Arhiv družboslovnih podatkov

Gradivo vezano na raziskavo

Slovenska razprava na Twitterju pred volitvami v Evropski parlament 2019

ADP-IDNo: TWITEP19
DOI: https://doi.org/10.17898/ADP_TWITEP19_V1

Uporaba podatkov

(***Hydrating data)***

Uporaba podatkov

Datoteka vsebuje seznam ID številk vseh uporabljenih tvitov oz. "*dehydrated tweets*".

Vsak, ki želi pridobiti bazo podatkov, lahko z uporabo *Twitter API* uporabljene tvite ponovno zajame iz Twitterja (oz. podatke "*hidrira*"). S tem lahko sami ustvarite bazo podatkov, ki je uporabljena v raziskavi.

Za "hidracijo" baze podatkov se lahko uporabijo namenski programi, kot sta na primer *hydrator* (https://github.com/DocNow/hydrator/) ali *twarc* (https://github.com/DocNow/twarc).

Proces "hidriranja" podatkov je natančneje opisan na povezavah:

- https://theneuralblog.com/hydrating-tweet-ids/
- https://programminghistorian.org/en/lessons/beginners-guide-to-twitter-data#hydrating

V nadaljevanju sledi opis postopka hidracije tvitov, povzet iz spletnega dnevnika The Neural Blog (Lamsal, 2021) v originalu.

> ***Hydrating tweet IDs***
>
> *Twitter has become an ultimate source of data for researchers. Its API is relatively straightforward and easy to use. Pulling tweets from Twitter is not as difficult as it seems. The variants of Twitter API lets the researchers access its live feed (streaming API) or search through tweets (search API).*
>
> *However, researchers must follow certain terms and conditions while dealing with the data retrieved from Twitter. Although researchers are allowed to access and pull data from its live feed or search and pull older tweets, they are not allowed to share the raw data with a third party. Only the tweet IDs, user IDs and/or message IDs can be shared publicly. Other parties who want to access the dataset you're sharing must hydrate the IDs in order to re-create a fresh raw dataset. The process of retrieving a tweet's complete information using its ID is known as the hydration of a tweet ID.*
>
> *If you came across a list of tweet IDs on the internet and would like to hydrate those IDs; you'll need to have access to the Twitter API. For this purpose, a Twitter developer account is a must. Go to Twitter's Developers Portal and signup for a dev account. Once your account is approved you can easily use third-party tools to access the Twitter API and hydrate the IDs.*
>
> *There are multiple desktop applications and libraries available for hydrating tweet IDs. You're here reading this article definitely means that you would want a suggestion on this. I personally use the twarc python library to deal with hydration. However, if you are someone who would prefer not to deal with even a single line of code, I'd suggest you use the Hydrator Desktop Application. Both of these handle the rate limits for you.*

"Rate limits?" Well, Twitter has rate limits placed on the number of times you can call its API every "window period." The window period varies depending upon the version/type of API you're using.

How many tweet IDs can you hydrate in a day?

The calculation goes something like this for the standard

`statuses/lookup`
*endpoint (which has a per 15 min window size): the total number of 15 min windows in a day) * the number of requests allowed per window * max number of tweets that can be retrieved in every request. Unless it's a paid developer account, we cannot increase the number of requests.*

The number of requests allowed per window is 900 for

`statuses/lookup`
*endpoint; therefore the endpoint gives around 8,640,000 tweets every 24 hours. Here's the calculation: (1440/15) * 900 * 100. Where 100 is the max number of tweets that can be retrieved in every request.*

Using Hydrator Desktop Application

Head out to the releases section of Hydrator, and download the application based on the machine you have (Windows, Mac, Linux). And install the application.

Once the application is installed, you'll be presented with a request to link your Twitter account so that you can retrieve data from Twitter. Click on "Link Twitter Account." A browser window opens up so that you can authorize the application to access your Twitter account. Once you verify the authorization request on Twitter, you'll be presented with a PIN which should be entered in the field provided by Hydrator so as to complete the authorization.

Once the authorization is completed, visit the "Add" section of the Hydrator.

In the "Add" section, you need to select a tweet ID file. Most of the files you find online with a list of tweet IDs would be TXT or CSV files. Both formats are supported by this application. Go ahead, and add the tweet IDs file you'd want to hydrate. Fill out the "Title", "Creator", ... , fields as appropriate.

Once you've filled out all the required details specific to your dataset, click "Add Dataset". The application will start hydrating the tweet IDs. Once the hydration is finished, you can convert the resulting dataset into a CSV file.

Using twarc python library

The official GitHub repo for twarc is here. If you're comfortable with python, you'll take no more than 2-3 minutes to start hydrating the tweet IDs.

Let's use

`pip` to install the twarc library.
$ pip3 install twarc

If you do not have `pip` already installed in your system, install it first with the command below.

$ sudo apt install python3-pip

Once `twarc` is installed, you're ready to start writing your script for hydrating a list of tweet IDs. Before getting started with the script, let's configure twarc with your Twitter API keys. For this, visit Twitter's Developer portal, create a new application, and get

`CONSUMER_KEY`,
`CONSUMER_SECRET`,
`ACCESS_TOKEN`,
`ACCESS_TOKEN_SECRET` of the newly created application. Once you have all the keys, use the following command to `configure` twarc.
$ twarc configure

Or, instead of configuring the keys from the command line, you can also set the keys while you write your script. Anyway, let's write our very first hydration script.

Import `twarc` and initialize the keys.

```
from twarc import Twarc
consumer_key=""
consumer_secret=""
access_token=""
access_token_secret=""
```

Importing a TXT/CSV file that contains a list of tweet IDs.

```
t = Twarc(consumer_key, consumer_secret, access_token, access_token_secret)
for tweet in t.hydrate(open('tweet_ids.csv')):
```

After this, you'll need to decide what tweet data you'd want to extract from the hydration. In our case let's print

`tweet text`,
`tweet id` and
`location` from where the tweet was probably tweeted (if available; hence the `if` block).

```
print(tweet['text'])
```

print(tweet['id'])
if tweet['place']:
print(tweet['place']['country'])

Now you can play around with Twitter data as per your needs. Suppose you want to know when a tweet was created, then you'd simply query the

`['created_at']`
attribute of the tweet data dictionary. For user's name, you'd query
`['user']['name']`
, for user's location, you'd query
`['user']['location']`
, and so on.

For more information on Twitter objects, visit this official Twitter page.

Here's the complete code:

from twarc import Twarc
consumer_key=""
consumer_secret=""
access_token=""
access_token_secret=""
t = Twarc(consumer_key, consumer_secret, access_token, access_token_secret)
for tweet in t.hydrate(open('tweet_ids.csv')):
print(tweet['text'])
print(tweet['id'])
if tweet['place']:
print(tweet['place']['country'])

Viri

Lamsal, R. (2021). Hydrating tweet IDs. The Neural Blog. https://theneuralblog.com/hydrating-tweet-ids/. Dostopano 7. 4. 2023.