

Adil Moujahid

[Follow @AdilMouja](#)

Published

Mon 21 July 2014

[← Home](#)

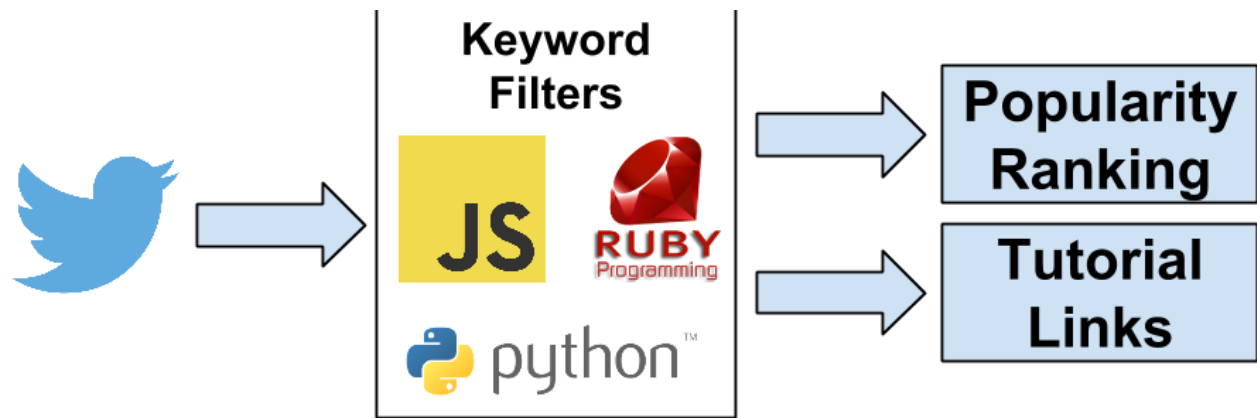
An Introduction to Text Mining using Twitter Streaming API and Python

// tags [python](#) [pandas](#) [text mining](#) [matplotlib](#) [twitter](#)

Text mining is the application of natural language processing techniques and analytical methods to text data in order to derive relevant information. Text mining is getting a lot attention these last years, due to an exponential increase in digital text data from web pages, google's projects such as [google books](#) and [google ngram](#), and social media services such as Twitter. Twitter data constitutes a rich source that can be used for capturing information about any topic imaginable. This data can be used in different use cases such as finding trends related to a specific keyword, measuring brand sentiment, and gathering feedback about new products and services.

In this tutorial, I will use Twitter data to compare the popularity of 3 programming languages: Python, Javascript and Ruby, and to retrieve links to programming tutorials. In the first paragraph, I will explain how to connect to Twitter Streaming API and how to get the data. In the second paragraph, I will explain how to structure the data for analysis, and in the last paragraph, I will explain how to filter the data and extract links from tweets.

Using only 2 days worth of Twitter data, I could retrieve 644 links to python tutorials, 413 to javascript tutorials and 136 to ruby tutorials. Furthermore, I could confirm that python is 1.5 times more popular than javascript and 4 times more popular than ruby.



1. Getting Data from Twitter Streaming API

API stands for Application Programming Interface. It is a tool that makes the interaction with computer programs and web services easy. Many web services provides APIs to developers to interact with their services and to access data in programmatic way. For this tutorial, we will use Twitter Streaming API to download tweets related to 3 keywords: "python", "javascript", and "ruby".

Step 1: Getting Twitter API keys

In order to access Twitter Streaming API, we need to get 4 pieces of information from Twitter: API key, API secret, Access token and Access token secret. Follow the steps below to get all 4 elements:

- 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".

Step 2: Connecting to Twitter Streaming API and downloading data

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"

#This is a basic listener that just prints received tweets to stdout.
class StdOutListener(StreamListener):

    def on_data(self, data):
        print data
        return True

    def on_error(self, status):
        print status

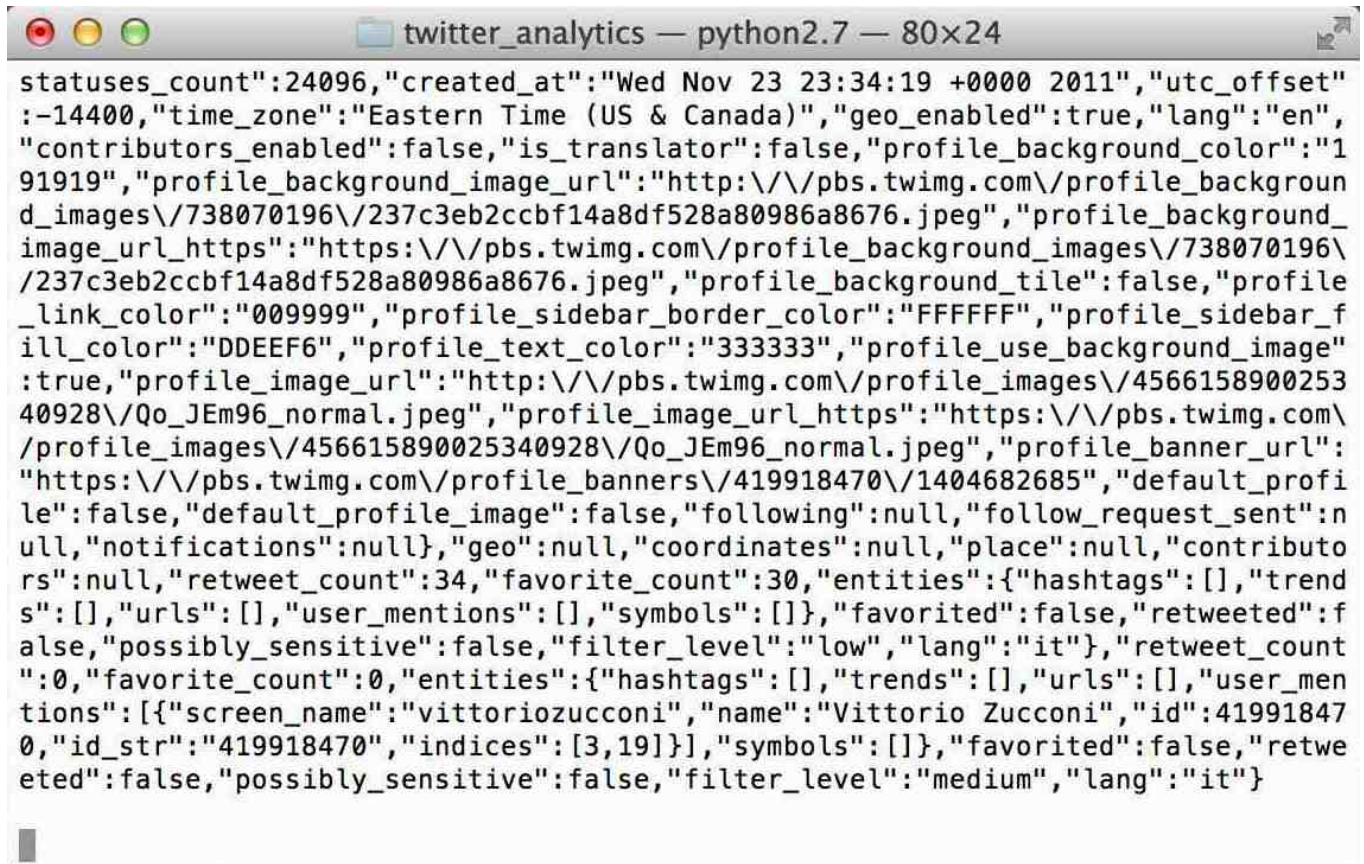
if __name__ == '__main__':

    #This handles Twitter authentication and the connection to Twitter Streaming API
    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: 'python', 'javascript',
    'ruby'
    stream.filter(track=['python', 'javascript', 'ruby'])

```

If you run the program from your terminal using the command: `python twitter_streaming.py`, you will see data flowing like the picture below.



```

status_count":24096,"created_at":"Wed Nov 23 23:34:19 +0000 2011","utc_offset"
:-14400,"time_zone":"Eastern Time (US & Canada)","geo_enabled":true,"lang":"en",
"contributors_enabled":false,"is_translator":false,"profile_background_color":"1
91919","profile_background_image_url":"http://pbs.twimg.com/profile_backgroun
d_images/738070196/237c3eb2ccbf14a8df528a80986a8676.jpeg","profile_background_
image_url_https":"https://pbs.twimg.com/profile_background_images/738070196/
237c3eb2ccbf14a8df528a80986a8676.jpeg","profile_background_tile":false,"profile
_link_color":"009999","profile_sidebar_border_color":"FFFFFF","profile_sidebar_f
ill_color":"DDEEF6","profile_text_color":"333333","profile_use_background_image"
:true,"profile_image_url":"http://pbs.twimg.com/profile_images/4566158900253
40928/Qo_JEm96_normal.jpeg","profile_image_url_https":"https://pbs.twimg.com/
profile_images/456615890025340928/Qo_JEm96_normal.jpeg","profile_banner_url":
"https://pbs.twimg.com/profile_banners/419918470/1404682685","default_profi
le":false,"default_profile_image":false,"following":null,"follow_request_sent":n
ull,"notifications":null},"geo":null,"coordinates":null,"place":null,"contributo
rs":null,"retweet_count":34,"favorite_count":30,"entities":{"hashtags":[],"trend
s":[],"urls":[],"user_mentions":[],"symbols":[]},"favorited":false,"retweeted":f
alse,"possibly_sensitive":false,"filter_level":"low","lang":"it"},"retweet_count
":0,"favorite_count":0,"entities":{"hashtags":[],"trends":[],"urls":[],"user_men
tions":[{"screen_name":"vittoriozucconi","name":"Vittorio Zucconi","id":41991847
0,"id_str":"419918470","indices":[3,19]}],"symbols":[]},"favorited":false,"retwe
eted":false,"possibly_sensitive":false,"filter_level":"medium","lang":"it"}

```

You can stop the program by pressing Ctrl-C.

We want to capture this data into a file that we will use later for the analysis. You can do so by piping the output to a file using the following command: `python twitter_streaming.py > twitter_data.txt`.

I run the program for 2 days (from 2014/07/15 till 2014/07/17) to get a meaningful data sample. This file size is 242 MB.

2. Reading and Understanding the data

The data that we stored `twitter_data.txt` is in JSON format. JSON stands for JavaScript Object Notation. This format makes it easy to humans to read the data, and for machines to parse it. Below is an example for one tweet in JSON format. You can see that the tweet contains additional information in addition to the main text which in this example: "Yaayyy I learned some JavaScript today! #thatwasntsohard #yesitwas #stoptalkingtoyourself #hashbrown #hashtag".

```
{
  "created_at": "Tue Jul 15 14:19:30 +0000
  2014",
  "id": 489051636304990208,
  "id_str": "489051636304990208",
  "text": "Yaayyy I learned some
  JavaScript today! #thatwasntsohard #yesitwas #stoptalkingtoyourself #hashbrown
  #hashtag",
  "source": "\u003ca href=\"http://twitter.com/download/iphone\"
  rel=\"nofollow\" \u003eTwitter for
  iPhone\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": 2301702187,
    "id_str": "2301702187",
    "name": "Toni
    Barlettano",
    "screen_name": "itsmetonib",
    "location": "Greater NYC
    Area",
    "url": "http://www.tonib.me",
    "description": "So Full of Art | \nToni Barlettano Creative
    Media +
    Design",
    "protected": false,
    "followers_count": 8,
    "friends_count": 25,
    "listed_count": 0,
    "created_at": "Mon
    Jan 20 16:49:46 +0000
    2014",
    "favourites_count": 6,
    "utc_offset": null,
    "time_zone": null,
    "geo_enabled": false,
    "verified": false,
    "statuses_count": 20,
    "lang": "en",
    "contributors_enabled": false,
    "is_translator": false,
    "is_translation_enabled": false,
    "profile_background_color": "C0DEED",
    "profile_background_image_url": "http://abs.twimg.com/images/themes/theme1/bg.png",
    "profile_background_image_url_https": "https://abs.twimg.com/images/themes/theme1/bg.png",
    "profile_background_tile": false,
    "profile_image_url": "http://pbs.twimg.com/profile_images/425313048320958464/Z2GcderW_normal.jpeg",
    "profile_image_url_https": "https://pbs.twimg.com/profile_images/425313048320958464/Z2GcderW_normal.jpeg",
    "profile_link_color": "0084B4",
    "profile_sidebar_border_color": "C0DEED",
    "profile_sidebar_fill_color": "DDEEF6",
    "profile_text_color": "333333",
    "profile_use_background_image": true,
    "default_profile": true,
    "default_profile_image": false,
    "following": null,
    "follow_request_sent": null,
    "notifications": null,
    "geo": null,
    "coordinates": null,
    "place": null,
    "contributors": null,
    "retweet_count": 0,
    "favorite_count": 0,
    "entities": {
      "hashtags": [
        {
          "text": "thatwasntsohard",
          "indices": [40, 56]
        },
        {
          "text": "yesitwas",
          "indices": [57, 66]
        },
        {
          "text": "stoptalkingtoyourself",
          "indices": [67, 89]
        },
        {
          "text": "hashbrown",
          "indices": [90, 100]
        },
        {
          "text": "hashtag",
          "indices": [101, 109]
        }
      ],
      "symbols": [],
      "urls": [],
      "user_mentions": []
    },
    "favorited": false,
    "retweeted": false,
    "filter_level": "medium",
    "lang": "en"
  }
}
```

For the remaining of this tutorial, we will be using 4 Python libraries `json` for parsing the data, `pandas` for data manipulation, `matplotlib` for creating charts, and `re` for regular expressions. The `json` and `re` libraries are installed by default in Python. You should install `pandas` and `matplotlib` if you don't have them in your machine.

We will start first by uploading `json` and `pandas` using the commands below:

```
import json
import pandas as pd
import matplotlib.pyplot as plt
```

Next we will read the data in into an array that we call `tweets`.

```
tweets_data_path = '../data/twitter_data.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

```

We can print the number of tweets using the command below. For the dataset that I prepared, the number is 71238.

```
print len(tweets_data)
```

Next, we will structure the tweets data into a pandas DataFrame to simplify the data manipulation. We will start by creating an empty DataFrame called `tweets` using the following command.

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

Next, we will add 3 columns to the `tweets` DataFrame called `text`, `lang`, and `country`. `text` column contains the tweet, `lang` column contains the language in which the tweet was written, and `country` the country from which the tweet was sent.

```

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

```

Next, we will create 2 charts: The first one describing the Top 5 languages in which the tweets were written, and the second the Top 5 countries from which the tweets were sent.

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



Join my Data in Practice Newsletter

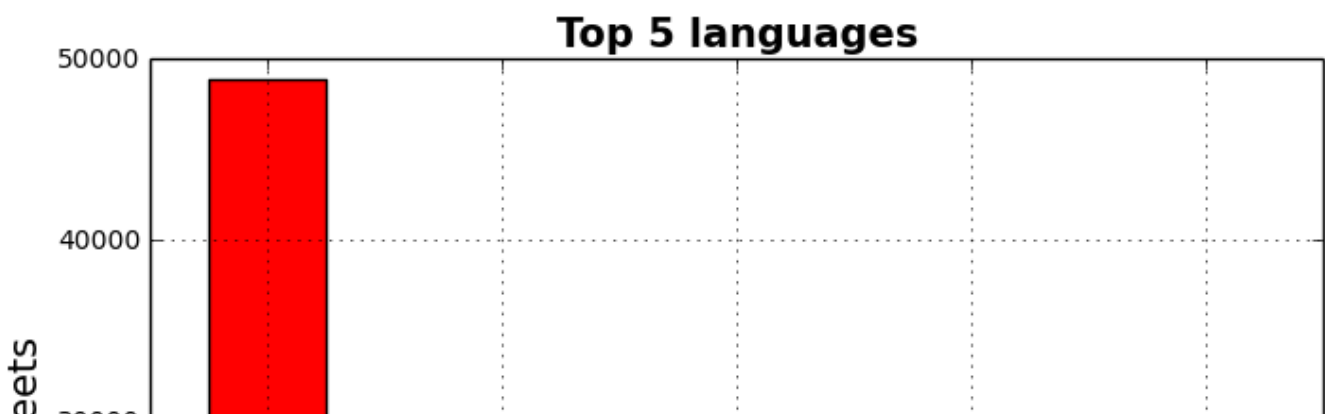


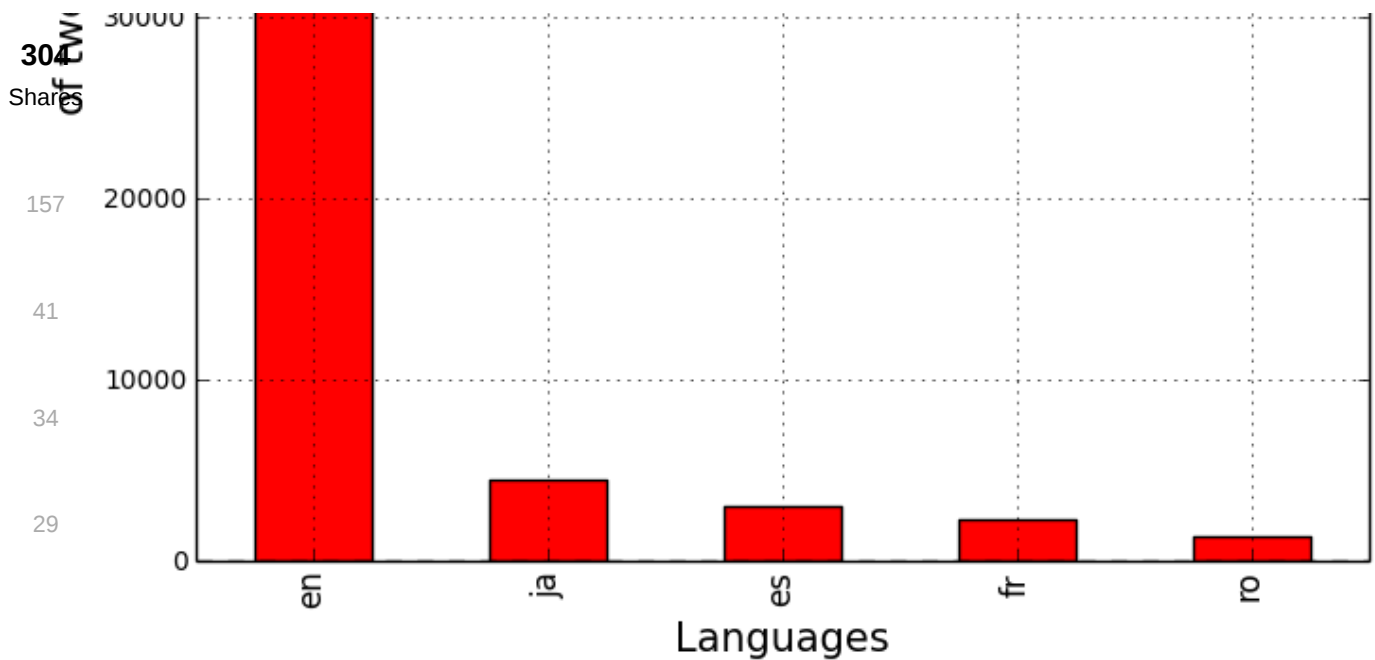
SUBSCRIBE NOW

```

ax.set_title('Top 5 languages', fontsize=15, fontweight='bold')
tweets_by_lang[:5].plot(ax=ax, kind='bar', color='red')

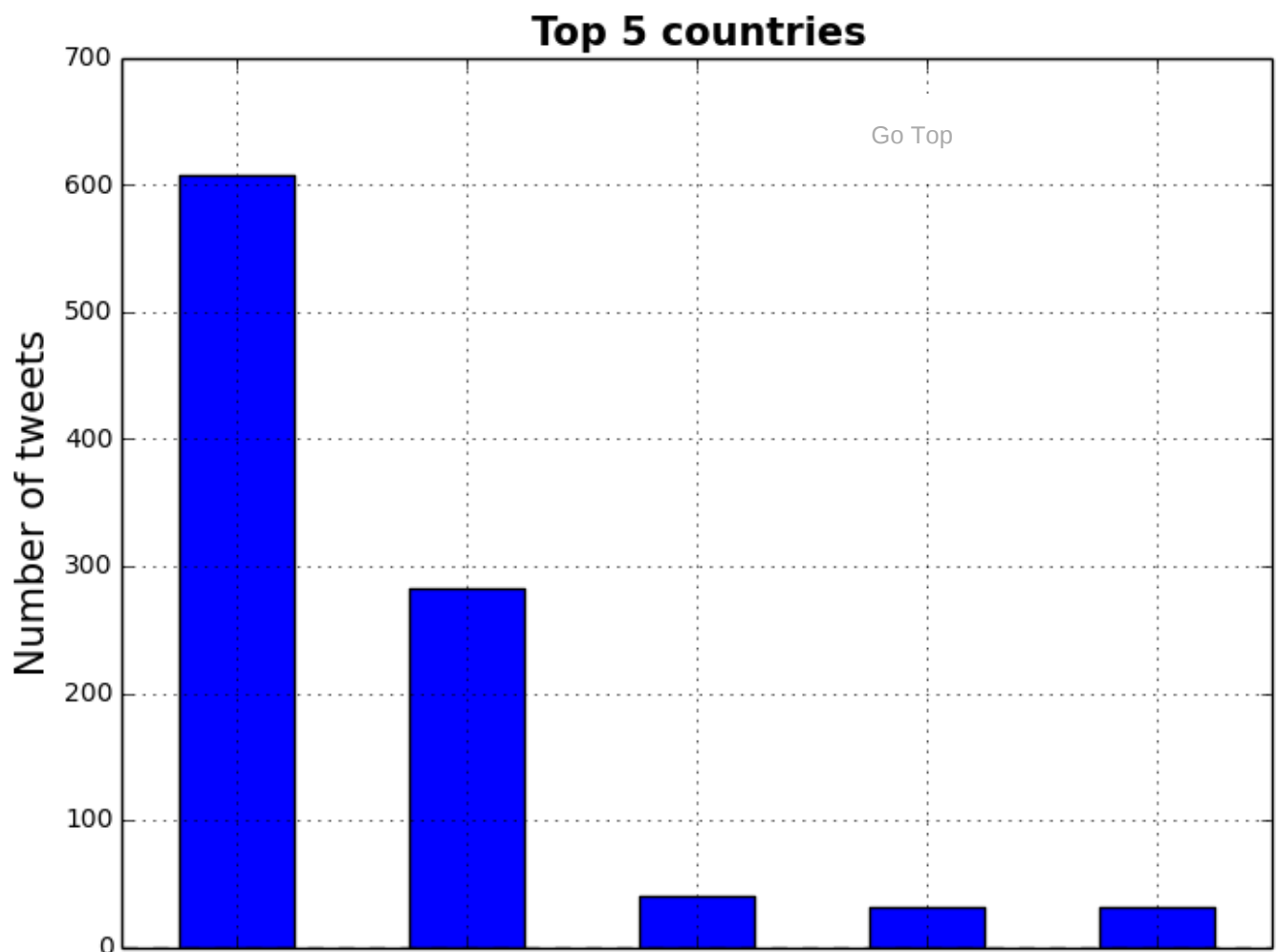
```





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

```
fig, ax = plt.subplots()
ax.tick_params(axis='x', labelsz=15)
ax.tick_params(axis='y', labelsz=10)
ax.set_xlabel('Countries', fontsize=15)
ax.set_ylabel('Number of tweets', fontsize=15)
ax.set_title('Top 5 countries', fontsize=15, fontweight='bold')
tweets_by_country[:5].plot(ax=ax, kind='bar', color='blue')
```





3. Mining the tweets

Our main goals in these text mining tasks are: compare the popularity of Python, Ruby and Javascript 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 manipulate the data easily.
- Target tweets that have "programming" or "tutorial" keywords.
- Extract links from the relevant tweets

Adding Python, Ruby, and Javascript tags

First, we will create a function that checks if a specific keyword is present in a text. We will do this by using [regular expressions](#). Python provides a library for regular expression called `re`. We will start by importing this library

```
import re
```

Next we will create a function called `word_in_text(word, text)`. This function returns True if a word is found in text, otherwise it returns False.

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

Next, we will add 3 columns to our tweets DataFrame.

```
tweets['python'] = tweets['text'].apply(lambda tweet: word_in_text('python', tweet))  
tweets['javascript'] = tweets['text'].apply(lambda tweet: word_in_text('javascript', tweet))  
tweets['ruby'] = tweets['text'].apply(lambda tweet: word_in_text('ruby', tweet))
```

We can calculate the number of tweets for each programming language as follows:

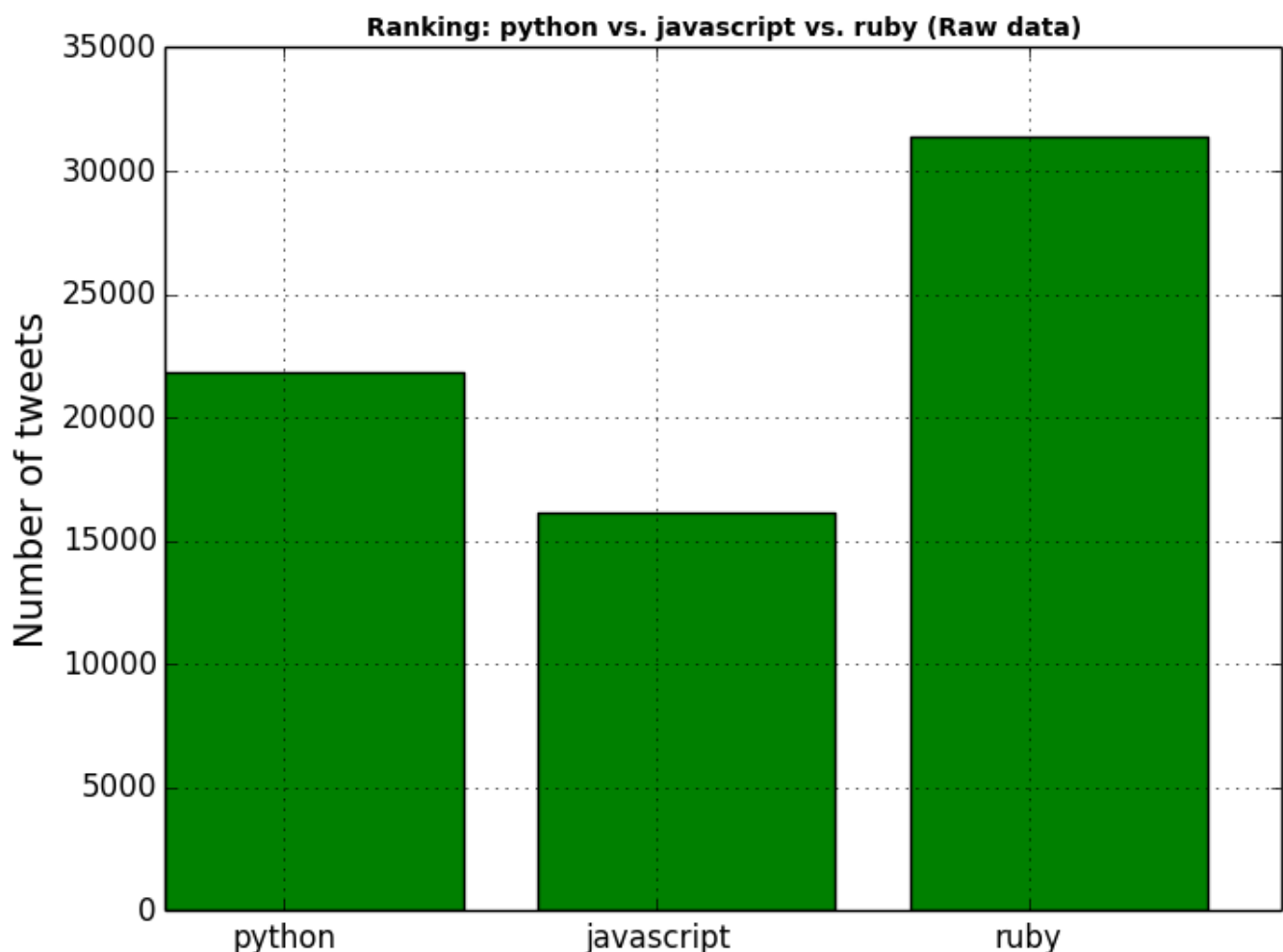
```
print tweets['python'].value_counts()[True]  
print tweets['javascript'].value_counts()[True]  
print tweets['ruby'].value_counts()[True]
```

This returns: 21839 for python, 16154 for javascript and 31410 for ruby. We can make a simple comparison chart by executing the following:

```
prg_langs = ['python', 'javascript', 'ruby']
tweets_by_prg_lang = [tweets['python'].value_counts()[True], tweets['javascript'].value_counts()[True], tweets['ruby'].value_counts()[True]]

x_pos = list(range(len(prg_langs)))
width = 0.8
fig, ax = plt.subplots()
plt.bar(x_pos, tweets_by_prg_lang, width, alpha=1, color='g')

# Setting axis labels and ticks
ax.set_ylabel('Number of tweets', fontsize=15)
ax.set_title('Ranking: python vs. javascript vs. ruby (Raw data)', fontsize=10, fontweight='bold')
ax.set_xticks([p + 0.4 * width for p in x_pos])
ax.set_xticklabels(prg_langs)
plt.grid()
```



This shows, that the keyword ruby is the most popular, followed by python then javascript. However, the tweets DataFrame contains information about all tweets that contains one of the 3 keywords and doesn't restrict the information to the programming languages. For example, there are a lot tweets that contains the keyword ruby and that are related to a political scandal called [Rubygate](#). In the next section, we will filter the tweets and re-run the analysis to make a more accurate comparison.

Targeting relevant tweets

We are interested in targeting tweets that are related to programming languages. Such tweets often have one of the 2 keywords: "programming" or "tutorial". We will create 2 additional columns to our tweets DataFrame where we will add this information.

```
tweets['programming'] = tweets['text'].apply(lambda tweet: word_in_text('programming', tweet))
tweets['tutorial'] = tweets['text'].apply(lambda tweet: word_in_text('tutorial', tweet))
```

We will add an additional column called `relevant` that takes value `True` if the tweet has either "programming" or "tutorial" keyword, otherwise it takes value `False`.

```
tweets['relevant'] = tweets['text'].apply(lambda tweet: word_in_text('programming', tweet) or
word_in_text('tutorial', tweet))
```

We can print the counts of relevant tweet by executing the commands below.

```
print tweets['programming'].value_counts()[True]
print tweets['tutorial'].value_counts()[True]
print tweets['relevant'].value_counts()[True]
```

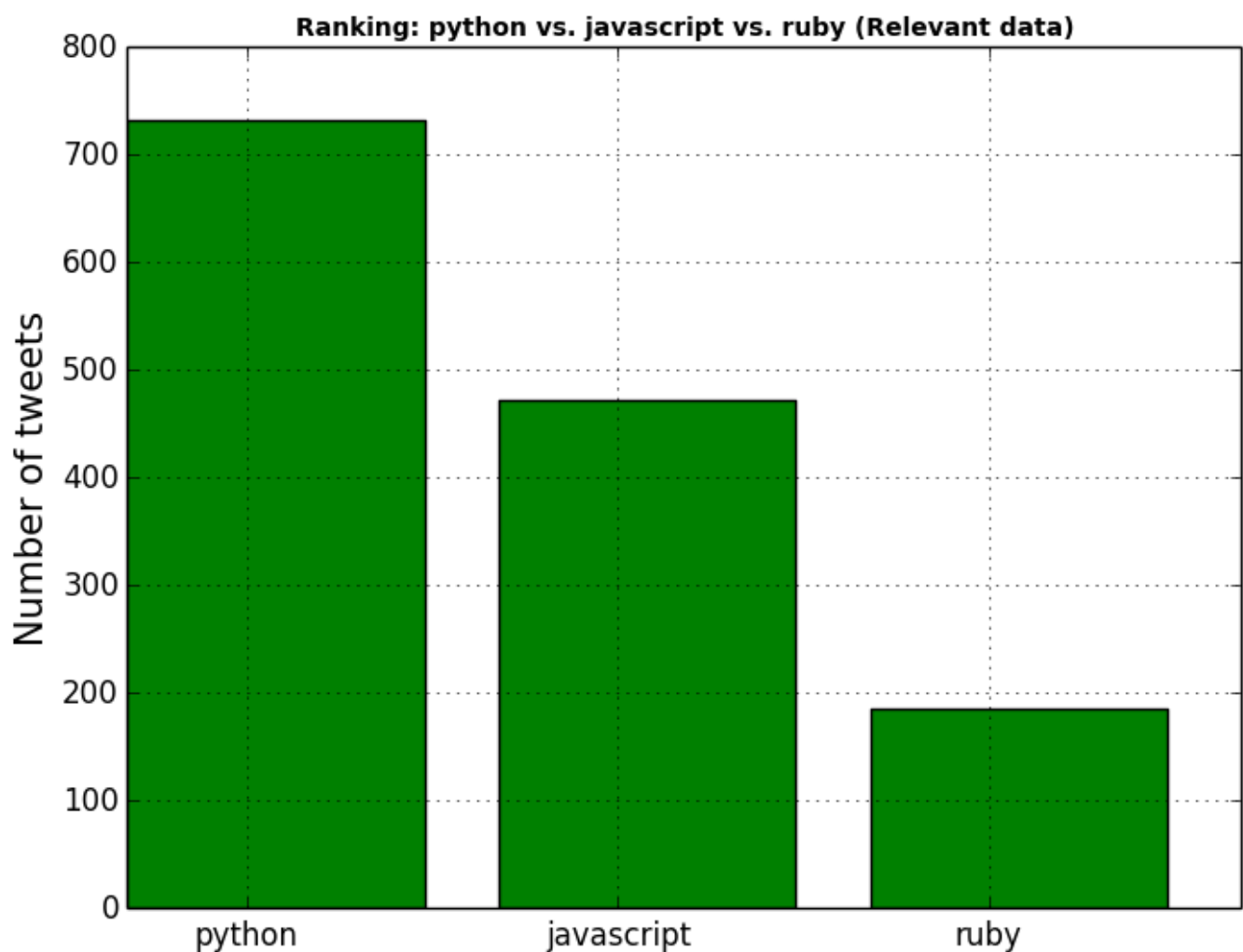
This returns, 871 for programming column, 511 for tutorial column, and 1356 for relevant column.

We can compare now the popularity of the programming languages by executing the commands below.

```
print tweets[tweets['relevant'] == True]['python'].value_counts()[True]
print tweets[tweets['relevant'] == True]['javascript'].value_counts()[True]
print tweets[tweets['relevant'] == True]['ruby'].value_counts()[True]
```

Python is the most popular with a count of 732, followed by javascript by a count of 473, and ruby by a count of 185. We can make a comparison graph by executing the commands below:

```
tweets_by_prg_lang = [tweets[tweets['relevant'] == True]['python'].value_counts()[True],
                      tweets[tweets['relevant'] == True]['javascript'].value_counts()[True],
                      tweets[tweets['relevant'] == True]['ruby'].value_counts()[True]]
x_pos = list(range(len(prg_langs)))
width = 0.8
fig, ax = plt.subplots()
plt.bar(x_pos, tweets_by_prg_lang, width, alpha=1, color='g')
ax.set_ylabel('Number of tweets', fontsize=15)
ax.set_title('Ranking: python vs. javascript vs. ruby (Relevant data)', fontsize=10,
fontweight='bold')
ax.set_xticks([p + 0.4 * width for p in x_pos])
ax.set_xticklabels(prg_langs)
plt.grid()
```



Extracting links from the relevant tweets

Now that 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.

```
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.

```
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.

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

We can now print out all links for python, javascript, and ruby by executing the commands below:

```
print tweets_relevant_with_link[tweets_relevant_with_link['python'] == True]['link']
print tweets_relevant_with_link[tweets_relevant_with_link['javascript'] == True]['link']
print tweets_relevant_with_link[tweets_relevant_with_link['ruby'] == True]['link']
```

This returns 644 links for python, 413 links for javascript, and 136 for ruby. Below are some python related links

- <http://t.co/WmTccp3rb1>
- <http://t.co/5qE3vPAy7N>
- <http://t.co/1rvmhqPsXD>
- <http://t.co/S9aq2AahjH>
- <http://t.co/ORg6IL8qXT>
- <http://t.co/EnK2UIDcJ8>
- <http://t.co/gtu9WVQCLK>
- <http://t.co/xvMTzqLGg0>
- <http://t.co/bgMZ0jlpA7>
- <http://t.co/O03VrRyEAb>
- <http://t.co/CfWYefZML7>
- <http://t.co/N3iU2ZYa2z>
- <http://t.co/S9aq2AahjH>
- <http://t.co/ytms7bcsQV>

Conclusion

In this tutorial, we covered many techniques used in text mining. The code provide in this post could be modified to create a deeper analysis or could be adapted to another use case. For those who want to go further in text mining, I recommend to follow up by studying regular expressions

You can find the source code from this tutorial in this [github repository](#) [github link](#).

references

- http://en.wikipedia.org/wiki/Text_mining
- http://en.wikipedia.org/wiki/Word-sense_disambiguation
- http://en.wikipedia.org/wiki/Regular_expression

Subscribe to my Data in Practice Newsletter

Subscribe

58 Comments

adilmoujahid.com

Login ▾

♥ Recommend 2

🔗 Share

Sort by Best ▾



Join the discussion...



Ashwin Murali • 8 months ago

Adil,

Thank you SO much for this wonderful post. I've been taking a keen interest in Data Sciences and Mining of late and this was like a godsend article to pick up and try something at a pace that I could grasp and understand.

Thanks once again!

10 ^ | ▾ • Reply • Share >



Adil Moujahid Mod ➔ Ashwin Murali • 8 months ago

Cruisemaniac,

Thank you very much for the kind words. I'm glad you liked the post. Stayed tuned, more interesting stuff coming :)

Adil

^ | ▾ • Reply • Share >



Keith Wilson • 8 months ago

This is awesome! Thank you!!

7 ^ | ▾ • Reply • Share >



Dave Roma • 5 months ago

Excellent! I really like that you exemplified not only how to stream but an actual useful mining scenario. Nice job!

5 ^ | v • Reply • Share >



Hussein Ghaly • 8 months ago

Great Tutorial, thanks Adil!

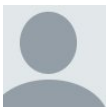
5 ^ | v • Reply • Share >



Adam Hughes • a month ago

Awesome. Why can't Twitter post something like this on their API page? An example is worth 1000 call signatures.

2 ^ | v • Reply • Share >



Yolandi Chia • 2 months ago

Hi Adil,

Thank you for this tutorial. Everything was working fine until I got this error:

```
x_pos = list(range(len(prg_langs)))  
NameError: name 'prg_langs' is not defined
```

Do you have any advice of how to fix this? I'm having trouble figuring out what it means.

Thanks!!

2 ^ | v • Reply • Share >



Yogesh Kamble • a month ago

This is very nice and neat tutorial. One can start learning data mining from your blog. Thank you very much.

1 ^ | v • Reply • Share >



Amar Parkash • 2 months ago

Really nice !! ...thanks a lot

1 ^ | v • Reply • Share >



Maarten Keulemans • 3 months ago

Hi Adil, thanks man! Wonderful tutorial!

1 ^ | v • Reply • Share >



bobby mcmuffin • 8 months ago

i think you forgot:

```
import matplotlib.pyplot as plt
```

1 ^ | v • Reply • Share >



Adil Moujahid Mod ➔ bobby mcmuffin • 8 months ago



Adil Moujahid • 1 month ago

It's corrected now. Thanks !!

^ | v • Reply • Share >



Hathal • 11 days ago

thanx for the great Post. I do have a question regarding gathering the data from twitter using the terminal. How to avoid the timeout which limit the amount of tweets gathered ?

^ | v • Reply • Share >



pramod kumar • 16 days ago

i'm getting a error for the above code "missing paranthesis in call to print"
i would appreciate any help

^ | v • Reply • Share >



Guest • 17 days ago

I can't seem to display the graphs.I am running the script via cmd.Am I doing anything wrong ? The tweet_by_country and other 3 files are generated but I don't see any graphs.Can you please help me ?

^ | v • Reply • Share >



William Greene → Guest • 12 days ago

Hi Guest, i'm unsure what OS you are using, in the code it shows the pictures are being encoded in the png format. Try adding that extension to the tweet files and opening again?

```
plt.savefig('tweet_by_lang', format='png')
```



^ | v • Reply • Share >



Andrew • 20 days ago

Hey Adil, very good tutorial! How can you filter after the time_zone as well ?

^ | v • Reply • Share >



gaurhari dass • 22 days ago

Hi Adil do u have idea about how to use advanced search api?

^ | v • Reply • Share >



Ankit Gupta • a month ago

Hi Adil, excellent post!

On executing the code under section 1, I got the following errors -

```
PS C:\Python27\mvWork> python \streamTwitter.py
```


Traceback (most recent call last):

File ".\streamTwitter.py", line 27, in <module>

stream.filter(track=['python', 'javascript', 'ruby'])

File "build\bdist.win32\egg\tweepy\streaming.py", line 428, in filter

File "build\bdist.win32\egg\tweepy\streaming.py", line 346, in _start

File "build\bdist.win32\egg\tweepy\streaming.py", line 239, in _run

File "C:\Python27\lib\site-packages\requests-2.5.2-py2.7.egg\requests\sessions.py", line 447, in request

[see more](#)

^ | v • [Reply](#) • [Share](#) >



Ankitha • a month ago

hi i am getting 401 error can u please suggest how to fix this error

^ | v • [Reply](#) • [Share](#) >



William Greene → [Ankitha](#) • 12 days ago

Need more details, sure you have right credentials for twitter api? is your 401 error occurring when trying to retrieve data? consider posting error output/pics.

^ | v • [Reply](#) • [Share](#) >



Ankitha → [William Greene](#) • 12 days ago

thanks allot for the reply., issue fixed :) there was some problrm in the twitter authentication.

^ | v • [Reply](#) • [Share](#) >



Karthik Srinivasan • a month ago

Thank you very much. Really useful stuff.

^ | v • [Reply](#) • [Share](#) >



me0 • a month ago

hi, thanks for the script: I wonder why I'm getting an error SyntaxError: Missing parentheses in call to 'print'?

where to put your python script? in the /tweepy folder?

^ | v • [Reply](#) • [Share](#) >



Rahul Sitaraman • a month ago

Dear Adil,

This is the output of my Python Shell

```
>>> ===== RESTART
```

```
=====
```

```
>>>
```

```
9511
```

Traceback (most recent call last):

File "/home/rahul/kabutar.py", line 19, in <module>

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

File "/home/rahul/kabutar.py", line 19, in <lambda>

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

KeyError: 'text'

```
>>>
```

Can you please help me with this.

Regards

Rahul

^ | v • Reply • Share ›



سلوى • a month ago

Bonjour,

j'ai un problème dans l'utilisation de matplotlib, le module numpy n'est pas reconnu alors que je l'ai déjà installé

Hello,

I have a problem in installation of matplotlib, it asks me to install numpy but i've already installed it.

^ | v • Reply • Share ›



Ayush Kumar • a month ago

In the very first script to fetch the tweets, what should I do if I want data about all the tweets I can get, not filtered by any keywords??

^ | v • Reply • Share ›



santoshi • a month ago

```
x_pos = list(range(len(prg_langs)))
```

NameError: name 'prg_langs' is not defined

can anyone suggest me..thank you..

^ | v • Reply • Share ›



Adil Moujahid Mod ➔ santoshi • a month ago

You should add: prg_langs = ['python', 'javascript', 'ruby']

It is fixed now

1 ^ | v • Reply • Share ›



santoshi ➔ Adil Moujahid • 23 days ago



I am getting this error

Traceback (most recent call last):

File "D:\Python27\tweetd.py", line 19, in <module>

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

File "D:\Python27\tweetd.py", line 19, in <lambda>

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

TypeError: 'int' object has no attribute '__getitem__'

^ | v • Reply • Share >



TD → santoshi • a month ago

I guess prg_langs is the array ['python', 'javascript', 'ruby']. Right ?

1 ^ | v • Reply • Share >



Matteo Manca • a month ago

Amazing ! very useful.

Thanks a lot

^ | v • Reply • Share >



hind kader • a month ago

bon article merci bccp

^ | v • Reply • Share >



Neha Sood • 2 months ago

hello sir, can u please tell how to extract tweets based on multiple keywords.

keyword1 AND keyword2 AND keyword3.... also, i would like to extract data on the basis of location. keyword1 AND location

^ | v • Reply • Share >



Ayoub Massoudi • 2 months ago

Hi Moujahid,

Thank you a lot for this excellent tutorial.

I got an error when I try :

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

AttributeError: 'list' object has no attribute 'value_counts'????

Is there something that i missed? is value_counts a pandas method and if so why dont you call pandas for this???

I'm using python 2.7.6

Thank you for helping

Ayoub

^ | v • Reply • Share >

**si_salah** • 3 months ago

great tutorial , thank you verry much !

^ | v • Reply • Share >

**chiranjeevi** • 3 months ago

```
tweets['python'] = tweets['text'].apply(lambda tweet: word_in_text('python', tweet))
tweets['javascript'] = tweets['text'].apply(lambda tweet: word_in_text('javascript', tweet))
```

```
tweets['ruby'] = tweets['text'].apply(lambda tweet: word_in_text('ruby', tweet))
```

i am facing problem in the above code :(i am unable to clear it off can you help me out?

below is the error i am facing :)

Traceback (most recent call last):

File "C:/Users/CHIRANJEEVI/Desktop/My New Project/mining.py", line 42, in
<module>

```
tweets['python'] = tweets['text'].apply(lambda tweet: word_in_text('python', tweet))
```

File "C:\Python34\lib\site-packages\pandas\core\series.py", line 2058, in apply

```
mapped = lib.map_infer(values, f, convert=convert_dtype)
```

File "pandas\src\inference.pyx", line 1046, in pandas.lib.map_infer
(pandas\lib.c:56997)

File "C:/Users/CHIRANJEEVI/Desktop/My New Project/mining.py", line 42, in
<lambda>

```
tweets['python'] = tweets['text'].apply(lambda tweet: word_in_text('python', tweet))
```

File "C:/Users/CHIRANJEEVI/Desktop/My New Project/mining.py", line 33, in
word_in_text

```
text = text.lower()
```

AttributeError: 'map' object has no attribute 'lower'

^ | v • Reply • Share >

**disqus_Uheal01PgF** → chiranjeevi • 2 months ago

If you are using Python 3.x, replace the line `tweets['python'] = tweets['text'].apply(lambda tweet: word_in_text('python', tweet))` with `tweets['text'] = [tweet['text'] for tweet in tweets_data]`

^ | v • Reply • Share >

**Adil Moujahid** Mod → chiranjeevi • 3 months ago

You should use Python 2.7 and not Python 3.4



Rajiv Bajpai • 3 months ago

Hi Adil, Sorry to bother u again,

```
RAJIVs-MacBook-Pro:python rajivbajpai$ python twitter_streaming.py
```

Traceback (most recent call last):

File "twitter_streaming.py", line 2, in <module>

```
from tweepy.streaming import StreamListener
```

ImportError: No module named tweepy.streaming

this is the error, couldn't understand what is going wrong :(

^ | v • Reply • Share ›



Adil Moujahid Mod ➔ Rajiv Bajpai • 3 months ago

Python didn't find Tweepy library. Did u install Tweepy?

^ | v • Reply • Share ›



Rajiv Bajpai • 3 months ago

Hi Adil ,thanks for such a nice tutorial.I am very new in this field,when m trying to run the code i got this error access_token_secret = "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
^

SyntaxError: EOL while scanning string literal

can u please suggest some help

^ | v • Reply • Share ›



Adil Moujahid Mod ➔ **Rajiv Bajpai** • 3 months ago

Hi Rajiv,

You should get your API keys from Twitter and insert the values in the python code. Follow the instructions in [Step 1: Getting Twitter API keys](#).

Adil

^ | v • Reply • Share ›



Antonio Falcone • 3 months ago

Hi Adil,

Is it possible to set a parameter "until" to return tweets generated before the given date?

Thank you

^ | v • Reply • Share ›



Adil Moujahid Mod → Antonio Falcone • 3 months ago

Hi Antonio,

In this tutorial, I used the Twitter Streaming API. This API is used to stream realtime data.

If you want to get historical data, you should use the Twitter Search API.

However, this will get only a few tweets. If you want to get more tweets, then you should buy them from Twitter data resellers like DataSift.com. Below is the link to the search API documentation.

<https://dev.twitter.com/rest/p...>

^ | v • Reply • Share >



Satrio • 4 months ago

This is a great tutorial. I have a problem in this particular line of code "tweets['text'] = map(lambda tweet: tweet['text'], tweets_data)". Python gave this warning: "Cannot set a frame with no defined index and a value that cannot be converted to a Series". I'm not sure what's happening. Can you give me some explanation? Thanks

^ | v • Reply • Share >



Adil Moujahid Mod → Satrio • 4 months ago

I think you're using an older version of Pandas library. Upgrade to the latest version of Pandas and try again.

^ | v • Reply • Share >



disqus_Uheal01PgF → Adil Moujahid • 2 months ago

Hi I am facing the same issue with the latest pandas. Is there something that could have changed in pandas that might be causing this?

^ | v • Reply • Share >



Shreyas Arur • 4 months ago

Hey Adil this is a great tutorial thank you so much. I just have one question: Is there a way for me to get all tweets in a particular region? Here you have retrieved only the tweets which have the keywords python, javascript and ruby. I want to get all the tweets regardless of topic.

^ | v • Reply • Share >



Adil Moujahid Mod → Shreyas Arur • 4 months ago