Web Scraping with Scrapy

SCC.413: Applied Data Mining

Week 14, 6th February.

Here we will use a popular Python web scraping tool, **Scrapy**, to spider websites, in this case forums, and download a dump of HTML for future processing.

1. Ensure you have downloaded all of the associated code (either from the git repository or Moodle space). Open a terminal and cd to the scrapy lab folder. This is probably best placed in your h-drive.

2. We will be using a python web scraping library called Scrapy. This should hopefully be already set up, but if it's not, simply run this command:

   ```
   $ pip install scrapy
   ```

   If you're still having trouble, there's installation instructions here https://doc.scrapy.org/en/latest/intro/install.html#intro-install. Scrapy has pretty good documentation and an official tutorial which should help if you're stuck (https://doc.scrapy.org/en/latest/intro/tutorial.html).

3. Inside the directory you made, set up a scrapy project by running this command:

   ```
   $ scrapy startproject <name of project>
   ```

   If you cd into the created folder, you will see that it has created various files and folders for you.

4. Now copy the two spiders we have provided (*flat_earth_spider.py* and *forum_spider.py*) into the "spiders" directory. This folder is where you should put all spiders that you make. Bear in mind that every spider should subclass "scrapy.Spider".

5. You should scrape websites responsibly. You don't want to accidentally take the site down! We can bear this in mind by applying rate limiting: limiting the rate of requests we send. For this exercise, we will limit the number of concurrent requests to 1. We also want to follow the "robots.txt" policy of the site we are scraping so we do not attempt to access forbidden pages. To do both these things, add the following two lines to "settings.py":

```
ROBOTSTXT_OBEY = True
CONCURRENT_REQUESTS = 1
```

6. We've provided you with a spider called "forumspider" in the file "forum_spider.py". This will scrape the first page of a forum board and dump the first page of each topic into a folder called "forum-dump".

   Before you run the spider, you'll have to specify the forum you want to scrape. The spider should work with any Simple Machines Forum (smf) site. We've provided a list of a few examples you can try in the file "list-of-boards.txt". Simply copy one of these URLs into the urls list in the "start_requests" method.

   To run the spider simply enter this command into the terminal while inside your directory.

   ```
   $ scrapy crawl forumspider
   ```

7. You should see a folder has been created containing dumps of the topics from this board. Now have a look at the code and try and understand how it works.

   The "start_requests" method tells the scraper what to do to begin with. In this case we just yield a request of the specified url[1]. This request is given a callback function which is the function called to handle the response to this request.

   The "board_parse" and "topic_parse" functions are called to handle responses. The "board_parse" method loops through all the links on a board and makes a request for each link to a topic, giving the "topic_parse" callback function, and then dumps itself into a file. The "topic_parse" method only dumps to a file.

8. Currently the spider only scrapes the first page of each topic from the first page of the board. Your task is to extend this spider to scrape multiple pages of topics and/or boards. You do not have to scrape absolutely everything, just be able to go beyond the first page.

   To do this you will need to work out how to locate and make a request for the URL of the next page. The Scrapy tutorial (linked above) will be a useful guide, particularly in how to get elements from the html.

9. **Advanced:** There are many more ways to improve this spider. Feel free to try out some of the other ways you could extend it, e.g.:

   (a) Getting multiple boards.

   (b) Changing it for other forums that aren't smf.

   (c) Applying the spider to other websites that aren't forums.

[1] Don't panic at the use of yield instead of return. Yield is used when a function returns a generator object. Read this for more info: https://pythontips.com/2013/09/29/the-python-yield-keyword-explained/. Don't worry about it too much, just think of it as behaving similarly to return.

10. Now you've had a go at a basic spider, you may be interested to see a more general one. We've provided a file called "flat_earth_spider.py" which scrapes an entire forum, following every link and dumping everything it wants into a folder. This is useful if you want to see every accessible page on a website, in this case a forum, but you don't necessarily know its exact structure. The code should run the same way as the previous example but with the name "flatearth". It's a bit rough and ready, and there will be some errors. Scrapers need to be resilient to errors as they will happen often. Thankfully scrapy handles most of this for us. Currently it's tailored towards a specific forum, but the principles should be applicable to any site.