Adil Moujahid

Follow @AdilMouja

Published

Mon 21 July 2014

←Home

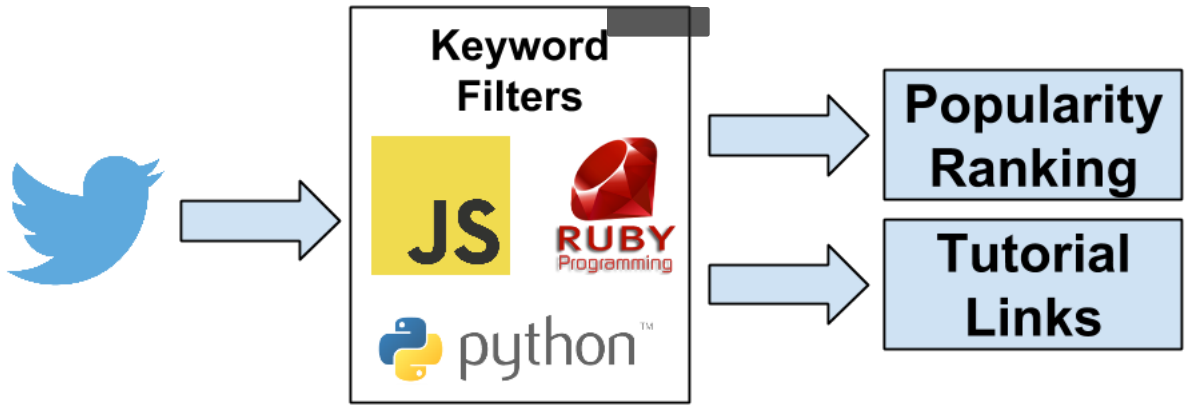
An Introduction to Text Mining using Twitter Streaming API and Python

// tags python pandas text mining matplotlib twitter api

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 explaing 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"

Go Top

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 authetification 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.

You can stop the program by pressing Ctrl-C.

Go Top

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`.

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_t
ranslator":false,"is_translation_enabled":false,"profile_background_color":"C0DEED","profile_background_image_url":"http:\/\/abs.twimg.com\/images\/themes\/theme1\/b
g.png","profile_background_image_url_https":"https:\/\/abs.twimg.com\/images\/themes\/theme1\/bg.png","profile_background_tile":false,"profile_image_url":"http:\/\/p
bs.twimg.com\/profile_images\/425313048320958464\/Z2GcderW_normal.jpeg","profile_image_url_https":"https:\/\/pbs.twimg.com\/profile_images\/425313048320958464\/Z2Gcd
erW_normal.jpeg","profile_link_color":"0084B4","profile_sidebar_border_color":"C0DEED","profile_sidebar_fill_color":"DDEEF6","profile_text_color":"333333","profile_u
se_background_image":true,"default_profile":true,"default_profile_image":false,"following":null,"follow_request_sent":null,"notifications":null},"geo":null,"coordina
tes":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, adn `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()
```

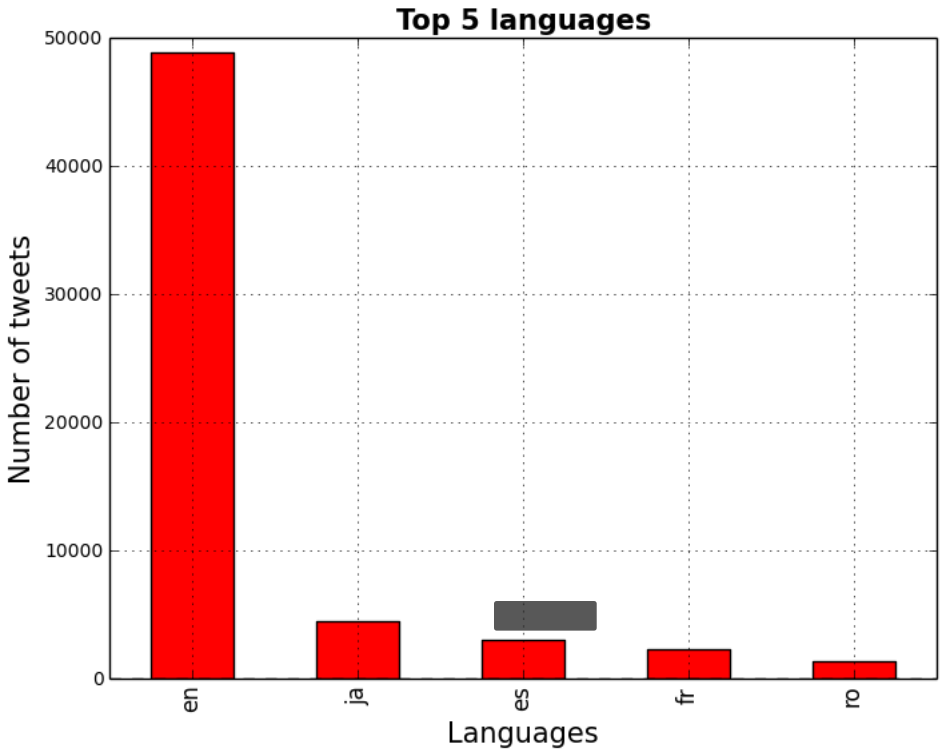
Next, we will add 3 columns to the `tweets` DataFrame called `text`, `lang`, and `country`. `text` column contains the tweet, `lang` column contains the language in which the tweet was written, and `country` the country from which the tweet was sent.

```
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)
```

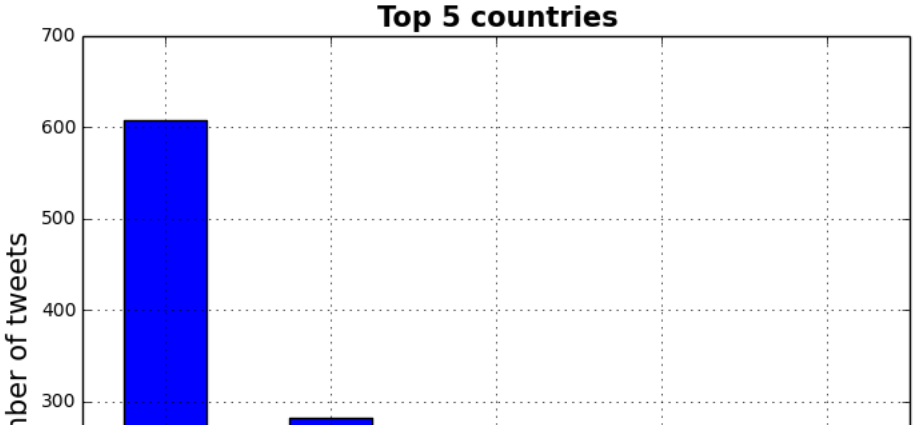
Next, we will create 2 charts: The first one describing the Top 5 languages in which the tweets were written, and the second the Top 5 countries from which the tweets were sent.

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

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

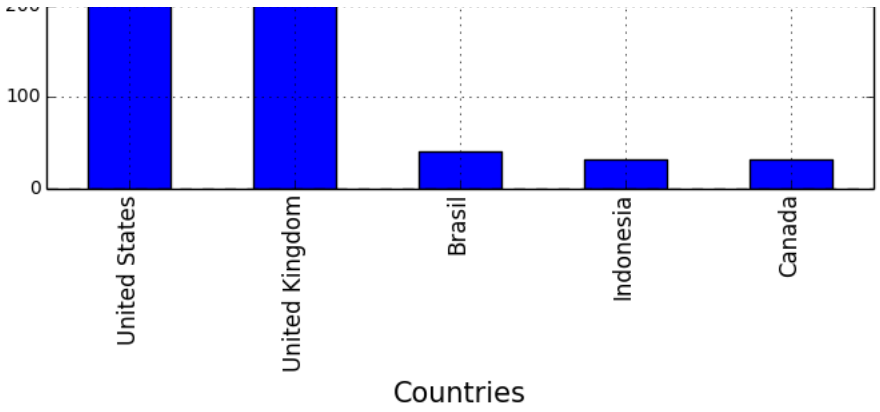
Go Top




```
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 5 countries', fontsize=15, fontweight='bold')
tweets_by_country[:5].plot(ax=ax, kind='bar', color='blue')
```



Go Top



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 manipualte the data easily.
- Target tweets that have "pogramming" or "tutorial" keywords.
- Extract links from the relevants 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 return `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))
```

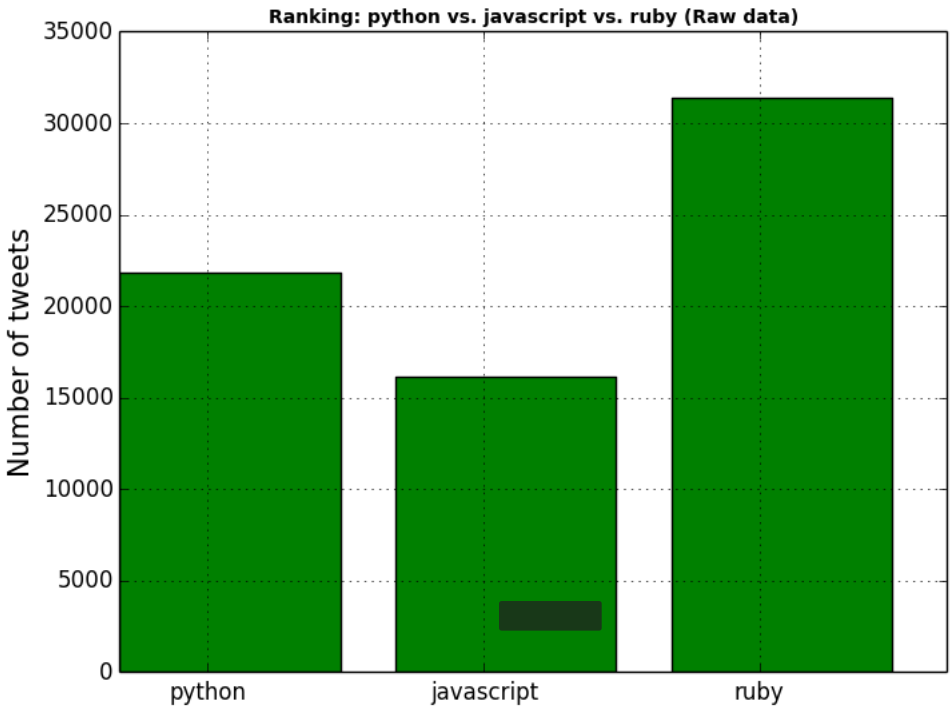
We can calculate the number of tweets for each programming language as follows:

```
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 comparaison 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()
```

Go Top



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 intersted in targetting 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 take 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]
```

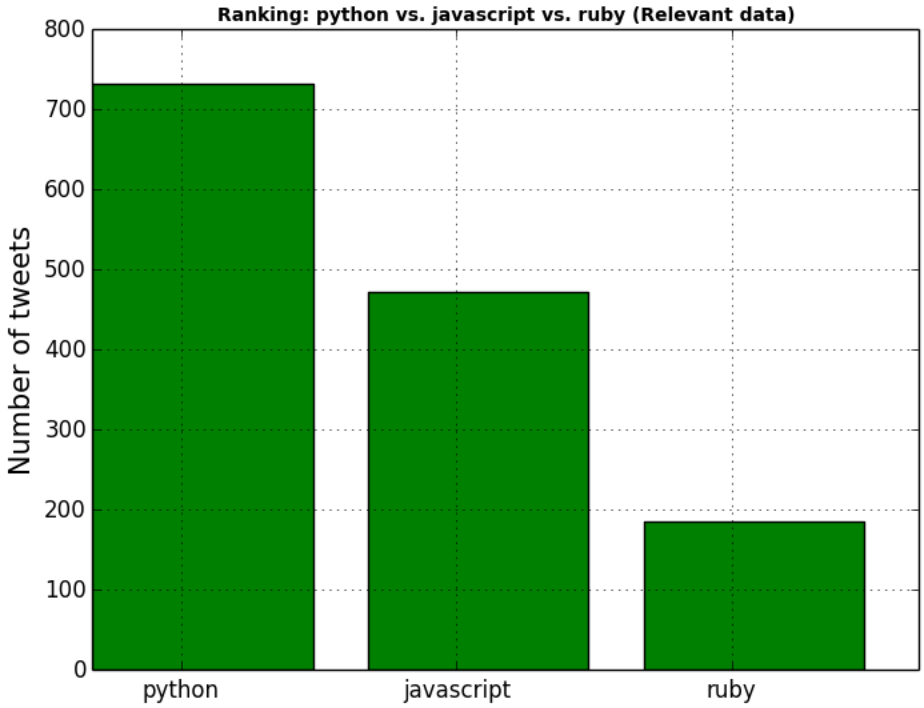
Go Top

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

```
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 comparaison 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 relevants tweets

Ashwin Murali • 3 years 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!

13 • Reply • Share ›

Adil Moujahid Mod → Ashwin Murali • 3 years 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 • 3 years ago

This is awesome! Thank you!!

Dave Roma • 3 years ago

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

6 ^ | v • Reply • Share ›

Hussein Ghaly • 3 years ago

Great Tutorial, thanks Adil!

6 ^ | v • Reply • Share ›

RTD • a year ago

This is a great tutorial for streaming with Python. The code works well if you have python 2.7. However some code changes need to be made when working with Python 3.5. In Python 3+, many processes that iterate over iterables return iterators themselves. Here is a solution if you are getting map object error

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))

Hope this helps.

4 ^ | v • Reply • Share ›

Kara S ➚ RTD • a month ago

Thank you.

^ | v • Reply • Share ›

Adam Hughes • 3 years ago

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

3 ^ | v • Reply • Share ›

Yolandi Chia • 3 years 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!!

3 ^ | v • Reply • Share ›

bobby mcmuffin • 3 years ago

i think you forgot:
import matplotlib.pyplot as plt

3 ^ | v • Reply • Share ›

Adil Moujahid Mod ➚ bobby mcmuffin • 3 years ago

It's corrected now. Thanks !!

^ | v • Reply • Share ›

Sonja • 2 years ago

Hi, Adil, Thanks very much for the tutorial, I got //tweets['text'] = map(lambda tweet: tweet['text'], tweets_data)

KeyError: 'text'// error at first and solved it like you suggested but then I got error //tweets['country'] = map(lambda tweet: tweet['place']['country'] if tweet['place'] != None else None, tweets_data)

KeyError: 'place'//.

2 ^ | v • Reply • Share ›

macwanjason ➚ Sonja • 2 years ago

You usually run across the KeyError when Python cannot find a specified key. This is often the case with JSON generated by the Twitter API that certain fields/keys will not be present for some tweets.

Instead of :

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

Replace this with:

Go Top

Similarly, say you are looking for a key that is nested two or more levels deep, you can chain multiple .get() functions like below.
tweets['child'] = map(lambda tweet: tweet.get('grandparent', {}).get('parent', {}).get('child') , tweets_data)

A more specific example:
tweets['user'] = map(lambda tweet: tweet.get('user', {}).get('name'),tweets_data)
1 • Reply • Share ›

maggieB ➚ macwanjason • 2 months ago
I am using python 3.4
when I do the same exact thing I get this error.
Traceback (most recent call last):
File "C:/Users/user/Desktop/TESTING.py", line 63, in <module>
tweets['text'] = list(map(lambda tweet: tweet.get('text', None),df2))
File "C:/Users/user/Desktop/TESTING.py", line 63, in <lambda>
tweets['text'] = list(map(lambda tweet: tweet.get('text', None),df2))
AttributeError: 'str' object has no attribute 'get'
I need some help please
• Reply • Share ›

Cedric Oeldorf ➚ Sonja • 2 years ago
Hi Sonja,

did you find a solution to the KeyError: 'place' in the end?
• Reply • Share ›

pandi meena • a year ago
hi adil, i want to know about what are the algorithms used in this code or projects....it is urgent
1 • Reply • Share ›

Yogesh Kamble • 3 years ago
This is very nice and neat tutorial. One can start learning data mining from your blog. Thank you very much.
1 • Reply • Share ›

Amar Parkash • 3 years ago
Really nice !! ...thanks a lot
1 • Reply • Share ›

Maarten Keulemans • 3 years ago
Hi Adil, thanks man! Wonderful tutorial!
1 • Reply • Share ›

Ahmed BESBES • 2 years ago
Once again, this is an amazingly written tutorial . Thank you very much .
Please keep on this good work .
It would be awesome to have some future tutorials about some machine learning applications using Python.
1 • Reply • Share ›

Adil Moujahid Mod ➚ Ahmed BESBES • 2 years ago
Glad you liked it! I'm thinking of writing some machine learning tutorials. Email me if you have an interesting dataset in mind. I might use it to build a machine learning use case around it.
• Reply • Share ›

Ahmed BESBES ➚ Adil Moujahid • 2 years ago
I don't have any dataset in mind right now. but , it would be cool for example to reproduce a use case given in Kaggle
• Reply • Share ›

King King Ofhearts • 14 days ago
i got everything right but my graphs don't pop. why ? please, can anybody help me?
• Reply • Share ›

Nisy Maile • a month ago
Hi, Just wondering if there's any way i can extract data for sentiment analysis in a period of five years from twitter
• Reply • Share ›

Go Top

niloufar zamani • 2 months ago

text lang
0 <map object="" at="" 0x00000058d795b668=""> <map object="" at="" 0x00000058d795bd68="">
1 <map object="" at="" 0x00000058d795b668=""> <map object="" at="" 0x00000058d795bd68="">

i would appreciate if you could help me with this problem.
thanks in advance.

Reply • Share ›

Lorenzo Romani • 2 months ago

Hi. I am using Anaconda with Python 3.6. I get stuck at this point:

```
>>> tweets = pd.DataFrame()
>>> 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)
>>> 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)
<matplotlib.text.text object="" at="" 0x0000028b48e9f518="">
>>> ax.set_ylabel('Number of tweets' , fontsize=15)
<matplotlib.text.text object="" at="" 0x0000028b49029be0="">
>>> ax.set_title('Top 5 languages', fontsize=15, fontweight='bold')
<matplotlib.text.text object="" at="" 0x0000028b493476d8="">
>>> tweets_by_lang[:5].plot(ax=ax, kind='bar', color='red')
<matplotlib.axes._subplots.axessubplot object="" at="" 0x0000028b48e92208="">
```

the graph does not pop up and I get this matplotlib messages.

please, can anybody help me?

Reply • Share ›

Shafiq Ahmed • 2 months ago

Hi Adil,

How do I get Tweets for any company? I work for an international Customer care center and have multiple clients. Basically I want to show our clients about what their customers are talking on social media. For this I basically need to know how to extract tweets. I have created API and secret code. Please help!

Reply • Share ›

Mahesh Kumar • 3 months ago

hello, i found this tutorial very useful.. thx for all the information provided.

I have extracted tweets and stored it in a text file. But when i try to count the number of tweets that are present in the file using the segment from ur code i always get the answer as 0.. though the file is not empty.. Why is this happening ? i have attached the code segment here

```
import json
import pandas as pd
import matplotlib.pyplot as plt
tweets_data_path= '/home/hduser/twfilegen1.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
print (len(tweets_data))
```

Any help will be appreciated. thanks

Reply • Share ›

prerna • 3 months ago

what is the duration of the data extracted ??

Reply • Share ›

Go Top

N thanks for this useful post !!

Reply • Share ›

Sajid Samsad • 5 months ago

Hi,
what if apart from reading tweets from .txt file I fetch it from mongod instance.
Then what would happen to this code:
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

Reply • Share ›

Ayse • 5 months ago

Hi Adil,

nice article!

I have problems in the line:
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

it seems that lines aren't loading proper and is throwing exceptions and then stopping, I would really appreciate if you can help me further.

Reply • Share ›

Ayse • 5 months ago

Adil,

Thank you for that great article but I have problems with that code: It seems that tweet is just getting one single line

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

Reply • Share ›

Sahar Nasiri • 5 months ago

Adil,

Thank you for the great tutorial. It helps a lot.

I have an error which I have know idea how to handle, would please help me?

This is the error:

File "C:\Python27\lib\threading.py", line 530, in __bootstrap_inner
self.run()
File "C:\Python27\lib\threading.py", line 483, in run
self.__target(*self.__args, **self.__kwargs)
File "C:\Python27\lib\site-packages\tweepy\streaming.py", line 294, in _run
raise exception
ConnectionError: HTTPSConnectionPool(host='stream.twitter.com', port=443): Max retries exceeded with url: /1.1/statuses/filter.json?

Go Top

the actively refused it ;)))

Reply • Share ›

PRITI SHARMA • 5 months ago

Hi Adil
Is there any way to extract all hashtags and their tweets for a certain period of time?

Reply • Share ›

Chandresh Maurya • 6 months ago

Nice post. One suggestion: Clarify the meaning of each line of the first code.

Reply • Share ›

Mathurin • 7 months ago

How do you get past the ssl problems with this on windows? TweepError: Failed to send request: ("bad handshake: Error([('SSL routines', 'ssl3_get_server_certificate', 'certificate verify failed')],)",)

Reply • Share ›

niranjan deshpande • 7 months ago

Hi,

I am getting error in json.loads command

" return self.scan_once(s, idx=_w(s, idx).end())

JSONDecodeError: Expecting value"

PS: i am using python 3.5

could please help me resolve this issue

Reply • Share ›

Mahendra Tipale • 8 months ago

Nice article. One can directly dump data output to mongodb or elasticsearch for analysis. Its nice tutorial to start with twitter streaming! Thanks.

Reply • Share ›

Amar • 9 months ago

Hi Adil,
Great explanation.
I am using Anaconda 3 and I was successful in streaming live tweets for your example of python, jana script and ruby. However, the code refused to store the stream into 'twitter_data.txt' file. This file was not created and because of which i coudnt work on rest of the code.
Can you please explain why this is happening? I have set path and directory. The error is in the path though.

Reply • Share ›

Soumia Mk • 9 months ago

thanks Adil for this tutorial
I do like that but about printers
so
I have downloading the data (12.6 MB) and the number of tweets = 2689
Now :

```
#Adding Printers columns to the tweets DataFrame
print 'Adding Printers tags to the data\n'
tweets['Hp'] = tweets['text'].apply(lambda tweet: word_in_text('Hp', tweet))
tweets['epson'] = tweets['text'].apply(lambda tweet: word_in_text('epson', tweet))
tweets['Kyocera'] = tweets['text'].apply(lambda tweet: word_in_text('Kyocera', tweet))
tweets['Solidoodle'] = tweets['text'].apply(lambda tweet: word_in_text('Solidoodle', tweet))
tweets['Solid Scape'] = tweets['text'].apply(lambda tweet: word_in_text('Solid Scape', tweet))
```

like that for exemple
and I have written all whats u do about :

```
#Analyzing Tweets by prrinters: First attempt
#Analyzing Tweets by Language
#Analyzing Tweets by Country
```

but I have this probleme:
TypeError: Empty 'DataFrame': no numeric data to plot??????????????????

Reply • Share ›

Go Top

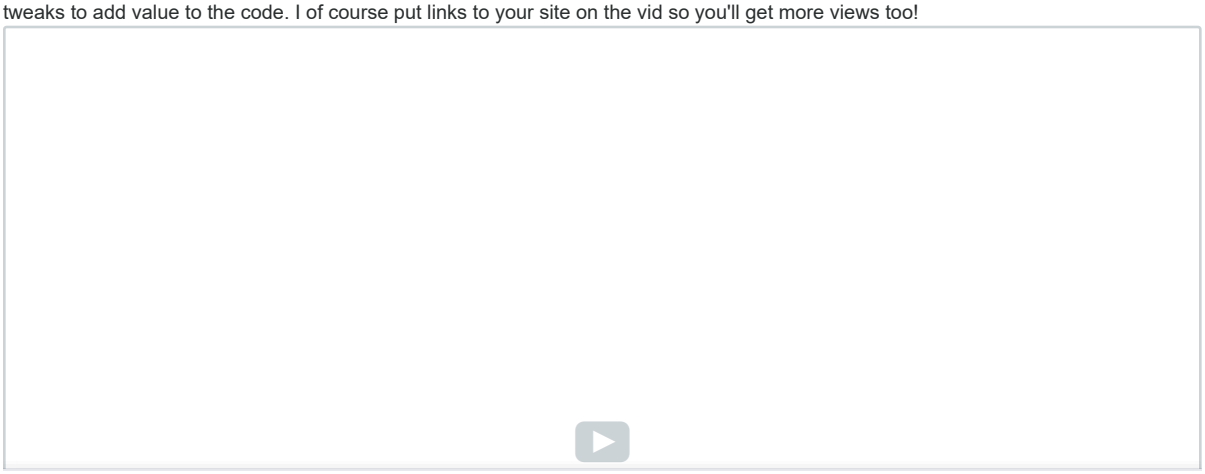
Ioannis Thibaios • 10 months ago

which is the process i should do?

Reply • Share ›

Hid • 10 months ago

Fantastic post! So useful that I set up my own and made a follow up video explaining how I set mine up, plus a few additional details / tweaks to add value to the code. I of course put links to your site on the vid so you'll get more views too!



see more

Reply • Share ›

Jay • a year ago

Hi, thank you very much for such a great post. I just started applying it into my research with Twitter data over the presidential debate.

By the way, do you know how to filter out emojis/emoticons in collected tweets ? and only analyze emojis/emoticons associate with certain keywords or hashtags?

Thanks.

Reply • Share ›

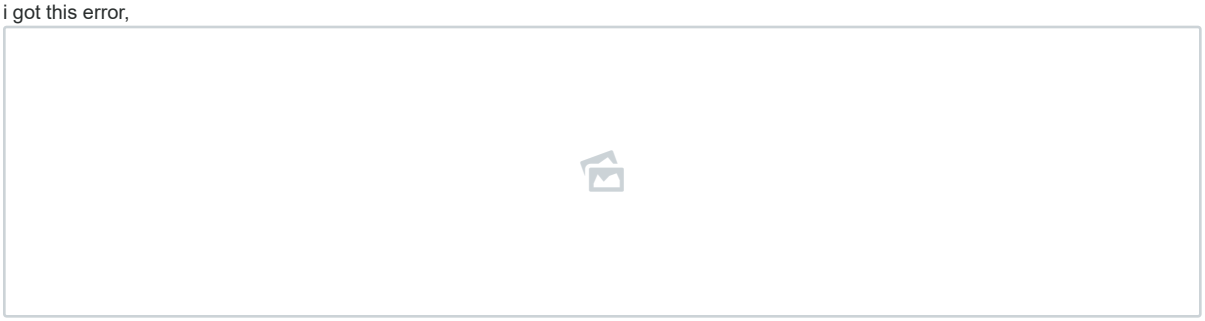
aparna • a year ago

it was very informative thank u for it. i would like to know about how can i get all the tweets of a particular trend and store it in a txt file thank you

Reply • Share ›

Andes • a year ago

i got this error,



it said AttributeError: 'map' object has no attribute 'lower'
my code was downloaded from your github.
i'm using python 3.5 i have downloaded pandas & matplotlib

Reply • Share ›

Branko Markovic8 ➚ Andes • 4 months ago

Hello! I stumbled upon this great example of using Python/Twitter combination. I get the exact same error like you, but still cannot find solution. I see that you manage to fix. How did you do it? I'm Python 3.4. user. Any link where this example could be downloaded or see in total? Thank you, BM.

Reply • Share ›

Blake Porter ➚ Andes • a year ago

I got that error if the text was None. I was able to get it to work by doing this.

Go Top

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

• Reply • Share ›

Nathan Prows ➚ Blake Porter • 8 months ago

thanks that fixed the issue I was having too

1 • Reply • Share ›

Andes ➚ Blake Porter • 7 months ago

thank you

• Reply • Share ›

Load more comments

ALSO ON **ADILMOUJAHID.COM**

Hacking Education with Python - Data Mining Coursera for Popular Courses

10 comments • 2 years ago•

Avatar **Dan Luba** — General Helpfulness Factor: 3000. Thank you.

Interactive Data Visualization with D3.js, DC.js, Python, and MongoDB

140 comments • 3 years ago•

Avatar **Isadora Almeida** — Fixed it!I had to manually put all the 44 fields from the csv to the headerline

Hello World! // Adil Moujahid // Data Analytics and more

2 comments • 3 years ago•

Avatar **hasna chalabi** — Hi Adile and congratulations for youi have to stream tweets from specific country what is the code should I do in python please? because i want later shown it in the …

Baseball Analytics: An Introduction to Sabermetrics using Python // Adil Moujahid // Data Analytics and more

4 comments • 3 years ago•

Avatar **AJ** — This post is so excellent. Thank you and I can't wait to dig into the stats!

Subscribe **Add Disqus to your site** **Add Disqus** **Add** **Privacy**

Go Top