COMP 479/6791
Information Retrieval and Web Search

Assignment 4

Demo File

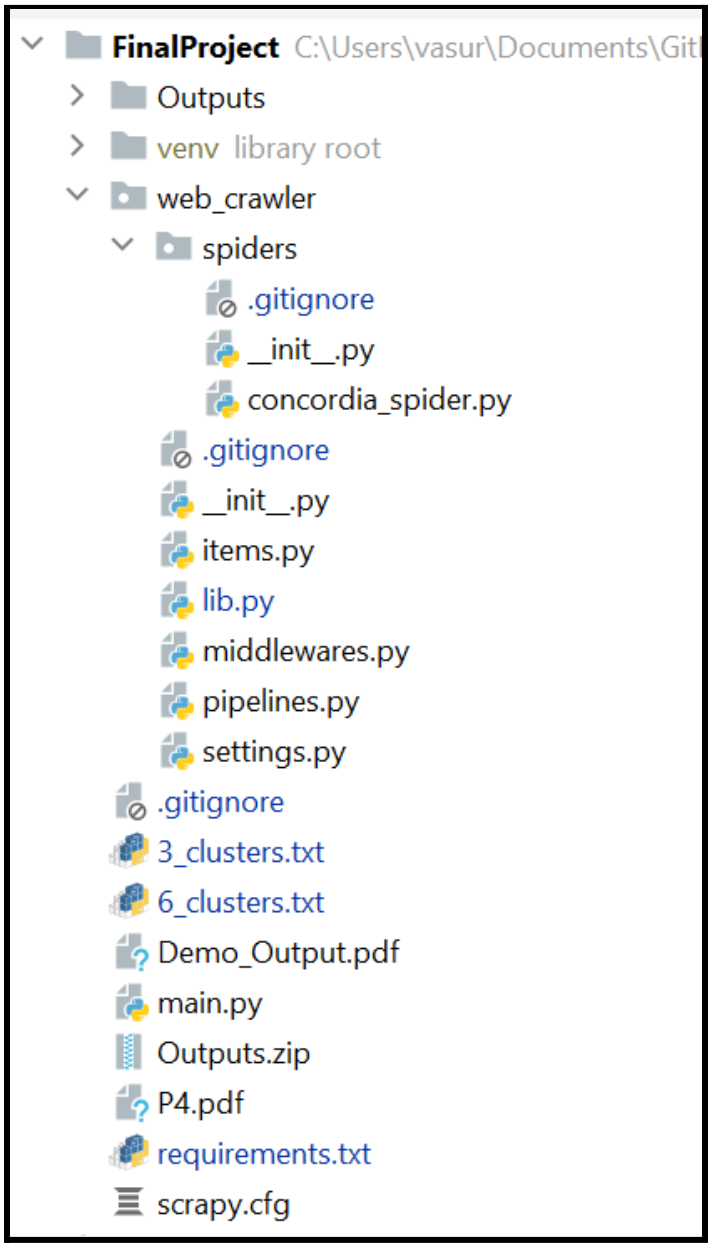
Under the guidance of Dr. Sabine Bergler

Vasu Ratanpara 40135264

December 2021

Project Files

These are the files that are part of my project. The work of each of the files is explained below. The working process of each file is mentioned in detail under the **Project Report**. This Document is more about how to use this project and which files contain what and why.



[The Project Structure]

Note: The project will scrap only https://concordia.ca. The Output will be generated under the "Outputs" folder automatically when you will run main.py.

1. main.py (Executable)

This file has a crawler, K-mean and Afinn all in one. When you will run it you will see the following output in the console. If you want to go with Default options just press Enter key to skip the input otherwise write appropriate input.

The 1st option asks if you want to run a crawler and crawl the website from starch. If you will skip this It will use HTML files from the Outputs folder which were written scrapped for testing. The testing set was built by scraping 100 pages from concordia.ca.

The Second Option will ask for how many words you want to print for each cluster. The default count is 50.

```
=> [If you want to use "Default" the just press Enter]

Do you want to run crawler? (otherwise, we will use data.json from previous run) [Default:No]:

Enter How many term you want to print per cluster? [Default:50]:
```

```
=> [If you want to use "Default" the just press Enter]

Do you want to run crawler? (otherwise, we will use data.json from previous run) [Default:No]:

Enter How many term you want to print per cluster? [Default:50]:10
```

2. concordia_spider.py

This file has the main crawling logic. It will be run by the crawler process in main.py. It will start the scrapper from https://concordia.ca and it will keep scrapping the pages till It will reach the upper bound which is set by the user at run time. From each page, it will extract all links for other HTML pages from the current pages and It will save those URLs to process further. Each page will be saved under the Output in the HTML files folder.

3. pipelines.py

This file will save the item object which is made of title, URL and body of HTML page. It will open the data.json file which may contain the data from other pages and append the information of the current page into data.json. Which will be used by k-means and Afinn.

4. lib.py

This is a general-purpose file that contains the functions which are commonly used as helper functions. It has IO functions and run-time measuring functions.

5. Outputs (Folder)

This is the folder where you will find the final outputs of the crawler, HTML files and data.json file which contains the data from all pages which were crawled.

Thanks for reading my demo file carefully. If you have any questions feel free to email me at vasu.ratanpata@mail.concordia.ca or on Moodle. I would be more than happy to guide you with my work. My testing output is attached as a PDF named Demo_Output.pdf.