Facebook Profile Social Analyzer

Abdullayev Elnur, Kagramanyan George, Nguyen Steven and Swimberghe Jan

1. INTRODUCTION

This application aims to collect and analyze the reactions of any user’s friends. We are motivated by a situation, in which most of the end-users of Facebook Social Network (FB) are not going to know their real connections. For example, many users have numerous friends in their list and no clear knowledge about the interactions between themselves and these friends. This knowledge is necessary for answering the question, “How often do I interact with my friends on Facebook?”. This type of analysis is curious for common users but mostly useful to popular users, who do not have “person” profiles, which deprives them of Facebook’s analytics platform. It also comes from the practical issue, of many FB users taking a lot of time to go through their friends list to try and figure out who has not interacted with them and to un-friend these people.

Therefore, we wrote this app to help these people have a clear understanding of the interaction between them and their friends, how frequently they received comments or reactions from their friends, or which friends they interacted with most. We wrote this application using Python as the programming language and Github as a sub-version control. In this application, we discovered different ways of retrieving data, the ways to control this received data, and the receiving processes. After that, we dealt with data extraction and created graphs of these interactions. Finally, we deploy into a web-style framework(Flask) that makes the app usable online as a FB application.

2. FACEBOOK DATA STRUCTURE AND RETRIEVING MECHANISM

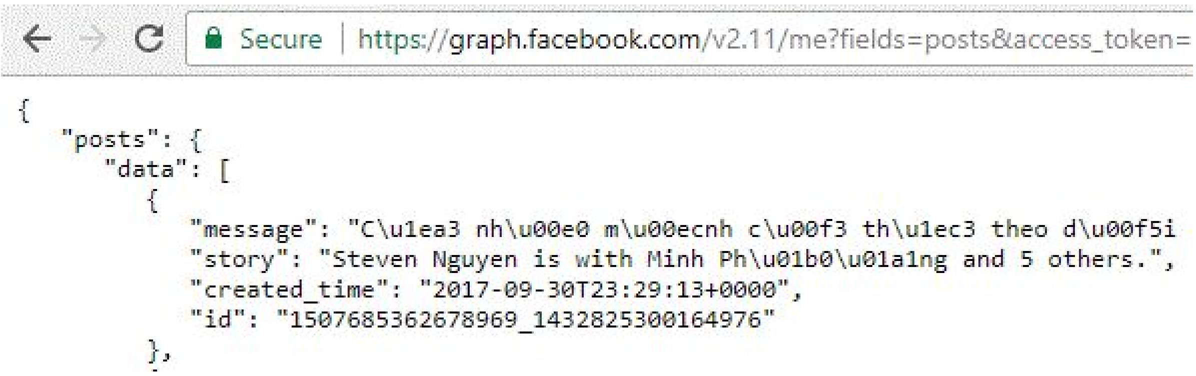
2.1. Data retrieving mechanism

In this application, we use Graph API (version 2.11) supported by Facebook to retrieve data. Facebook (FB) allows applications to retrieve a personal profile or a page with limited data fields. However, applications first must get the user token from the user profile. We obtained this by design a login webpage, registered as FB App. By this way, every Facebook user could be invited to login through this webpage and accept our requested data fields. After all, we would have that user token and their permissions to retrieve necessary data, in our case, are posts, comments, and reactions.

We will request directly to FB Graph API (https://developers.facebook.com/tools/explorer/) with necessary fields to retrieve the data.

2.2. Data structure

The user token obtained from the user login and acceptance would be transferred to our Main class to retrieve data. The would-be-retrieved FB data is in JSON format (Fig. 1). There are 2 levels of data, we only take care of the 2nd level that is “data”. The Fig. 1 shows the JSON data retrieve from a posts field.


Secure | https://graph.facebook.com/v2.11/me?fields=posts&access_token=

```
{
   "posts": {
      "data": [
         {
            "message": "C\u1ea3 nh\u00e0 m\u00ecnh c\u00f3 th\u1ec3 theo d\u00f5i
            "story": "Steven Nguyen is with Minh Ph\u01b0\u01a1ng and 5 others.",
            "created_time": "2017-09-30T23:29:13+0000",
            "id": "1507685362678969_1432825300164976"
         },
```

Figure 1. Example of a return of JSON (fields=posts)

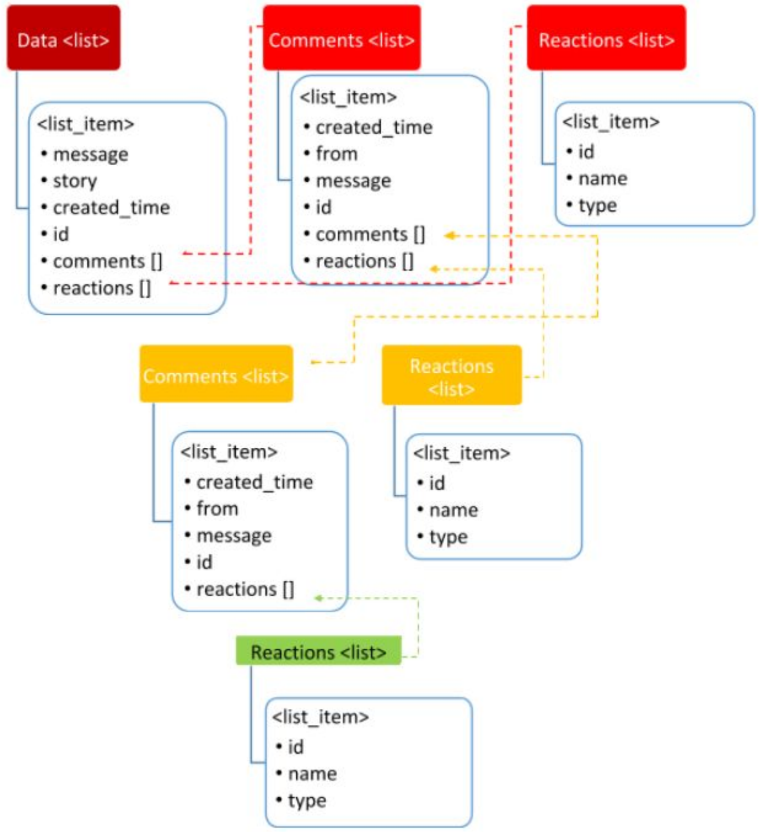


Figure 2. The maximum deep of data level could be supported by one retrieving time from

(Dark Red: Level 0; Red: Level 1; Orange: Level 2; Green: Level 3)

As mentioned above, in phase 2, those data would be extracted to have the id and get other fields. Other fields would be appended continue to the “id” in this figure. In short, we will have a list of contents [list] (fig. 2), with each content have a list of message, story, created_time, id, comments [list], reactions [list].

2.3. Data storing and encryption

In the first phase, data would be retrieved from FB Graph API under HTTPS connection. The filename is a string obtained from a hash of SHA-256 followed by a Base64 URL-safe-encoding.

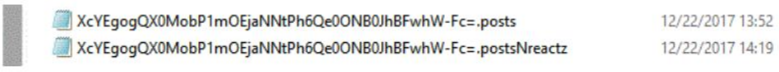

XcYEgogQX0MobP1mOEjaNNtPh6Qe0ONB0JhBFwhW-Fc=.posts 12/22/2017 13:52
XcYEgogQX0MobP1mOEjaNNtPh6Qe0ONB0JhBFwhW-Fc=.postsNreactz 12/22/2017 14:19

Figure 5. Data storing on file system

XcYEgogQX0MobP1mOEjaNNtPh6Qe0ONB0JhBFwhW-Fc=.posts - Notepad

Figure 6. An encrypted file opened in Notepad

3. WORKING ENVIRONMENT AND ACCESS TO THE APPLICATION

This program was developed and tested under Python 2.7.x and 3.6.x versions. The repository of this project is available at the following link https://github.com/vinhtien/apc.

The web application consists of two web pages, a “welcome” page and the one where the graphs of the user are shown. To be able to see the graphs, the user is required to grant permission to the application for accessing his posts. However, since this facebook application is in developer mode and thus available only for the users that explicitly were added to it by the creator on the “facebook for developers” platform. That said, if the reader wants to access our application he can provide us his facebook username to be able to do so.

4. MODULES AND CLASSES

4.1.1. Core

This module is used to retrieve the JSON data from Facebook, handling errors and retries.

4.1.2. FileIO

This module is used to handle file reading and writing. This file also ensures that data would be saved and open in UTF-8 encoding. Filename and data content would be encrypted and decrypted by this class.

4.1.3. Main

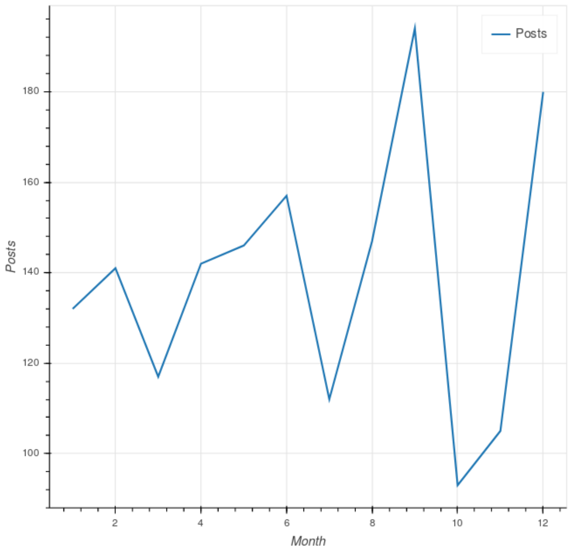
This module implements other modules (Core, FileIO) to retrieve, save and load data from the internet or from the saved data files.

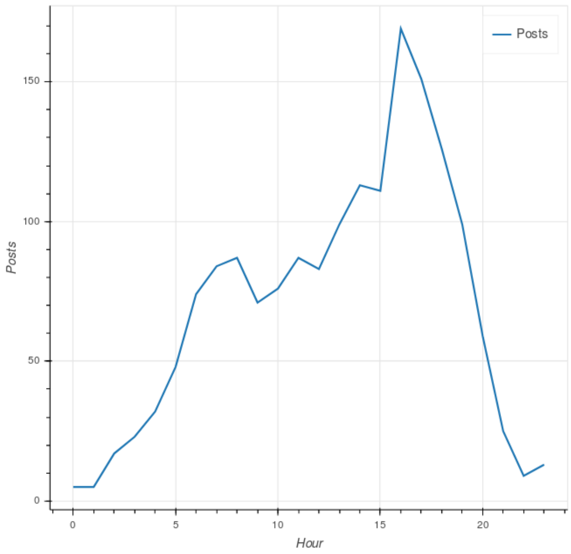
4.1.4. Extraction

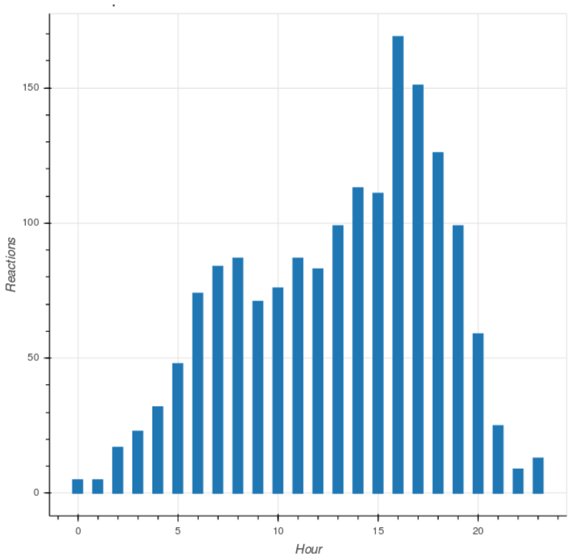
Makes pandas.DataFrames of json output.

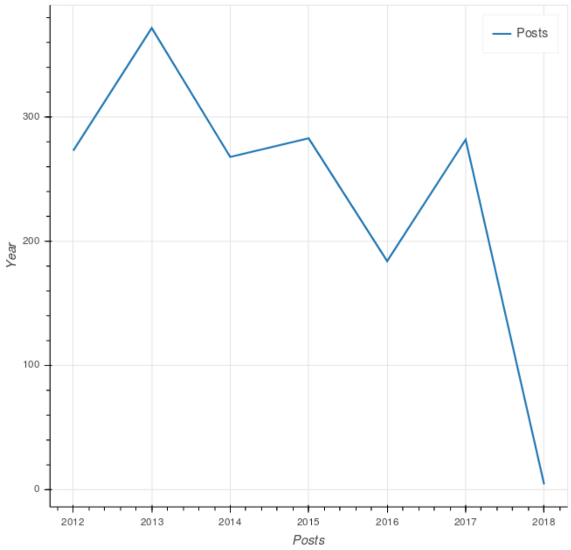
4.1.5. Charts

Handles plotting from pandas dataframes. Samples could be seen below.


Posts
180
160
140
120
100
Posts
2
4
6
8
10
12
Month


Posts
150
100
50
0
Posts
0
5
10
15
20
Hour


150
100
50
0
Reactions
0
5
10
15
20
Hour


Posts
300
200
100
0
Year
2012
2013
2014
2015
2016
2017
2018
Posts