Adil Moujahid

Published

Mon 21 July 2014

←Home

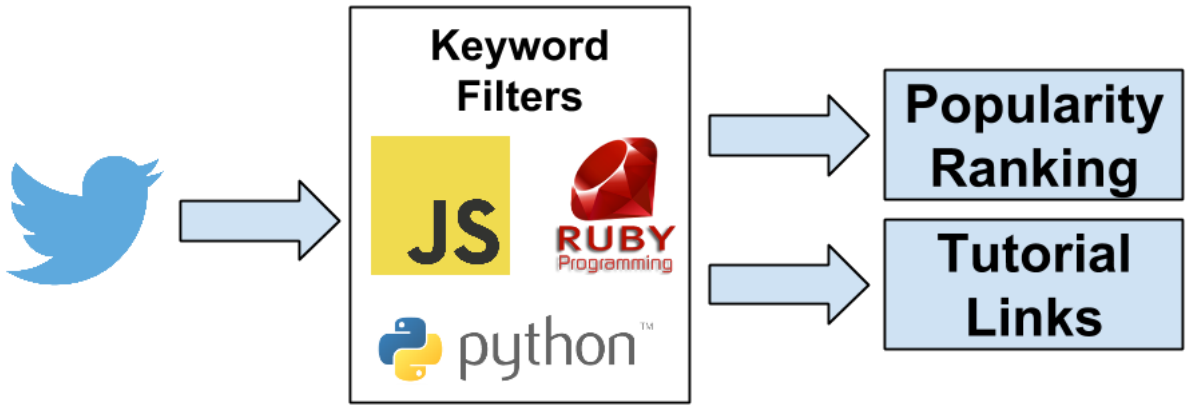
ı Introduction to Text Mining using Twitter Streaming API and Python

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"
- 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

Go Top

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

390 Shares
riables that contains the user credentials to access Twitter API
ess_token = "ENTER YOUR ACCESS TOKEN"
ess_token_secret = "ENTER YOUR ACCESS TOKEN SECRET"
sumer_key = "ENTER YOUR API KEY"
sumer_secret = "ENTER YOUR API SECRET"
167

is is a basic listener that just prints received tweets to stdout.
ss StdOutListener(StreamListener):
132
    def on_data(self, data):
        print data
        return True
30
    def on_error(self, status):
        print status


__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.

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"`.

Go Top

{"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"}

he remaining of this tutorial, we will be using 4 Python libraries `json` for parsing the data, `pandas` for data manipulation, `matplotlib` for creating ts, adn `re` for regular expressions. The `json` and `re` libraries are installed by default in Python. You should install `pandas` and `matplotlib` if you have them in your machine.

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

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

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

```
ets_data_path = '../data/twitter_data.txt'

ets_data = []
ets_file = open(tweets_data_path, "r")
or 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.

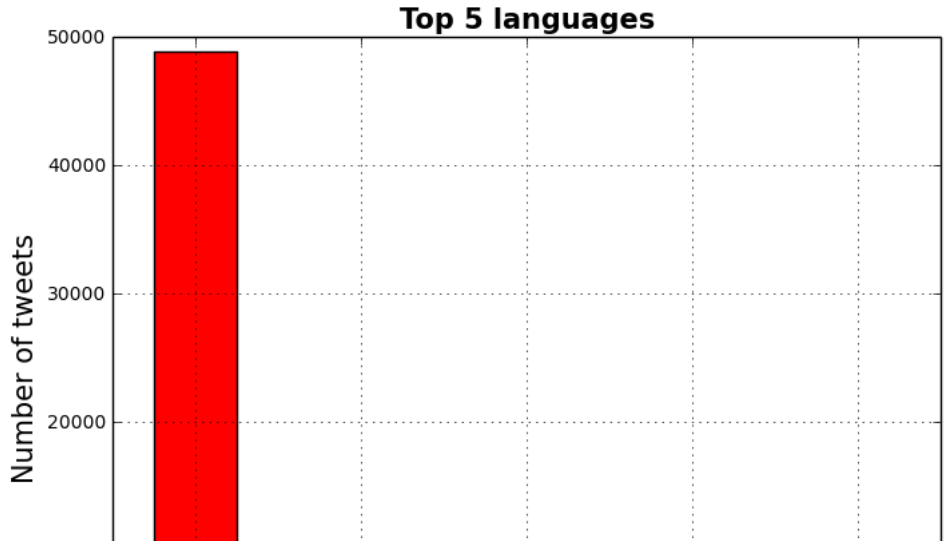
```
tweets = pd.DataFrame()
```

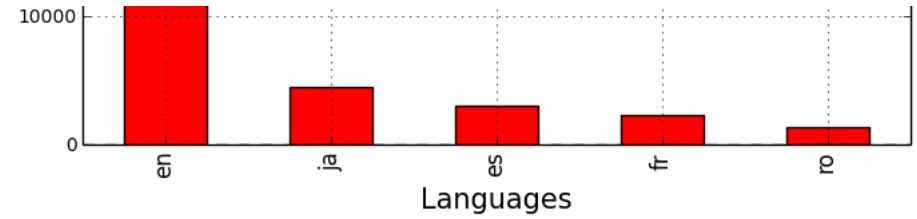
Next, we will add 3 columns to the `tweets` DataFrame called `text`, `lang`, and `country`. `text` column contains the tweet, `lang` column contains the language in which the tweet was written, and `country` the country from which the tweet was sent.

```
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)
```

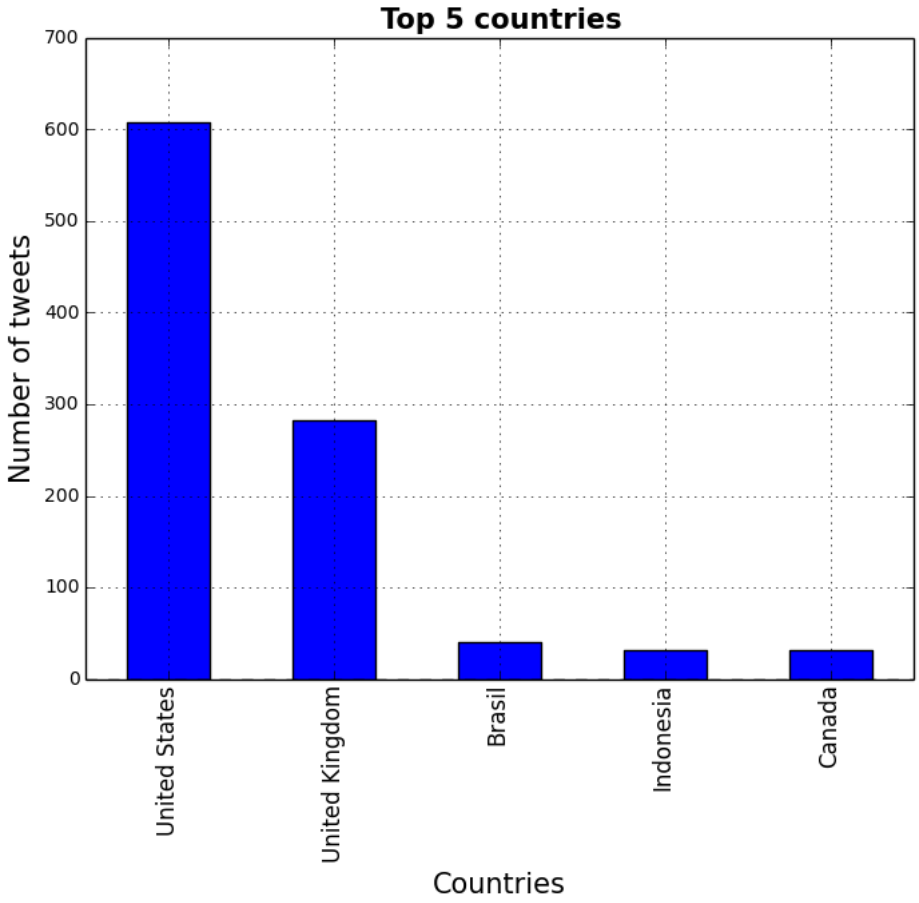
Next, we will create 2 charts: The first one describing the Top 5 languages in which the tweets were written, and the second the Top 5 countries from which the tweets were sent.

```
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')
```






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

, ax = plt.subplots()
tick_params(axis='x', labelsize=15)
tick_params(axis='y', labelsize=10)
set_xlabel('Countries', fontsize=15)
set_ylabel('Number of tweets' , fontsize=15)
set_title('Top 5 countries', fontsize=15, fontweight='bold')
ets_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 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
```

Go Top

```
    return False
```

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

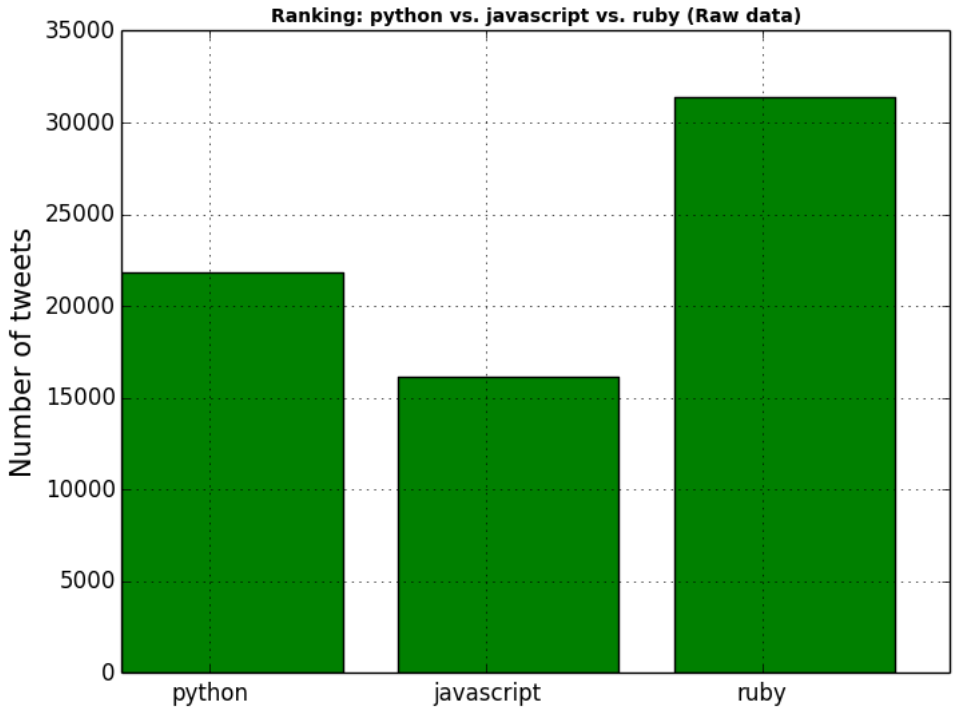
```
tweets['python'] = tweets['text'].apply(lambda tweet: word_in_text('python', tweet))
tweets['javascript'] = tweets['text'].apply(lambda tweet: word_in_text('javascript', tweet))
tweets['ruby'] = tweets['text'].apply(lambda tweet: word_in_text('ruby', tweet))
```


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

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


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


```
;_langs = ['python', 'javascript', 'ruby']
:ets_by_prg_lang = [tweets['python'].value_counts()[True], tweets['javascript'].value_counts()[True], tweets['ruby'].value_counts()[True]]

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

;etting axis labels and ticks
 set_ylabel('Number of tweets', fontsize=15)
 set_title('Ranking: python vs. javascript vs. ruby (Raw data)', fontsize=10, fontweight='bold')
 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 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.

Go Top

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

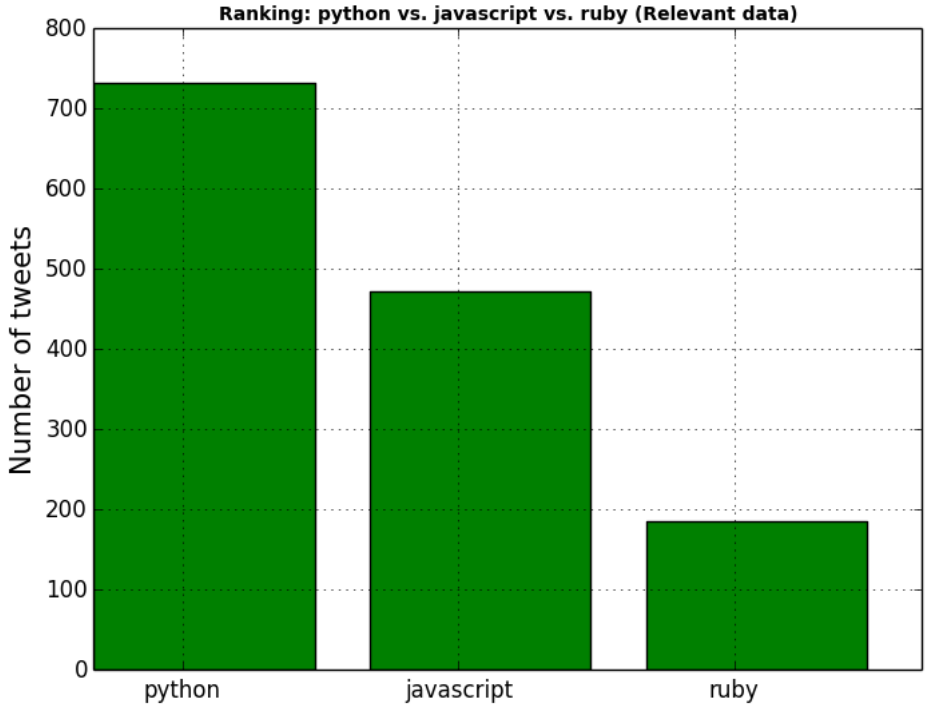
This returns, 871 for `programming` column, 511 for `tutorial` column, and 1356 for `relevant` column.

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]
```


on 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 executing the commands below:

```
ets_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]]
os = list(range(len(prg_langs)))
dth = 0.8
g, ax = plt.subplots()
t.bar(x_pos, tweets_by_prg_lang, width,alpha=1,color='g')
 set_ylabel('Number of tweets', fontsize=15)
 set_title('Ranking: python vs. javascript vs. ruby (Relevant data)', fontsize=10, fontweight='bold')
 set_xticks([p + 0.4 * width for p in x_pos])
 set_xticklabels(prg_langs)
t.grid()
```



Extracting links from the relevants 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']
```

Go Top

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/EnK2UlDcJ8
- 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

onclusion

is 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 ited to another use case. For those who want to go further in text mining, I recommend to follow up by studying regular expressions.

can find the source code from this tutorial in this github repository github link.

ferences

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

Yolandi Chia • 2 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 ›

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 • a year 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:

tweets['text'] = map(lambda tweet: tweet.get('text', None),tweets_data)

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 ^ | v • Reply • Share ›

Cedric Oeldorf ➚ Sonja • a year ago

Hi Sonja,

did you find a solution to the KeyError: 'place' in the end?

^ | v • Reply • Share ›

Adam Hughes • 2 years 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 ›

bobby mcmuffin • 3 years ago

i think you forgot:
import matplotlib.pyplot as plt

2 ^ | v • Reply • Share ›

Adil Moujahid Mod ➚ bobby mcmuffin • 3 years ago

It's corrected now. Thanks !!

^ | v • Reply • Share ›

RTD • 10 months 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.

1 ^ | v • Reply • Share ›

Yogesh Kamble • 2 years 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 ›

Go Top

Amar Parkash • 2 years ago
Really nice !! ...thanks a lot
1 • Reply • Share ›

Maarten Keulemans • 2 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 ›

Mathurin • 21 days 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 • 24 days 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 • a month 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 • 2 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 • 2 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:

Go Top

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

Reply • Share ›

Ioannis Thibaios • 4 months ago

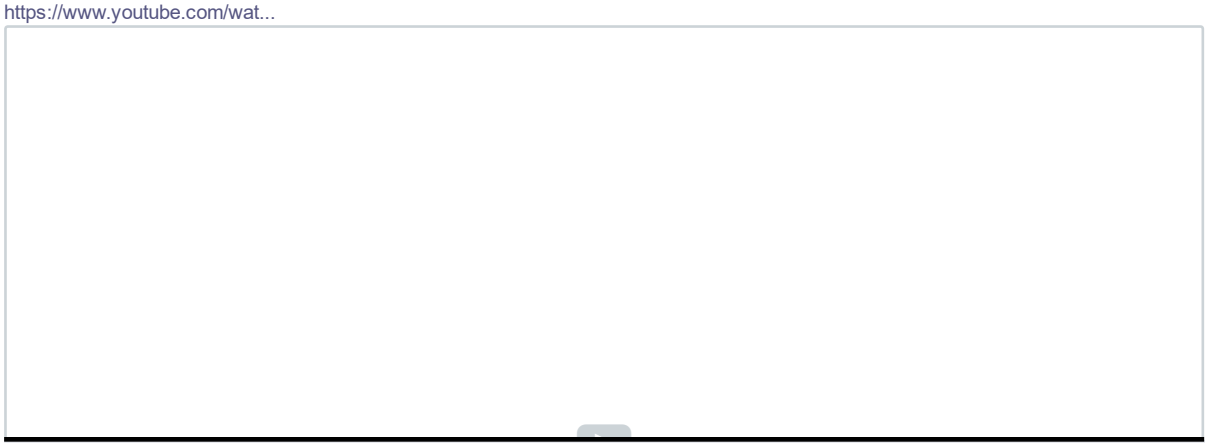
Hi Adil,

Thank you for this post. I want to ask, which algorithm is used in this project? Also, if i want to filter the tweets related to certain dates, which is the process i should do?

Reply • Share ›

Hid • 4 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!
https://www.youtube.com/wat...



see more

Reply • Share ›

Jay • 4 months 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 ›

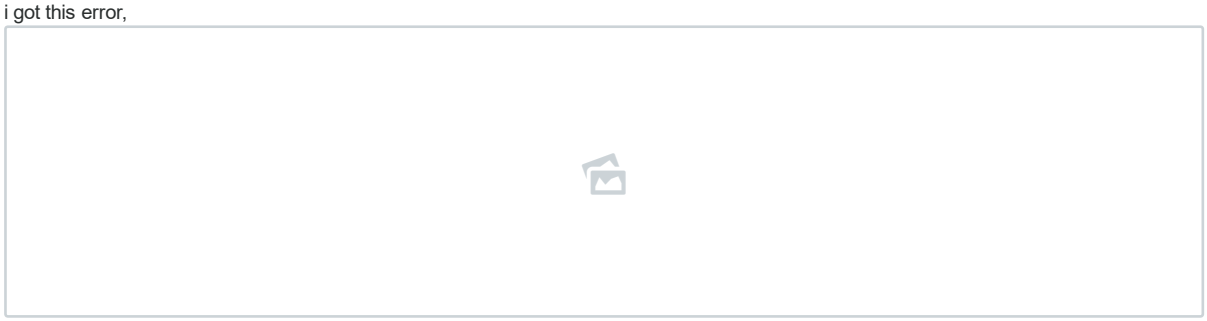
aparna • 5 months ago

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 • 5 months 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 ›

Blake Porter ➚ Andes • 4 months ago

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

```
def word_in_text(word,text):
if text == None:
return False
word = word.lower()
text = text.lower()
match = re.search(word,text)
```

Go Top

if match:
return True
else:
return False

• Reply • Share ›

Nathan Prows ➚ Blake Porter • a month ago
thanks that fixed the issue I was having too
1 • Reply • Share ›

Andes ➚ Blake Porter • 5 days ago
thank you
• Reply • Share ›

custom writing reviews • 6 months ago
That is why text mining gets lots of attention nowadays especially that people could get relevant ideas and other important information that would help them with their experience in using twitter and be able to dig up tweet counts that you needed to know about.
• Reply • Share ›

Ken Lin • 6 months ago
Hi,

You mentioned that the total tweet count is 71238. And after calculating the instance of each keywords, the results are 21839 for python, 16154 for javascript and 31410 for ruby.

21839 + 16154 + 31410 = 69403, which doesn't add up to 71238. So what happened to the rest of the the tweets? When I tried the tutorial myself, my numbers also don't add quite add up to the total tweet count.
• Reply • Share ›

Furkan Öyken ➚ Ken Lin • 4 months ago
Hi,

Maybe some of the tweets contains more than one tag and the streamlistener catches the tweet more than once for example;

"I believe #python and #ruby can work well together."

So streamlistener catches the tweet for python and ruby seperately and adds to the dataset.

I actually have no idea about how it is working. This is just my best guess.
• Reply • Share ›

Kush • 7 months ago
Hi, its a great post. You are looking for particular word like "python","javascript'. I have made a list with the words which are similar to python, java also. Now i want to comapare this list to my tweets. How can i do that? I have tried this
for i in len(python_similar_words):
tweets['sports_keeda']= tweets['clean_text'].apply(lambda tweet: check_word_in_text(python_similar_words[i],tweet))

But its not predicting right output. I mean if there is a tweet having similar words of python_similar_words,its giving false
• Reply • Share ›

Chirag Khandelwal • 7 months ago
Hi Adil,

I have tried the script and it working fine for me.. the only problem is script stops running after 2 Hours. can you please help me to setup things so that script will run continue until I stop it.
I am running script script on Ubuntu 14.04 server.

Thanks
• Reply • Share ›

Jhonatas Kleinkauff • 7 months ago
Good work Adil! Thanks!
• Reply • Share ›

pandi meena • 7 months ago
hi adil, i want to know about what are the algorithms used in this code or projects....it is urgent
• Reply • Share ›

Rahul Shukla • 7 months ago
Hello ! After starting my script, I am getting the following error after like 10 minutes every time:

Exception in thread Thread-1:
Traceback (most recent call last):
File "/usr/local/Cellar/python/2.7.11/Frameworks/Python.framework/Versions/2.7/lib/python2.7/threading.py", line 801, in __bootstrap_inner
self.run()

Go Top

File "/usr/local/Cellar/python/2.7.11/Frameworks/Python.framework/Versions/2.7/lib/python2.7/threading.py", line 754, in run
self.__target(*self.__args, **self.__kwargs)
File "/usr/local/lib/python2.7/site-packages/tweepy/streaming.py", line 294, in _run
raise exception
ProtocolError: ('Connection broken: IncompleteRead(0 bytes read, 512 more expected)', IncompleteRead(0 bytes read, 512 more expected))

Any help would be of great use !

Reply • Share ›

D8amonk • 8 months ago

167

Excellent tutorial, Adil. Thank you very much for your time and effort in completing this.

132

I'd really appreciate any knowledge you can share on how to pull for a specific date range. As I speak, the script is feeding the _stream file into the .txt for preparation and analysis, but I can't see what range is being pulled; therefore, I don't know when to stop. Is there a way to do this, and/or build-in a "stop after X number of JSON objects are returned" functionality?

30

Thanks again!

Reply • Share ›

Adil Moujahid Mod ➚ D8amonk • 8 months ago

Hi D8amonk,
You can do so by adding your condition to the StdOutListener class. Below is the source code for getting 100 tweets from the streaming api.

```
MAX_NUM_TWEETS = 100
class StdOutListener(StreamListener):
def __init__(self):
self.count = 0

def on_data(self, data):
self.count += 1
print data
if self.count > MAX_NUM_TWEETS:
return False
return True

def on_error(self, status):
print status
```

1 Reply • Share ›

Jake Steele • 8 months ago

very useful! Good for the ground level. Wish there was more stuff out there like this

Reply • Share ›

kaan • 8 months ago

hi, i can't plot the histograms why is that??? after i run the code it just prints number of tweets in my file and returned exit...
pls help!!!!

Reply • Share ›

CMHealingAngel ➚ kaan • 3 months ago

use the below at the end of the plotting code block

```
plt.show()

For Multiple plots or figure windows use
fig1 = plt.figure(1)
# plotting codes goes here
plt.figure(1).show()

fig2 = plt.figure(2)
# plotting codes goes here
plt.figure(2).show()
```

Reply • Share ›

Jonathan Yaniv • 8 months ago

Amazing.
I'm trying to do something similar but with endless data.
Using Amazon kinesis

If you know something about it I'd be glad to have such an example too

Reply • Share ›

abbey • 10 months ago

Hi Adil,

Thank you so much for this wonderful post.

Go Top

Now I wanted to extract the text of the tweets based on the "id","text","lang=en" and "location" from the JSON file.
Can give any idea regarding this.

• Reply • Share ›

paridhi soni • a year ago

whr r the graph supposed to be displayed??

• Reply • Share ›

Ashutosh Gaurav • a year ago

167

Hi Adil

Great Post. Now I wanted to extract the text of the tweets based on the location/coordinate of the tweets (e.g. all the text of the tweet

132

tweeted from New Delhi containing words "Python, Ruby or JavaScript". Can give any idea regarding this.

• Reply • Share ›

30

abbey ➔ Ashutosh Gaurav • 10 months ago

Hi Ashutosh,
Did you get help to your question. Pls can you shared it.

Thanks

• Reply • Share ›

Load more comments

Go Top