Assignment 3: M3 — Report

Authors: Andy Park (ID# 29379609), Kyler Frazier (ID# 92061283)
Class: UCI, CS 121, Prof. Alberto Krone-Martins

To start, here's a short list of queries and how we made their results better:

1. **master of software engineering**
   At first, we did not have 2grams implemented. software and engineering are both relatively common words in the corpus, but when we added the 2gram, it really narrowed down the results to pages that had the entire query string in them.
2. **machine learning**
   When we made an index of the index to make queries run faster, we first partitioned the index by starting character. However, there was an overwhelming number of tokens that began with the letter "m", which made this query very slow. We fixed it to partition the index by number of tokens instead.
3. **https://aiclub.ics.uci.edu/**
   At first, we had not only forgotten to add the URL of the websites to the index, but we also did not properly tokenize them. This meant that it would only give results for exact url matches, but not close matches. We remedied both of these issues by tokenizing, stemming, and adding URLs both from the document title and the href tags.
4. **Cristina Lopes**
   Our search result improved significantly after implementing tf-idf and cosine similarity. Before, our search engine would typically only retrieve documents that had only one instance of the query.
5. **ACM**
   The search result has improved after adding more weights to special tags, such as title, h1, h2, bold, etc…
6. **Artificial Intelligence**
   There were no more near or exact duplicates search results after adding simhash. Before, there were a lot of related websites that appeared.
7. **1+1**
   At first, we only tokenized the alphanumeric sequences. After using nltk's tokenizer, which includes special characters, we got more useful results.
8. **CML**
   CML is frequently in the title of websites, but since it doesn't appear that often it was getting buried by other links. However, we updated our indexer to give more weight to tokens that are in the title, which significantly helped.
9. **Astronomy**
   Astronomy websites are far and few in this corpus, however, they tend to link to one another. By adding weight to links to other websites, we were able to give the query an extra boost.
10. **This is a very long query**
    We found two issues with this query. The first is that it was slow. We found that with longer queries, our program ran for too long. In response to this, we indexed the index

even further (made the “meta-index” longer and the sub-indexes shorter). This reduces search time for each token in the query. In addition, we had an issue that our program was filtering out search results that only had all 2grams in them. This was a quick fix, but an important one nonetheless.

11. **Hero strawberry**
    This is an odd one. Building off of the last query, our 2grams were filtering out or documenting exact 2gram matches. I would be surprised if any documents had “hero strawberry” as an exact match! Now, it gets actual results, those that detail hero and strawberry, not necessarily both together.

Here are a few more to try!

- hackathon 2019
- Test
- Publications
- Undergraduate Research
- ics.uci.edu
- Netflix
- Awards Ceremony
- Course schedule
- python, c++, java
- Print statement python
- a b c d e f g h i j k l m n o p q r s t u v w x y z
- abcdefghijklmnopqrstuvwxyz

We might also note that the vast majority of these queries run in less than 10 milliseconds!