COMP 598 Homework 6 – API Data Collection

40 pts
Assigned Oct 28, 2021
Due Nov 5, 2021 @ 11:59 PM

Non-standard (i.e., built-in) python libraries you can use:

- pandas
- requests and json
- beautifulsoup

Task 1: Data Collection & Bias (20 pts)

You've been given the task of assessing the average length of a Reddit post title. But since we can't collect ALL Reddit posts, we're going to try two different ways of sampling posts to assess their length:

- Sample 1: Collect the 1000 newest posts from the 10 most popular subreddits by subscribers: funny, AskReddit, gaming, aww, pics, Music, science, worldnews, videos, todayilearned. This means you collect 100 posts from each.
- Sample 2: Collect the 1000 newest posts from the 10 most popular subreddits by # of posts by day: AskReddit, memes, politics, nfl, nba, wallstreetbets, teenagers, PublicFreakout, leagueoflegends, unpopularopinion. This means you collect 100 posts from each.

Compute average post title length (measured as number of characters/post text) for each scenario.

This task should be split into two scripts: **`collect.py` and `compute_title_lengths.py`**

collect.py does the work of collecting the data for BOTH of the sampling approaches. It should store the data (as received from Reddit) in files sample1.json and sample2.json (respectively). **Reddit's API wraps posts into a key called "children", so sample1.json and sample2.json files must store the contents of the children tag**. Collecting and dumping posts into these files can work however you like – we will not grade it directly.

The generated sample JSON files should look like this:
https://gist.github.com/hannelita/eb06c75a7b98cc2e16cb8727039e4911

compute_title_lengths.py should accept an input file containing one JSON dict per line (corresponding to a subreddit post) and output the average post title length – in essence, it should accept one of the sample.json files produced by the collect.py script. The input JSON dict should respect ***exactly*** the format returned by reddit's API. The script is called like: python3 compute_title_lengths.py <input_file>.

Both scripts should be placed under the src/ folder of the template. Check the README.md for hw6 for further instructions.

Important aside: Once you've implemented the scripts and computed the average title lengths, reflect for a moment about how these two quite reasonable definitions of random posts have produced different results (this is ungraded – but an important part of this task). In a real project, you'll have to pick and be prepared to defend only one approach. This is the challenge that data scientists run into all the time. Drop in for office hours and let me know which you'd pick and why. Or is there another approach that you think would be better? I'm eager to discuss and debate.

Tips:

- If the post title doesn't contain any text, it still counts as a post … just with text length of zero.
- Note that you are collecting POSTS, not comments (which are the responses to posts).
- To collect the posts you'll want to use the /new API endpoint.
- make sure to set the User-Agent in your get requests … see API guidelines here: https://github.com/reddit-archive/reddit/wiki/API.
- In order to get this data, you may need to authenticate to Reddit. Instructions here: https://github.com/reddit-archive/reddit/wiki/OAuth2
- The libraries requests + json will be your friends.

Task 2: Scraping relationships (20 pts)

In lecture, we began work on a system for scraping the whosdatedwho website. Here, you need to finish that system.

Write a script collect_relationships.py that collects the relationships for a set of celebrities provided in a JSON configuration file as follows:

```
python3 collect_relationships.py -c <config-file.json> -o <output_file.json>
```

where config-file.json contains a single JSON dictionary with the following structure (the exact path and list of celebrities can, obviously, change):

```
{
      "cache_dir": "data/wdw_cache",
      "target_people": [ "robert-downey-jr", "justin-bieber" ]
}
```

In the configuration example above, we are targeting two people, "robert-downew-jr" and "justin-bieber", and we are caching their info under data/wdw_cache.

Your script will then go and fetch the relationships for the target individuals. Note that the target people are indicated using the identifier that follows **"/dating/**". All pages visited MUST be cached in the cache directory specified – as described in the lecture. This means that, if run twice on the same config file, it will use data exclusively from the cache the second time.

The output format for the file is:

```
{
      "robert-downey-jr": [ "person-1", "person-2", "person-3" ],
      "justin-bieber": []
}
```

Where the identifiers in the list are the people the person had a relationship with. If the person has had no relationships, then they will have an empty list.

Submission Instructions

Please check hw6 README.md file and directory.