Information Retrieval Analysis

John Kirschenheiter
Jesse Shellabarger

Curation

To aid in our ability to search through the documents, we did a little bit of curation. First, we used an online tool (https://www.phpjunkyard.com/tools/html-to-text.php) to convert the html from the wikipedia page into plain text. We then removed some of the text pertaining only to Wikipedia and not to the article. We stripped the title of the page of any text relating to wikipedia so that our header search would be more accurate. Additionally, we converted the entire article to lowercase to make the search case insensitive. We also removed any of the lines that seemed to have excessive white space in there start. These lines were a result of the online html converter we used, and mostly referenced unuseful data. Finally, we did our best to remove the references section of the article. This proved difficult, as the syntax wasn't standardized. In the end, we simply removed the last section of the page in an attempt solve the problem. While this solution isn't perfect, it worked well enough to give use good results.

Retrieval Method

We started off by implementing the BM25 indexing algorithm. We started here because we believed it would end up being the most relevant part of our information retrieval system. After completing BM25 we found that it did not perform as well as we had hoped. Document length had a larger effect on the results than we would have liked. It was so extreme that when searching for “Washington” the algorithm would suggest the article on Fillmore because it was much shorter and also mentioned Washington frequently.

To fix this we added a component to search the titles of the articles for terms in the search query. We made this component contribute much more to the final score of the document than the BM25 search, so that a search for “Washington” would be sure to return the page on Washington.

At this stage we were unhappy with our inability to search for phrases effectively. Searching “Civil War President” would look at only at the frequencies of the individual words and not consider the phrase as a whole. To remedy this, we implemented another search methods using Skip Bigrams. This search looks for phrases in the search query within the documents. To aid in successfully finding these phrases in the documents, a single word is allowed to be inserted into the query phrase. We weighted this search more than the BM25 index, but less than our header search.

Results

The final results were satisfactory. The search operates quite quickly, and the results are all relevant to the searches given (with the exception of nonsense searches). This is with the exception of searches where five relevant documents do not exist. Because we always display five results, the search of “Civil War President” will return more than just the article on Lincoln.

We decided to output the top five search results since, for many searches, more than one president fulfills the search terms. At the same time, we didn’t want to overwhelm the reader with the scores of every article. Our results were so reliable that we got the exact same matches for many of the test searches given to us in the assignment. That being said, our program did somewhat diverge when very common words were searched such as “I”, “the”, and “an”. This is probably a result of the BM25 being weighted differently from the test program.