20 QUERIES

WARD - 20391420- UC IRVINE - CS 121 - F2020

```
[
"MOSFET", # Term that Does not Exist in Dataset
"Dingo ate me baby", # Query with terms that exist in dataset,
but whole phrase does not
#From Top Common Words
"support document",
"browser",
"sourcer",
"cbcl",
# From Least Common Words
"lawks",
"lawler",
"lave-man",
# Query From Bold / Heading / Title
"breast cancer wisconsin"
"language for distributed embedded systems"
"Ai club"
# Long Queries
the university of california irvine ai club workshop
# Assumedly Common Words (more matching documents)
"master of software engineering",
"computer science",
"informatics",
"computable plant",
"a", # One Letter Query
"krisberg org" # Query Of Common and Least Common Term
"kovarik@mcmail.cis.mcmaster.ca", # Long single word
]
```

POOREST PERFORMERS

```
the university of california irvine ai club workshop 1259.263499999996ms
the university of california irvine ai club workshop 653.0355000000014ms
sourcer 432.52720000000267-ms
computer science 608.8585999999978ms 424.3296000000001ms
a 638.1358000000006ms
master of software engineering 295.1747000000005ms
```

*** In final performance they were all brought down to around 50ms and certainly below 300ms.***

Methods of Improvement

I. General Improvements
   A. Cut down the amount of iterations for the query TF-IDF by using set() of terms rather than each term if repeated in query
   B. Removed exact duplicates with **MD5 hashing**. Attempted to implement SIMHASH, but could not solve O(n^2) hamming distance problem.
   C. Removed IF checks and replaced with try except "better to ask for forgiveness than for permission"
   D. Replaced dicts with defaultdict to remove key checks

II. Lecture Derived Search Improvements
   A. **Minimal seek positions,** made sure not stored in no more than 2-3 files/positions per file
   B. **Pre-generated "maps"** in their own JSON files, allowing for modularity in testing, and when loaded in memory before search time, allows for "access over calculation"
      1. {docID:url},
      2. {url:docID},
      3. {docID:bolded_words},
      4. {docID:links},
      5. {docID:page_rank},
      6. {docID:hash},
      7. [docID],
      8. [termID],
      9. {term:corpus_frequency},
      10. {bolded_word:[docID]},
   C. **During document pruning** process, method of **iterating through sorted query TF-IDF** of terms to append documents to selection was used.
   D. Set union each subset of docID's until the minimum document requirement is met. If it goes over, prune via set intersection.
   E. Stripping stop words from query if query is sufficiently long / ratio of stop words to contextual words is high
   F. Accuracy and speed was improved by utilizing **bold / title / heading** in tandem with base TF-IDF
   G. Implemented **PageRank** for additional accuracy.