COMP3009J – Information Retrieval
Programming Assignment

This assignment is worth **20% of the final grade** for the module.

Due Date: Monday 27th May 2019 at 08:00

Before you begin, download and extract the file ``lisa.zip'' from Moodle. This contains several files that you will need to complete this assignment. The README.txt file describes the files contained in the archive and their format.

Part 1: BM25 Model

For this assignment you are required to implement the **BM25** of Information Retrieval. You must create a program (using Python) that can do the following.

1. **Extract the documents** contained in the LISA document collection. These are contained in the lisa.all.txt file. You should divide the documents into terms in an appropriate way. The strategy must be documented in your source code comments.

2. Perform **stopword removal**. A list of stopwords to use is contained in the stopword_list.txt file that is on Moodle.

3. Perform **stemming**. For this task, you may use the code that has been posted on Moodle (porter.py).

4. The first time your program runs, it should create an appropriate **index** so that IR using the BM25 method may be performed. Here, an index is any data structure that is suitable for performing retrieval later.

This will require you to calculate the appropriate weights and do as much pre-calculation as you can. This should be stored in an external file in some human-readable format. Do not use database systems (e.g. MySQL, SQL Server, etc.) for this.

5. The other times your program runs, it should load the index from this file, rather than processing the document collection again.

6. Accept a query on the command line and return a list of the 15 most relevant documents, according to the BM25 IR Model, sorted beginning with the highest similarity score. The output should have three columns: the rank, the document's ID, and the similarity score. A sample run of the program is contained later in this document. The user should continue to be prompted to enter further queries until they type "QUIT".

It is **ESSENTIAL** that this can be run as a standalone program, without requiring an IDE such as IDLE, etc. You can assume that the any files you need from Moodle will be in the same directory as your program (which will be the working directory) when it is run. Do not use absolute paths in your code.

Non-standard libraries (other than the Porter stemmer provided) may not be used.

Part 2: Evaluation

For this part, your program should use the standard queries that are part of the LISA collection to evaluate the effectiveness of the BM25 approach. The user should be able to select this mode by changing how they run the program.

For example, if users want to enter queries manually, they should be able to run (assuming your program is called search.py):

```
python search.py -m manual
```

Or to run the evaluations:

```
python search.py -m evaluation
```

For the evaluations, the standard queries should be read from the ``lisa.queries.txt’’ file. An output file should be created (named ``evaluation_output.txt’’). Each line in this file should have three fields (separated by spaces), as follows:

1. Query ID.
2. Document ID.
3. Rank (beginning at 1 for each query).

A sample of what this file should look like is shown below.

After creating this file, your program should calculate and print the following evaluation metrics (based on the relevance judgments contained in the ``lisa.relevance.txt’’ file:

- Precision
- Recall
- P@10
- R-precision at R=0.4
- MAP

What you should submit

Submission of this assignment is through Moodle. You should submit a single .zip archive containing the following:

- The source code for your program. This should be contained in one file only.
- A README.txt file with brief instructions on how to use your program (examples below).

Sample README file

Name: Zhang San
UCD Student ID: 13123452

By default, this system searches the LISA collection using the BM25 model, with queries supplied by the user.

A mode can be selected by using the -m flag. Possible options are “manual” and “evaluation”.

Selecting “evaluation” will run and evaluate the results for all queries in the lisa.queries.txt file.

Contact

If this specification is unclear, or you have any questions, please contact me by email (david.lillis@ucd.ie).

Sample Run (Manual)

```
$ python search.py -m manual
Loading BM25 index from file, please wait.
Enter query: library information conference

Results for query [library information conference]
1 928 0.991997
2 1109 0.984280
3 1184 0.979530
4 309 0.969075
5 533 0.918940
6 710 0.912594
7 388 0.894091
8 1311 0.847748
9 960 0.845044
10 717 0.833753
11 77 0.829261
12 1129 0.821643
13 783 0.817639
14 1312 0.804034
15 423 0.795264

Enter query: QUIT
```

Note: In all of these examples, the results, and similarity scores were generated at random for illustration purposes, so they are **not** correct scores.

Sample Run (Evaluation)

```
$ python search.py -m evaluation
Loading BM25 index from file, please wait.

Evaluation results:
Precision: 0.1382
Recall:    0.4124
P@10:      0.6210
MAP:       0.2532
```

Sample Evaluation Output File (First 15 lines)

```
1 408 1
1 1151 2
1 679 3
1 889 4
1 1068 5
1 1031 6
1 464 7
1 1185 8
1 292 9
1 751 10
1 117 11
1 94 12
1 1238 13
1 115 14
1 959 15
```