tweet_collector

September 24, 2019

1 Let's collect tweets!

Run these cells after each other, and the collection of tweets will automatically continue from the last saved tweet.

1.0.1 Parameters

[1]:
```
# Parameters for saving tweets
tweet_per_file = 1000
max_n_files = 100
dir_path = '../data/tweets_NY_150km'

# Parameters of the query (tweepy API.search())
q           ='*'
geocode     ='40.7128,74.0060,150km'
tweet_mode  ='extended'
lang        ='en'
result_type = 'recent'
```

1.1 Imports

[2]:
```
import tweepy
import json
import os
import re

from IPython.display import display, clear_output
```

1.2 Load Twitter credentials, access API

[4]:
```
twitter_credentials = json.load(open('./keys.json', 'r'))['twitter1']
CONSUMER_KEY = twitter_credentials['consumer_key']
CONSUMER_SECRET = twitter_credentials['consumer_secret']
token_key    = twitter_credentials['token_key']
token_secret = twitter_credentials['token_secret']
```

OAuthHandler vs. AppAuthHandler AppAuthHandler is much better for data retrieval. It has a higher rate limit, and even if it reaches the limit, it waits automatically until it can request more data.

[5]:
```
# Authenticate twitter Api
auth = tweepy.AppAuthHandler(CONSUMER_KEY, CONSUMER_SECRET)
api = tweepy.API(auth, wait_on_rate_limit=True, wait_on_rate_limit_notify=True)
if (not api):
    print ("Can't Authenticate")
# auth.set_access_token(token_key, token_secret)
```

1.3 Get the ID of the last saved tweet

[6]:
```
file_names = os.listdir(dir_path)
all_tweets = []
for file_name in file_names:
    file_path = dir_path + '/' + file_name
    with open(file_path, 'r', encoding='utf-8') as file:
        all_tweets.append(json.load(file)[0])

ids = [tweet['id'] for tweet in all_tweets]
last_id = max(ids)
first_id = min(ids)
print('Num. of files :', len(file_names))
print('First ID :', first_id)
print('Last ID  :', last_id)
```

```
Num. of files : 5
First ID : 1172558832842792960
Last ID  : 1176485641116368898
```

1.4 Collect tweets

[7]:
```
i = 0
i_file = len(file_names)



c = tweepy.Cursor(api.search,
                  q = q,
                  geocode = geocode,
                  tweet_mode = tweet_mode,
                  lang = lang,
                  since_id = last_id,
#                  max_id=first_id,
                  result_type = result_type
                 )

tweets = []
```

```
for tweet in c.items():

    #get full text for retweets and normal tweets too
    try:
        text = tweet.retweeted_status.full_text
    except AttributeError:
        text = tweet.full_text

    #save certain attributes (other than text)
    tweets.append(
      {
          'id':tweet.id,
          'text':text,
          'created_at':str(tweet.created_at),
          'author_name':tweet.author.name,
      })

    #save every #tweet_per_file number of tweets to a json
    i += 1
    if i > (tweet_per_file-1):
        with open(dir_path + '/' + 'tweets_{:03d}.json'.format(i_file), 'w',␣
 ↪encoding='utf-8') as file:
            json.dump(tweets, file, ensure_ascii=False, indent=4)
        i_file += 1
        i = 0
        tweets = []

    clear_output(wait=True)
    display('{}/{}'.format(i_file, i))

    if i_file > (max_n_files):
        break
```

```
'5/58'
```