COMP 479

Project 3 Report

Web crawling, scraping, indexing,

querying, and ranking of

Concordia university research department

By: Abdullatif Dal'ab

ID:27880960

Presented to Dr. Sabine Bergler

December 2, 2019

Fall 2019

Table of Contents

1. Analysis of crawling concordia.ca/research URL using Oncrawl

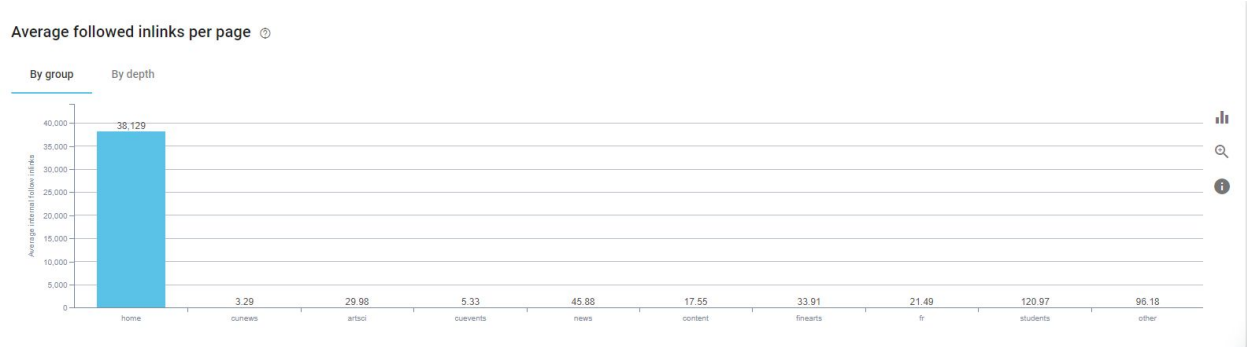
Oncrawl is a technical SEO platform that offers a numberof solutions, and one of them is web crawling. Not only that, it also analyzes the content of website being crawled and generates a report with useful information. I tried a number of tools before selecting Oncrawl, but I won't delve into the details now as I will be discussing these attempts in the issues faced section of this paper.

To make the program work, all you have to provide is the root URL (https://concordia.ca/research). The program stores all the URLs crawled in a csv file, which you can then download.
In the first attempt, I ran the crawler for 2 hours and generated around ~4520 unique url links. I used that as a test set.

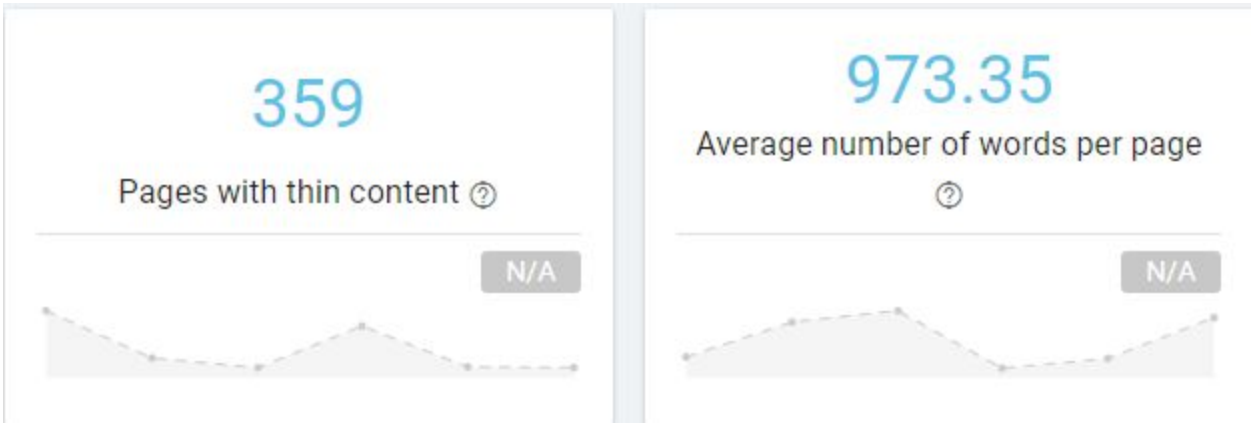
In my second run, I ran the program for 14 hours, and generated around ~40000 unique urls which I chose to use in the creation of my final index. Since there is a limit on the number of pages that can be visited (Free trial), I couldn't run the program much more than that and had to settle for around 40000 documents.

After creating the final index and testing my system, I realized that there were a lot of duplicates being returned. Meaning that even if the urls were unique, a lot of their content was not. I had to filter all of these duplicate files, and I ended up with a final index consisting of ~17000 documents. The final results returned barely had any duplicates after filtering when these 17k documents were used to build the final index.

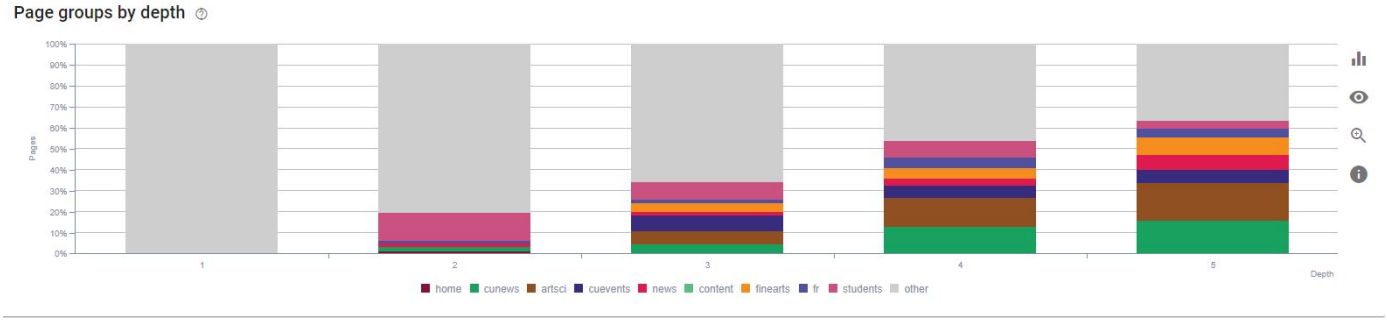
Here are some statistics describing the crawling results of the second run:



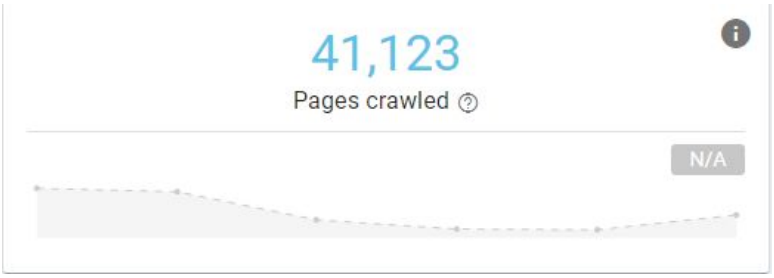
This shows the average number of links received by pages. As you can see, the homepage had a disproportionately larger number of links.



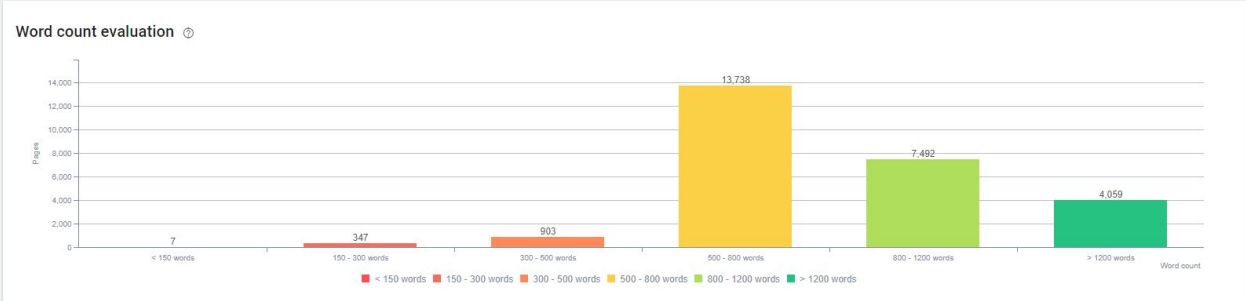
As for the average number of words per page extracted, the number is approximately 973.35.



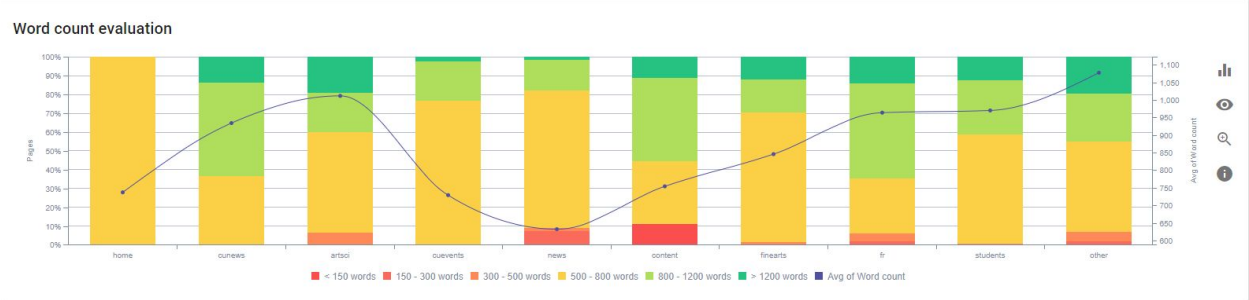
As the crawler goes deeper, the number of pages start to vary. In depth 2, most of the pages were parts of the news section of the website, but as we go to depths (4 and 5), the majority of the pages were parts of sections like artsci (Arts and computer science). That means the deeper the crawler goes, the better it will be for the information retrieval system being built, as text content from that section of the website is highly relevant to information needs that have to be satisfied (i.e. AI researchers)



Number of pages crawled is 41,123 which is almost equal to the number of documents created for the final index (before filtering duplicates)



This graph shows the average number of words on pages.



This graph shows the distribution of number of words per section (the average being around 900 as shown earlier).

2. Creation of documents

After downloading the csv file that holds all the links crawled, my program reads it line by line, extracts the urls, stores them in a list, makes sure that they all are part of the concordia.ca domain, and then filters any url duplicates.

In the code below, the list of filtered_links is iterated through. If a link doesn't work, it will be identified and skipped. A dictionary called docID_link_map is used to create a mapping between document IDs and links. Boilerpipe (Extractor class) is used to extract text content from the url links. That text content is then stored in a text file (which is the final document).

```
docID_link_map = {}

headers={'User-Agent': 'Mozilla/5.0'}

for i,link in enumerate(filtered_links):
    try:
        extractor = Extractor(extractor='ArticleExtractor', url=link)
        extracted_text = extractor.getText()

    except:
        print("link:"+str(i)+ " " +link)
        print("Current document:")
        print(i)
        continue

    docID_link_map[int(i+1)]=link

    os.chdir("documents-ai")

    text_file = open(str(i+1), "w")

    text_file.write(extracted_text)

    text_file.close()

    os.chdir("..")
```

3. Creation of final index

After all the urls are looped through, the documents created will serve as an input to the Spimi function:

```
os.chdir("documents4-oc")
spimi_obj = Spimi()
for txt_file in documents:
     spimi_obj.SPIMI_INVERT(txt_file,stem=True,stopw_150=False)
```

```
MEMORY FULL - WRITING TO DISK.
MEMORY FULL - WRITING TO DISK.
MEMORY FULL - WRITING TO DISK.
MEMORY FULL - WRITING TO DISK.
```

Each block will consist of around 1000 documents (that's the parameter set). The blocks are then merged into a single merged block, and that will serve as the final index. The final_index will be saved onto disk and submitted as a deliverable.

4. Queries tested to satisfy 3 information needs

First information need: **Which departments have AI research?**

The queries tested are:

Query1: department artificial intelligence concordia research

Query2: department artificial intelligence research researcher engineering machine learning

Query3: department research big data ai concordia

Analysis

Query1: department artificial intelligence concordia research

Results @10:

Note: that the results show follow this structure [# of Document, Document ranking weight]

[(30841, 962.5606007696398),

(26871, 949.8557620881151),

(19951, 937.6610006131345),

(24308, 924.973993656531),

(13428, 912.2566733383641),

(6514, 899.5412261015653),

(12018, 887.3466477761266),

(21744, 874.9471366234667),

(14576, 862.2368429142776),

(16239, 849.5207799523987)]

Document content analysis:

Document: 26871

Relevant content:

"Dilara Baysal is a Ph.D. **Student at Concordia University's Social and Cultural Analysis** program. Her current research interests include the changing forms of work and management in the context of new technologies such as automation and artificial intelligence."

Document: 19951

"Dr. Chunjiang An is an Assistant Professor in the **Department of Building, Civil, and Environmental Engineering at Concordia University**

M. Hu, G. Huang, W. Sun, Y. Li, D. Ding, C. An, X. Zhang, and T. Li, Multi-objective Ecological Reservoir Operation Based on Water Quality Response Models and Improved Genetic Algorithm: A Case Study in Three Gorges Reservoir, China, Engineering Applications of Artificial Intelligence (Elsevier), 36, 332-346 (2014)."

Query2: department artificial intelligence research researcher engineering machine learning

Results @10:

[(12478, 251.05167838678938),

(26871, 225.65185976762305),

(13781, 201.1616822888468),

(26036, 175.7416580416431),

(6514, 150.34236513415468),

(6033, 125.85251807237526),

(34637, 100.44675452371919),

(21484, 75.32245975796543),

(34918, 49.91578126184317),

(13889, 25.425913549365845)]

Document content analysis:

Document: 12478

Relevant content:

“Martin reports that new hire Michael Hallett, a professor in the **Department of Biology,**

will be exploring machine learning and artificial intelligence”

Document: 13781

Relevant content:

“Fatima Amara works on another highly impactful project:

the study of novel solar energy applications to buildings and infrastructure.

The research is piloted by Andreas Athienitis , professor in the **Department of Building, Civil and Environmental Engineering** and NSERC/Hydro-Québec Industrial Research Chair in Optimized Operation and Energy Efficiency toward High Performance Buildings. Using machine learning and artificial intelligence, she is able to add a statistical data-analytical dimension to a building, from which she pulls important clues regarding the behaviours of both occupants and buildings.”

Query3: artificial intelligence ai research concordia faculty department university

Results @10:

[(33565, 258.7621532100718),

(11292, 238.69117445447435),

(26871, 218.61039560194226),

(24308, 199.23363627513785),

(10259, 179.15232403933504),

(6514, 159.1621234173823),

(33326, 139.78561239771815),

(10093, 119.70430016191533),

(25004, 99.69530850394206),

(1480, 79.61263681643665)]

Document content analysis:

Document: 33565

Relevant content:

“Concordia University seeks to appoint a Canada Research Chair (CRC) Tier 2,
a research intensive tenure-track faculty position, in Computational Physics
The selected candidate will receive a tenure-track faculty appointment in the **Department of Physics ,**in the **Faculty of Arts and Science** , and is expected to become a member of the Centre for Research in Molecular Modeling (CERMM) at Concordia. Experience, or a strong interest, in using artificial intelligence (AI) methods and multi-scale modeling in computational physics/biophysics would be an asset.”

Document: 25004

Relevant content:

“Artificial Intelligence, Human-Computer Communication, Pattern Recognition
Professional associations:PhD Dr. Ching Y. Suen is the Director of CENPARMI and the Concordia Honorary Chair on AI & Pattern Recognition.
His research projects have been funded by the **ENCS Faculty** and the Distinguished Chair Programs at Concordia Universit”

Second information need: **Which researchers are working on AI research?**

The queries tested are:

Query1: concordia computer science artificial intelligence engineering university researcher computing

Query2: natural language processing researcher

Query3: researcher concordia deep learning phd engineering

Analysis

Query1: concordia computer science artificial intelligence engineering university researcher computing

Results @10:

[(26871, 157.5556548564489),

(949, 131.85745874763847),

(31283, 105.033512234989),

(6514, 78.20942435382204),

(6033, 52.511625661462865),

(34918, 25.69782353034346)]

Document content analysis:

Document: 949

Relevant content:

“Many other Concordia researchers with expertise in AI-related fields are also associated with CENPARMI, including **Sabine Bergler**, who investigates meaning and context behind words; **Adam Krzyzak**, who conducts research on handwriting analysis and facial recognition; Tien

Bui, whose major projects include building computers that mimic human vision; **Tristan Glatard** and **Marta Kersten-Oertel**, both of whom are involved in medical image analysis; **Nawwaf Kharma**, MSc 16, a specialist in nature- inspired computing; **Tiberiu Popa** and **Charalambos Poullis**, leaders in visual computing; and **Leila Kossem**, a natural language-processing expert
who is the current vice-president of the Canadian Artificial Intelligence Association"

Document: 6514

Relevant content:

"**Antonio Crespo** is a passionate learner, a lover of the skies, and an enthusiast of human behavior studies. He holds two bachelor degrees, BSc Aeronautical Sciences and BSc Social Sciences (Sociology, Political Science), and a Master degree in Computer Science. Throughout more than 25 years working with the Armed Forces and the Aviation Industry, he was assigned to several technical, military, managerial, executive and diplomatic positions, which includes a mandate in one of the specialized United Nations organizations. As a researcher, he has been actively working with Artificial Intelligence for the last eleven years. Antonio is currently conducting two AI related research projects, the first one targeting machine learning and sustainable development, and the second one focusing on aircraft automation and autonomy. He is also a Graduate Science Teaching and Learning Fellow, within a Concordia
CTL program aimed at the enhancement of STEM learning processes. Communication, Leadership and Management"

Query2: natural language processing researcher

Results @10:

[(33790, 120.51758971570958),

(26871, 105.28609946192421),

(949, 90.52679888632926),

(31283, 75.27287641477862),

(6514, 60.01889337098953),

(6033, 45.25977179364042),

(23820, 30.010196888342495),

(34918, 14.75913276458623)]

Document content analysis:

Document: 31283

Relevant content:

“1972, an emerging researcher named **Ching Yee Suen** joined Concordia as an assistant professor in the Faculty of Engineering and Computer Science. Having recently completed his doctoral research project — building a platform to “teach” computers to read multi-font documents with a voice output for the blind — he was fascinated by letters and characters. “Our machine was one of the first of its kind to scan documents
and read characters,” Suen says.Ching Yee Suen is director of Concordia’s Centre for Pattern Recognition and Machine Intelligence”

Document: 26871

Relevant content:

“**Dilara Baysal** is a Ph.D. Student at Concordia University's Social and Cultural Analysis program.Her current research interests include the changing forms of work and management in the context ofnew technologies such as automation and artificial intelligence”

Query3: researcher concordia deep learning phd engineering

Results @10:

[(10102, 130.2351807893794),

(949, 113.91024731162511),

(31283, 97.59175352668551),

(6033, 81.2731973865194),

(11151, 64.95917841377683),

(2958, 48.920818483518225),

(12652, 32.59378634009286),

(5412, 16.312412135657354)]

Document content analysis:

Document: 10102

Relevant content:

“**Parnian Afshar** is a PhD candidate in information systems engineering .

Her research aims to use deep learning-based cancer radiomics to improve detection,

diagnosis and prognosis capacities.’

Third information need: **What AI research is being conducted at Concordia?**

The queries tested are:

Query1 = concordia university artificial intelligence research paper graduate conducted

Query2 = concordia artificial intelligence research neural network deep learning

Query3 = ai research concordia

Analysis

Query1: concordia university artificial intelligence research paper graduate conducted

Results @10:

[(4511, 107.08645939474773),

(26871, 85.35389222125066),

(15923, 64.14914183600933),

(6514, 42.40889800254895),

(34918, 21.20445872314173)]

Document content analysis:

Document:

Relevant content: 4511

“23.Stathopoulos, T., (1983) "**Artificial Intelligence in Simple Beam Design**",

Journal of Structural Engineering, ASCE, Vol. 109, No. 9, p. 2225, Proc. paper 18207.”

Query2: concordia artificial intelligence research neural network deep learning

Results @10**:**

[(18950, 107.58476591117675),

(11460, 80.92900926271241),

(10259, 53.62098267741364),

(12018, 26.45185196966367)]

Document content analysis:

Document: 10259

Relevant content:

“created the IAPR ICDAR Awards, to honour both young and established outstanding researchers in the field of **Document Analysis and Recognition**. He has always been fascinated by letters and characters, ever since he started his doctoral research on teaching the computer to read **multifont documents with a voice output for the blind**.Research areas **Computer analysis and recognition of documents**”

Query3: ai research concordia

Results @10:

[(25599, 682.7112824985373),

(11249, 676.1435224441726),

(29169, 669.571428662156),

(8678, 663.0034400408281),

(1480, 656.4368182686878),

(14787, 649.8662758566022),

(21435, 643.2943771098338),

(13243, 636.7223935670512),

(13239, 630.1634296930455),

(33206, 623.5962578882013)]

Document content analysis:

Document: 25599

Relevant content:

“Associate Director, Milieux Institute for Arts, Culture and Technology

Director, xmodalimmersion, **machine learning**, new media theory, STS, performance, haptics, senses, embodiment, digital audio, sensory studies, sensory ethnography, interaction design, media art history, sound design, **critical AI studies**, posthumanism, philosophy of technology, enactive cognition, research-creation”

5. Comparison of results with a teammate

My teammates queries for the 3 information needs are the following:

Information need 1:

Query1: ai research

Query2: artificial intelligence research

Query3: artificial intelligence department

Information need 2:

Query1: artificial intelligence department research

Query2: artificial intelligence paper

Query3: artificial intelligence topic

Information need 3:

Query1: machine learning

Query2: deep learning

Query3: artificial concordia

These queries have returned a lot of false positives (e.g. documents discussing AI but not answering which departments have AI). Although my queries were more specific (helped in returning relevant documents), some of the documents that were omitted because of the length of the query, were actually relevant (these documents were returned using my teammates queries).

To compare our results, I queried my teammates following three queries (for each information need):

Query1: artificial intelligence department

Query2: artificial intelligence research

Query3: artificial concordia

I then saved the top 10 documents of these queries, and compared them to his top 10 documents (same queries).

Out of all the comparisons, only one identical document was found. (document 14832 in my final index). It was returned for query 2, and had the following content:

"CENPARMI members regularly present research papers at key conferences worldwide. CENPARMI often participates in industry conferences such as:
International Conference on Pattern Recognition and
Artificial Intelligence (ICPRAI)"

We concluded that it is only normal that we have different documents and we believe that is primarily because:

1- We used different crawlers

2- Our indexes are of different sizes

3- Different implementations of our ranking functions

6. Testing different ranking schemes and analyzing results

Ranking schemes

Two ranking schemes were tested for this report: TF-IDF and BM25.

Here's an example of the top 10 documents returned for the following query:

Department artificial intelligence concordia research

```
In [9]: query1 = "department artificial intelligence concordia research" # concordia is important cuz other sc
        retrieve_docs((query1),tfidf=False,num_ret=10)

Out[9]: [(30841, 962.5606007696398),
         (26871, 949.8557620881151),
         (19951, 937.6610006131345),
         (24308, 924.973993656531),
         (13428, 912.2566733383641),
         (6514, 899.5412261015653),
         (12018, 887.3466477761266),
         (21744, 874.9471366234667),
         (14576, 862.2368429142776),
         (16239, 849.5207799523987)]

In [10]: query1 = "department artificial intelligence concordia research" # concordia is important cuz other sc
         retrieve_docs((query1),tfidf=True,num_ret=10)

Out[10]: [(30841, 6536.363740562001),
          (26871, 6450.358954501975),
          (19951, 6364.354168441949),
          (24308, 6278.349382381923),
          (13428, 6192.3445963218965),
          (6514, 6106.33981026187),
          (12018, 6020.335024201844),
          (21744, 5934.330238141818),
          (14576, 5848.325452081792),
          (16239, 5762.320666021766)]
```

As you can see, the weights of BM25 (top) and TFIDF (bottom) are different, however, the ranking is identical. This conclusion is true for all the queries tested.

In order to figure out why, I looked into the BM25 algorithm to see how the parameters k, b average length of document, and length of document affect the weight. It seems that the parameters term frequency and document frequency (params shared in both schemes) have the largest weight in determining the end result. That means ranking will be preserved even if the ranking weight is different between the ranking schemes.

Let's take a closer look into BM25's formula:

```
"Formula steps:"

idf = np.log(N/document_frequency)
numerator = (k1+1)*term_frequency
denomenator = k1*((1-b)+ (b*(doc_length/avg_doc_length))) + term_frequency

ratio = numerator/denomenator

weight = idf * ratio

return weight
```

The ratio's numerator is multiplied by term frequency, while the ratio's denominator only takes the sum of term frequency. That means the numerator will often be > than the denominator. The idf doesn't change in BM25.

Comparing this to tf-idf's formula: tf * idf, we can see that the values change, but not significantly so that the rankings are affected.

Usefulness of results on average

On average precision @10 is around 40%. So 3 - 4 of the top 10 documents fulfill the information need while the rest fulfill the queries needs.

For the queries that returned 50 to 100 documents, the number of relevant documents increased, but not dramatically. Out of 100 documents, there would be around 10 relevant ones that fulfill the information need.

The number of documents ~17000 and the terms in the queries are the two main factors that determine how well the system performs. More specific queries return better results, but much

less documents. Some of the documents lost might indeed fulfill the information need, but are discarded anyways because of the AND operation.

I believe that the system can be improved a lot by increasing the number of documents that the index is built on and by adding phrases into the index (e.g: machine learning, deep learning, artificial intelligence) or positional indexing.

7. Crawling aitopics website and creating a limited AI index

To generate the AI index, I crawled aitopics, and generated around 3000 documents. Instead of using boilerpipe to extract text content, I saved everything in these documents because a lot of AI terms were in the tags of these pages, and boilerpipe discarded those.

After creating the AI-index, I generated the following document frequency index:

```
{'artificial': 3594,
 'intelligence': 3594,
 'machine': 3509,
 'learning': 3499,
 'ai': 3605,
 'neural': 3346,
 'networks': 2563,
 'cognitive': 1241,
 'supervised': 489,
 'unsupervised': 462,
 'semantic': 3605,
 'analysis': 3604,
 'chatbot': 2516,
 'science': 3409,
 'algorithm': 3438,
 'data': 3605,
 'mining': 3063,
 'big': 3118,
 'turing': 366,
 'analytics': 1485,
 'cluster': 201,
 'engineering': 2466,
 'reinforcement': 1754,
 'nlp': 524,
 'deep': 3192,
 'classification': 2777,
 'regression': 1596,
 'probability': 2445,
 'gradient': 1082,
 'machines': 2106}
```

In the ranking function (tf-idf or bm25) the following code determines which document frequency is used:

```
"Find document frequency"
if term in ai_index.keys():
    document_frequency = ai_index[term]
else:
    document_frequency = len(Merged_block[term])
```

If the term is not an AI term, the regular document frequency from the main index is used. However, if it is any of the terms mentioned in the previous page, the document frequency is retrieved from the AI index and is used for ranking.

Here is an example of the results:

```
In [45]: query1 = "artificial intelligence"

In [48]: # Old df weights
         retrieve_docs((query1),tfidf=False,num_ret=10)

Out[48]: [(17402, 2897.974674452672),
          (22513, 2886.733611141023),
          (11249, 2875.473195625473),
          (14832, 2864.212962234033),
          (19951, 2852.9492674494963),
          (20461, 2841.73772648784),
          (29676, 2830.475833519786),
          (21484, 2819.2242945687285),
          (8678, 2807.9841587674196),
          (12261, 2796.750007675136)]

In [52]: # Ai index df weights
         retrieve_docs((query1),tfidf=False,num_ret=10)

Out[52]: [(17402, 1215.959398262353),
          (22513, 1211.2427910878355),
          (11249, 1206.5181237647414),
          (14832, 1201.7935322950038),
          (19951, 1197.067499183405),
          (20461, 1192.3631883795106),
          (29676, 1187.6379057089262),
          (21484, 1182.91693543015),
          (8678, 1178.2007145665102),
          (12261, 1173.4869863632612)]
```

The first query uses the initial index document frequency weights while the second query uses the document frequency weights of the AI index. The rankings remained identical (true for all

tested queries), however, the weights have indeed changed. One observation noted is that using the AI document frequencies, the variance between the ranking weight of documents is lower. If this is incorrect, then it's likely that there is a bug in the query function that I didn't have the time to diagnose.

8. Problems encountered and issues resolved

Problems encountered:

- Websphinx returning a small number of pages ~ 1000
- Websphinx crashing regularly midway through crawling
- Time delays caused by aitopics crashing
- Changes required to be done on spimi to accommodate new tasks
- A lot of duplicates
- tf-idf and BM25 rankings are identical
- AI-index df ranking and concordia-index rankings are identical

Issues resolved:

- Used oncrawl to crawl concordia.ca/research and generate documents
- Duplicate document filtration
- Crawling aitopics midnight (Not many people crawling and so it didn't crash abruptly)
- Fixed spimi to accommodate new tasks

I have been unable to diagnose the problem of Identical rankings. The ranking functions implementations seem correct as they're based on the books formulas. They do help return relevant documents as well. The issue is most likely with the querying function as a single term query returns different rankings using tfidf and bm25.

Individual team contributions

The team-work part in this project consisted of only exchanging queries and comparing results, as well as discussions on the use of online crawlers.

9. Conclusion

The number of documents collected, the size of the index, and the terms in a query play a big role in determining whether a system will be able to satisfy an information need.