IFN647: Assessment 2 Variation

Title: Search Engine Technology project
Brief description: Project to implement software using techniques based on search engine technology, and evaluation of the technique implemented.
Individual or Group: Individual or team of two or three (maximum).
Weight: 40%
Due Dates:
Variation Proposal: Tuesday 25th September 2018
Demonstration: Wednesday 24thOctober 2018, during the lecture.
Report: Tuesday 30th October 2017

Outline

After a successful trial in 2017, we have decided to offer IFN647 students the possibility to propose and implement a variation to the second assessment. Such option is subject to eligibility (see below). The individualized projects have to feature a number of functionalities similar to the mainstream project assessment, and will be graded against the same set of criteria. Typically, these individualized projects will use a different collection of documents than the Baseline Project, will require specific query analysis, and will add an extra load of creating evaluation materials for this collection (queries and relevance judgements). It is important to note that this additional load has no impact on marking, and it is understood that the students undertaking these projects are motivated beyond the basic assessment for the unit (e.g. prototyping an innovative product, developing a search functionality to enhance their IFN701 or IFN702 project, or participating in a shared task).
Additionally, students who undertake a variation for the assessment will be required to present their work in class in week 12 or week 13.

Eligibility

To be eligible to submit a variation for assessment 2, you need to **submit a variation proposal describing your objectives and your design by the 25th of September 2017**. Please follow the template provided at the end of this document. Feel free to discuss your plan before submission with your tutors or the lecturer. It is also a good idea to submit your plan earlier than the due date in order to receive feedback early and make adjustments where necessary.
Only students with an approved variation may proceed to submit the variation for assessment. If the variation is not approved, the students will proceed with the mainstream project description.

Core requirements

Functionality: the project must meet at least the minimum systems requirements presented below.
Evaluation: A strategy for evaluating the software must be defined and implemented. If the evaluation strategy does not follow the trec_eval standards, the program has to be delivered with evaluation tools and materials (gold standards, measures against gold standards, etc.) as well as any relevant software and appropriate documentation.

Interface: You are strongly encouraged to offer a web-based demonstration/application, as it will be a valuable showcase to add to your development portfolio when you are applying for a software development position. You may alternatively chose to provide a desktop-based interface or application. The system developed must include some minimum requirements for the interface, however, you are encouraged to be creative in the development process.

Final Presentation– Due October 24th 11am

The final presentation should include the motivations for your project, details of the general architecture, and information relevant to your implementation (for example use of external libraries, novel algorithms or formulas for relevance, rules or regular expressions, etc.). You will also need to include in your demonstration of the software, the evaluation strategy adopted and the final evaluation results. The expected duration for your presentation is 15 minutes. If you wish, you may provide a pre-recorded video presentation. In this case, make sure you provide it well in advance to ensure there are no technical issues encountered on the day of presentation.

Minimum System Requirements

You are required to produce an application that allows users to index, search and retrieve documents from your collection based on a specific information need. It is assumed that you will produce a user friendly Graphical User Interface. The functionality of the application is broken up into the following tasks:

Task 1: Index

- On start-up your application should enable the user to specify two directory paths. The first directory path will contain the source collection. The other directory path will contain the location where the index will be built.
- The user will then be able to indicate that they would like to create the index from the specified collection (e.g. through a button click or similar), which is then done programmatically.
- The application should build the index and report the time it takes to index to the user.

Task 2: Search

- Once the index has been built the user must be able to search it, based on the task description.
- The query needs to be pre-processed.
- The application should match the final query to relevant documents and rank the documents according to relevance. The application should report how long it took to search the index and display this on the GUI (include the time required for query creation)
- The application should display on the GUI how many relevant documents are returned from the search, as well as some information about the documents (for example, the document title, or the first line of the abstract if applicable).
- As part of the application's implementation you are required to offer both the baseline scoring functionality of Lucene (or equivalent search engine of your choice or of your making), and an improvement to this baseline scoring functionality.

Task 3: Browse Results

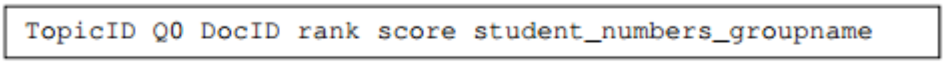
- The user should be able to move backwards and forwards thorough the ranked list of results, viewing a fixed number of results at a time.
- Alternatively, the result needs to be organized in an intelligent fashion (with categories, via a map, or some easily recognized construct)

Task 4: Retrieve Results

- Using an appropriate interface control, the user should be able view the document. This can either open in the existing window or a new window.

Task 5: Save Results

- The user must have the option of saving the list of the retrieved results in a text file. To do this the user will need to specify the name of the file to save the results and query identification. New results should be appended to the end of an existing results file.
- The format of the text file should be compatible with the *trec_eval* program. The format of which is as follows:

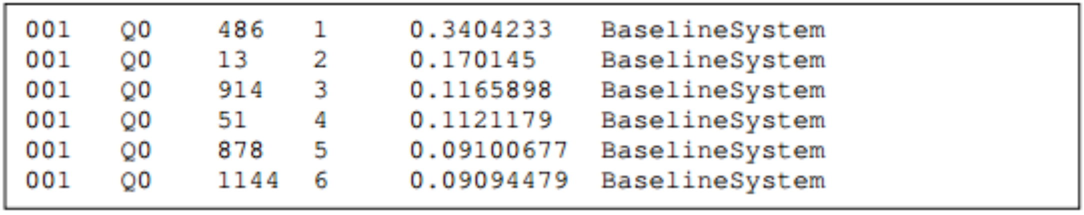


```
TopicID Q0 DocID rank score student_numbers_groupname
```

Where:

- ➢ TopicID is the query identification as entered by the user;
- ➢ Q0 is simply the two characters Q0;
- ➢ DocID is the respective document id;
- ➢ Rank is the rank of the file as returned by your application;
- ➢ Score is the relevancy score of the file as determined by your application;
- ➢ Student_numbers_groupname are the QUT student numbers for all members of the group (delimited by underscores) and a group name (e.g 0123456798_0987654321_ourteam). This is used to identify the result file generated from your application.

All parameters are separated by whitespace. A sample of the results file from the BaselineSystem is provided next.



```
001    Q0     486    1      0.3404233    BaselineSystem
001    Q0     13     2      0.170145     BaselineSystem
001    Q0     914    3      0.1165898    BaselineSystem
001    Q0     51     4      0.1121179    BaselineSystem
001    Q0     878    5      0.09100677   BaselineSystem
001    Q0     1144   6      0.09094479   BaselineSystem
```

Note that in order to be compatible with *trec_eval* the results file needs to be a unix formatted file. You can use the program ***dos2unix*** to assist with this.

Task 6: Advanced features

Depending on your collection of task, you will be implementing some additional processing to the collection, the queries or a radically different ranking function. These advanced features may include classification, information extraction, translation, transcription, query expansion and weighting, complex use of Lucene's fields, query intent prediction, and probably others we are yet to imagine.

Assessment Criteria

The assessment criteria are separated into two parts:

1. System functionality, as defined in the variation proposal.
2. Written report, following the instruction of section 4 of the mainstream project description. The written report may include any part of the variation proposal.

Marks will be assigned as follows. Please note that marks may be deducted for reasons which include:

- failing to meet supplied criteria
- poor communication
- inadequate descriptions,
- poorly considered projects

Functionality Assessment Criteria

Part	Marks
Functionality as per specification in variation proposal	50

Report Assessment Criteria

Part	Marks
Statement of Completeness	2
Design	10
Changes to Baseline	8
System Evaluation	10
Comparison to Baseline	10
User Guide	3
Advanced Features	8
Report Total	50

Total Marks

Part	Marks
Functionality	50
Report	50
Report Total	100

Project Plan (template)

Student Name:

Aims of the project

Describe here in one or two sentences what your project is about, and how it is related to another project of yours. This may be a future project that has not yet been developed. However, it must be a natural extension, further development, or have some inherent relation to this existing project.
(For example. "Creating a complete search engine" or "the aim of this project is to develop a query expansion module to acknowledge variations in first name spellings for the QUT search engine" or "implementing a search engine for a collection of OCR'd documents").

Collection

Describe the documents that your users will be able to access through your system, and how you will acquire this collection of documents.

Task/information need

You need to describe

- the type of queries that your users will be able to enter in your system. You must also provide examples of 5 queries and the expected documents that they should return (discuss this with the teaching team if your collection is very large).
- The pre-processing that will occur to handle your queries
- The information that the users will need to see from the documents.

Note: There are no marks for the document collection and evaluation materials, however it is a requirement for project variations.

Project design

Draw a schema with the various components, inputs, outputs and resources of your system. Include in the schema existing libraries that you will use, existing libraries that you will modify (and what you will modify) and new components that you will create entirely.
You may add footnotes to your schema to explain more some important parts, and you can also add subparts to the schema to describe how some parts of the process you will implement will work. You may also include use cases where relevant.

Programming language(s) used

System requirements

These must be derived from the minimum system requirements. They will serve as a basis for assessing the functionality of the system. This should clearly include the advanced features that your system will be implementing.

Evaluation strategy

If you already have an evaluation strategy in mind, feel free to include it here for feedback. But you don't have to have it all figured out just yet.