Assignment 3: M3 Test Queries

Ashley Nguyen, Leo Siu, Rudy Xie, Muye Chen

Poor Performing Queries:

1. Hack at UCI
2. Ligma : Words that were not found at all in the index crashed the front end
3. commit the change
4. “cristina lopes “ : Test case used for testing whitespace in between and around tokens in the query. To address this test case, we stripped each token and if there was more than one token and an empty token was in the token set (‘’), then we removed this token.
5. Peter the anteater does not like crista lopes but at least there are spicy leaks in the science research library and a apple
6. Mark Baldwin
7. software engineering BS academic calendar
8. master of software engineering
9. Apple of my eye
10. how hard is cs 121

Well Performing Queries:

1. Research
2. Anteater
3. Email
4. Zot
5. Machine Learning
6. Plaza Verde
7. Python
8. Sql
9. Programming
10. ICS

Improvements Made:

The main improvement to be made was how we structured our inverted index to match with the query. The index was initially built as one large heaped pickle file with terms as keys and postings as values, however that led to problems when searching for postings for each query term. In our final improvement we separated our inverted index into two files. First, the MergedIndex text file stores the terms and postings in the form of [term: {post.docId} {post.docName} {post.tf} {post.tag_weight} <> {post.docId2} {post.docName2} {post.tf2} {post.tag_weight2}....] where <> is our separator. We then created a separate index file with [{term1:seekValue} {term2:seekValue}...] to store our offsets of where each token’s posting was located in the inverted index’s text file. This helped greatly reduce I/O file open overhead and also iterating through the text file to find the postings. With this, calculating the tf-idf scoring was much easier and there was no need to find the intersection of all the documents with ranked retrieval.