Dublin City University
School of Computing
CA4009: Search Technologies
Laboratory Session 1

November 2018

Module Coordinator: Gareth Jones

Laboratory Tutors: Abhishek Kaushik, Procheta Sen

1 Introduction

The purpose of the initial laboratories for CA2009 Search Technologies is to enable you to explore the text indexing and ranked information retrieval methods covered in the module, see the Section 3 of the lecture notes on Text Retrieval. The laboratories will cover: analysis of the statistics of text documents, indexing of documents, ranking algorithms used to support effective search, and evaluation of the effectiveness of these algorithms via standard metrics using experimental test collections. The technical details of the algorithms used are described in the lectures notes and you should consult these in the first instance if you are not sure what the laboratory instructions are referring to.

In the first laboratory you will explore the statistics of a collection of documents. and observe the use of text search requests on the collection with standard ranking algorithms.

2 Laboratory Reports and Submission

You should create a electronic report file for this laboratory.

- Include the title and date of the laboratory, and your names at the beginning of your report.
- For each activity described below, enter your answers into your report file, making clear which section your response relates to.
- At the end of the laboratory session you must upload your report to the CA4009 loop page via the link for this laboratory.
- If you do not finish the exercises to your satisfaction, you can complete the assignment in your own time, and submit a second extended report. The latest date for submission of the extended report is one week after the laboratory session.

3 Test Collections and Evaluation Software

The laboratory is based around a standard information retrieval test collection, and standard software used for research and development in information retrieval.

3.1 TREC Test Collection

The information retrieval test collection used for the laboratory is based on one the standard datasets created for the Text REtrieval Conference (TREC) introduced in lectures. The collection consists of:

- A collection of about 500,000 news articles taken mainly from the *LA Times*. The edition of the paper containing these news stories was published about 20 years ago, so you probably won't be familiar with most of the stories. The use of old news documents is quite common in natural language technology research since publishers are more willing to let researchers use them for free or at low cost.
- A set of search requests (referred to at TREC as *topics*). Each topic has three fields:
  - Title: A short statement of an information need. Typically one sentence in length. You can think of this statement of the information need as being a rich version of a query to a web search engine.
  - Description: A longer more detailed statement describing the information need. Typically several sentences in length.
  - Narrative: A still longer statement, typically describing the attributes of a relevant document for this topic. This sometimes also explains the attributes of documents which should be regarded as non-relevant.
- Relevance Assessments: A list of documents manually judged to be relevant or non-relevant for each topic. The relevance judgements were created using a pooling procedure as described in the lecture notes. Any document in the collection which does not appear in the relevance list for a topic, has not been judged for this topic and is assumed to be non-relevant. Each topic may have one or more or no relevant documents. The list of relevant documents for each topic in the test collection is combined together in a single file conventionally referred to as the "qrel file".

3.2 Indexing and Information Retrieval

The documents in this TREC collection have been indexed for this laboratory using *Lucene*, an open source toolkit used for research in information retrieval and development of commercial information retrieval systems. You can find more information about Lucene here: `https://lucene.apache.org`.

The documents were indexed by applying standard methods of stop word removal (a standard list of 571 stop words) and Porter stemming. Using the indexed file, the document collection can be searched using several standard information retrieval ranking algorithms. Stop word removal and stemming are also automatically applied to search queries entered into interface.

The indexed collection can be accessed for the laboratory exercises via a web interface.

3.3 Evaluation Software

Information retrieval systems can be evaluated using a range of wide range of standard metrics. The most commonly used are the *precision*, *recall* and *mean average precision* based measures introduced in the Section 3: Text Retrieval of the Search Technologies lecture notes.

As described in lectures, precision and recall measures for an information retrieval system can be calculated using a test collection of the form outlined above. To evaluate the metrics, the document collection is indexed using the information retrieval system, each search request is applied to the indexed documents in turn, the ranked output for each request created by the information retrieval system is collected, and the

evaluation metrics calculated using the ranked output and the qrel file which indicates which documents are relevant.

A standard software application called "trec_eval" is available to compute these standard evaluation metics for TREC formatted a qrel file and a suitably formatted ranked list. You will learn about using trec_eval in a later laboratory session.

4 Document Collection Statistics

As described in lectures, information retrieval ranking algorithms for text documents are generally based on statistics of the document collection.

The purpose of the first section of the laboratory is to examine and try to interpret some of the information in the indexed document collection.

- Login to your School of Computing linux account.
- Open a web browser and access the indexed TREC collection via the following URL `http://136.206.48.185:8084/IRModelGenerator/search.jsp`. This URL is only accessible on the DCU campus.

  For this section of the laboratory select the "view collection stats" option to view come of the collection statistics.

  This displays a list of the terms in the document collection ranked by the number of documents that each term appears in and also shows the total number of occurrences in the collection.
- Examine the occurrence count information for the terms in the list that you can see, consider the values and what they tell you about the terms and their likely usefulness for information retrieval. Enter the results of your analysis in your report file for today's laboratory.

5 Interactive Searching using Lucene

The purpose of the next part of the laboratory is to search the TREC document collection using Lucene via the web interface.

- Select the option "search" interface, and select one of the ranking functions: tf-idf or BM25. Default values for $k = 1.2$ and $b = 0.75$ are present for BM25, but you can change these in the interface.
- Enter a short search request into the query box.
- Examine the list of retrieved documents.

  This displays the ID within the TREC collection of each document., and a text snippet for each document. The snippets are based on sentences containing the query terms. You can open each of the original documents by clicking on the document ID.
- Explore the relationship between the words that you entered into the query box, the contents of the snippets, the contents of the documents, the ranking function that you chose and the ranking of the documents. See if you can gauge how the term weights and document lengths might have produced the ranked output that your see. You can enter more search requests to explore this relationship further.
- Search again using the other ranking function. Do you notice anything different about the retrieved lists?

- Enter your observations about the relationship between the search requests, the snippets, the documents and either ranking functions into your laboratory report.

6 Coding Exercise

For the final part of this laboratory you will work with a small programme which calculates word frequencies in a single document.

You can download a template file *Word_Counter* from the laboratory section of the loop page. This program is available in both Java and Python, you can use whichever you prefer. You will be downloading a zip file which you need to uncompress in your directory.

You need to implement three functions into this program.

- Return the top k terms sorted by their frequency in the file. You should enable the value of k to be specified as an input parameter to your program.
- Extend your program to ensure that all tokens are completely in lower case, and return the top k terms.
- Perform stemming and return the top k terms.

For your lab report describe using examples the outputs from each version of your program.
Please submit your completed program using the appropriate link on the loop page.