Venkata Sai Abhishek Gogu - U00825706
Madhurya Gandi - U00843034
Information Retrieval Assignment 1

Lucene experience: Briefly describe your Indexer and Index Searcher built using the Lucene API. Include search enhancements you made along with justification for its effectiveness.

Indexer-Once Lucene is set up, we extracted the raw data and separated into different files from the cranfield data set. These files need to be indexed using the Lucene Indexer. The first question encountered during indexing is what analyzer should we use. The built-in analyzers available—WhitespaceAnalyzer, SimpleAnalyzer, StopAnalyzer, KeywordAnalyzer, and StandardAnalyzer. Among them we chose StandardAnalyzer because of the cranfield dataset. The dataset mostly contains strings and not only the meaning of the sentence needs to be preserved but also need to be effective in removing stop words. This analyzer has a JFlex-based grammar underlies it, tokenizing with cleverness for the following lexical types: alphanumerics, acronyms, numbers, words with an interior apostrophe, serial numbers. StandardAnalyzer also includes stop-word removal, using the same mechanism as the StopAnalyzer. After which the documents are created by adding necessary fields into it. Lucene allows influencing search results by "boosting" in more than one level-Document Level Boosting, Document Field Level Boosting and Query Level Boosting. We tried the first two types of boosting and noticed a better performance only for certain type of queries.

Index Searcher- Once a user enters the query, it is parsed and then analysed using the same analyzer which is used during indexing for better performance. We used a basic search as it can give generalised results for all types of queries. We tried several other query parsing such as- WildCard query, Fuzzy query, phrase query, regex query,etc. Once query is parsed, it is sent to the index searcher which returns the topdocs of the hits sorting by score(or relevance). TopDocs class in Lucene can be used to get several results like-The total number of hits for the query and the top hits for the query. Using these results we can display the best search results to the user.

Benchmark output: Present the results of five queries in the benchmark, preferably in the form of a table or any other form suitable for convenient visualization

Query	Precision	Recall	F measure
1	0.2068966	0.2068966	0.2068966
2	0.2	0.2	0.2
3	0.55556	0.55556	0.55556
4	0.66667	0.66667	0.66667
5	0.2	0.2	0.2