George Gabricht - 56735102
Huy Minh Tran - 64272325
David Lee - 001037870

CS 121 - Project 3
Milestone 3 Report

Queries:
Beneath each query are the metrics that we improved along our journey.

1. What would you do for a Klondike bar
    a. Poor Retrieval Times
2. Donald Bren school of informatics
    a. Poor Results and Times
3. Graduate level courses
    a. Poor Results
4. Careers in software engineering
    a. Poor Retrieval Times
5. Masters of software engineering university of California Irvine
    a. Poor Retrieval Times
6. Machine learning algorithms
    a. Poor Results and Times
7. How much wood would a wood chuck chuck if a wood chuck could chuck wood
    a. Poor Retrieval Times
8. Cracking the coding interview
    a. Poor retrieval times
9. On campus Starbucks locations
    a. Poor Results
10. Past present and current prediction technology
    a. Poor Retrieval Times
11. What is the history of the university of California Irvine
    a. Poor Retrieval Times
12. Biomedical engineering university of California San Diego
    a. Poor Retrieval Times

George Gabricht - 56735102
Huy Minh Tran - 64272325
David Lee - 001037870

13. Human genome mapping and its applications
    a. Poor Retrieval Times
14. The interdisciplinary studies university of California Irvine provides
15. Which programming languages are used most frequently
    a. Poor Retrieval Times
16. The evolution of web crawling technique
    a. Poor Results and Times
17. What are digital forensic techniques
    a. Poor Results and Times
18. How surveillance cameras capture and track daily life
    a. Poor Results and Times
19. Obfuscation and optimization of android
    a. Poor Results and Times
20. What does the virtual reality solution offer
    a. Poor Result and Times

PLEASE SEE NEXT PAGE FOR DETAILED EXPLANATIONS

George Gabricht - 56735102
Huy Minh Tran - 64272325
David Lee - 001037870

When we began retrieving these queries, there were many issues with both the ranking effectiveness and retrieval efficiency. We began our ranking with simple tf-idf scoring, with no log normalization and no cosine similarity. As the project went on, we implemented both of these concepts, which seem to have improved our rankings in certain regards.

These Queries were chosen primarily because they either did not pull up any relevant pages or were very slow to load. When we began, we would write our index to string and then write it to a file. Soon thereafter we decided to split our index by term, each file containing its own term. This proved to be effective on our local machines but when tested in Openlab, we hit the iNode Disk Quota 60k file limit.

At this point, we were forced to redesign our index structure on disk. We managed to split our index based on the first letters in the term, which produced significantly fewer files while keeping file sizes relatively small. To speed up the retrieval process, common stop words and champion lists of size 50 were implemented for each term, resulting in less computation time and significantly less time to sort results. Along the way, many implementations were tested, including multithreaded file loading and even preloading the 2000 most common terms across documents the index.

The biggest improvement to retrieval times was in utilizing the pickle module to serialize our index files for lightning speed retrieval. While our search engine is far from perfect, we are very proud of how far we have come and in the experience we gained while developing this project. I have included in the zip folder both versions of our code. However, for purposes of Speed, Memory, Correctness and overall Project Requirements, outside of Openlab, the standard fileset is the proper one. (indexer_unload_pickle.py and retriever_unload_pickle.py)