How to Scrape Large Amounts of Reddit Data

Matt Podolak Follow
Feb 14 · 5 min read

Photo by Markus Spiske on Unsplash

In this article, I'm going to show you how to use Pushshift to scrape a large amount of Reddit data and create a dataset. I define "large" as a set of data between 50,000–500,000 items. Data queries for over 500,000 items will be covered in a separate article as these will increase the risk of an out of memory error occurring, depending on the amount of available memory.

Why should you create a dataset from Reddit data? Reddit provides a public forum for communities with similar interests to discuss and exchange ideas, with a large majority of this data being text data. Creating a Reddit dataset of posts and comments from a specific subreddit will allow you to extract various insights, from sentiments towards different stock tickers, to trending topics in the news. One example could be creating a dataset of comments from r/WallStreetBets and looking for what stock tickers are most frequently mentioned over a particular timespan.

By the end of this article, you will have created a dataset of 100,000 comments from r/WallStreetBets created between Dec 1/20 and Feb 1/21. First, I'll start by discussing the different ways to get Reddit data and why we will be using Pushshift.

Reddit Data

There are 2 main ways to retrieve data from Reddit, using either the Reddit or Pushshift API.

The Reddit API is great but only allows users to pull a limited amount of recent comments or submissions from a few different streams for a subreddit, such as `hot`, `new`, `top`, etc. Due to this, using the Reddit API is not recommended when you need to create a large dataset.

Now you might be wondering, what exactly is Pushshift? Pushshift is a service that ingests new comments and submissions from Reddit, stores them in a database, and makes them available to be queried via an API endpoint.

Pushshift has a few drawbacks that are worth mentioning. Since Pushshift is a third-party service with a single maintainer, downtime can happen, it's rare, but it can cause small windows of time during which comments and submissions are not ingested and stored. Occasionally these gaps are backfilled with data, but not always. The second drawback is that the data is NOT real-time since Pushshift has to ingest and store the comments and submissions. The delay in ingesting comments or submissions varies and can be up to 2 days, so if you have a need for real-time data, I would recommend using the Reddit API (via `praw`) instead.

Using Pushshift

To retrieve data from Pushshift you make API requests to a specific endpoint with parameters specifying what query you want it to run. You can work directly with the API or use a Python wrapper. There are two main wrappers created in Python for Pushshift, `psaw`, and `pmaw`. For the creation of large datasets I would recommend using `pmaw`, it's a package I created that is highly optimized for extracting large amounts of data, running 1.79x faster than `psaw` from the benchmarks I performed with up to 400,000 submissions. `pmaw` has built-in rate-limiting, pagination, and runs requests on multiple threads, all we have to do is define our query based on the Pushshift endpoint parameters.

Data Retrieval

Now it's finally time to start writing some code! The rest of this tutorial will be using Python version 3.6, which you can download here. I would recommend running your code in a Jupyter Notebook since it allows for fast iterations, however, any environment or IDE will be fine.

First, install the `pmaw` and `pandas` packages, `pandas` will allow us to work with the data once it is retrieved using `pmaw` and export it to a CSV file.

```
pip install pmaw pandas
```

Import the packages and create an instance of the `PushshiftAPI` object with the default parameters.

```
import pandas as pd
from pmaw import PushshiftAPI

api = PushshiftAPI()
```

The `search_comments` method will be used to request Reddit comments from Pushshift. Using keyword arguments we define a date range (with `before` and `after`) `subreddit`, and a `limit` on the number of comments returned. The number of comments returned will be ≤ `limit` depending on how many comments are available on Pushshift for the provided arguments.

The `before` and `after` arguments in `pmaw` only accept dates in the epoch time format, which is the number of seconds that have elapsed since 00:00:00 UTC on Jan 1, 1970.

```
import datetime as dt
before = int(dt.datetime(2021,2,1,0,0).timestamp())
after = int(dt.datetime(2020,12,1,0,0).timestamp())
```

We convert our dates (Dec 1/20 and Feb 1/21) to epoch time using the `datetime` library. The `datetime` method accepts a date in the following format `year`, `month`, `day`, `hour`, `minute`, `second`, `microsecond` . `timestamp()` is used to return the epoch time value for the `datetime` objects that were created.

Now that we have defined our keyword arguments, we are ready to start retrieving comments from Pushshift.

```
subreddit="wallstreetbets"
limit=100000

comments = api.search_comments(subreddit=subreddit, limit=limit,
before=before, after=after)

print(f'Retrieved {len(comments)} comments from Pushshift')
```

This will take some time to run as we will be retrieving 100,000 comments. Under the hood, `pmaw` makes several API requests to Pushshift that each return a maximum of 100 comments, with requests being subjected to a rate-limit of 60 requests per minute. In optimal conditions, this would take 27m 46s to complete, however, requests are often

rejected and need to be re-sent.

When `search_comments` has finished retrieving the comments from Pushshift, a `Response` generator object will be returned. Using `pandas` we create a comment `DataFrame` with the `Response`.

```
comments_df = pd.DataFrame(comments)

# preview the comments data
comments_df.head(5)
```

The final step of creating this Reddit dataset is storing it locally so the data for future use. The comments can be saved in a CSV file with the `to_csv` method.

```
comments_df.to_csv('./wsb_comments.csv', header=True, index=False,
columns=list(comments_df.axes[1]))
```

Congrats, you just created and saved your first Reddit dataset. You can load the CSV into your future programs with `pandas` for any downstream processing or analysis that you'd like to perform. Using `pmaw`, and following steps similar to these, I was able to extract 700,000 posts and 9.5 million comments from r/WallStreetBets, that you can find on Kaggle. In the next article, I'll be reviewing how you can use `pmaw` to create Reddit datasets (ranging from 500 thousand to 10 million items) while avoiding potential memory errors.

If you run into any problems during this process, please open an issue on the `pmaw` GitHub, leave a comment, or reach out to me on LinkedIn.

Reddit Data API Python Data Science