KENNESAW STATE UNIVERSITY

Web Crawler

CS 7263
Information Retrieval

Summer 2025

Instructor – Dr. Jiho Noh

Michael Rizig

I. ABSTRACT

The goal of this project is to build a web scraper that will crawl through the course textbook and collect all text pages from the textbook. The crawler should then extract the pages titles and contents and store each unique page in a library of pages. Finally the crawler should perform corpus processing and generate statistics about all crawled and downloaded pages.

II.DATA

For this project, we initially start our crawl on the html webpage for the course textbook*: Christopher D. Manning, Prabhakar Raghavan and Hinrich Schütze, Introduction to Information Retrieval, Cambridge University Press. 2008.* Link: https://nlp.stanford.edu/IR-book/information-retrieval-book.html. This book has approx. 192 unique webpages, which is the min target number of webpages for this scraper.

III. CODE BREAKDOWN

Below is a breakdown of each python file and an explanation of what each function of the corresponding class does.

1. crawler.py

is_link_ok_to_follow(self,link):

This function checks if input link is a valid string, checks if link is already crawled via the crawled pages library, removes anchors, queries and '/', checks if link is not a relative, and checks if the link ends with '.html'. if all these conditions are met, returns true, else returns false.

get_html(self,url):

This function opens the passed URL using urllib.urlopen, loads the HTML code into a string variable and returns the string.

discover_urls(self, soup, pageurl):

This function Loops through each <a> tag in the html in soup object and calls **check_relative_link** to ensure link is not relative, then calls **is_url_ok_to_follow** to ensure link is ok to follow, it then adds the url to the frontier queue and cleans using **clean_url.**
No return.

check_relative_link(self,url,pageurl):

This function checks if the url has the full url by checking how many "/" exist in the url. If it is 1 or less, it appends the url to the home pageurl.
Returns the complete url.

clean_url(self,url):

this function cleans the url by removing anchor tags and queries.
Returns cleaned url.

extract_info(self,url,soup):

This function calls **clean_url** to clean input url, then puts the url into a dict with the key as the cleaned url and the value as a tuple of the urls title and text content. Soup is used to extract title and text.
No return.

save_webpages(self,library):

This function loops through the library of crawled urls and saves them 1 by 1 with a unique file name into the 'webpages/' folder.
No return

crawl(self):

This function loops while the frontier queue is not empty, checking each url with **is_url_ok_to_follow**. It then uses **get_html** to grab the first url in the queues HTML. With this HTML as a string it creates a new soup object. It then calls **extract_info** and **discover_urls** with this soup object to extract text and discover all links on this page. Finaly when the frontier queue is empty, it calls **save_webpages** to save the library into text files.
No return

2. stats.py:

process(self):

This function goes through each file in the directory "webpages/" and generates the following statistics:

- Total words
- Total words (stop words removed)
- Total unique words
- Average page length
- Top 30 words
- Top 30 words (stop words removed)

The function then saves these as class variables.
No return

print_statistics(self):

This function simply prints the generated statistics from **process.**
No return

IV: OUTPUT

```
(.venv) michaelrizig@Michaels-MacBook-Pro webscraper % python3 stats.py
Total words:  214715
Total words (stopwords removed): 9874
Unique words:  9903
Average Page Length:  735.3253424657535
Top 30 words (with stopwords):
        1: 'the' Count: 11642 Documents: 290
        2: 'of' Count: 7108 Documents: 289
        3: 'a' Count: 4915 Documents: 280
        4: 'in' Count: 4717 Documents: 290
        5: 'and' Count: 4218 Documents: 283
        6: 'to' Count: 4010 Documents: 290
        7: 'is' Count: 3724 Documents: 287
        8: 'for' Count: 2563 Documents: 277
        9: 'we' Count: 2020 Documents: 242
        10: 'that' Count: 1858 Documents: 258
        11: 'as' Count: 1621 Documents: 260
        12: 'are' Count: 1395 Documents: 241
        13: 'this' Count: 1389 Documents: 251
        14: 'an' Count: 1109 Documents: 287
        15: 'be' Count: 1095 Documents: 235
        16: 'document' Count: 1070 Documents: 190
        17: 'documents' Count: 1036 Documents: 196
        18: 'with' Count: 1034 Documents: 239
        19: 'index' Count: 1029 Documents: 288
        20: 'on' Count: 991 Documents: 241
        21: 'query' Count: 987 Documents: 166
        22: 'by' Count: 856 Documents: 228
        23: 'can' Count: 799 Documents: 214
        24: 'retrieval' Count: 784 Documents: 136
        25: 'at' Count: 751 Documents: 288
        26: 'it' Count: 751 Documents: 225
        27: 'page' Count: 749 Documents: 286
        28: 'not' Count: 733 Documents: 211
        29: 'from' Count: 709 Documents: 209
        30: 'or' Count: 655 Documents: 208
Top 30 words: (without stopwords)
        1: 'document' Count: 1070 Documents: 190
        2: 'documents' Count: 1036 Documents: 196
        3: 'index' Count: 1029 Documents: 288
        4: 'query' Count: 987 Documents: 166
        5: 'retrieval' Count: 784 Documents: 136
        6: 'page' Count: 749 Documents: 286
        7: 'term' Count: 652 Documents: 155
        8: 'information' Count: 645 Documents: 148
        9: 'classification' Count: 639 Documents: 68
        10: 'terms' Count: 634 Documents: 161
        11: 'text' Count: 595 Documents: 126
        12: 'next' Count: 592 Documents: 284
        13: 'web' Count: 575 Documents: 106
        14: 'clustering' Count: 558 Documents: 41
        15: 'figure' Count: 541 Documents: 137
        16: 'contents' Count: 539 Documents: 285
        17: 'previous' Count: 538 Documents: 283
        18: 'search' Count: 538 Documents: 123
        19: 'model' Count: 522 Documents: 105
        20: 'may' Count: 521 Documents: 285
        21: 'section' Count: 463 Documents: 179
        22: 'two' Count: 461 Documents: 157
        23: 'case' Count: 455 Documents: 285
        24: 'set' Count: 448 Documents: 136
        25: 'one' Count: 445 Documents: 174
        26: 'example' Count: 434 Documents: 154
        27: 'number' Count: 427 Documents: 157
        28: 'relevance' Count: 415 Documents: 67
        29: 'vector' Count: 412 Documents: 96
        30: 'book' Count: 377 Documents: 289
(.venv) michaelrizig@Michaels-MacBook-Pro webscraper %
```