Queries:
cristina lopes
machine learning
acm
ics course schedule
introduction to python
donald bren hall
computer science faculty
uci software engineering
research
uci ics help

master of software engineering
ics student portal
contact information
login
important sources
how to apply
calendar
resources for new ics students
systems
computer science documents

The 10 queries in red performed poorly, primarily due to their vague wording and overall generic nature, which resulted in a large number of irrelevant queries. For example, queries such as “login”, “contact information”, and “calendar” appear in thousands of documents, which makes it very difficult to rank the truly relevant ones. The performance of these bad queries was also due to the large size of the index, which ended up slowing down the query execution time.

We ended up improving these queries by first splitting the index into more parts. Originally, it was “a-f”, “g-l”, “m-r”, and “s-z”, but we split the queries into more pieces, ranging from “a-z” or 26 different files. This makes it so that reading a particular file is much quicker and easier since there is much less to cover per file. In addition, we added the checking of defragmented links and if the link has been visited before to increase the accuracy of the index.

We also moved away from JSONs and utilized Pickles to preserve the same data, but in a binary format to speed up the process. We kept the same 26 alphabetical files in pickle format and created 26 offset files to hold the positions of each term of that corresponding file. We used Pickles specifically for partial access to keep track of the positions of each term within the pickle. For our search queries, we found its particular position and used seek() to get to that position and load() to load the postings without loading the whole file into memory. This overall improved our bad queries and in general improved the running time of our search engine.