Information Retrieval system using Lucene

Neel Rajkumar Mishra (224143), Abhinav Srivastava (223683), Mohit Jaripatke (224651), Rajatha Nagaraja Rao (223758)

TEAM ROCKET

December 19, 2018

Versions used :
Lucene : 7.6
Java : 1.8.0

1 Introduction:

Information Retrieval system takes the user query and outputs a file relevant to the query of the user. It is similar to the search engines we come across daily which displays the results based on the search query of the user. This functionality is useful to get information from tons of document and display the most matching results first.

2 User steps:

Please note: Java packages and jdk should be installed on the user's machine.

- To run the jar file correctly keep the jar application and lucene library in the same folder.
- Command Line Argument input format :

java -jar IR P.jar [path to document folder] [path to index folder] [VS/OK] [query].

- *path to document folder* is where the files is stored.
- *path to index folder* is where the indexed file as a directory would be stored.
- *VS/OK* is choosing the model on which ranking of the documents would be based. There are two options to choose from Vector Space model and Okapi BM25 model.
- *query* is the term which the user wants to search among the documents

3 Working of the code:

The code will first navigate to the path of document and then read the HTML files i.e. file ending with '.html' extension. Also it will read texts file i.e. is ending with '.txt' extension. After reading the files the system would index all files it came across and form a directory, Porter stemmer is used to stem the contents of the page at the same time the user inputted query would also be stemmed and searched across all the documents. After this based on the model selected, ranking would be calculated and the top 10 documents would be displayed with their scores to user.
Below are the library functions and their purpose:

org.apache.lucene.queryparser. This function is used to convert the input string into a lucene query format using java

org.apache.lucene.search.IndexSearcher It scans the indexes and adds it to the existing index of documents.

org.apache.lucene.analysis.Analyzer Used for analyzing the stream of tokens.

org.apache.lucene.search.Query Base class for all kinds of queries

org.apache.lucene.store.FSDirectory Base class for Directory implementations that stores indexes

org.apache.lucene.search.TopDocs Stores the maximum hits of documents returned by Index Searcher

org.apache.lucene.index.IndexWriter Creates and maintains an index

org.jsoup.Jsoup Used to extract and parse HTML from a query string