Aaron Shan ID: 920177055

Jinzhuang Li ID: 919483572

ECS 152A

11/25/2024

Code files:

1. Crawl_har.py: This is part two, this file is to crawl har files through the top-1m.csv
2. Crawl_har_withGPT: This is part two, this is to crawl har files with gpt suggestions
3. find_3rdParty_cookies.py: This file is to find the top 10 third-party domain and cookies
4. find_3rdParty_cookies_withGPT.py: This file is to find the top 10 third-party domain and cookies with gpt suggestions
5. output(folder): this is please to store 1533 har files and the log file.

Part 1. Crawler har files

1. Based on the basic structure of selenium_proxy.py from the week 6 discussion and any code from the week 6 discussion, we can have a basic understanding of the logic of the crawler.
2. We are going to store the process in the log file. This is because we need to save proof in a file for any future checking. There is no way or possible just to print out the process on the terminal. It will cause my laptop to crash (it crashed due to this type of issue) so we have to write a function to set up logging.

3. Due to the question asking, we need to automatically crawl the top 1000 websites from the top-1m.csv file therefor we need to write a function to read sites from top-1m.csv and we need to create a path that lets our program access the file.
4. However, there are some websites that we can't access (crawl), this will lead to the read time out. Therefore, we set 1800 as the number of websites that we need to access.
5. In the terminal result, when we check how the process goes or make sure there is no error when the code is running to crawl, so when we read every 100 websites, we will show a report to see how many HAR files (websites we successfully access) and how many failed and we will also show the progress status. We are adding this to show in the terminal is to make sure the code is running to the end. After the crawling is completed, we will store the har files in the path: output/har_files. But the progress will after will get some error because there are some websites were failed to reached so it will lead to the wrong number

```
aaronshan@AarondeMacBook-Pro-2 hw2 % python3 part1.py
Successfully loaded 1800 sites
Starting crawl process...
Progress: 100/1800 sites processed
Successful: 96, Failed: 4
Progress: 200/1800 sites processed
Successful: 188, Failed: 12
Progress: 300/1800 sites processed
Successful: 281, Failed: 19
Progress: 400/1800 sites processed
Successful: 369, Failed: 31
Progress: 500/1800 sites processed
Successful: 455, Failed: 45
Progress: 600/1800 sites processed
Successful: 545, Failed: 55
Progress: 700/1800 sites processed
Successful: 641, Failed: 59
Progress: 800/1800 sites processed
Successful: 737, Failed: 63
Progress: 1100/1800 sites processed
Successful: 1006, Failed: 94
Progress: 1200/1800 sites processed
Successful: 1098, Failed: 102
Progress: 1400/1800 sites processed
Successful: 1213, Failed: 187
Progress: 1500/1800 sites processed
Successful: 1306, Failed: 194
Progress: 1600/1800 sites processed
Successful: 1401, Failed: 199 Progress: 1800/1800 sites processed
Successful: 1533, Failed: 267

Crawling completed!
Total sites processed: 1800
Successful crawls: 1533
Failed crawls: 267
HAR files saved in: output/har_files
aaronshan@AarondeMacBook-Pro-2 hw2 %
```

6. Because we store all the progress reports in the log file, we can make sure each site can successfully be accessed and also we can know which websites failed to crawl. (please check the log file for the complete report)

   Here is an example:

```
2024-11-24 01:32:46,900 - INFO - Crawling 1757/1800: http://moviesda8.com
2024-11-24 01:34:46,913 - ERROR - Error crawling moviesda8.com:
HTTPConnectionPool(host='localhost', port=62598): Read timed out. (read timeout=120)
2024-11-24 01:34:49,225 - INFO - Crawling 1758/1800: http://onlar.az
2024-11-24 01:36:47,106 - INFO - Successfully saved HAR for http://onlar.az
```

Part 2: Identify the top 10 most commonly seen third-party across all sites and Identify the top 10 cookies

1. For the second part, we need to analyze the HAR files that we crawl.
2. In the code, we need to check the second-level domain and then we check the third party.
3. In addition, we are running the code on the Mac, I checked with Google and ChatGPT, I need to set SSL context for macOS to run my code.
4. During my original implementation, my idea was to find third-party and cookies by a theory that is “counting every instance of a third-party cookie and every instance of a third-party domain request” (I asked ChatGPT, it provide some recommendations of label and counting information – please check below for GPT section for part 2)
5. The total har files that we have are 1533 files / 1800 har files that we read through.
6. To make sure that we process the code correctly we did the same steps as what we did for part one which is we set a mark on each 100 to see if the code can analyze and read all

the files correctly. Then we get our answers.

```
aaronshan@AarondeMacBook-Pro-2 hw2 % python3 part2.py

Analyzing 1533 HAR files...
Processed 100/1533 files
Processed 200/1533 files
Processed 300/1533 files
Processed 400/1533 files
Processed 500/1533 files
Processed 600/1533 files
Processed 700/1533 files
Processed 800/1533 files
Processed 900/1533 files
Processed 1000/1533 files
Processed 1100/1533 files
Processed 1200/1533 files
Processed 1300/1533 files
Processed 1400/1533 files
Processed 1500/1533 files

=== Top 10 Most Common Third-Party Domains ===
Rank  | Domain                                  | Occurrences
------------------------------------------------------------
1     | www.googletagmanager.com                |        2246
2     | pagead2.googlesyndication.com           |        2223
3     | static.flashscore.com                   |        1778
4     | m.media-amazon.com                      |        1452
5     | static.xx.fbcdn.net                     |        1306
6     | www.google-analytics.com                |        1220
7     | securepubads.g.doubleclick.net          |        1175
8     | www.googleadservices.com                |        1167
9     | www.google.com                          |        1126
10    | cdn.cookielaw.org                       |        1085

=== Top 10 Most Common Third-Party Cookies ===
Rank  | Cookie Name                             | Occurrences
------------------------------------------------------------
1     | ar_debug                                |         676
2     | IDE                                     |         408
3     | datr                                    |         171
4     | sb                                      |         168
5     | fr                                      |         168
6     | ps_n                                    |         163
7     | MR                                      |         151
8     | MUID                                    |         237
9     | SRCHUSR                                 |         149
10    | _SS                                     |         150
aaronshan@AarondeMacBook-Pro-2 hw2 % python3 part2_2.py
```

7. Double claims: we follow the idea of "count the same cookie multiple times from the same site if it appears in multiple requests" to get the cookie. After ChatGPT and using the recommendations from it, we get other answers.
8. **Let's talk about the intended functionality by referencing Cookiepedia**
    - Ar_debug: there is not yet any general information about this cookie based on its name only. As shown on Cookiepedia, the main purpose of this cookie is unknown.
    - IDE: This domain is owned by Doubleclick (Google). The main business activity is: Adwords is Google's real-time bidding advertising system or platform. The main use of this cookie is to provide a user with relevant ads therefore it is a Targeting/Advertising cookie.
    - Datr: This cookie name is related to Facebook and is used to determine the immediate web browser which is used to connect to Facebook irrespective of the user logged in and is the primary cookie for Site Integrity and Security. The overall function of this cookie is for the strictly necessary purpose.
    - Sb: There is no general information about this cookie and the main purpose is unknown
    - Fr: This advertising cookie lodged in the Facebook platform is responsible for a host of advertising products including real-time bidding from third-party advertisers. The main purpose of this cookie is a Targeting/Advertising cookie.
    - Ps_n: There is no general information about this cookie and the main purpose is unknown

- Mr: This is a Microsoft MSN 1st party cookie which we apply to evaluate the utilization of the site for statistics purposes. This cookie stands for performance mainly.
- Muid: This cookie is employed by Microsoft principally as a unique user identifier. It can also be set by embedded Microsoft scripts. Thought to share across a host of domains that belong to the Microsoft platform and track users. The main goal of this kind of cookie is to perform as an analytics cookie, as well as Targeting/Advertising.
- Srchusr: This is a Microsoft MSN 1st-party cookie that assures the website of a correct performance on the platform. The main image of this cookie is Functionality.
- _SS: This is a Microsoft MSN 1st-party cookie that provides assurance to the website for a correct performance on the platform. The main image of this cookie is Functionality.

Part 1: Crawler har files with Chatgpt

- Based on the long time running for collecting the HAR files, we need to make sure our code can run during the original testing. To avoid some huge issues, our output information matter in the log file that we created, or the output on the terminal, is the same.
- Based on what we asked for GPT, we made some suggestions, we added more error handling to make sure we collect all the files correctly and access the website. When we

successfully access the website, we add a new message that will be written into the log file which is the code “ logging.info(f” Successfully saved HAR file for {url}.”

- Most of what we changed is the writing format of error handling to improve the readability. Even though I gave the code to GPT to let it suggest some ideas we only accepted a few of them to write another version with GPT ideas.
- Here is the output that we get (we make it be same as the output of the original one.

```
aaronshan@AarondeMacBook-Pro-2 hw2 % python3 part1.py
Successfully loaded 1800 sites
Starting crawl process...
Progress: 100/1800 sites processed
Successful: 96, Failed: 4
Progress: 200/1800 sites processed
Successful: 188, Failed: 12
Progress: 300/1800 sites processed
Successful: 281, Failed: 19
Progress: 400/1800 sites processed
Successful: 369, Failed: 31
Progress: 500/1800 sites processed
Successful: 455, Failed: 45
Progress: 600/1800 sites processed
Successful: 545, Failed: 55
Progress: 700/1800 sites processed
Successful: 641, Failed: 59
Progress: 800/1800 sites processed
Successful: 737, Failed: 63
Progress: 1100/1800 sites processed
Successful: 1006, Failed: 94
Progress: 1200/1800 sites processed
Successful: 1098, Failed: 102
Progress: 1400/1800 sites processed
Successful: 1213, Failed: 187
Progress: 1500/1800 sites processed
Successful: 1306, Failed: 194
Progress: 1600/1800 sites processed
Successful: 1401, Failed: 199 Progress: 1800/1800 sites processed
Successful: 1533, Failed: 267

Crawling completed!
Total sites processed: 1800
Successful crawls: 1533
Failed crawls: 267
HAR files saved in: output/har_files
aaronshan@AarondeMacBook-Pro-2 hw2 %
```

- Here are some detail in the log files:

```
2024-11-23 23:43:14,185 - INFO - Crawling 1701/1800: http://avto.net
2024-11-23 23:43:15,225 - INFO - Successfully saved HAR for http://avto.net
2024-11-23 23:43:15,232 - INFO - Crawling 1702/1800: http://82xnxx.com
2024-11-23 23:44:42,246 - ERROR - Error crawling 82xnxx.com: Message: timeout: Timed out
receiving message from renderer: 28.880
```

Part2: Identify the top 10 most commonly seen third-party across all sites and Identify the top 10 cookies with CHATGPT

- After we updated the code based on the idea and needed steps from CHATGPT, we updated the code. In the updated code, we follow the idea provided by ChatGPT, which is to "count each site only once per cookie, regardless of how many times the cookie appears on that site."
- From the original implementation, we get

```
        for cookie_full, count in self.all_third_party_cookies.most_common():
            domain, name = cookie_full.split(':', 1)
            if name not in seen_names:
                seen_names.add(name)
                total_occurrences = self.cookie_name_count[name]
                print(f"{rank:<5d} | {name:<35} | {total_occurrences:>11d}")
                rank += 1
                if rank > 10:
                    break
```

We count total occurrences of cookies but combine all instances from the same domain

- From the GPT idea implementation, we get:

```
    # Convert cookie site counts to a sorted list
    cookie_counts = [(cookie, len(sites)) for cookie, sites in self.cookie_sites.items()]
    cookie_counts.sort(key=lambda x: x[1], reverse=True)
```

We count the number of unique sits where each cookie appears

- We also based on the suggestion made some changes to the code. In the original code,

```
def __init__(self, har_dir='output/har_files'):
    self.har_dir = har_dir
    self.all_third_parties = Counter()
    self.all_third_party_cookies = Counter()
    self.cookie_name_count = Counter()  # New counter for cookie names only
```

But in the GPT idea implementation, we change to

```
def __init__(self, har_dir='output/har_files'):
    self.har_dir = har_dir
    self.all_third_parties = Counter()
    # Track which sites each cookie appears on
    self.cookie_sites = {}
```

this lead to clear readability and more clearer version.

There are many changes between the two files, please check the code file.

- Here are the output will be shown:

```
aaronshan@AarondeMacBook-Pro-2 hw2 % python3 part2_2.py

Analyzing 1533 HAR files...
Processed 100/1533 files
Processed 200/1533 files
Processed 300/1533 files
Processed 400/1533 files
Processed 500/1533 files
Processed 600/1533 files
Processed 700/1533 files
Processed 800/1533 files
Processed 900/1533 files
Processed 1000/1533 files
Processed 1100/1533 files
Processed 1200/1533 files
Processed 1300/1533 files
Processed 1400/1533 files
Processed 1500/1533 files

=== Top 10 Most Common Third-Party Domains ===
Rank  | Domain                                  | Occurrences
-------------------------------------------------------------
1     | www.googletagmanager.com                |        2246
2     | pagead2.googlesyndication.com           |        2223
3     | static.flashscore.com                   |        1778
4     | m.media-amazon.com                      |        1452
5     | static.xx.fbcdn.net                     |        1306
6     | www.google-analytics.com                |        1220
7     | securepubads.g.doubleclick.net          |        1175
8     | www.googleadservices.com                |        1167
9     | www.google.com                          |        1126
10    | cdn.cookielaw.org                       |        1085

=== Top 10 Most Common Third-Party Cookies ===
Rank  | Cookie Name                             | Number of Sites
-----------------------------------------------------------------
1     | ar_debug                                |         676
2     | _ga                                     |         474
3     | __cf_bm                                 |         413
4     | IDE                                     |         408
5     | receive-cookie-deprecation              |         242
6     | MUID                                    |         237
7     | _gcl_au                                 |         234
8     | _gid                                    |         205
9     | datr                                    |         171
10    | fr                                      |         168
```

Based on the data, we can see the top 10 most common third-party domains are the same but the top 10 cookies are different.

- Based on the data, we think the GPT idea implementation approach (counting by sites) is probably more meaningful for analyzing cookie prevalence across the web, as it shows how many different websites use each cookie rather than how many times each cookie appears.
- Now let’s talk about what each cookie means based on the Cookiepedia

- Ar_debug: there is not yet any general information about this cookie based on its name only. As shown on Cookiepedia, the main purpose of this cookie is unknown.
- _ga: we found this cookie but there are no matches on the Cookiepedia. But we searched on Google, it said _ga is used for long-term user identification.
- __cd_bm: we found this cookie but there are no matches on the Cookiepedia. But we searched on Google, it said it is a cookie used on end-user devices by Cloudflare to support the Bot Management
- IDE: This domain is owned by Doubleclick (Google). The main business activity is: Adwords is Google's real-time bidding advertising system or platform. The main use of this cookie is to provide a user with relevant ads therefore it is a Targeting/Advertising cookie.
- Receive-cookie-deprecation: There is no general information about this cookie and the main purpose is unknown
- MUID: This cookie is employed by Microsoft principally as a unique user identifier. It can also be set by embedded Microsoft scripts. Thought to share across a host of domains that belong to the Microsoft platform and track users. The main goal of this kind of cookie is to perform as an analytics cookie, as well as Targeting/Advertising.
- _gcl_au: this is utilized by Google AdSense to test advertisement effectiveness traversing from a specific Web page that utilizes their services. The main use of this cookie is for Targeting/Advertising.

- _gid: On Cookiepedia, it only shows the main purpose of the cookie is performance. But if we search on Google, it is for advertising purposes like serving and rendering ads or personalizing ads.
- Datr: This cookie name is related to Facebook and is used to determine the immediate web browser which is used to connect to Facebook irrespective of the user logged in and is the primary cookie for Site Integrity and Security. The overall function of this cookie is for the strictly necessary purpose.
- Fr: This advertising cookie lodged in the Facebook platform is responsible for a host of advertising products including real-time bidding from third-party