Information retrieval & Web Search
Assignment

All work *must* be your own to be considered for grading.

Your application must be submitted in a single .zip archive in the format,
Student#_Name_Assignment1.zip (this should contain you and your group member name and student number.)
And for those without group member then only your name and student number should be on it.

Only 2 Members in a group.

You MUST include a readme file detailing how to compile and run your application

(5 marks: based on the quality and accuracy of your readme file).

The Document to use for the application is IRW1.txt (this is your source document)

Resources:
A working sample of 2000 documents is provided in the file IR1.txt.
Each document is formatted as
Document ID
TITLE
TEXT
****
Document ID
TITLE
TEXT
****
A sample set of 5 queries is provided in IR1_Queries.txt. Each query is identified by number and each query ends with the pattern " ## "

Part 1: (40 marks)

Your application must read in all files in a folder located in the same folder as the application is run from. (for example: Assignment1/Assignment1.class , should have Assignment1/files as a folder - you must specify the setup in your readme file!). All text files in the folder should be pre-processed and used to prepare the Index.
The application should output a TF-IDF Weighted Term document Incident matrix to use for the information retrieval step. This can be stored in any format you want.
The application *must* output a text file containing an Inverted Index which will be used to assess the quality of the pre-processing stage of your application.

Note: after the initial command line execution, your application should not require any human interaction to finish the pre-processing stage.

Use whatever library you want for the Stopwords and Stemming.

Part 2: (40 marks)
Your application must read in a series of queries from a text file with each query separated by a line of asterisks (**********). Each time the query module is run it should output a ranked list of relevant documents using the Vector Space model of Information Retrieval. The ranked list should include the Cosine similarity score of the relevant documents as well as the document number and document title. The Query module should not require the preprocessing step to be re-run each time a query is made (make two applications, 1 for processing, 1 for querying). The query module should not require human input once it has been started. The ONLY argument that should be required is the name of the query file to be read in.

Commenting and Presentation (15 marks)
You MUST comment your code. The quality of your comments can be awarded up to 15 marks. Code must be clear with whitespace to separate logical sections of the applications. Your output must be easily read and clearly presented.

Rules:
You can use any coding language of your choice.

If your code does not compile according to your readme, you will be docked 30 marks. It MUST compile.

You will be awarded marks based on the quality of results and the time taken to finish outputting the file for part 1 and , timed separately, to complete each query run in part 2.

Your application will be tested against another Document file and queries. To see how well it handles file access and processing.

Best of luck…