lionel Messi'].value_counts())
```

```
False    305
True      11
Name: Cristiano Ronaldo, dtype: int64
False    314
True       2
Name: Luis Suarez, dtype: int64
```

```
False    311
True       5
Name: Neymar, dtype: int64
False    244
True      72
Name: Lionel Messi, dtype: int64
```

In [21]:

```
R_tweets_by_Fb_player = [tweets[tweets['relevant'] == True]['Cristiano Ronaldo'].value_counts()[Tru
e],
                      tweets[tweets['relevant'] == True]['Luis Suarez'].value_counts()[True],
                       tweets[tweets['relevant'] == True]['Lionel Messi'].value_counts()[True],
                      tweets[tweets['relevant'] == True]['Neymar'].value_counts()[True],
                     ]
x_pos = list(range(len(R_tweets_by_Fb_player)))
width = 0.8
fig, ax = plt.subplots()
plt.legend(R_tweets_by_Fb_player)
plt.bar(x_pos, R_tweets_by_Fb_player, width,alpha=1,color=pack_color)
# Setting axis labels and ticks
ax.set_ylabel('Number of tweets', fontsize=15)
ax.set_title('Ranking: Cristiano Ronaldo vs. Luis Suarez vs Lionel Messi  vs Neymar', fontsize=10,
fontweight='bold')
ax.set_xticks([p + 0.01 * width for p in x_pos])
ax.set_xticklabels(Fb_player)
plt.grid()
```



Extracting links from the relevants tweets

In this part we extracted the relevant tweets, we want to retrieve links to programming tutorials. We will start by creating a function that uses regular expressions for retrieving link that start with "http://" or "https://" from a text. This function will return the url if found, otherwise it returns an empty string.

In [22]:

```
def extract_link(text):
    regex = r'https?://[^\s<>"]+|www\.[^\s<>"]+'
    match = re.search(regex, text)
    if match:
        return match.group()
    return ''
```

Next, we will add a column called link to our tweets DataFrame. This column will contain the urls information.

In [23]:

```
tweets['link'] = tweets['text'].apply(lambda tweet: extract_link(tweet))
```

Next we will create a new DataFrame called tweets_relevant_with_link. This DataFrame is a subset of tweets DataFrame and contains all relevant tweets that have a link.

In [24]:

```
tweets_relevant = tweets[tweets['relevant'] == True]
tweets_relevant_with_link = tweets_relevant[tweets_relevant['link'] != '']
```

We can now print out all links for football player by executing the commands below:

In [25]:

```
print("---------Link of Cristiano Ronaldo---------")
print(tweets_relevant_with_link[tweets_relevant_with_link['Cristiano Ronaldo'] == True]['link'])
print("---------Link of Luis Suarez---------")
print(tweets_relevant_with_link[tweets_relevant_with_link['Luis Suarez'] == True]['link'])
print("---------Lionel Messi---------")
print(tweets_relevant_with_link[tweets_relevant_with_link['Lionel Messi'] == True]['link'])
print("---------Link of Neymar---------")
print(tweets_relevant_with_link[tweets_relevant_with_link['Neymar'] == True]['link'])
```

```
---------Link of Cristiano Ronaldo---------
373     https://t.co/xYNBU3idcK
1352    https://t.co/eYWpsJ7XLn
4905    https://t.co/pIqKs66R4H
4915    https://t.co/1flncGqa2l
5710    https://t.co/zGbL1BtDiK
Name: link, dtype: object
---------Link of Luis Suarez---------
1573    https://t.co/g7xaLpMJZL
5965          https://t.co/8rA…
Name: link, dtype: object
---------Lionel Messi---------
373     https://t.co/xYNBU3idcK
375     https://t.co/XncHQxy9u3
448     https://t.co/jEWLT0qasF
509     https://t.co/WGWOJvpVX3
776     https://t.co/JHpoMbdSV4
1080    https://t.co/LqI9wCxuha
1313                https://t.…
1428        https://t.co/WfiyW…
1581        https://t.co/HfG9G…
1658    https://t.co/wtMkMSeDrz
1681    https://t.co/qnxlZ9ndY3
1984    https://t.co/zo9LJWVlOA
2224    https://t.co/pARNK1dJje
2298    https://t.co/NsefOoLFAT
2304    https://t.co/XncHQxy9u3
2337    https://t.co/zo9LJWVlOA
2344    https://t.co/7445uioON9
2732    https://t.co/1EEWindnyk
2813    https://t.co/NKfIikecJR
2872           https://t.co/Zi…
2960    https://t.co/lH3FpHrMEb
3073        https://t.co/HfG9G…
3355    https://t.co/OIRFzC4q0R
3633    https://t.co/XncHQxy9u3
3873    https://t.co/vs2fOx0l2S
3878    https://t.co/Vc8gKFaCOh
3897    https://t.co/p6ohWuleYw
3940    https://t.co/bs69fXUNdr
3956    https://t.co/XncHQxy9u3
4339    https://t.co/XncHQxy9u3
4492    https://t.co/KpkqrQbkJp
4526    https://t.co/S82OXzzt6E
4535    https://t.co/j5R3OxlCdw
4541    https://t.co/Qtx89psYLy
4551    https://t.co/xnP1dJ6GgQ
4657    https://t.co/rsv5IhHruQ
4672    https://t.co/nxCuG7Shjn
4905    https://t.co/pIqKs66R4H
5020    https://t.co/XncHQxy9u3
5021    https://t.co/Lctkzn30Eu
5027    https://t.co/kXkbLtYs61
5109        https://t.co/HfG9G…
5144        https://t.co/6jlTV…
5381    https://t.co/TjaQJSWkmO
5459    https://t.co/XncHQxy9u3
```

```
5459    https://t.co/XncHQxy9u3
5468     https://t.co/3duhtbFU…
5547    https://t.co/iFEDTrni9f
5670    https://t.co/x0pcSWYGhL
5706    https://t.co/XncHQxy9u3
5761    https://t.co/AGoEb83GSz
5809        https://t.co/HfG9G…
5992    https://t.co/uhAkGvlUxv
5996    https://t.co/R0pmhGptVw
6005        https://t.co/HfG9G…
Name: link, dtype: object
---------Link of Neymar---------
32      https://t.co/mVAa0AsmiZ
76      https://t.co/a52TItj7h6
1666    https://t.co/JXuNIg9W9j
4264    https://t.co/rCxHJcEKRd
Name: link, dtype: object
```

Reference

- https://matplotlib.org
- https://apps.twitter.com
- https://developer.twitter.com
- http://www.tweepy.org
- http://en.wikipedia.org/wiki/Text_mining
- http://en.wikipedia.org/wiki/Word-sense_disambiguation
- http://en.wikipedia.org/wiki/Regular_expression
