Shaun Chiang #15657686

Assignment 3 M3

Queries I tested: Cristina lopes, Machine learning, Recursion, Yunan chen, Syntax errors, Alex thornton, Python programming basics, Course restrictions, ACM, Normal distribution stats, CS 161, Bren hall construction, Multithreading, Graduation requirements, Master of software engineering, Analysis of algorithms, Computer science at uci, Quantum computing research, Uci wics general meetings, Computing in healthcare

Queries that performed well; good results and good performance

1. Cristina lopes
2. Machine learning
3. Recursion
4. Yunan chen
5. Syntax errors
6. Alex thornton
7. Python programming basics
8. Course restrictions

Queries that perform poorly; their results weren't that relevant to what was searched.

9. ACM
10. Normal distribution stats
11. CS 161
12. Bren hall construction
13. Multithreading
14. Graduation requirements

What I did to improve this

Often times, websites that were data dumps, word dumps, or contained large amounts of text were in the top query results (because their raw tfidf scores were high). In order to combat this and get more relevant searches, I decided to modify the top 20 query result's tfidf scores based on how many unique term tokens were on the website. Long webpages that high tfidf scores, but contained a disproportionately large quantity of terms, had their tfidf scores lowered. This way, I could better rank URLS that were "information dense" relative to the search query. My modified tfidf score per document was calculated as below.

Modified score = raw tfidf score / $\log_{10}$(num unique terms on webpage)

More Queries that perform poorly; they processed too slowly

15. Master of software engineering
16. Analysis of algorithms
17. Computer science at uci
18. Quantum computing research
19. Uci wics general meetings
20. Computing in healthcare

What I did to improve this

These queries returned results in longer than 300ms, likely because they contained stop words and common tokens whose terms had several thousands of postings in my tfidf scoring file. Because of this, in my tfidf file, I decided to limit how many postings could be associated with a term. Each term now has a maximum of 3,000 postings associated with it. The postings with the highest tfidfs are included in this list of 3,000. I chose 3,000 max postings per term because I had to make a tradeoff between speed and performance, and 3,000 provided a good balance of both.