Crawling Exercise

Rita Pessoa Correia

March 19, 2021

1 Framework, Programming Language and IDE

In order to accomplish this exercise goal of developing a simple crawling for Wikipedia, I decided to use Selenium framework. I chose this software, because I've already worked a little bit with it, so I have some basic knowledge about its usage and tools.

The first step was to download the Selenium WebDriver (`https://chromedriver.chromium.org/downloads`) according to my Chrome's version and operative system.

Then, as the programming language, I decided to use Python, because it's one of the languages that I am more comfortable with. Thus, and because Python 3 was already present in the local machine, I had do install the selenium package through `pip3`, with the following command: `pip3 install selenium`.

As the IDE (Integrated Development Environment) I used VSCode, because I think it's intuitive, clean, light and has some good extensions.

2 Program Developing

After making the necessary imports for my Python program, I implemented my url starting point as the portuguese Wikipedia main page (`https://pt.wikipedia.org/wiki/Wikip%C3%A9dia:P%C3%A1gina_principal`), and defined a 10 seconds timeout.

After that, I divided the program in two parts:

1. **Get the number of references** to "Universidad Carlos III de Madrid" in Wikipedia;
2. **Get the number of students** extracted from Wikipedia.

Between almost each operation, I had to declare a `implicit_wait` because the crawling process happens too fast in the browser, and sometimes it doesn't give the page enough time to completely update and process.

2.1 Getting the Number of References

For this purpose, I had to manually go to the Wikipedia page, and inspect the element correspondent to the main search bar, and copy its `XPath`. After the selenium driver finds the element, it has to call the `click()` method, in order to automatically click on the bar, and then

in the `send_keys` method with "Universidad Carlos III de Madrid" as an argument, so it can write that phrase and search it after clicking the correspondent button in an identical way.

After clicking on the search button, the browser will already be on another page, relative to the search results of the given phrase. Then, I just had to find the `XPath` of the element that pointed to the number of references to "Universidad Carlos III de Madrid" in the Web page, and copy it into a variable to be printed on the terminal, as we can see if figure 1.

```
rpecorreia@MBP-de-Rita ~ % /usr/local/bin/python3 /Users/rpecorreia/Desktop/WebCrawling/main.py
1) 10 results
```

Figure 1: Printing the number of references to "Universidad Carlos III de Madrid" in Wikipedia.

2.2 Getting the Number of Students

To get the number of students according to Wikipedia, the process is quite similar. First, through the `XPath`, the program had to find and click on the element that opens the "Universidad Carlos III de Madrid" main Wikipedia page from the results page, to seek the number of students.

Then, and because the number of students is on the first paragraph of the "Universidad Carlos III de Madrid" Wikipedia web page, the program found that paragraph, again due to its `XPath`, and stored it on a variable called `search3`. Then, I stored in another variable called `selection`, the number of students, and printed it in the terminal, as we can see in figure 2.

```
rpecorreia@MBP-de-Rita ~ % /usr/local/bin/python3 /Users/rpecorreia/Desktop/WebCrawling/main.py
1) 10 results
2) 8.419 students
```

Figure 2: Printing the number of students according Wikipedia.