**IR Phase 3 Requirements**

The project in its entirety is now presented.

You started off the project (Phase1) with a prior homework which was to submit 10-20 queries to a search engine and store 150-200 documents that were results of those queries. This formed the corpus of your project and you were to store which docs were returned by which query. The project as a whole is to implement a "search engine" that will search the corpus and return documents that you feel will match the query. In Phase 1, you implemented an inverted index for all the documents. This created an index list with pointers to the corpus. Then in Phase2, you provided (or and ONLY here you can use someone else's s/w for) Porter's algorithm to understand which words in the index are related to which words, thus allowing for a match based on Porter's algorithm words in query that are "similar" to the words in the index list. The usage of Porter's algorithm should be an option that the user can control on the command line as a flag/switch. Using Porter's algorithm means that the words in your query will not only return those documents that have the word itself (as stored in the original inverted index) but also return any documents that have words that according to Porter's algorithm shares the same root as the word in the query.

Regardless of whether the user invoked Porter's algorithm or not, (the user will set a flag at command line for either option) your "search engine" should display on a GUI (any suffices, including using the simplistic Java AWT or Java Swing set) a results page which returns lists of the document names followed by snippets of the document following each document name, and any other information useful (line number/position in document? document stats?--I will leave this up to you).

Whether the results are read and/or displayed on a GUI or using a text file should be another option flag from command line.

This "output" switch should have three choices: GUI alone, Output Text file alone, or Both.

The default should be output textfile only. The user should be allowed to supply the name of the input and output files but you should create one as a default in case no name was provided. Finally compute the recall and precision of your search engine for each of the original queries to see how many of the original docs related to the query that are in your corpus were returned by your engine.

In summary, Your system should run from the command line and read in a query or series of separate queries from an input text file and you should likewise (in addition to the GUI) maintain an output text file which prints out the queries, the results with the snippet text (and any useful stats) provided by your system and the computation of recall and precision for each query, after listing the query and the returned documents. Persistence should be provided (manner up to you) so that the inverted index is not computed each time your system runs, but can load the originally computed inverted index into memory when the system loads.

Submit all relevant files (code, documentation, user manual, installation text, input and output) by attaching them to one email from Qmail to Class Project email using the following Subject Line:

Subject: IR Phase 3 Submission <Due-Date-Of-Phase3>