PROJECT REPORT

Design:

The major part of this phase's design is taken from phase 2 of this project which included the inverted index construction for the given document collection, and all the methods that are required have been implemented as instructed in the class. As a result, almost all the relevant documents from the relevance judgement file are being retrieved. The ranks of the documents might slightly differ. In this phase the following four components or methods where implemented.

1. Tf_Idf Scheme
2. Normalization Scheme
3. Cosine Similarity Scheme
4. Precision and Recall Scheme

Tf_Idf:

- In this method the term frequency which we already had in the inverted index where taken based on the terms that are appearing in the query.
- For query, a separate inverted index was constructed.
- Since Inverse document frequency or idf is same for both document collection and query, the idf that was calculated for document was used for query as well.
- Idf was calculated using the formulae explained in class which idf = Log10 (N/idf).
- Finally, the calculated idf was multiplied by the term frequencies in the inverted index to find out score of the term.

Normalization:

- For normalization, the Tf_Idf values where first divided by their Euclidean distance.
- Euclidean distance was calculated by squaring each Tf_Idf, adding them and then applying square root, then later Tf_Idf of each document is divided by its Euclidean distance, the same is done for the query as well.

Cosine Similarity:

- For finding the cosine similarity the normalized terms in a document collection for a particular document are multiplied by the normalized terms in the query document.
- After which the cosine similarity is found.

Precision and Recall:

- For precision the number of relevant documents from the retrieved documents are taken and are divided by the total documents that are retrieved.
- For recall the number of relevant documents are taken and are divided by the total relevant documents from the relevancy judgement file.

Data Structures/Classes:

1. Map<String, Entry<Integer, Map<Integer, Integer>> - A HashMap to store inverted index.
2. Map<String, Map<Integer, Double>> – A HashMap to store term and it Tf_Idf scores.
3. Math Class – The math class was used to calculate the logarithm values and to apply square root.
4. Decimal Format Class – Decimal format class was used to limit the number of decimal values to 6.

Time Taken:

- Time taken by the search engine is approximately 7.10 seconds.

System Performance among different query settings:

Query	Settings	Docs Retrieved	Relevant Docs Retrieved	Total Relevant Docs	Precision	Recall
352	Title	1113	2	2	0.001795	1.000000
353	Title	125	7	10	0.055556	0.700000
354	Title	446	6	9	0.013423	0.666667
359	Title	873	1	1	0.001144	1.000000

352	Title + Description	1898	2	2	0.001053	1.000000
353	Title + Description	1897	8	10	0.004215	0.800000
354	Title + Description	2387	8	9	0.003350	0.888889
359	Title + Description	1505	1	1	0.000664	1.000000
352	Title + Narrative	3764	2	2	0.000531	1.000000
353	Title + Narrative	3372	10	10	0.002965	1.000000
354	Title + Narrative	2119	8	9	0.003774	0.888889
359	Title + Narrative	2652	1	1	0.000377	1.000000

Table 1: Performance of the Search Engine

Name: **Sri Sravya Tirupachur Comerica**

ID: **11259523**