Introductory Investigation into Information Retrieval using Lucene and the Cranfield Documents.

Breandán Kerin
14310166
M.A.I. Computer Engineering,
Trinity College Dublin
kerinb@tcd.ie

1. ABSTRACT

The Cranfield Documents are a set of documents developed by Cyril W. Cleverdon in order to evaluate the efficiency of indexing systems. [1] A project based on the Lucene Search Engine (**SE)** was developed to provide a greater understanding into the Lucene library and its capabilities within the realm of information retrieval (**IR**). Different methods of indexing and ranking the model have been investigated as well as field boosting.

1 IMPLEMENTATION

For this project, a SE was implemented using the Lucene IR library and developed in Java. To complete this project, there were several components that were needed to be completed, these are as follows:

1. **Load and Parse supplied document collection**
   The Cranfield documents were read in from the file *cran.all.1400* and parsed based on the document tags that are provided within this document. These tags are *'.I' – Document ID, '.T' – Document Title, '.A' – Document Author, '.B' – Document Bibliography, '.W' – Document Written Content*
   The above tags were used to identify and subsequently parse the various fields within each document, and were used to produce a list of Lucene Documents.
1. **Index the document collection**
   The list of documents were then indexed using Lucene *IndexWriter*. As part of the process involved in indexing the documents, both the *Analyzer* and *Similarity* model are supplied to process the document prior to being added to the Index. The Index was stored in memory using *RAMDirectory* for the purpose of performance.
2. **Load and Parse the supplied queries**
   The *cran.qry* file is parsed based on the tags: *'.I' – Query ID, '.W' – Query Written Content*
   It should be noted that when creating the query object, the query ID used was an incremental counter rather than the id supplied due to its non-linearity. This produces a list of queries that are used to query to document described above. A searcher was then defined using the Analyzer and Similarity model that was used previously when creating the Index in the IndexWriter.
4. **Execute the queries over the Cranfield documents**
   The queries supplied are executed in chronological order over the index created in step 2. This then produces a ranked list of relevancy scores for each query and for the top 1000 document values; which is Lucene's maximum number of relevant documents which can be ranked.
5. **Generate Mean Average Precision and Recall**
   "trec_eval is a tool that is used to evaluate rankings… that is sorted by relevance" and it was used to compute the MAP and Recall for the SE. [2]

2 EXPERIMENTAL DESIGN DETAILS

During the design and implementation of this project, there were several factors that could be examined and implemented to improve the results obtained. The methods implemented and tested were:

1. **Ranking model**
   The implementation developed allows the user to choose from a selection of implemented ranking models which are: BM25, Boolean, VSM/Classic, LM Dirichlet and Sweet Spot. Several methods for ranking were provided such that comparisons and conclusions about performance could be devised quickly and easily and such that an optimal combination of ranking and analyzer models could be identified quickly.
2. **Analyzers**
   Similar to the ranking models used, there were several implementations used for the Analyzer models which allowed the user to devise an optimal combination of ranking and analyzer models for the given SE.
   The Analyzer models used are: MorfologicAnalyzer, StopAnalyzer, SimpleAnalyzer, StandardAnalyzer, WhitespaceAnalyzer and CustomAnalyzer (Combines

LowerCaseFilter, TrimFilter, SnowballFilter using EnglishStemmer and StopFilter using StandardAnalyzer.ENGLISH_STOP_WORDS_SET).

3. **Field Boosts**
   A field boost was implemented in order to test and evaluate its performance on the SE. The purpose of which is to add bias weighting to the different sections of the document being queried. This is discussed more in the RESULTS AND DISCUSSION section below.

3 RESULTS AND DISCUSSION

As noted above, trec_eval was used to calculate a value for the MAP and Recall for the SE. The results for the MAP for the combinations of Ranking models and Analyzers are illustrated in *Table 1.1*. A plot for the Rank Model vs Size of Recall for the MorfologicAnalyzer is shown below.

For succinctness only this one chart is shown and highlights the trend that as the number of documents considered in Recall grow from 5 - 1'000, the Recall value grows to be approx. 0.93 - 0.96 from approx. 0.2 - 0.4.

The same results were shown in the graphs that didn't fit in the report, that for each model and analyzer combination the Recall improves to the point of relatively near convergence among all of the models; 0.9331-0.9631.

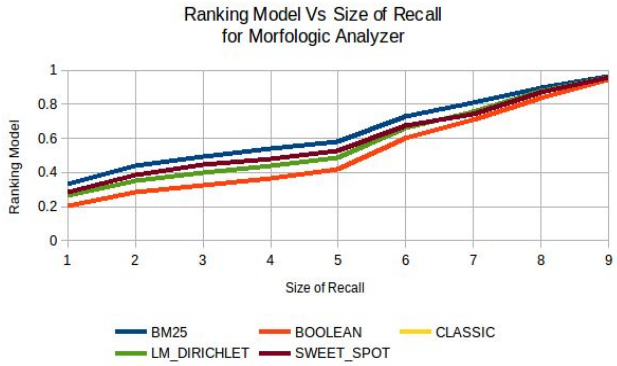
Although Field Boosts were briefly mentioned, trial and error showed the ideal boosting was: *title = 0.2*, *content = 0.8*, all others = 0.0. These values will change based on the SE implementation, the queries and the documents being searched

4 CONCLUSIONS

The best performing model was BM25 Custom: MAP = 0.4297, Recall = 0.9548. The worst performing model was LM Dirichlet Whitespace: MAP = 0.212, Recall =0.9381. Both of which used a static field boosting shown above.

This shows that the implemented solution performs well enough while not being massively performant but well enough to illustrate that a good understanding of Lucene has been achieved from this project.

Other methods which could have been implemented if time had provided for are: Query Expansion, Machine Learning models (e.g. Learning to Rank), implement more advanced ranking and analyzing models or a combination of more models previously used could.



REFERENCES

[1] Cyril W. Cleverdon, "Glasgow IDOM - Cranfield collection," 1960's. [Online]. Available: http://ir.dcs.gla.ac.uk/resources/test_collections/cran/. [Accessed 12 10 2018].

[2] Rafael Glater, "Learn how to use trec_eval to evaluate your information retrieval system", May 25 2016, Available: http://www.rafaelglater.com/en/post/learn-how-to-use-trec_eval-to-evaluate-your-information-retrieval-system. [Accessed 21 10 2018]

Access to AWS can be done by SSHing to:
ssh demonstrator@ec2-34-240-83-205.eu-west-1.compute.amazonaws.com
PWD: luceneproject1

2 **Table 1.1 -** *Illustrates the MAP score for each combination of Ranking and Analyzer model. As it can be seen the Custom Analyzer and Boolean Ranking model produce the highest MAP at 42.97%.*

	BM25	BOOLEAN	CLASSIC	LM_DIRICHLET	SWEET_SPOT
MORFOLOGIC	0.4059	0.2364	0.3513	0.3112	0.3513
STOP	0.4066	0.2834	0.3794	0.3001	0.3794
SIMPLE	0.4067	0.2371	0.3527	0.3107	0.3527
STANDARD	0.4069	0.2837	0.3767	0.3006	0.3767
CUSTOM	0.4297	0.2860	0.2860	0.3119	0.4010
WHITESPACE	0.3674	0.2120	0.3027	0.2120	0.2914