
Facebook Page Post Scraper

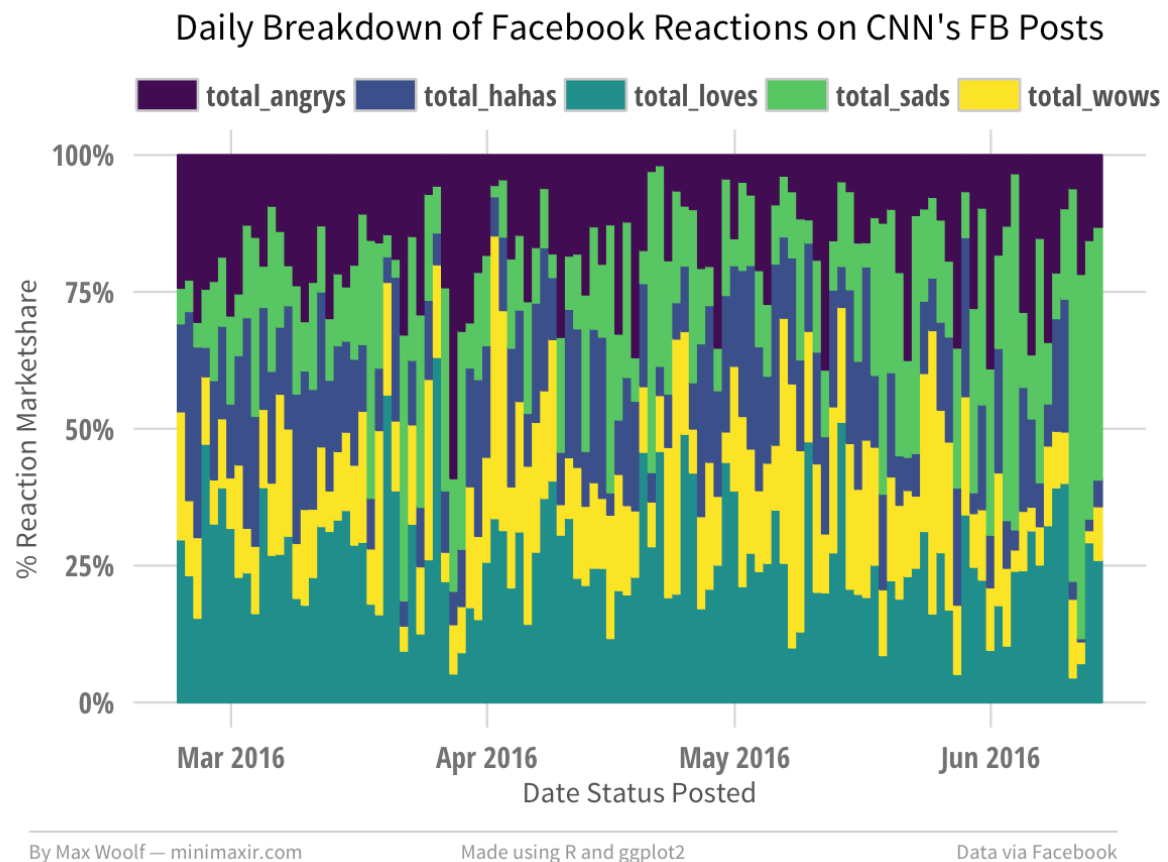
UPDATE December 2017: Due to a bug on Facebook's end, using this scraper will only return a very small subset of posts (5-10% of posts) over a limited timeframe. Since Facebook now owns CrowdTangle, the (paid) canonical source of historical Facebook data, Facebook doesn't have an incentive to fix the linked bug.

On December 12th, a Facebook engineer commented that they are developing a new endpoint for scraping posts chronologically. I will refactor this script once that happens. Until then, there likely will not be any PRs accepted.

num_reactions	num_comments	num_shares	num_likes	num_loves	num_wows	num_hahas	num_sads	num_angrys
4	0	3	3	0	1	0	0	0
1846	614	760	1573	148	12	2	111	0
1197	106	254	1021	44	1	131	0	0
963	318	99	639	94	10	100	2	119
3079	611	863	2108	43	38	883	5	3
7802	552	1425	6778	767	248	2	7	0
12812	1920	3474	10790	720	20	1283	1	0
2192	155	181	1887	41	1	261	2	0
4475	105	526	3714	731	0	4	26	0
1295	353	511	1143	74	0	78	0	0
8931	272	2910	6264	1317	55	5	1287	3
3326	314	535	3129	81	23	93	0	0
11049	360	3383	8732	1635	12	660	1	9
6282	1808	1417	4649	20	360	1238	13	2

A tool for gathering *all* the posts and comments of a Facebook Page (or Open Facebook Group) and related metadata, including post message, post links, and counts of each reaction on the post. All this data is exported as a CSV, able to be imported into any data analysis program like Excel.

The purpose of the script is to gather Facebook data for semantic analysis, which is greatly helped by the presence of high-quality Reaction data. Here's quick examples of a potential Facebook Reaction data visualization using data from CNN's Facebook page:



Usage

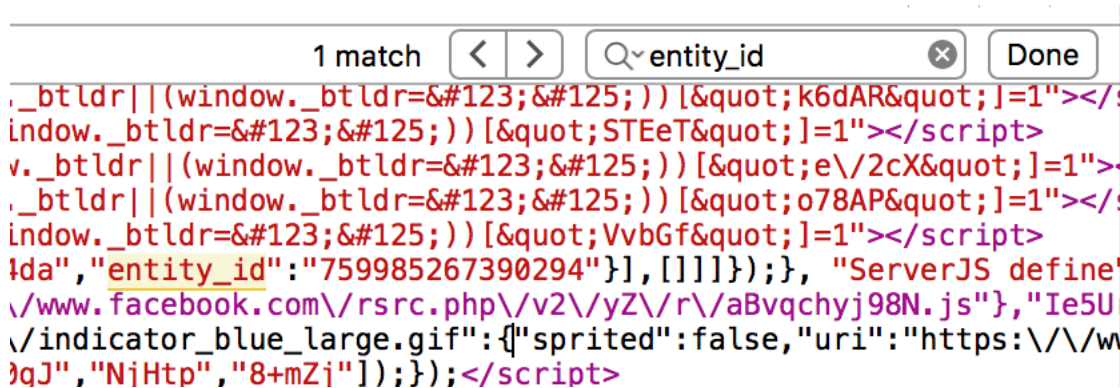
Scrape Posts From Public Page

The Page data scraper is implemented as a Python 2/3 script in `get_fb_posts_fb_page.py`; fill in the App ID and App Secret of a Facebook app you control (I strongly recommend creating an app just for this purpose) and the Page ID of the Facebook Page you want to scrape at the beginning of the file. Then run the script by `cd` into the directory containing the script, then running `python get_fb_posts_fb_page.py` or `python3 get_fb_posts_fb_page.py`.

Scrape Posts from Open Group

To get data from an Open Group, use the `get_fb_posts_fb_group.py` script with the App ID and App Secret filled in the same way. However, the `group_id` is a *numeric ID*. For groups without a custom username, the ID will be in the address bar; for groups with custom usernames, to get the

ID, do a View Source on the Group Page, search for the phrase "`entity_id`", and use the number to the right of that field. For example, the `group_id` of Hackathon Hackers is 759985267390294.



Scrape Comments From Page/Group Posts

To scrape all the user comments from the posts, create a CSV using either of the above scripts, then run the `get_fb_comments_from_fb.py` script, specifying the Page/Group as the `file_id`. The output includes the original `status_id` where the comment is located so you can map the comment to the original Post with a `JOIN` or `VLOOKUP`, and also a `parent_id` if the comment is a reply to another comment.

Keep in mind that large pages such as CNN have *millions* of comments, so be careful! (scraping throughput is approximately 87k comments/hour)

Privacy

This scraper can only scrape public Facebook data which is available to anyone, even those who are not logged into Facebook. No personally-identifiable data is collected in the Page variant; the Group variant does collect the name of the author of the post, but that data is also public to non-logged-in users. Additionally, the script only uses officially-documented Facebook API endpoints without circumventing any rate-limits.

Note that this script, and any variant of this script, *cannot* be used to scrape data from user profiles. (and the Facebook API specifically disallows this use case!)

Known Issues

- UTF-16 text (CJK) sometimes fails.

-
- GIFs in comments will not appear for an App access_token. (it requires a User access_token for no apparent reason).

Maintainer

Max Woolf (@minimaxir)

Max's open-source projects are supported by his Patreon. If you found this project helpful, any monetary contributions to the Patreon are appreciated and will be put to good creative use.

For more information on how the script was originally created, and some tips on how to create similar scrapers yourself, see my blog post [How to Scrape Data From Facebook Page Posts for Statistical Analysis](#).

Credits

Peeter Tintis, whose fork of this repo implements code for finding separate reaction counts per this Stack Overflow answer.

Marco Goldin for the Python 3.5 fork.

License

MIT

If you do find this script useful, a link back to this repository would be appreciated. Thanks!