CS 432/532 Web Science: Assignment #2

Alexander C. Nwala

Bethany DeMerchant
16 February 2019

Contents

Problem 1:

Write a Python program that extracts 1000 unique (collect more e.g., 1300 just in case) links from Twitter. Omit links from the Twitter domain (twitter.com). Also note that you need to verify that the final target URI (i.e., the one that responds with a 200) is unique. You could have many different shortened URIs for www.cnn.com (t.co, bit.ly, goo.gl, etc.). You might want to use the streaming or search feature to find URIs. If you find something inappropriate for any reason you see fit, just discard it and get some more links. We just want 1000 links that were shared via Twitter.

Solution 1:

Links were collected using the `Tweepy` library[1] for Python. `Tweepy`'s `StreamListener` was used to stream Tweets. The `entities` objects of received Tweets were then checked for a URI. If a URI was present and did not include "twitter.com/", it was added to a list of URIs. The process was repeated until the length of the list was greater than 1500 items.

Links were checked for uniqueness and validity using the `Requests` library[2]. Each link was followed, including redirects, until receiving status code 200. The link resulting in a 200 code was then added to a list. After checking all links, the list was then converted to a dictionary and back to remove any remaining duplicated, as suggested by W3Schools[3]. After removing duplicates and nonexistent links, the result was a list of 1359 links.

Problem 2:

Download the TimeMaps for each of the target URIs. Create a histogram of URIs vs. number of Mementos (as computed from the TimeMaps). For example, 100 URIs with 0 Mementos, 300 URIs with 1 Memento, 400 URIs with 2 Mementos, etc. The x-axis will have the number of mementos, and the y-axis will have the frequency of occurrence.

Solution 2:

TimeMaps were downloaded using `curl http://memgator.cs.odu.edu/timemap/link/[URI]`. A Python script was used to run the command, capture the result, and output it to a file for each URI. Each TimeMap was then checked and occurrences of the term "memento" counted. The structure of the TimeMap means that the term "memento" appears only within appropriate `rel=` tags and the number of occurrences can be used as the number of mementos. A histogram was then made using `Matplotlib`[4] to show the frequency of each number of mementos.

As seen in Figure 1, most pages returned few mementos—1086 of 1359 returned no mementos. Figure 2 shows a detailed view of pages which returned between 0 and 20 mementos, representing 95% of the data.

[1] http://docs.tweepy.org/en/3.7.0/
[2] http://docs.python-requests.org/en/master/
[3] https://www.w3schools.com/python/python_howto_remove_duplicates.asp
[4] https://matplotlib.org/index.html

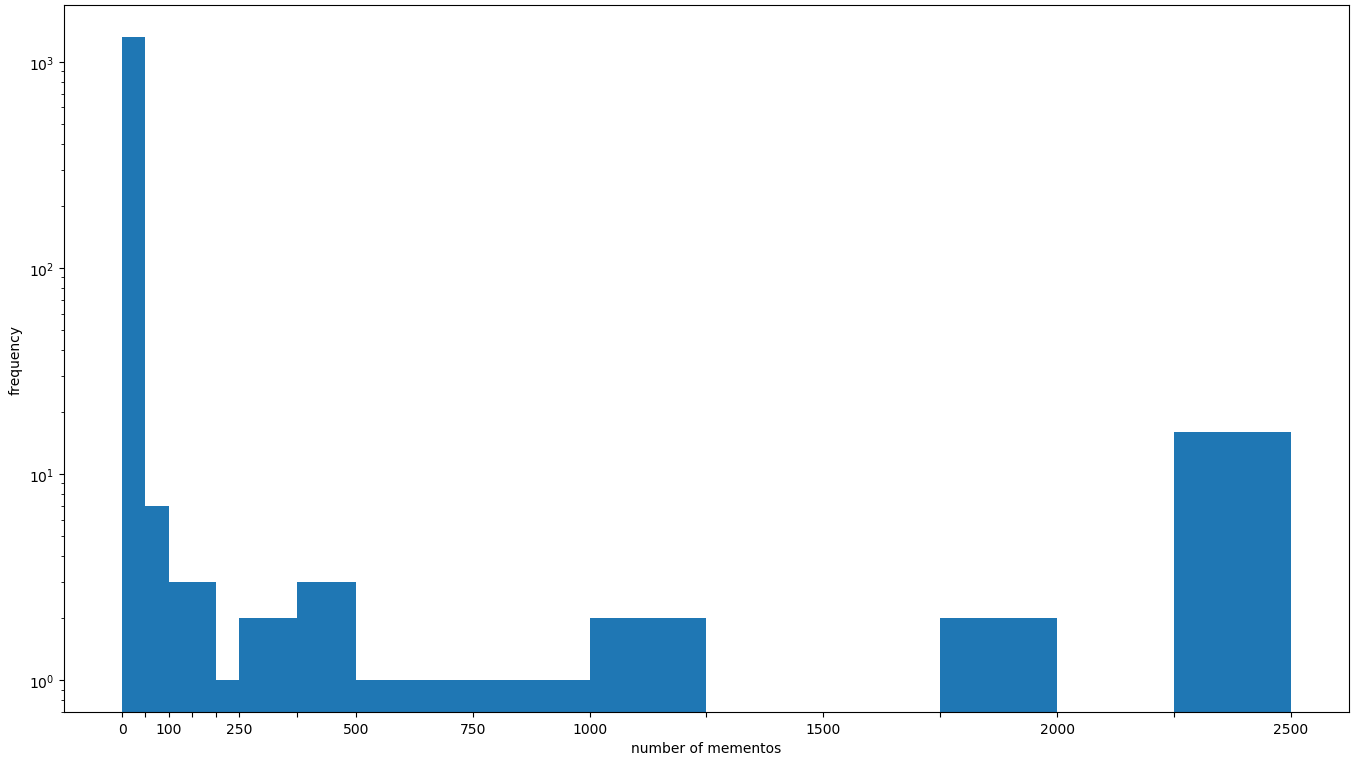

Figure 1: Histogram of all pages

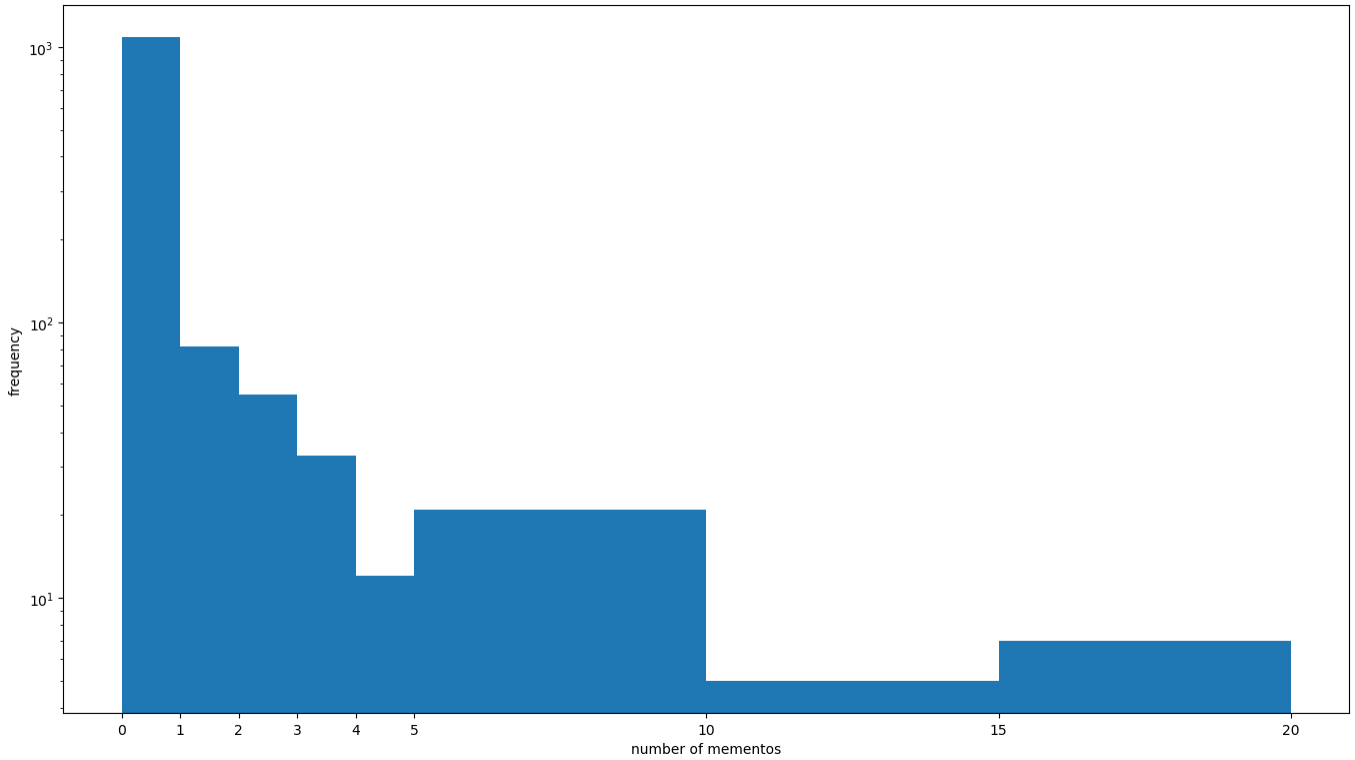

Figure 2: Histogram of pages with 20 or fewer mementos

Problem 3:

Estimate the age of each of the 1000 URIs using the "Carbon Date" tool. Note: you should use "docker" and install it locally. For URIs that have > 0 Mementos and an estimated creation date, create a graph with age (in days) on the x-axis and number of mementos on the y-axis. Not all URIs will have Mementos, and not all URIs will have an estimated creation date. Show how many URIs fall into either category.

Solution 3:

CarbonDate data was collected using:

```
docker container run --rm -it oduwsdl/carbondate ./main.py -l [URI]
```

A Python script was used to run the command, capture the result, and output it to a file for each URI. The result was then loaded as a JSON object and the URI and estimated creation date were recorded. The date was then compared to the date the data was downloaded[5] to determine the age of the page in days. `Matplotlib` was then used to plot the age of each page as related to the number of mementos available for that page. Figure 3 shows that there is a loose correlation between age and number of mementos—that is, older pages are more likely to have more mementos, and much more likely to have at least one.

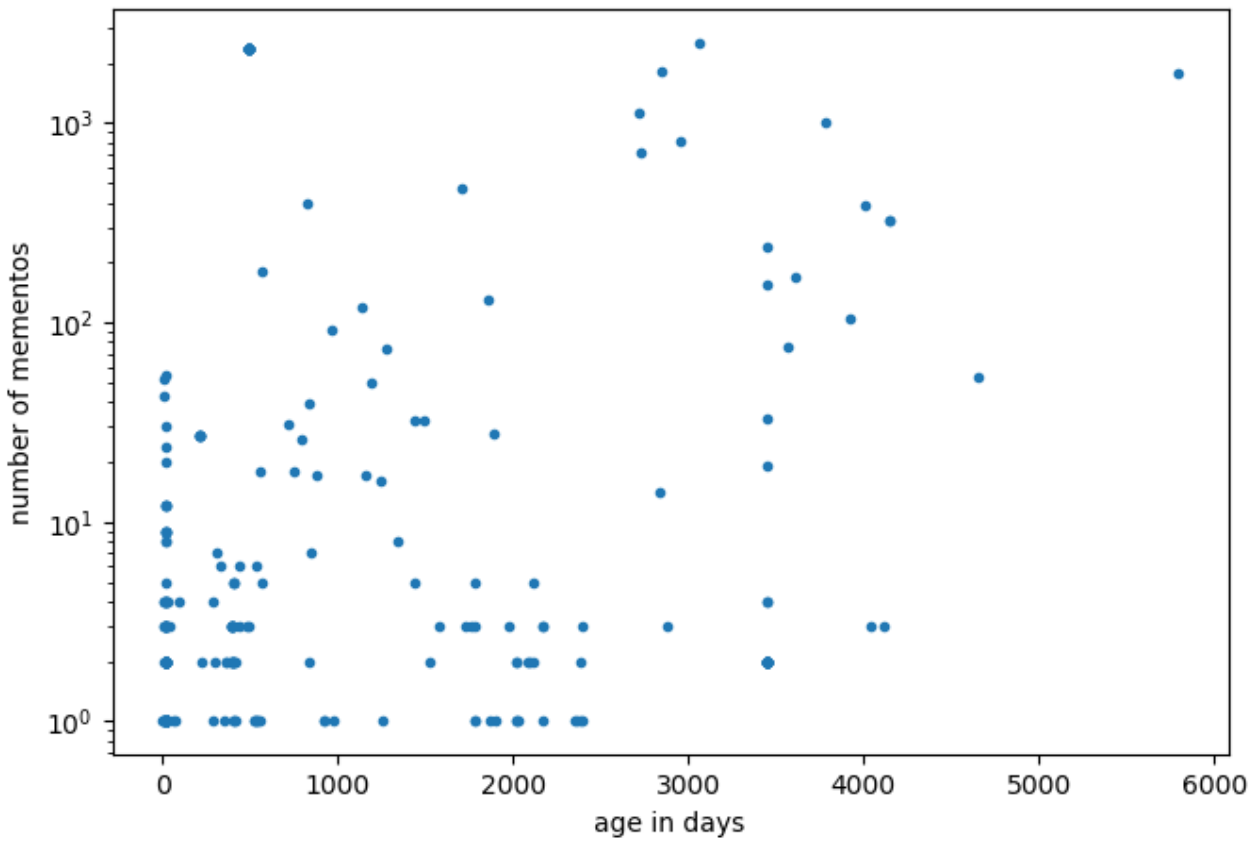

Figure 3: Age of page and number of mementos

[5] I actually downloaded it at a reasonable date, I promise! But I messed some things up and ended up having to do it over again, so I just used that date.