**CSCE5200 Information Retrieval**

Project Phase 2: Search Engine Index Construction

**Project Phase 2 Description:**

An IR Engine includes the following three major components:

* Text parser
* Indexer
* Retriever

For this phase, you are asked to build the Indexer. It will include building a forward index and an inverted index for a large document collection. You will use this infrastructure in the next phase which involves implementing the query processing and retrieval portion of the vector space model for IR.

**Forward Index**

The forward index stores a list of words for each document. The following is an illustration of the forward index:

|  |  |
| --- | --- |
| Forward Index | |
| **Document** | **Words** |
| 1 | cow 2; moon 4; sum 10; …….. |
| 2 | cat 1; hat 6; flower 4; ……. |
| 3 | dish 5; spoon 4; ………. |

In your built forward index file, stores the document information in the following format:

docID1: …; wordIdi: freq in docID1; wordIdi+1: freq in docID1; ……….

docID2: …; wordIdj: freq in docID2; wordIdj+1: freq in docID2; ……….

(only record the words (and frequencies) occurring in a document)

**Inverted Index**

The inverted index stores a list of documents for each word. The following is an illustration of the inverted index:

|  |  |
| --- | --- |
| Inverted Index | |
| **Word** | **Documents** |
| orange | Doc1: 3; Doc3: 4; Doc4: 5; …. |
| cow | Doc2: 5; Doc3: 8; Doc4:1; …… |
| computer | Doc5: 10; …. |
| moon | Doc7: 9; …. |

In your built inverted index file, stores the word information in the following format:

wordID1: docId1: freq in docID1; docId2: freq in docID2; ……….

wordID2: docId10: freq in docID10; docId12: freq in docID12; ……….

**Resources to be provided:**

* Stop words list and a Stemming Algorithm (Porter stemmer)
* TREC document collection (to be indexed).

What to submit:

* Source code via **Canvas** online submission.
* A project report (*maximum* ***2 pages***) that shows how you designed your IR Engine; you should also document how long it takes to index the given collection, and the total size of your index. The report should be submitted online along with the code.
* A ***readme.txt*** file like the one submitted for the first project. It should include the instructions to compile your code, the input arguments to be provided to your main program, etc.

**Testing:**

For this project, you are expected to index the entire corpus that will be made available to you. For testing, we will use several test files which will contain a few sentences.

Your program should be able to parse the test file, create the indices for the test file, provide an interface for the user to specify a term, and output the information in the inverted index of the test file corresponding to this term on the screen.

// Given the specified term, search the dictionary file and if exists, retrieve all the information about this query term from the created inverted index, including all documents it appears in and numbers of its appearance in each document.