TAUGHT BY:

PROF. NADA NAJI

INFORMATION RETRIEVAL

SEMESTER I, FALL 2016

December 9, 2016

ROHIT RAJ  
SAMANJATE SOOD  
Sumit BHANWALA

INTRODUCTION

The goal of this project is to design and build an Information retrieval system. The performance of this system is then evaluated and compared in terms of retrieval effectiveness.

The project has two phases. The first phase is indexing and ranking the corpus. The second phase evaluates the results of the first phase.

Phase I

The first task of the project is to build four different retrieval modules namely Lucene, BM25, tf-idf, and cosine similarity. These programs are then run on the CACM test-collection. This gives use 4 different set of results.

We then use BM25 as the base search engine and perform query expansion using Pseudo relevance feedback approach to produce another set of result.

For the next task we use BM25 as our search engine and perform Stopping based on the common words list given. These words are neither considered for document scoring nor considered as a query term. This gives us the sixth and final set of results for this phase. In the same task we also index the stemmed corpus and run the stemmed queries on this index.

Phase II

Here we produce another set of results by performing stopping on the BM25 model with query expansion. We then evaluate the seven results. The performance assessment is done in terms of retrieval effectiveness. We take into consideration effectiveness matrices like MAP, MRR, P@K, and Precision & Recall tables to evaluate the performance of the seven distinct runs.

For the final task, we generate a snippet of the relevant documents based on the query which is loosely based on Lunn’s text summarization technique [1].

LITERATURE AND RESOURCES

Phase I

Task 1

Lucene [2] – This is a java program that uses three external Lucene.jar files to perform functionalities like indexing, parsing, and retrieval of documents based on the given queries.

We have used the java Lucene libraries to perform search using the queries on the test collection provided, CACM [3].

TF-IDF – For this model we calculate the query term weight and the document term weight. The sum of the product of all such terms is take. To calculate the weights, we simply calculate the term frequency of the term in the document and divide it by the summation of the frequencies of all the terms in the document. This is then multiplied by the inverse document frequency which is the logarithm of the total number of document in the collection divided by the number of documents that term occurs in.

Cosine Similarity – For this model we take the dot product of the term weights for the matching query and document terms. This is then normalized by dividing the dot product with the product of the length of two vectors.

BM25 -

BIBLIOGRAPHY

[1] Search Engines, Information Retrieval in Practice – W. Bruce Croft et. al. page 216

[2] Lucene Version 4.7.2 Documentation - <http://lucene.apache.org/core/4_7_2/>

[3] CACM document collection - <http://www.search-engines-book.com/collections/>