CA6005
2023-2024

Assignment 1

Search Engine

Mark **50**% of the module mark

Groups You will be working independently

Submission deadline **Friday 15 March 2023 – 23:59**
If you do not adhere to the prescribed submission guidelines below, the lecturer assumes no responsibility if your assignment is overlooked and consequently is not corrected.

How to submit? Upload your assignment on Loop in PDF format only and as one single file. Be sure to clearly identify your name on the title page of your assignment. There will be a follow-on interview scheduled during which you will be asked to demonstrate your information retrieval system operating.

The name of the PDF file should **respect the following naming template**:

SurnameOfStudent_NumberOfStudent1 _CA6005_Assignment1.pdf
Example:
Gurrin_1768573_CA6005_Assignment1.pdf

Objectives You will apply the knowledge acquired during the first two courses of this module to implement a simple Information Retrieval system that is able to index a small collection of documents, perform queries over it, and generate an output in the form of a ranked list of documents. As part of this project, you will conduct also an evaluation of this system in order to assess its performance.

Description More specifically, the project will include the implementation of the following components:

- **Indexing.** Into some form of inverted index.
- **Ranking.** You need to implement all three different retrieval models: Vector Space Model, BM25, and one Language Model of your choice.
- **Evaluation**. You should compare the three different models that you have implemented. To generate the results of your experimental evaluation, you should use the TREC evaluation programme (*trec_eval*) that you can find here(https://github.com/terrierteam/jtreceval) dyn_eval. A description of the metrics computed by trec_eval can be found here (https://trec.nist.gov/pubs/trec16/appendices/measures.pdf). Please either compile trec_eval on your chosen platform or use dyn_eval which provides various platform executables.

In this project, you will be working with the Cranfield collection, a small collection of domain limited 1,400 abstracts and associated queries. The collection contains the collection with the following files:

- README.md - describing the collection

- cran.all.1400.xml - an XML encoded version of the original 1,400 document cranfield collection.
- cran.qry.xml - the queries that you should use to evaluate your search engine
- cranqrel.trec.txt - the relevance judgements in trec_eval format.

The collection is available from here: https://github.com/oussbenk/cranfield-trec-dataset

In order to evaluate your retrieval models with *trec_eval* or *dyn_eval* you will need to generate an output file for each of the models containing the output of your search engine. The output file is a simple text file, it will contain a sequence of lines, each line corresponding to a retrieved document per each query.

If you are returning the top 100 documents per each query, and given that the folder topics contains 162 topics, your file will contain up to 16,200 lines.

For trec_eval, each line needs to contain exactly the following information:

query_id iter <space> document_id <space> rank <space> similarity <space> run_id

Notice that the values of iter, rank, and run_id are irrelevant for the computation of the metrics, and *trec_eval* uses the similarity to compute the rank and resolve ties, so it is important to print the actual value of similarity.

You can use either Python or Java for this project. Your project will be evaluated in terms of the PDF file, the online demonstration, choices taken and rational behind them, and a review of your source code during the interview.

The assignment will be assessed as follows:

- 10% Indexing
- 10% Ranking-VSM
- 10% Ranking-BM25
- 10% Ranking-LM
- 10% Evaluation
- 30% Report structure and clarity
- 20% Interview defence and code review. We will have an online interview for all submissions in late March.

Notice that your marks are mostly calculated from your report, failure to reflect in the report all the important processes and choices made in the project will result in lower marks.

The report should be written in ACM sigconf template, either the latex template

https://www.overleaf.com/latex/templates/acm-conference-proceedings-primary-article-template/wbvnghjbzwpc

Or the Word template

https://www.acm.org/binaries/content/assets/publications/word_style/interim-template-style/interim-layout.docx

The report length should be between 5 and 8 pages (exceeding pages will not be considered and the corresponding grades will be lost. Report shorter than 5 pages will also be penalized.)

The report must contain a link to a repository of your code.

The report documents your project. You should provide a general architecture of your system and provide motivations behind specific choices that you had to take when implementing the different components of your system. When these choices were supported by scientific publications, you should provide references and include them in a Bibliography section.

Your report should contain at least the following sections:

- **Abstract**. A quick overview of the content of the report
- **Introduction**. An introduction to the problem and an overview of the architecture of your system.
- **Indexing**. A section describing the process of indexing the collection. Here you can include specific choices that you have made in terms of; document analysis and pre-processing indexing construction, data structures.
- **Ranking**. A section describing the search and ranking component of your system. Here you can provide details about how you implemented the retrieval and ranking of documents, as well as specific choices and motivation behind data structures.
- **Evaluation**. In this section, you should provide details about how you have tackled specifically the Cranfield collection, in terms of the document structure, query creation, pre-processing, etc. This section must provide a table with the evaluation results in terms of MAP, P@5, and NDCG. You must provide also a discussion of the results.
- **Conclusions**. This section provides an overview of the main findings of your project: what worked well and what did not, what you would change and how this work can be extended in the future.

Remember, this is an individual project, and students are supposed to work individually.

Your project code and your report will be checked for plagiarism.

We expect all of our students to conform to DCU Academic Integrity and Plagiarism Policy https://www.dcu.ie/system/files/2020-09/1_-_integrity_and_plagiarism_policy_ovpaa-v4.pdf

NOTE: You cannot use any external library or API for indexing and retrieval (e.g. Lucene, MG4J, Solr, ElasticSearch, Lemur, Terrier, etc.).