Twitter Scrapper

July 19, 2019

In [ ]:
```
import tweepy
from TwitterAPI import TwitterAPI
import json
import pandas as pd
from datetime import timedelta, date
import time

#shubhendu
consumer_key = "aM13nmKcUaJhgl8KVn25v89ad"
consumer_secret = "7v8R4WNjDEWZuGUvFHzOKbHHe77Vi4JTZIfyfUWJjajGRRSJh0"

access_token = "1150321422834601984-ZDvhH8jdXdeNEZvZN1nfwzcG00qXZg"
access_token_secret = "ysmdZKHCNkhKhLB1QUN3M39kWmaXjuukQVpgKxojMvlCD"

#vkumar.0101@gmail.com
# consumer_key = "eQ8ErZL3vEvZkQvTSKMPndhzb"
# consumer_secret = "ztZJo14QU1wFtREneeQ7HhBUbK2RidqIIHCvXKSZDjEqVMTprv"

# access_token = "115683208-cdY9Yru9aaStNHr9QtA2uLTn4khAqpoL33HvPKgm"
# access_token_secret = "4vheUcufR5nPw9WAYbMs0omnjc97KmYaN2fk3dYQ0O1UO"

#moreanmol@gmail.com
# consumer_key = "fBKRihVu5bNjOypeey39J8v6x"
# consumer_secret = "E4fkBQMCfO9WGg6ZDitWxFRAdWrjIPaL24RTNSpnTXIsfGdO6g"

# access_token = "1146388838534660096-iYJJ8NXMf16sNjOO6jd6HzsjvtjfVp"
# access_token_secret = "r16edV26xmYuxj1wyzOhgZnvqKcYjnXqxaMRcMv8nUk75"

auth = tweepy.OAuthHandler(consumer_key, consumer_secret)
auth.set_access_token(access_token, access_token_secret)

api = tweepy.API(auth)
print(api)
```

In [ ]:
```
import time

#for each response, filter out important attributes for dataframe
```

```
        def process_tweet(tweet) :
            tweet_date = tweet['created_at']
            ts = time.strftime('%Y-%m-%d %H:%M:%S', time.strptime(tweet_date,'%a %b %d %H:%M:%S
            tweet_id = tweet['id']
            tweet_text = tweet['text']
            if('extended_tweet' in tweet.keys()):
                tweet_text = tweet['extended_tweet']['full_text']
            user_id = tweet['user']['id']
            followers_count = tweet['user']['followers_count']
            friends_count = tweet['user']['friends_count']
            user_mentions = tweet['entities']['user_mentions']
            screen_names = [user_mention['screen_name'] for user_mention in user_mentions]
            screen_name = tweet['user']['screen_name']
            retweet_count = tweet['retweet_count']
            favorite_count = tweet['favorite_count']
            retweeted = tweet['retweeted']
            tweet_row = {'date':ts,
                         'tweet_id' : tweet_id,
                         'user_id' : user_id,
                         'followers_count' : followers_count,
                         'friends_count' : friends_count,
                         'user_mentions' : screen_names,
                         'screen_name' : screen_name,
                         'retweet_count' : retweet_count,
                         'favorite_count' : favorite_count,
                         'retweeted' : retweeted,
                         'full_text':tweet_text}
            return tweet_row
```

```
In [ ]: #Ref : https://stackoverflow.com/questions/1060279/iterating-through-a-range-of-dates-
        def daterange(start_date, end_date):
            for n in range(int ((end_date - start_date).days)):
                yield start_date + timedelta(n)
```

Fetch Swiggy and Zomato Tweets. Store raw data under swiggy/json and zomato/json folder. Processed selected data is stored under zomato and swiggy folders

Fetch data for Swiggy

```
In [ ]: #Ref : https://medium.com/@poconnell732/acquiring-free-historical-geo-located-data-fro
        api = TwitterAPI(consumer_key=consumer_key,
                         consumer_secret=consumer_secret,
                         access_token_key=access_token,
                         access_token_secret=access_token_secret)
        # PRODUCT = '30day' #Using 30day API
        # LABEL = '30daysdev' # sandbox name
```

```
PRODUCT = '30day' #use full archive API
LABEL = 'twitter30days'
web_request_count = 0

start_month_date = date(2019, 7, 6)
end_month_date = date(2019, 7, 7)

for date in daterange(start_month_date, end_month_date):
    list_tweets = []
    next_token = {}
    date_str = date.strftime("%Y%m%d")
    start_date = date_str + "0000"
    end_date = date_str + "2359"
    print(start_date)
    print(end_date)

    while ((next_token is not None) and (web_request_count<5)):
        print(next_token)
        if not next_token :
            req_dict = {'query' : '@swiggy_in OR @swiggyCares lang:en', 'fromDate': sta
        else :
            req_dict['next'] = next_token

        web_request_count += 1
        print('web_request_count: ', web_request_count)
        req = api.request('tweets/search/%s/:%s' % (PRODUCT, LABEL), req_dict)
        print('status_code: ', req.status_code)
        response = req.json()
        if('next' in response.keys()):
            next_token = response['next']
        else :
            next_token = None

        #print(response)
        results = response['results']
        with open('data/swiggy/json/' + date_str + '_' + str(web_request_count) + '.jso
            json.dump(results, f)

        #print(results)
        for tweet in results:
            tweet_row = process_tweet(tweet)
            list_tweets.append(tweet_row)
        df = pd.DataFrame(list_tweets)
    df.to_csv('data/swiggy/' + date_str + '.csv', index=False)
    print(df.shape)
```

Fetch data for Zomato

```
In [ ]: api = TwitterAPI(consumer_key=consumer_key,
```

```
                consumer_secret=consumer_secret,
                access_token_key=access_token,
                access_token_secret=access_token_secret)
# PRODUCT = '30day' #Using 30day API
# LABEL = '30daysdev' # sandbox name
# PRODUCT = 'fullarchive'
# LABEL = 'devbox1'
web_request_count = 0

start_month_date = date(2019, 6, 29)
end_month_date = date(2019, 6, 29)

for date in daterange(start_month_date, end_month_date):
    list_tweets = []
    next_token = {}
    date_str = date.strftime("%Y%m%d")
    start_date = date_str + "0000"
    end_date = date_str + "2359"
    print(start_date)
    print(end_date)

    while (next_token is not None) and (web_request_count<5):
        print(next_token)
        if not next_token :
            req_dict = {'query' : '@zomatocare OR @zomatoIn lang:en', 'fromDate': start
        else :
            req_dict['next'] = next_token

        web_request_count += 1
        print('web_request_count: ', web_request_count)
        req = api.request('tweets/search/%s/:%s' % (PRODUCT, LABEL), req_dict)
        print('status_code: ', req.status_code)
        response = req.json()
        if('next' in response.keys()):
            next_token = response['next']
        else :
            next_token = None

        results = response['results']
        with open('data/zomato/json/' + date_str + '_' + str(web_request_count) + '.jso
            json.dump(results, f)

        #print(results)
        for tweet in results:
            tweet_row = process_tweet(tweet)
            list_tweets.append(tweet_row)
        df = pd.DataFrame(list_tweets)
    df.to_csv('data/zomato/' + date_str + '.csv', index=False)
```

```
print(df.shape)
```

In [ ]: