Patrick Connolly

CS 429

Assignment 2

Measuring the time it takes to complete a query and display the results and compare it with higher numbers of terms searched is a good way to compare different query types. As the numbers of words in thee query increase, the longer it takes to retrieve results.

queries = [

"with without yemen ",

"germany invasion of poland",

"allied forces axis powers",

"america enters the war against germany",

"what happened to germany after the world war"

]

queries[0] has 2 non stop-words

queries[1] has 3

queries[2] has 4

queries[3] has 5

queries[4] has 6

**Results for exact query:**

N=2: 0.00299143791198730469 seconds

N=3: 0.00299191474914550781 seconds

N=4: 0.00398874282836914062 seconds

N=5: 0.00997304916381835938 seconds

N=6: 0.01595711708068847656 seconds

**Results for champion list query:**

N=2: 0.00000000000000000000 seconds

N=3: 0.00099730491638183594 seconds

N=4: 0.00000000000000000000 seconds

N=5: 0.00199460983276367188 seconds

N=6: 0.00598382949829101562 seconds

**Results for index elimination query:**

N=2: 0.00099754333496093750 seconds

N=3: 0.00000000000000000000 seconds

N=4: 0.00099778175354003906 seconds

N=5: 0.00099754333496093750 seconds

N=6: 0.00398921966552734375 seconds

**Results for cluster pruning query:**

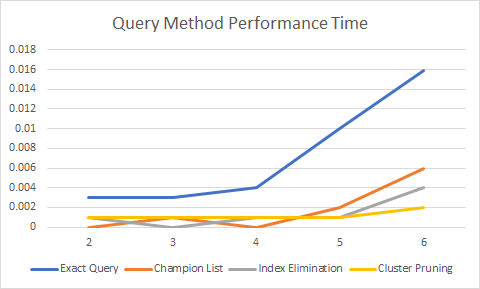
N=2: 0.00099730491638183594 seconds

N=3: 0.00099754333496093750 seconds

N=4: 0.00099754333496093750 seconds

N=5: 0.00099730491638183594 seconds

N=6: 0.00199460983276367188 seconds



As we can see, Cluster Pruning and Index Elimination do much better than exact retrieval and champion list. Some points have dropped due to maybe choice of query words. But the trends would remain constant for other queries.

Since we precompute the values of document vectors and magnitudes, it makes it much faster to calculate similarity between two documents.

**Choosing R values:**

Since we have to build the champions list in the indexing itself, we need to be careful of the choice of the value for R. We don’t know the value of K in the beginning therefore I chose the following method of calculating R.

**r = int(sqrt(len(S))) + 10,** where S is the number of documents

For the following document collection sizes S:

S = 10, r = 13

S = 100, r = 20

S = 500, r = 32

S = 10000, r = 120

For this collection size of 423 documents we will get at least r = 30 documents

This value of **r** becomes a problem for ***K > sqrt(S) + 10.***