Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

School of Computer Science and Statistics

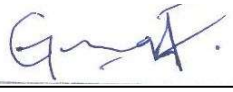
Assessment Submission Form

Student Name	GEORGE CHAVADY
Student ID Number	19305272
Course Title	M.Sc. COMPUTER SCIENCE – INTELLIGENT SYSTEMS
Module Title	CS7IS3 – INFORMATION RETRIEVAL AND WEB SEARCH
Lecturer(s)	GARY MUNNELLY, JOERAN BEEL, OWEN CONLAN
Assessment Title	Assignment 1 – Lucene and Cranfield
Date Submitted	28-2-2020
Word Count	405

I have read and I understand the plagiarism provisions in the General Regulations of the University Calendar for the current year, found at: http://www.tcd.ie/calendar

I have also completed the Online Tutorial on avoiding plagiarism 'Ready, Steady, Write', located at http://tcd-ie.libguides.com/plagiarism/ready-steady-write

I declare that the assignment being submitted represents my own work and has not been taken from the work of others save where appropriately referenced in the body of the assignment.

Signed:  Date: 28-2-2020

Introduction

Searching and searching techniques is deemed very important in the field of Information Retrieval and Web Search. Search Engines indexes content or Information Objects in the Web and help retrieve them in an efficient manner. The task that I have submitted helped me get an overall understanding of how IOs are scientifically indexed and thus retrieved efficiently from a large collection of objects such as the World Wide Web.

Keywords

Analyzers, tokens, index, scoring, field boosting.

Methodology

The steps taken to implement the Search Engine could be divided into two stages:

Stage 1:

- Download Cranfield Data
- Download Sample Index and Search files
- Implemented Indexing and search functionalities.

Stage 2:

- Parse content and queries
- Index content by different fields.
- Query and store the results.

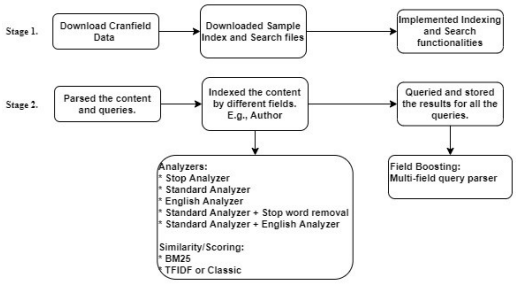


Figure 1: Steps to create a Search Engine using Lucene

Analyzers are the tools that creates tokens by parsing the documents. The different analyzers tested for indexing are Stop Analyzer, Standard Analyzer, English Analyzer, a combination of Standard Analyzer and Stop word removal and a combination of a Standard Analyzer and an English Analyzer.

Standard Analyzer:

This analyzer tokenizes based on sophisticated grammar and recognizes certain formats such as e-mail addresses and acronyms. It also converts words to lower case and removes stop words.

Stop Analyzer:

The Stop Analyzer divides text at non letter characters, lowercases, and removes stop words.

English Analyzer:

It is an analyzer for the English language. It tokenizes the documents after removing the standard stop words from the list of terms or words to be parsed.

Of all the analyzers, the combination of Standard analyzer and stop word removal gave the best results, a Mean Average Precision of '0.3501'.

The two different *scoring techniques* implemented and checked for are BM25 and Classic Similarity (TFIDF). Implementing scoring techniques didn't significantly improve the precision. *Field Boosting* is also implemented to give a higher weightage to different fields in the document that have been indexed. This has been implemented using a Multi-Field query parser. The boost values helped improve the performance of the query engine.

Finally, the results are evaluated with manually written relevance score using a tool called Trec_Eval.

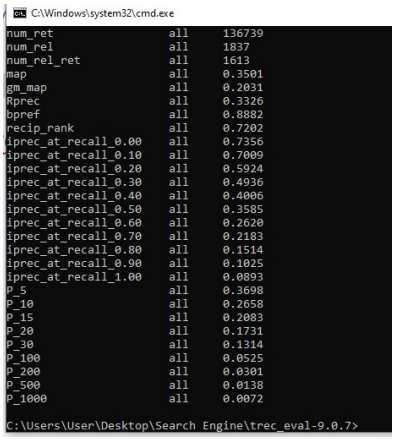


Figure 2: Precision scores generated by Trec_Eval

Resource Details

- Public DNS – 'ec2-3-80-189-102.compute-1.amazonaws.com'
- Github repository – 'https://github.com/georgejohnchavady/LuceneIR.git'
- .pem file – 'CS7is3.pem' is present in the github repository mentioned above.
- Command to ssh into the AWS instance – 'ssh -i <path to CS7is3.pem file> <user>@<public DNS>'
  E.g., ssh -i CS7is3.pem ubuntu@ ec2-3-80-189-102.compute-1.amazonaws.com