CS 121 Project 3 M 3 Test Query Document

Group Member (sorted by alphabetical order):

Chentao	Ye	48361846
Jay	Zhang	66051593
Jiawen	Ye	54035818
Yingchen	Zhou	55879895

Here are twenty queries that we used to evaluate our search engine:

1. Machine learning
2. Machine
3. Learn
4. UCI machine
5. Information retrieval
6. URL
7. Encrypt
8. Computer science
9. Time complexity
10. UCI Admission center
11. Algorithm
12. Computer Science basic algorithm
13. Optimization
14. Algorithm optimization
15. Data structure
16. Computer security cryptography network
17. Artificial Intelligent and machine learning
18. ICS student affairs office appointment
19. Programming language learning
20. Time and Memory balance

When we were using the queries like “machine learning”, “URL”, “computer science”, and “data structure”, we found that the time spent for our search engine to process the target URLs of these queries are relatively slower than the others. After we reviewed our code for many times, we found the less efficiency was caused by reopening the json files. Specifically, before our optimization, we had a index file that records number-ids linked to its json file name but not the web URLs. So every time we get the names of these json files, we have to reopen them in order to get the URL of the website and render it in our User interface. When the corresponding json files are relatively large and long files, it will take a certain amount of time to open them again. Therefore, in order not to waste time of reopening them, we modified our index format and recorded the number-ids with URLs directly, which can help our User Interface demonstrate the results efficiently when it knows the number-ids of target pages.

When our search engine only ranked by the frequency of term occurrence, we found that when a query is composed of several words, like “Computer security cryptography network”, “Artificial Intelligent and machine learning”, “ICS student affairs office appointment” and “Computer Science basic algorithm”, it is likely that the effectiveness of the query search will be affected by the frequent or never occurrence of one specific term. Some target URLs that we got at that time were not relative to our queries. Then we applied the TF-IDF Cosine algorithm into our ranking function which considered the rarity of the term and the similarity distance between specific query and web document. This helped us improve effectiveness of our search engine for long queries dramatically.

In addition, instead of using standard text files for recording our inversed index files, we used the json format to save them, because json file can efficiently read our index and convert all of them into a dictionary that can help us achieve the information we need in O(1) time. That is why our search engine can always process the result URLs in high efficiency.