Information Retrieval Report:

Name: Ilay cohen, ID:206515744, Email: Ilayc11@gmail.com

Name: Victor Gavrilenko, ID: 209406255, Email: ###########

GitHub: [GitHub Repo](https://github.com/Gavision97/Information-Retrieval---Wikipedia-Search-Engine-Project)

Key Experiments –

We started by building indexes using the InvertedIndex class and the file for gcp from assignment 3. We built indexes for the title, body, we built a dictionary for document and its length, and another one that contains a document title and its id for faster access.

After that we built two more indexes using PorterStemmer in our tokenization part, one index for title stemmed and body stemmed.

In each one of our tries, we used PageRank from assignment 3. The first thing we tried was CosineSim. We tried it multiple times – with stemmed query and with regular query, with stemming the indexes and without, we tried it with different weights. We saw that our results weren’t fast/precise enough. Next thing we tried was BM25.

Using BM25 and everything stemmed

Comparing duration of search and result quality vs Num of words

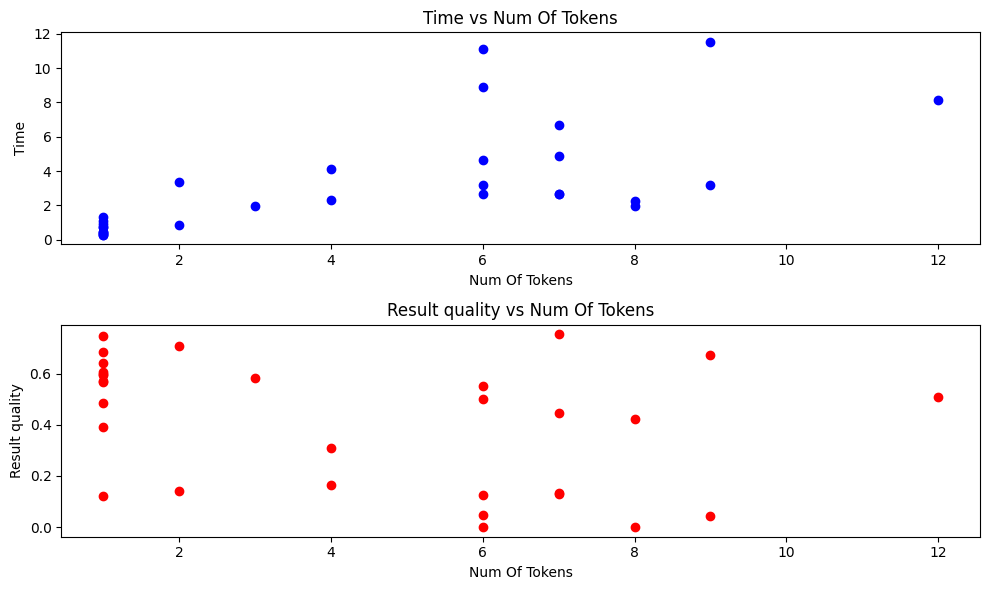
A graph of words with red dots

Description automatically generated

The slowest query was 'Who is considered the "Father of the United States"?' with a duration of 15.852468252182007. The fastest being ‘Bioinformatics’ with a duration of 0.5975043773651123.

The most precise query ‘When was the Gutenberg printing press invented?’ with precision of 0.754. the least precise were 'Who is the author of "1984”?, "What is the structure of the Earth's layers?" with a precision of 0.0. That is to be expected, we built our indexes so if a word didn’t appear more than 50 times it will not be included, that is to earn more speed on documents retrieval.

Next we added threads and async. We split our program to two asynchronous tasks- one for title and one for body. When each task generates a thread for each token in the query and each thread job is to generate a posting list. We can see that we were able to shave a few seconds from our slowest query and push the average down.



\*\*\*\*\*\*\*NEED TO ADD WE TRIED WORD2VEC\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

|  |  |
| --- | --- |
| Name | Size |
| body\_index\_.pkl | 4533953 |
| body\_index\_final.pkl | 12270824 |
| body\_stem\_index.pkl | 16732350 |
| title\_dictionary.pkl | 177080269 |
| title\_dl\_.pkl | 44396028 |
| title\_index\_.pkl | 480035 |
| title\_index\_final.pkl | 843370 |
| title\_stem\_index.pkl | 894252 |
| doc\_l2\_norm.pkl | 88844500 |
| pageRank.pkl | 33903328 |
| TOTAL: 10 objects, 362,984,869 bytes (346.122 MiB) | |