INFORMATION RETRIEVAL SYSTEMS

# Experiment Report

Name : Sai Vaishnavi Alla

ID : 30923115

Each submission is to include an **Experiment Report** of the tests that are conducted, the results of each test, the analysis and conclusion drawn from the experiments.

The main aim of this assignment is to create an Information Retrieval engine that will index the documents and also search for the nearest matching terms. I have used Java to create the System. Listed below is the procedure I have followed to create an Information Retrieval Engine.

1. The Program has two main functionalities :

* Search
* Index

2. Initially for doing the indexing I have used the regex function which is used in the splitting process and helps in tokenization. We identify various terms such as :

* Emails
* IP Digits
* Acronyms
* Quotes
* Equations

3. The Stemming process is done, and I have also removed the duplicates and stop words. This was done using Vector Operations. The Stemmer algorithm was used in the vector to adjectivity of the word and leaves us with the core word.

4. The directory files and the file containing all the tokenized words are loaded into the Hash Map.

5. For each term I have found the term count against every file in the directory and stored it in the Hash Map table. A multi key hash map table is used so that the term count can be obtained using a key of term and file.

6. Within the Hash Map we calculate the idf(Inverse document frequency) and the contents of the Hash Map are stored in a Invert Table. l

7. This final output for the indexing is an Invert table.

**Search**

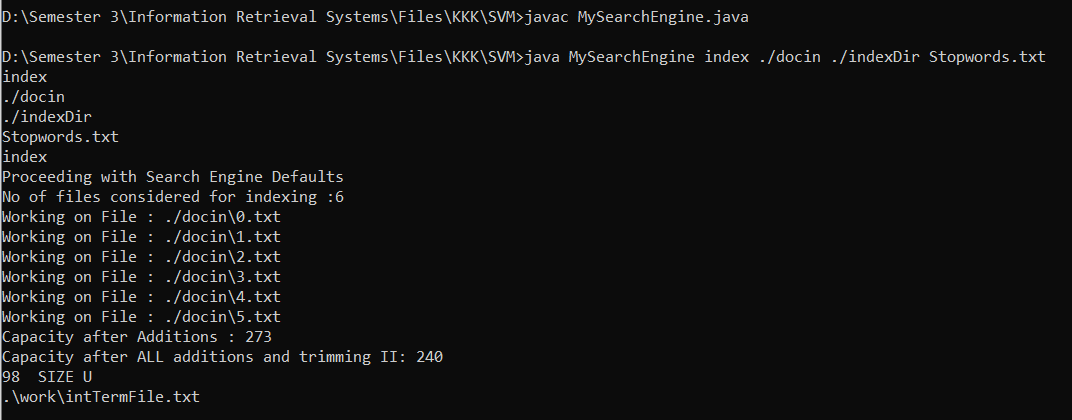
1. For the search part we give input as the search functionality , number of documents and query words as the input.

1. Invert file loaded into the Hash table. Instead of loading the term count , the term weight is loaded. The term weight is calculated using the formula tf \*idf. The idf can be calculated as follows.

3. The required query is also loaded into Hash table along with its term weight.

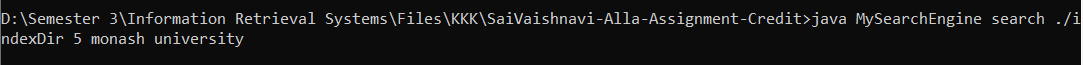
4. Now that we have i\*df , term count and term weight available to us cosine similarity will be calculated .I have then sorted the result using the Quick sort algorithm.

5. The program is also designed to run without the command line parameters. All information will be picked from the configuration file. A minor alteration should be made in the configuration file .



In the above example for indexing I have taken only 6 documents for testing. The file size is also displayed. For the files we get the invert tab as follows





I have used the above command for searching the most favourable term in the document.

From the above program I have analysed that larger documents take longer time to run.