**COMP 479 PROJECT 4 REPORT**

**Jixuan Li 40073785**

1. Crawling tools

In the project, I choose crawler4j as the crawling tool.

The configuration is shown here:

//save the crawled page to ./data

config.setCrawlStorageFolder("./data");

config.setPolitenessDelay(50);

config.setMaxDepthOfCrawling(15);

//I crawled 2 differnet set of pages,

//one for development, size = 50 “./development\_data”

//one for production,size = 20000 “./Project\_4\_Java\_crawl/data”

config.setMaxPagesToFetch(20000);

config.setIncludeBinaryContentInCrawling(false);

config.setResumableCrawling(false);

//here is the code of following robot exclusion rule

PageFetcher pageFetcher *= new* PageFetcher(config);  
RobotstxtConfig robotstxtConfig *= new* RobotstxtConfig();  
RobotstxtServer robotstxtServer *= new* RobotstxtServer(robotstxtConfig, pageFetcher);  
CrawlController controller *= new* CrawlController(config, pageFetcher, robotstxtServer);

//The seed path of the crawler

controller.addSeed("https://www.concordia.ca/");

In this part, I showed my crawling tool configuration, how I followed the robot exclusion rule, and how I limited the size of my development set and the production set.

the code of the crawler is shown here:

String fileName *=* "./data/" *+* counter*++ +* ".html";  
*try*{  
 BufferedWriter out *= new* BufferedWriter(*new* FileWriter(fileName));  
 out.write("<url>");  
 out.write(url);  
 out.write("</url>");  
 out.write("<MainText>" *+* html *+* "</MainText>");  
 out.close();  
} *catch* (*IOException* e) {  
 e.printStackTrace();  
}

I worte the ID of the html file as the file name, and wrote the url of the page into html file.

In general, I got 50 docs for dev, while 14852 for production

1. Text extraction

In this part, I used beautiful soup to extract the content from all the docs.

See: ./project4.py

*def* block\_document\_segmenter(file\_stream):  
 *for* iD, html\_doc *in* file\_stream:  
 soup = BeautifulSoup(html\_doc, 'html.parser')  
 *yield* iD, clean(soup.get\_text())

1. Create the tf\_td\_inverted index.

*def* build\_tf\_df\_dictionary(file\_path):  
 print("Building tf\_df\_dictionary")  
 docFrequency\_F = {}  
 word\_counter = 0  
 *for* iD, term *in* parse\_tokens\_from\_documents(block\_document\_segmenter(block\_reader(file\_path))):  
 word\_counter += 1  
 *if* term *in* docFrequency\_F:  
 posting\_list = docFrequency\_F.get(term)  
 *if* int(iD) *in* posting\_list:  
 posting\_list[int(iD)] = posting\_list[int(iD)] + 1  
 *else*:  
 posting\_list[int(iD)] = 1  
 *else*:  
 docFrequency\_F[term] = {int(iD): 1}  
  
 *for* key *in* docFrequency\_F.keys():  
 docFrequency\_F[key] = collections.OrderedDict(docFrequency\_F[key])  
  
 completed\_dict = {}  
 *for* key *in* docFrequency\_F.keys():  
 new\_key = key + "/" + str(docFrequency\_F[key].\_\_len\_\_())  
 completed\_dict[new\_key] = sorted(docFrequency\_F[key].items(), key=*lambda* x: x[1], reverse=*True*)[:50]  
  
 *return* completed\_dict

by this code, I created the dictionary as per professor’s requirement:

each posting list size is limited by 50, and showing the highest tf posts. And it includes all the vocabularies I crawled

This is the sample item from the dictionary, the enitre dictionary is at ‘./required\_tf\_td\_dictionary.json’

"schedules/13340": [[4426, 11], [11256, 7], [11627, 7], [3792, 7], [4301, 7], [5956, 7], [6940, 7], [757, 7], [10486, 6], [11660, 6], [1948, 6], [2027, 6], [2136, 6], [4060, 6], [4473, 6], [6147, 6], [6213, 6], [6578, 6], [7539, 6], [8369, 6], [1004, 5], [10103, 5], [10151, 5], [10169, 5], [10179, 5], [10667, 5], [10706, 5], [10770, 5], [10776, 5], [1088, 5], [10953, 5], [11344, 5], [11476, 5], [11629, 5], [11687, 5], [11709, 5], [121, 5], [12304, 5], [12346, 5], [1241, 5], [12455, 5], [12669, 5], [12721, 5], [12888, 5], [12891, 5], [12897, 5], [13089, 5], [13224, 5], [13313, 5], [13522, 5]]

1. Formulate 2 diffenrent queries for each information needs, and show the top 15 results

The completed result will be available at the attachment(./search\_result.docx), here we only show the best result link

1. which researchers at Concordia worked on COVID 19-related research?

test\_query\_a = ["COVID", "researcher"]

RSV:

title: Surveillance takes on many different forms during a pandemic, according to Concordia researcher

url: <https://www.concordia.ca/news/stories/2020/05/26/surveillance-takes-on-many-different-forms-during-a-pandemic-according-to-concordia-researcher.html?c=%2Fcoronavirus%2Fupdates>

TF\_IDF

title: Updates

url: <https://www.concordia.ca/coronavirus/updates.html>

test\_query\_b = ["COVID", "related", "research"]

RSV:

title: Researcher Coronavirus FAQ

url: <https://www.concordia.ca/research/coronavirus-faq.html>

TF\_IDF

title: COVID-19 research bulletin, March 20, 2020

url: <https://www.concordia.ca/cunews/main/covid/2020/03/20/covid-19-research-bulletin-march-20.html>

1. which departments at Concordia have research in environmental issues, sustainability, energy and water conservation

test\_query\_c = ["water", "conservation", "environmental", "energy", "department"]

RSV:

title: CREATE program

url: <https://www.concordia.ca/research/water-energy/about/create.html>

TF-TDF:

title: Concordia grad’s innovative research uses chemistry to capture solar energy

url: <https://www.concordia.ca/cunews/main/stories/2020/08/12/concordia-grads-innovative-research-uses-chemistry-to-capture-solar-energy.html>

test\_query\_d = ["research", water", "environment", "sustainability", "energy"]

RSV:

title: CREATE program

url: <https://www.concordia.ca/research/water-energy/about/create.html>

TF-IDF:

title: Urban water consumption will increase due to climate change, Concordia research shows

url: <https://www.concordia.ca/news/stories/2020/07/29/urban-water-consumption-will-increase-due-to-climate-change-concordia-research-shows.html>

1. Report on different behaviour of the ranking schemes, any issues with tf ranked posting
2. For tf-idf ranking, it counts the word frequency of a document, and use it as the ranking criteria. But when a document like a webpage, the header, footer, and the menu words can affect the tf and idf. For example, when the header contains COVID-19 information, each of the page will contains COVID-19, and that make df meaningless. Similarly, when we try to search the word in the menu, like ‘Concordia’, it will increase the term frequency in the doc. Also for a webpage, the length of header and footer, and links in the document may longer than the main text, so the search may fall into searching the text in irrelevant contents.
3. No semantic analysis
4. Challenge queries:

See the full query results in (./search\_result.docx)

# ['water', 'management', 'sustainability', 'Concordia'],

RSV:

Result 1

ID: 12760

score: 14.225976278638663

title: ‘We need to find better ways to protect our water'

url: https://www.concordia.ca/cunews/main/stories/2017/02/21/water-source-of-life-engineering-climate-change-expert-ali-nazemi.html

=================================

Result 2

ID: 4400

score: 13.148834511790133

title: CREATE program

url: https://www.concordia.ca/research/water-energy/about/create.html

=================================

TF-IDF

Result 1

ID: 11675

score: 0.06363751181826606

title: Urban water consumption will increase due to climate change, Concordia research shows

url: https://www.concordia.ca/news/stories/2020/01/21/urban-water-consumption-will-increase-due-to-climate-change-concordia-research-shows.html?c=%2Fnews%2Ftopic

=================================

Result 2

ID: 5421

score: 0.05691101920364353

title: Urban water consumption will increase due to climate change, Concordia research shows

url: https://www.concordia.ca/news/stories/2020/01/21/urban-water-consumption-will-increase-due-to-climate-change-concordia-research-shows.html?c=%2Fabout%2Fsustainability%2Fresearch

# ['Concordia', 'COVID-19', 'faculty'],

RSV:

Result 1

ID: 3875

score: 9.528781919180958

title: Important information regarding access to faculty offices

url: <https://www.concordia.ca/cunews/offices/vps/hr/2020/07/02/important-information-regarding-access-to-faculty-offices.html?c=%2Fcoronavirus%2Fupdates>

TF-IDF:

Result 1

ID: 3378

score: 0.03943468282125535

title: Annonces

url: <https://www.concordia.ca/fr/coronavirus/annonces.html>

Due to the page limit, other query results please see attachments.