Logbook

**Thursday 8th December**

I met with Iris and discussed my plan for the project. She gave me some useful links for natural language processing that I had a look over. I still need to precisely formulate my project question/topic.

**Tuesday 13th December**

I have started collecting the data for my natural language processing project about books. It is closely inspired by the “Guardian top-100 novels” suggestion mentioned in the course, however, I want to get more books to work with.

I have found that amazon doesn’t support scraping their website (and the API is not completely free) so I need to find a different website. I have ended up with Goodreads (also as mentioned in the course) but I still want to get a lot more books to work with. I have found their “lists” of books to be nicely accessible for scraping a managed to get 1249 books (name, link to books page) from the largest list “to-read”. Even though it is said to contain over 2bn books, it turns out that goodreads website doesn’t bother showing more than the first 1249 books (25 pages with 50 books, last cutoff with only 49). This is still a good amount.

Major issue I have encountered is that the webpages of individual books are using javascript and so using BeautifulSoup alone won’t be enough there. I have therefore decided to include “learning the Selenium package” into my project and have started learning it.

**Wednesday 14th December**

I have learned how to use the “Selenium” library for scraping website that use javascript to load dynamically, which is exactly the case for the goodreads pages of individual books. I admit this was a big struggle and took me a long time. I have managed to load the goodreads website using the webdriver. Now I need to figure out how to scrape the data from it.

**Thursday 15th December**

I have learned how to use the “Scrapy” library its “Selector” class to find data on a website using xpaths. I have managed to write the code that scrapes all the info I want for a book, including the description. Now the issue is to make it consistent and error proof so that I can run a single script to scrape all ~1200 books at once and produce once csv file.

**Friday 16th December**

I have realized that the goodreads “shelves” of books contain duplicates. I have added a filter so only books with unique url links are stored. There still may be duplicates but those books will then have 2 different pages with 2 different comment sections. Looking at the output from the “to-read” shelf, there are now 1207 books with unique links (out of the 1249) and of those a quick look in excel suggests all are actually unique (sorting names by A to Z and looking at repeat rows).

I have also found that to be able to see 1249 books per shelf, you need to be logged in. I have used the selenium package to make my code log itself in so that when scraping it can access all the visible books in a list. This is not completely automated because sometimes there is a “im not a robot” check and then I need to fill that out myself before the script can continue. Previously I had no issue with this since I wasn’t scraping using the webdriver. I use the “goodreads” login before every scrape or crawl now because I am pretty sure goodreads has reduced what is visible with login to restrict scraping, including a different limit of how many request/min you can send to their server.

**Saturday 17th December**

I put it all together and made it error proof enough that I can run all crawl through all 1200 book pages at once and scrape them. I can now finally start working with the data.