
NETWORK SCIENCE OF ONLINE INTERACTIONS

Chapter 2 exercises +
Data Gathering

Joao Neto

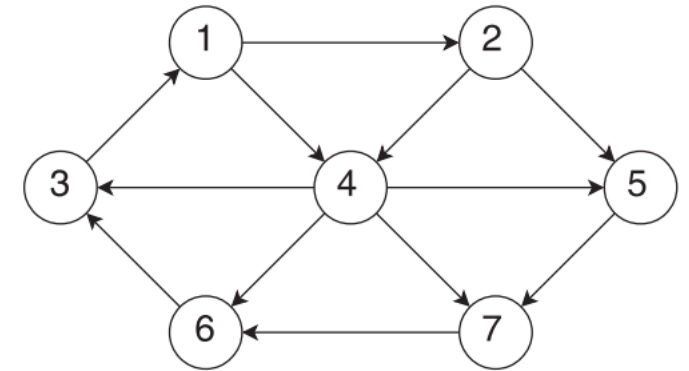
03/May/2023

BOOK EXERCISES – CHAPTER 2

- Exercises: 2.24, 2.35, 2.36, 2.37, 2.38

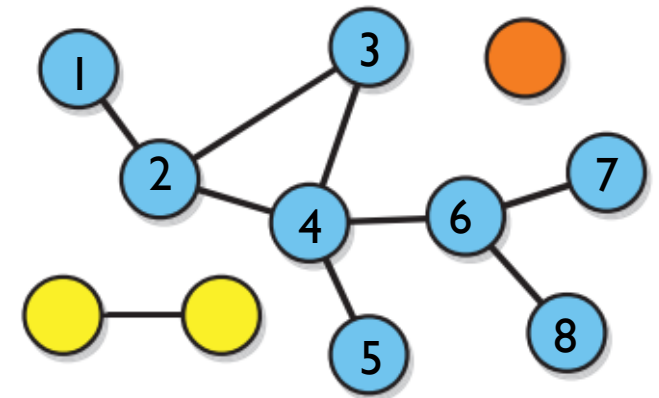
2.24 Consider an undirected version of the network in Figure 2.16. What is the diameter of this network?

- $\ell_{max} = 2$



2.35 Consider the undirected network in Figure 2.4. Compute the shortest-path length for each pair of nodes in the giant component.

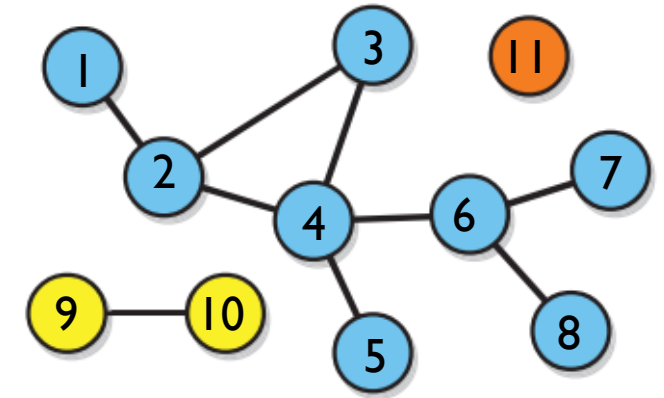
$$\ell = \begin{pmatrix} 0 & 1 & 2 & 2 & 3 & 3 & 4 & 4 \\ & 0 & 1 & 1 & 2 & 2 & 3 & 3 \\ & & 0 & 1 & 2 & 2 & 3 & 3 \\ & & & 0 & 1 & 1 & 2 & 2 \\ & & & & 0 & 2 & 3 & 3 \\ & & & & & 0 & 1 & 1 \\ & & & & & & 0 & 2 \\ & & & & & & & 0 \end{pmatrix}$$



BOOK EXERCISES – CHAPTER 2

2.36 Consider the undirected network in Figure 2.4. Compute the clustering coefficient for each node such that it is defined.

- C requires $k_i > 1$
 - undefined: 1, 5, 7, 8, 9, 10, 11
- $C_i = 2 \times \frac{\text{\#triangles}}{k_i(k_i-1)}$
- $C_2 = 2/6, C_3 = 2/2, C_4 = 2/12, C_6 = 0$



BOOK EXERCISES – CHAPTER 2

2.37 Consider the network example in Figure 2.12. Compute the shortest-path length for each pair of nodes, and the average shortest-path length for the network.

■ with code:

```
G = nx.Graph()
G.add_edges_from([('d','e'),('e','c'),('e','b'),('c','b'),('c','g'),
('b','g'),('b','h'),('g','f'),('g','a'),('f','a')])

APL = nx.average_shortest_path_length(G)
shortest_paths = dict(nx.shortest_path_length(G))
```

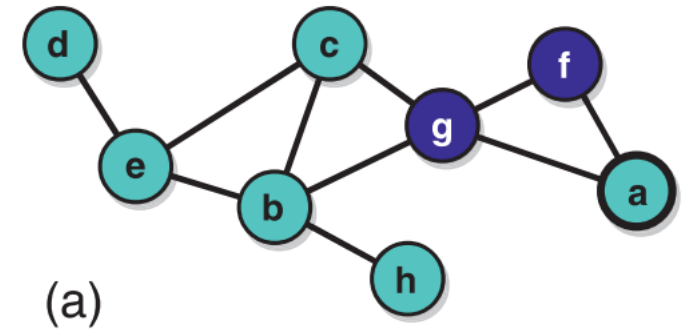
✓ 0.0s Python

```
print(APL)
shortest_paths
```

✓ 0.0s Python

2.0

```
{'d': {'d': 0, 'e': 1, 'c': 2, 'b': 2, 'g': 3, 'h': 3, 'f': 4, 'a': 4},
'e': {'e': 0, 'd': 1, 'c': 1, 'b': 1, 'g': 2, 'h': 2, 'f': 3, 'a': 3},
'c': {'c': 0, 'e': 1, 'b': 1, 'g': 1, 'd': 2, 'h': 2, 'f': 2, 'a': 2},
'b': {'b': 0, 'e': 1, 'c': 1, 'g': 1, 'h': 1, 'd': 2, 'f': 2, 'a': 2},
'g': {'g': 0, 'c': 1, 'b': 1, 'f': 1, 'a': 1, 'e': 2, 'h': 2, 'd': 3},
'h': {'h': 0, 'b': 1, 'e': 2, 'c': 2, 'g': 2, 'd': 3, 'f': 3, 'a': 3},
'f': {'f': 0, 'g': 1, 'a': 1, 'c': 2, 'b': 2, 'e': 3, 'h': 3, 'd': 4},
'a': {'a': 0, 'g': 1, 'f': 1, 'c': 2, 'b': 2, 'e': 3, 'h': 3, 'd': 4}}
```



```
import pandas as pd
df = pd.DataFrame.from_dict(shortest_paths, orient='index')
df
```

✓ 0.0s Python

| | d | e | c | b | g | h | f | a |
|---|---|---|---|---|---|---|---|---|
| d | 0 | 1 | 2 | 2 | 3 | 3 | 4 | 4 |
| e | 1 | 0 | 1 | 1 | 2 | 2 | 3 | 3 |
| c | 2 | 1 | 0 | 1 | 1 | 2 | 2 | 2 |
| b | 2 | 1 | 1 | 0 | 1 | 1 | 2 | 2 |
| g | 3 | 2 | 1 | 1 | 0 | 2 | 1 | 1 |
| h | 3 | 2 | 2 | 1 | 2 | 0 | 3 | 3 |
| f | 4 | 3 | 2 | 2 | 1 | 3 | 0 | 1 |
| a | 4 | 3 | 2 | 2 | 1 | 3 | 1 | 0 |

```
df['d'].mean()
```

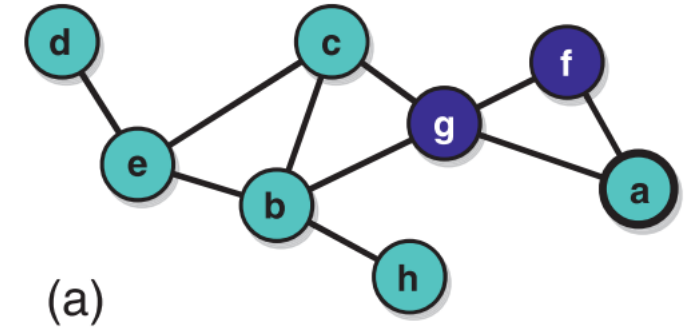
✓ 0.0s Python

2.375

BOOK EXERCISES – CHAPTER 2

2.38 Consider the network example in Figure 2.12. Compute the clustering coefficient for each node such that it is defined, and for the network.

- NetworkX: The value of c_u is assigned to 0 if $\deg(u) < 2$.



```
G = nx.Graph()
G.add_edges_from([('d','e'),('e','c'),('e','b'),('c','b'),('c','g'),
                  ('b','g'),('b','h'),('g','f'),('g','a'),('f','a')])

C_avg = nx.average_clustering(G)
C = nx.clustering(G)

print(C_avg)
C
```

✓ 0.0s Python

0.4583333333333333

```
{'d': 0,
 'e': 0.3333333333333333,
 'c': 0.6666666666666666,
 'b': 0.3333333333333333,
 'g': 0.3333333333333333,
 'h': 0,
 'f': 1.0,
 'a': 1.0}
```

```
K = dict(nx.degree(G))
df_degree = pd.Series(K)
valid_nodes = df_degree[df_degree>1].index.to_list()
valid_nodes
```

✓ 0.1s Python

```
['e', 'c', 'b', 'g', 'f', 'a']
```

```
df_clustering = pd.Series(C)
df_clustering[valid_nodes].mean()
```

✓ 0.0s Python

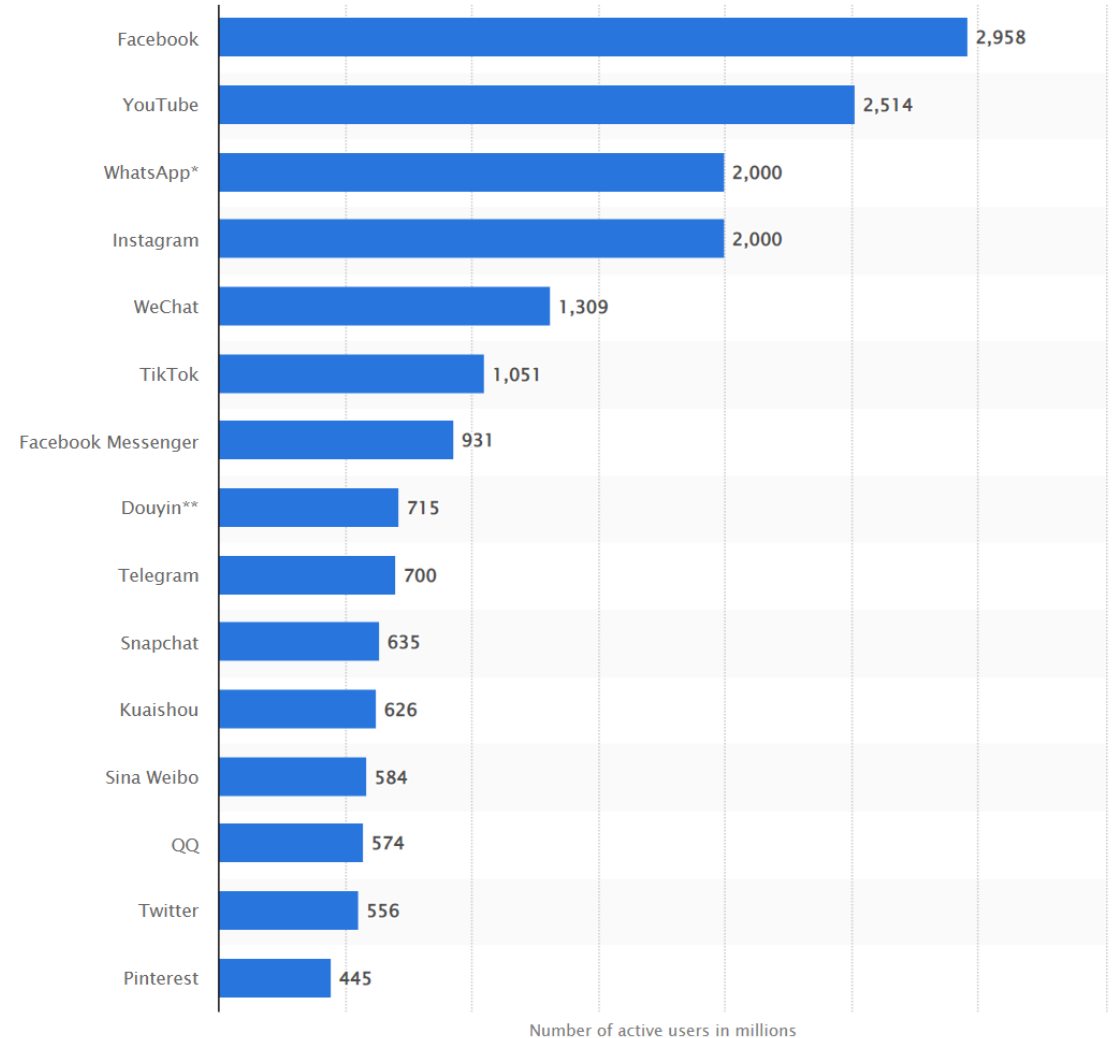
0.6111111111111111

BOOK EXERCISES – CHAPTER 2

- Questions?

DATA GATHERING

- There are **many** social media platforms
- They are **extremely** large
- How do we get and analyze data from these places?



DATA GATHERING

- Published datasets

- Citable reference
- Data is clean (or should be)
- Unlikely to be exactly what you want

- Platform API

- Targeted to what you want
- Data is usually clean
- Rate limits can limit scope of research
- Can always be killed by the platform

- Scraping

- Always possible
- Data is very dirty, lots of computational work

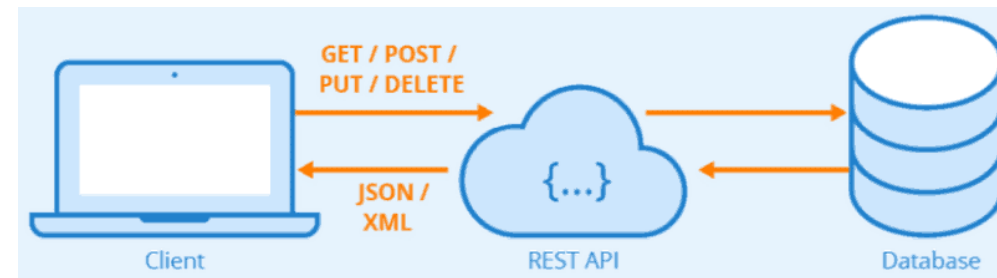
The Pushshift Reddit Dataset

Jason Baumgartner^{1,*}, Savvas Zannettou^{2,Ⓢ}, Brian Keegan³, Megan Squire⁴, Jeremy Blackburn^{5,Ⓢ}

¹Pushshift.io, ²Max Plank Institute, ³University of Colorado Boulder, ⁴Elon University, ⁵Binghamton University

*Network Contagion Research Institute, ⓈiDRAMA Lab

jason@pushshift.io, szannett@mpi-inf.mpg.de, brian.keegan@colorado.edu, msquire@elon.edu, blackburn@cs.binghamton.edu



Basic

For hobbyists or prototypes

- Rate limited access to suite of v2 endpoints
- 3,000 Tweets per month - posting limit at the user level
- 50,000 Tweets per month - posting limit at the app level
- 10,000 Tweets per month - read-limit rate cap
- 2 app IDs
- Login with Twitter
- Cost: \$100 per month

[Subscribe now](#)

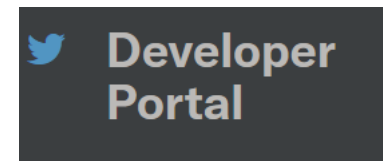


DATA GATHERING

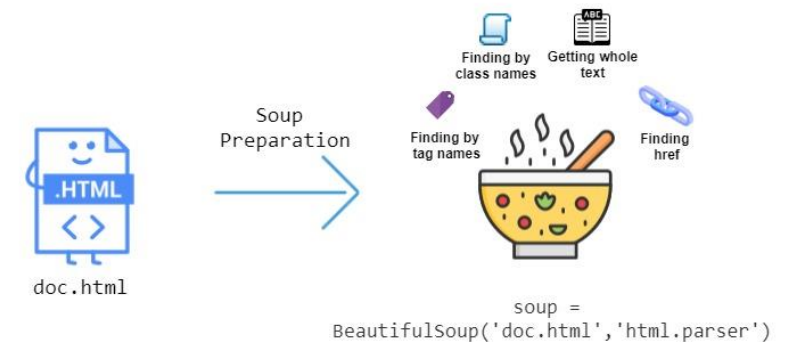
- Published datasets
 - Journal proceedings (e.g. ICWSM)
 - Zenodo
 - Groups and services (iDRAMA, SMAT, Communalytic)
- Platform API
 - Pure HTTP: requests package
 - Reddit: PRAW, Pushshift
 - Twitter: twarc, tweepy
 - Telegram: telethon, pyrogram
- Scraping
 - BeautifulSoup4, Selenium
 - Scraping/IP services (Bright Data, etc)

[Home](#) / [Search](#)

Search



authorized applications



PUBLISHED DATASETS

Reddit

- 15B comments
- 2B submissions
- 102M users
- 4.2M communities

Telegram

- 317M messages
- 27.8k channels
- 2.2M users

Voat

- 16.3M comments
- 2.4M submissions
- 113k users
- 7.1k communities

.win

- 1.6M comments
- 227k submissions
- 44.9k users
- 11 sites

TheDonald.win

- 1.1M comments
- 169k submissions
- 35.5k users

HackerNews

- 14.2M comments
- 2.9M stories
- 357k users

Parler

- 84.6M comments
- 98.5M submissions
- 2.4M users, 13.2M user profiles

Gab

- 14.7M comments
- 19.5M submissions
- 160k users

StackExchange

- 98.6M comments
- 35.4M answers
- 23M questions
- 5.5M users

Youtube

- 4.6B videos
- 214M channels

BitChute

- 3.1M videos
- 11M comments
- 61k channels

TikTok

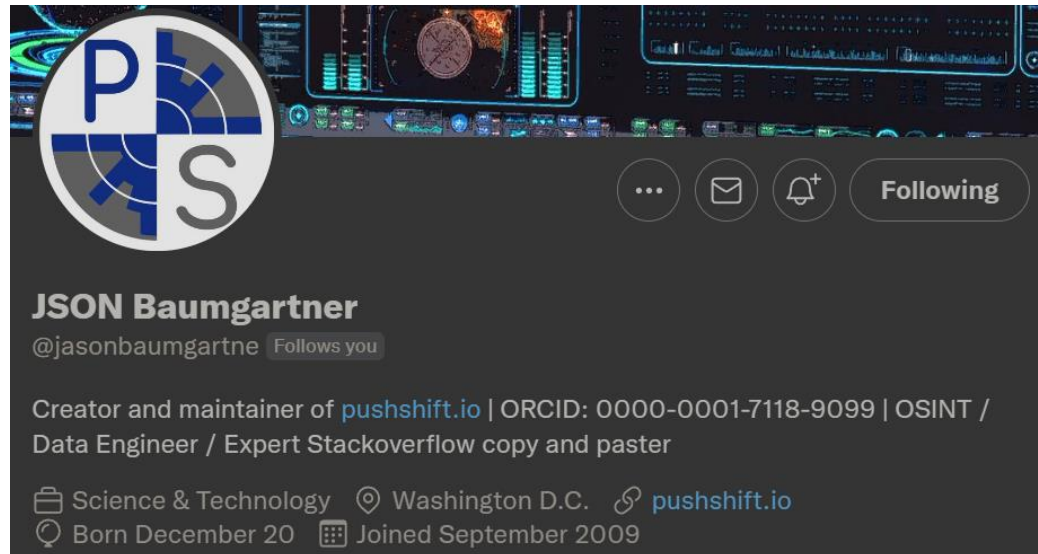
- 25M videos
- 7M users

4chan (/pol)

- 131M comments
- 3.4M threads

PUBLISHED DATASETS

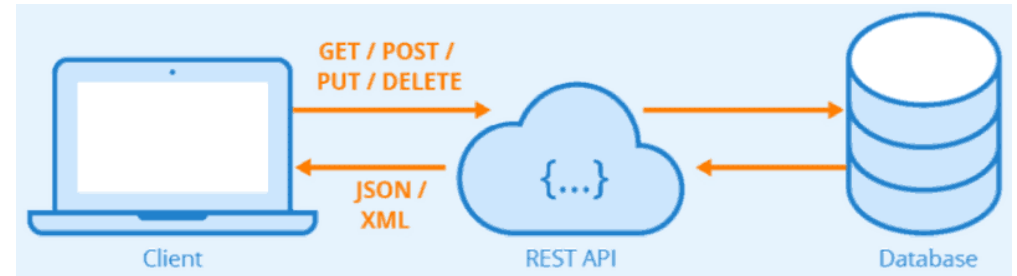
- How complete is this data?
 - Reddit is almost fully-sampled (sequential IDs)
 - Youtube is ~25% sampled (massive crawling project)
 - All the others are undersampled to varying (usually unknown) degree



- References
 - **Reddit:** J. Baumgartner et al., arXiv:2001.08435 [cs] (2020).
 - **4chan:** A. Papasavva et al., arXiv:2001.07487 [cs] (2020).
 - **Bitchute:** M. Trujillo et al., arXiv:2004.01984 [cs] (2020).
 - **Parler:** M. Aliapoulos et al., arXiv:2101.03820 [physics] (2021).
 - **Voat:** A. Mekacher & A. Papasavva, arXiv:2201.05933 [cs]. (2022).
 - **TikTok, Telegram, HackerNews, StackExchange:** <https://files.pushshift.io/>
 - **Youtube:** https://archive.org/details/archiveteam_youtube
 - **.Win:** https://ddosecrets.com/wiki/.Win_Network

PLATFORM API

- Application Programming Interface
 - Allows you to get targeted data, from some parameters
 - Data comes from someone else's server
- Authentication
 - Registering somewhere to get keys
- Endpoints
 - What data you can get
- Rate limits
 - Use a package to handle it (exponential backoff)
 - Errors, risk of suspension otherwise



List of Endpoints

| Endpoint | Description | Status |
|---|--|--------|
| /reddit/search/comment/ | Search Reddit Comments | Active |
| /reddit/search/submission/ | Search Reddit Submissions | Active |
| /reddit/submission/comment_ids/{base36-submission-id} | Retrieve comment ids for a submission object | Active |

PLATFORM API

- Example: Pokeapi
 - No authentication
 - Many endpoints
 - Generation
 - Pokemon

```
import requests

# gets the number of pokemon in generation 1
url = 'https://pokeapi.co/api/v2/generation/1'
response = requests.get(url)
data = response.json()
len(data['pokemon_species'])
```

✓ 0.1s Python

151

```
# gets pokemon that evolves into pikachu
url = 'https://pokeapi.co/api/v2/pokemon-species/pikachu'
response = requests.get(url)
data = response.json()
data['evolves_from_species']['name']
```

✓ 0.1s Python

'pichu'

Berries
Contests
Encounters
Evolution
Games
Items
Locations
Machines
Moves
| Pokémon
Abilities
Characteristics
Egg Groups
Genders
Growth Rates
Natures
Pokeathlon Stats
| Pokemon
Pokemon Location Areas
Pokemon Colors
Pokemon Forms
Pokemon Habitats
Pokemon Shapes
Pokemon Species
Stats
Types



The RESTful Pokémon API
Serving over 330,000,000 API calls each month!

Generations (endpoint)

A generation is a grouping of the Pokémon games that separates them based on the Pokémon they include. In each generation, a new set of Pokémon, Moves, Abilities and Types that did not exist in the previous generation are released.

GET <https://pokeapi.co/api/v2/generation/{id or name}/>


☐ View raw JSON (0.772 kB, 42 lines)

Generation (type)

| Name | Description | Type |
|-----------------|--|--|
| id | The identifier for this resource. | integer |
| name | The name for this resource. | string |
| abilities | A list of abilities that were introduced in this generation. | list NamedAPIResource (Ability) |
| names | The name of this resource listed in different languages. | list Name |
| main_region | The main region travelled in this generation. | NamedAPIResource (Region) |
| moves | A list of moves that were introduced in this generation. | list NamedAPIResource (Move) |
| pokemon_species | A list of Pokémon species that were introduced in this generation. | list NamedAPIResource (PokemonSpecies) |
| types | A list of types that were introduced in this generation. | list NamedAPIResource (Type) |
| version_groups | A list of version groups that were introduced in this generation. | list NamedAPIResource (VersionGroup) |

PLATFORM API

- Better example: Reddit
- Allows authenticated and anonymous use
 - Authenticated: 60 requests per min, account-based
 - Anonymous: 30 requests per min, IP-based
- Authentication:
 - <https://www.reddit.com/prefs/apps>



test-script
personal use script
p-jcoLKBynTLew

change icon

secret gko_LXELoV07ZBNUXrvWZfzE3al

name test-script

description

about url

redirect uri http://127.0.0.1:65010/authorize_callback

update app

developers reddit (that's you!) remove

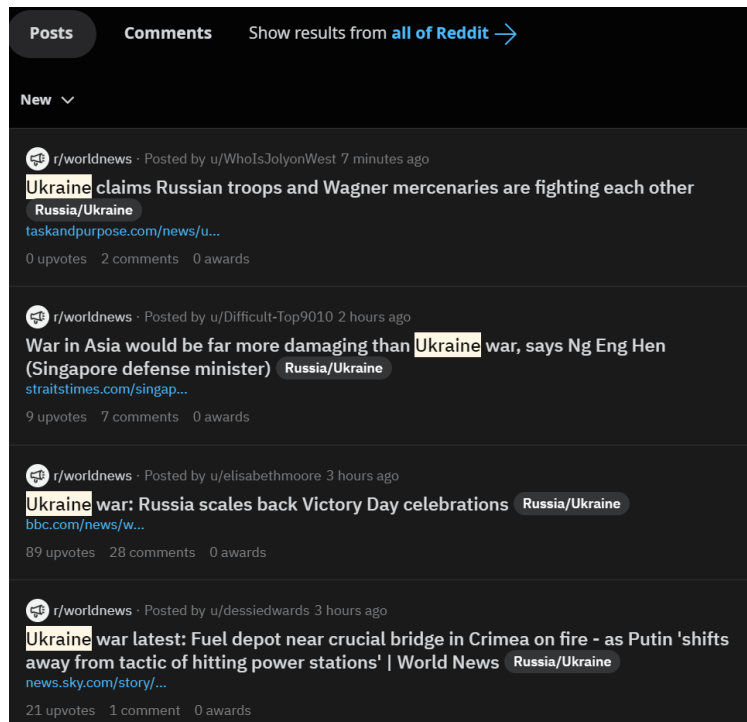
add developer:

delete app

```
1 // 20230503121515
2 // https://www.reddit.com/r/anarchychess/about.json
3
4 {
5   "kind": "t5",
6   "data": {
7     "user_flair_background_color": null,
8     "submit_text_html": "<!-- SC_OFF --><div class=\"md\"><p>Thank you for your
submission. If it is spam/advertisement, please remember we will remove
it.</p></div><!-- SC_ON -->",
9     "restrict_posting": true,
10    "user_is_banned": false,
11    "free_form_reports": true,
12    "wiki_enabled": true,
13    "user_is_muted": false,
14    "user_can_flair_in_sr": true,
15    "display_name": "AnarchyChess",
16    "header_img": "https://b.thumbs.redditmedia.com/duwlracKOIqV9ost8irrboVY7HJh_DXpJVqzL5RJRzw.png",
17    "title": "Chess Humour",
18    "allow_galleries": true,
19    "icon_size": [
20      256,
21      256
22    ],
23    "primary_color": "#bbdbbf",
24    "active_user_count": 2717,
25    "icon_img": "https://b.thumbs.redditmedia.com/yJYA2zLSrKRE-ovDpFmbuGwE-_3tCbD34lcc2W9ucDQ.png",
26    "display_name_prefixed": "r/AnarchyChess",
27    "accounts_active": 2717,
28    "public_traffic": false,
29    "subscribers": 451567,
```

PLATFORM API

- Reddit API package: PRAW
- Handles rate limiting
- Many, many endpoints
 - Check the Reddit API documentation



```
import praw

params = {"client_id": "YOUR_ID", "client_secret": "YOUR_SECRET", "user_agent":
"YOUR_APP_ID", "username": "YOUR_USERNAME", "password": "YOUR_REDDIT_PASSWORD"}

praw_session = praw.Reddit(client_id=params['client_id'],
                             client_secret=params['client_secret'],
                             password=params['password'],
                             user_agent=params['user_agent'],
                             username=params['username'])

#Get the last 10 posts from worldnews with the word "Ukraine"
for submission in praw_session.subreddit('worldnews').search('Ukraine',
sort='new', limit=10):
    print("{} (score: {})".format(submission.title[:50], submission.score))
```

✓ 1.3s Python

War in Asia would be far more damaging than Ukrain (score: 11)
Ukraine war: Russia scales back Victory Day celebr (score: 81)
Ukraine war latest: Fuel depot near crucial bridge (score: 21)
Finnish newspaper hides a secret room, detailing R (score: 476)
Russia comes out in support of Hindus; Lashes Ukra (score: 0)
Ukraine war: Russia scales back Victory Day celebr (score: 22)
/r/WorldNews Live Thread: Russian Invasion of Ukra (score: 843)
For the first time since it invaded Ukraine, Russi (score: 329)
Denmark to Send Military Assistance to Ukraine Wor (score: 393)
Denmark to make \$250 mln donation to Ukraine for m (score: 3854)

PLATFORM API

- The issues
 - Rate limits: max 100 objects per API call
 - Max 8.6M objects per day
 - But many calls only return one object, so max 86K per day
 - Typically 20% less than max
 - Evolving data (always record ingestion timestamp)
 - Impossible to fully-sample data
 - Repeated search doesn't work because rate limits
- Solution (for now): Pushshift
 - Get old data from Pushshift
 - Get updated data from Reddit

PLATFORM API

- Pushshift + PRAW
 - Gets sequential ids from Pushshift
 - Gets updated data from PRAW
 - Platforms have different rate limits

```
subreddit = 'picard'

#Gets data from pushshift
base_url = "https://api.pushshift.io/reddit/submission/search/"
params = {"subreddit": subreddit, "sort": "desc", "sort": "created_utc", "size": 20}
response = requests.get(base_url, params=params)
data = response.json()
#adds t3_ to the ids for praw
ids = ['t3_'+i['id'] for i in data['data']]
ids
```

✓ 1.2s

Python

```
['t3_134pvn0',
't3_134koj5',
't3_134k55g',
't3_134k4dp',
't3_134k2dc',
't3_134byvk',
't3_134bl3d',
```

```
import praw

params_example = {"client_id": "YOUR_ID", "client_secret": "YOUR_SECRET",
"user_agent": "YOUR_APP_ID", "username": "YOUR_USERNAME", "password":
"YOUR_REDDIT_PASSWORD"}

#Opens praw session
praw_session = praw.Reddit(client_id=params['client_id'],
client_secret=params['client_secret'],
password=params['password'],
user_agent=params['user_agent'],
username=params['username'])

# gets data from the submissions in id from praw
submissions = praw_session.info(fullnames=ids)
for submission in submissions:
    print(" (score: {}) {}".format(submission.score, submission.title[:70]))
```

✓ 2.3s

Python

Output exceeds the [size limit](#). Open the full output data [in a text editor](#)

```
(score: 0) Seven of Nine and Janeway actors still beefing?
(score: 977) Ageless beauty! ❤️
(score: 0) Picard is a Narcissist. there I've said it.
(score: 148) What is Mugatu meditation??
(score: 0) It's a shame we didn't get a true continuation of season 1.
(score: 0) Icheb is the character most deserving of resurrection from Picard, can
(score: 2) I can't see how the original Spacedock could possibly have gotten to a
(score: 0) Seems I'm the only one who didn't love the trench run scene
(score: 1) Obligatory Vanity Post
(score: 0) Oh Beverly...
(score: 57) Saying goodbye to Vadic
(score: 316) Your call...?
```

PUSHSHIFT DATA DUMPS

- Full sample dataset of all Reddit
 - 15B comments
 - 2B submissions
 - About 21TB uncompressed
 - Updated until 03/2023
- Sources
 - Monthly dumps (also on torrent)
 - <https://files.pushshift.io/reddit/>
 - Per-subreddit torrent:
<https://academictorrents.com/details/c398a571976c78d346c325bd75c47b82edf6124e>

Revisiting Reddit: an updated look from the Pushshift dataset

Joao P. Neto^{1,2*} and Riccardo Carlucci¹



Subreddit comments/submissions 2005-06 to 2022-12

Watchful1

The screenshot shows the Pushshift website interface. At the top, there are navigation tabs: Home, Technical (6/9), Comments (0), and Collections. On the right, there's a yellow button for 'Download 1.66TB' and a star icon with '0'. Below the navigation, a list of files is displayed under the heading 'reddit (39963 files)'. The list includes files like 'subreddits/0sanitymemes_comments.zst' (14.95MB), 'subreddits/0sanitymemes_submissions.zst' (5.83MB), 'subreddits/0x3642_comments.zst' (128.42kB), 'subreddits/0x3642_submissions.zst' (10.75MB), 'subreddits/0xPolygon_comments.zst' (11.85MB), 'subreddits/0xPolygon_submissions.zst' (4.28MB), and 'subreddits/1000ccplus_comments.zst' (7.50MB). On the right side of the page, there's an advertisement for 'Authentic JOBS' with the text 'Your new development career awaits. Check out the latest listings.' and a 'Hosted by users:' section featuring the 'Watchful1' profile.

Type: Dataset

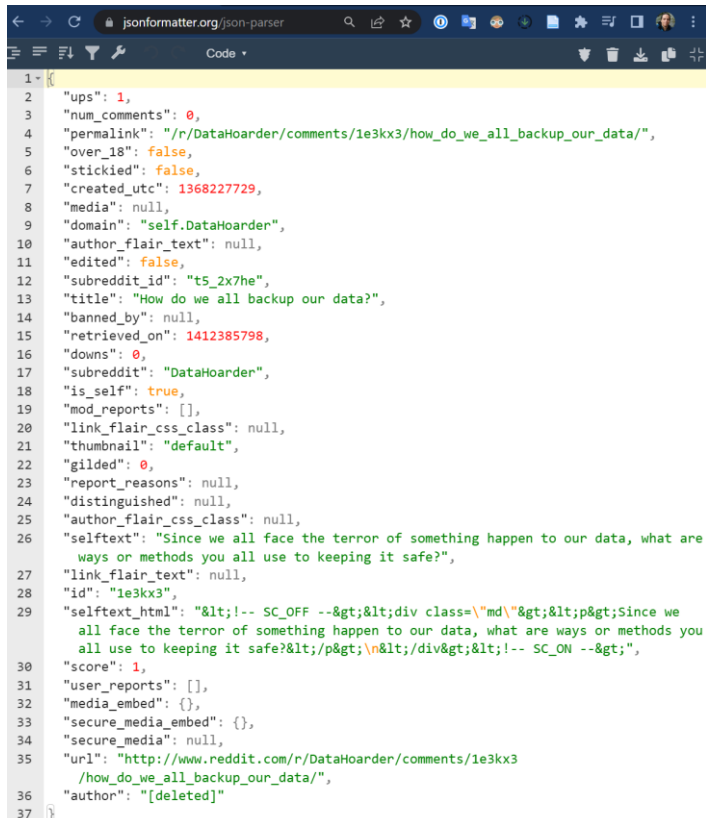
Tags: [reddit](#)

Abstract:

This is the top 20,000 subreddits from reddit's history in separate files. You can use your torrent client to only download the subreddit's you're interested in.

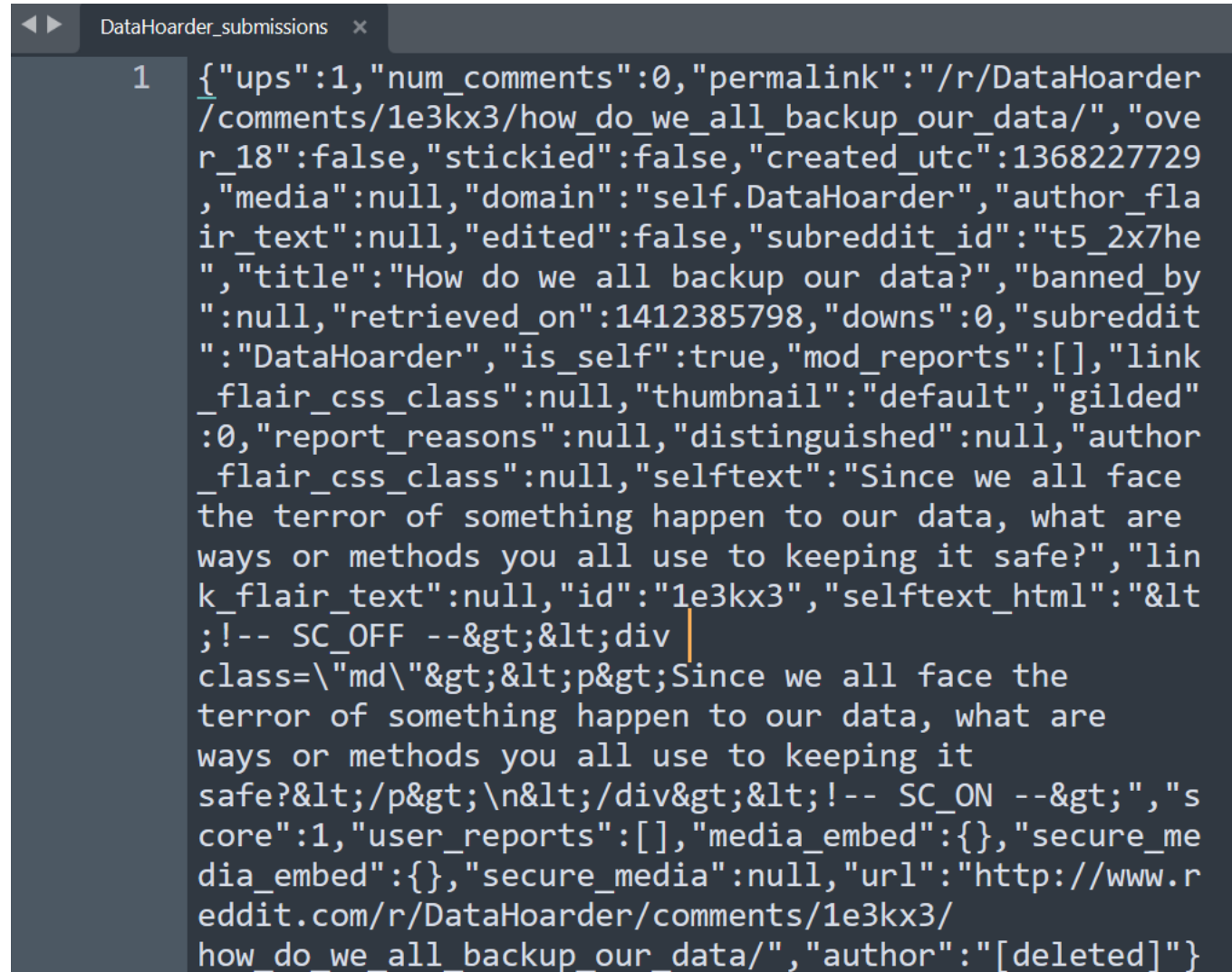
PUSHSHIFT DATA DUMPS

- Handling the data dumps
- Data is ndjson
- Python handles it natively (json)



A screenshot of a web browser displaying a JSON dump of a Reddit comment. The browser's address bar shows 'jsonformatter.org/json-parser'. The JSON data is formatted with syntax highlighting and line numbers. The comment is from the user 'DataHoarder' and contains a warning about data backup.

```
1 {
2   "ups": 1,
3   "num_comments": 0,
4   "permalink": "/r/DataHoarder/comments/1e3kx3/how_do_we_all_backup_our_data/",
5   "over_18": false,
6   "stickied": false,
7   "created_utc": 1368227729,
8   "media": null,
9   "domain": "self.DataHoarder",
10  "author_flair_text": null,
11  "edited": false,
12  "subreddit_id": "t5_2x7he",
13  "title": "How do we all backup our data?",
14  "banned_by": null,
15  "retrieved_on": 1412385798,
16  "downs": 0,
17  "subreddit": "DataHoarder",
18  "is_self": true,
19  "mod_reports": [],
20  "link_flair_css_class": null,
21  "thumbnail": "default",
22  "gilded": 0,
23  "report_reasons": null,
24  "distinguished": null,
25  "author_flair_css_class": null,
26  "selftext": "Since we all face the terror of something happen to our data, what are ways or methods you all use to keeping it safe?",
27  "link_flair_text": null,
28  "id": "1e3kx3",
29  "selftext_html": "<!-- SC_OFF --><div class=\"md\">&lt;p&gt;Since we all face the terror of something happen to our data, what are ways or methods you all use to keeping it safe?&lt;/p&gt;&lt;/div&gt;&lt;!-- SC_ON -->",
30  "score": 1,
31  "user_reports": [],
32  "media_embed": {},
33  "secure_media_embed": {},
34  "secure_media": null,
35  "url": "http://www.reddit.com/r/DataHoarder/comments/1e3kx3/how_do_we_all_backup_our_data/",
36  "author": "[deleted]"
37 }
```



A screenshot of a code editor showing a JSON dump of a Reddit comment. The editor has a tab titled 'DataHoarder_submissions'. The JSON data is displayed with syntax highlighting and line numbers. The comment is from the user 'DataHoarder' and contains a warning about data backup.

```
1 {
  "ups": 1, "num_comments": 0, "permalink": "/r/DataHoarder/comments/1e3kx3/how_do_we_all_backup_our_data/", "over_18": false, "stickied": false, "created_utc": 1368227729, "media": null, "domain": "self.DataHoarder", "author_flair_text": null, "edited": false, "subreddit_id": "t5_2x7he", "title": "How do we all backup our data?", "banned_by": null, "retrieved_on": 1412385798, "downs": 0, "subreddit": "DataHoarder", "is_self": true, "mod_reports": [], "link_flair_css_class": null, "thumbnail": "default", "gilded": 0, "report_reasons": null, "distinguished": null, "author_flair_css_class": null, "selftext": "Since we all face the terror of something happen to our data, what are ways or methods you all use to keeping it safe?", "link_flair_text": null, "id": "1e3kx3", "selftext_html": "&lt;!-- SC_OFF --&gt;&lt;div class=\"md\"&gt;&lt;p&gt;Since we all face the terror of something happen to our data, what are ways or methods you all use to keeping it safe?&lt;/p&gt;&lt;/div&gt;&lt;!-- SC_ON --&gt;", "score": 1, "user_reports": [], "media_embed": {}, "secure_media_embed": {}, "secure_media": null, "url": "http://www.reddit.com/r/DataHoarder/comments/1e3kx3/how_do_we_all_backup_our_data/", "author": "[deleted]"
}
```

PUSHSHIFT DATA DUMPS

- Suggested pipeline
 1. Download subreddit data dump
 2. Extract only necessary data from the json files
 3. Convert to pandas for handling
 4. Do network analysis in NetwokX
- Questions?