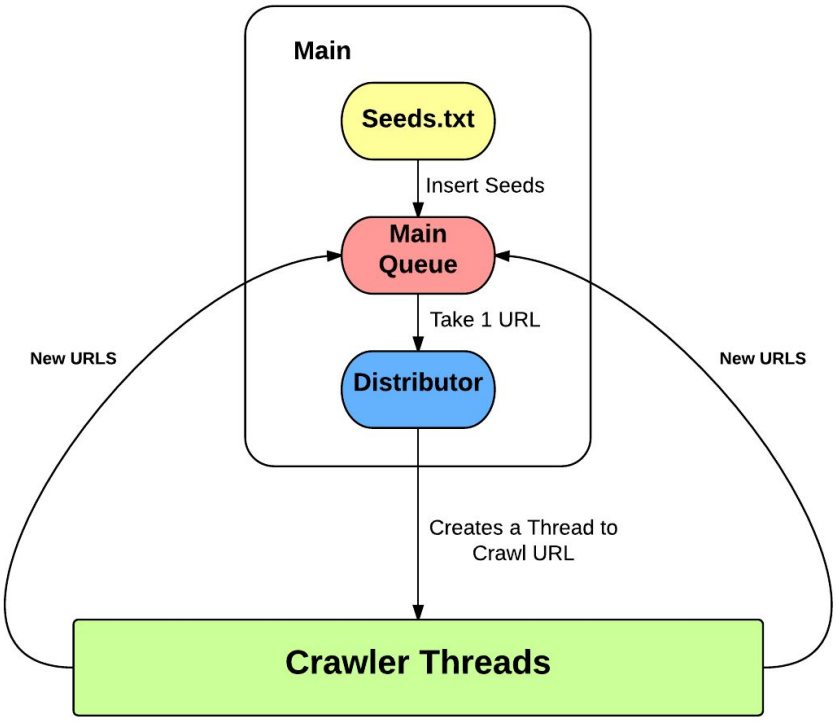
CS172: Introduction to Information Retrieval

Course Project: Web Crawler - Project 1

1. Overview of System:

a. Architecture

Diagram of implementation:



b. Crawler Strategy

i. First, the Main Queue of the class Crawl is created, which inside holds an int depth and a string URL.

ii. Then iterate through the seed.txt file and insert each line as a Crawl member into the Main Queue.

iii. The distributor loop inside of the main will then extract the Crawl from the front of the main Queue, and create a Crawler_Thread to work on it.

iv. The main distributor will then add the URL into the output text file.

v. Inside of the Crawler_Thread, the URL member of the Crawl class is passed in as a parameter to the Crawler_Thread, then using JSoup, grab the pages, parse, normalize, and then put them into a new Queue.

vi. Then a couple of User based normalizations will be done, such as removing bookmarks.

vii. Compare the Queue of compiled links with a hash set that has all of the previously found links, if it is not found and Depth is less than the wanted amount , ++Depth, and insert it into the back of the Main Queue.

viii. If the element was found inside of the hash set, or if the Depth was found to be equal or greater to the requested Depth, then do not add the element into the queue.

ix. Run until the requested number of pages is reached, or if there are no new unique pages to crawl.

c. Data Structures

i. Crawl Class:

1. **Depth:**
   This will be representative of the number of hops in the current crawler. When the number of hops equals to the number of hops requested originally by the arguments, then the crawler thread will not insert their findings into the queue.
2. **URL:**
   The URL that the Crawler Thread will work on.
3. **MAX Depth:**
   An int that will hold the requested depth that the arguments gave.
4. **out_file:**
   The file that is being written to is inside of the Crawler Thread.
5. **Thread_Share:**
   This class is included in the Crawl class so it can access it inside of the crawler thread to update the queue and check the incoming URLs to see whether or not they are repeats from before or not.

ii. Thread Share Class:

1. **Blocking Queue of Crawlers:**

   This is the main queue that will be shared among all threads. The Distributor will be grabbing from the queue and then creating a Crawler Thread that will have the Crawl taken from the queue as a parameter. The Crawler thread will then in turn crawl the URL inside of the Crawl class and insert its findings into the queue if it is lower than the wanted number of hops. A blocking queue is used as a blocking queue is as you would say, “Thread safe” and will work coordinately with the many threads.

2. **HashSet:**

   The HashSet will store the previously crawled pages. If an element that is found in the current crawl is found inside of the hash set, it is removed from the queue that will be inserted into the main queue of crawlers

3. **int title:**

   A number to label the titles of the pages that are being visited, so that duplicate names do not overwrite each other.

4. **int threadcount:**

   An int to keep track of how many concurrent threads are currently running and limit them as needed to.

2. Run Instructions:

./Web_crawler.sh <seed.txt> <10000> <6> <output-dir>

args[0] = input file

args[1] = num-pages

args[2] = hops-away

args[2] = output directory

3. Screenshots: