Information Retrieval System Report

Raffaele Cammi
Master's Degree in Computer Science Multimedia
University of Pavia

Exam: Information Retrieval

1. Introduction

This project focuses on the implementation of an Information Retrieval (IR) system based on the inverted index model. The main objective was to design a program capable of indexing a text collection and performing efficient term-based search. The system follows the classical Boolean retrieval framework, supporting both conjunctive (AND) and disjunctive (OR) queries.

The project was developed in Java and organized into several modules, including tokenization, stop word removal, stemming, and retrieval. The implementation also integrates key optimizations such as the Porter stemming algorithm and skip pointers for improved performance. The goal was not only to replicate theoretical IR models but also to understand how indexing structures, compression, and optimization affect the search process in practical applications.

2. Summary

The system is composed of modular components that reflect the standard IR pipeline. The `Tokenizer` splits text into tokens, the `StopWords` class removes frequent and semantically weak terms, and the `PorterStemmer` reduces words to their morphological root. After preprocessing, the `Indexer` constructs a dictionary that maps each term to a list of documents where it appears, known as the posting list. These lists are stored persistently on disk through the `IndexIO` class in three main files: `index.dict`, `docs.map`, and `collection.freq`.

At query time, the `Retriever` loads the inverted index and processes the user query, applying the same normalization steps (tokenization, stop word removal, stemming). The system supports Boolean queries with multiple terms and uses skip pointers to optimize intersections between posting lists. The graphical interface (`Gui.java`) provides an intuitive way to execute searches and view the resulting document paths.

Overall, the program achieves efficient retrieval and compact storage while maintaining simplicity and transparency in its design.

3. Critical Analysis

From a technical perspective, the system provides a complete yet minimal IR framework, emphasizing data structure design and modular programming. The use of a `HashMap` for the dictionary and `ArrayList` for posting lists offers straightforward implementation and fast access time. However, the approach may face scalability limits with large document collections due to memory usage and lack of compression.

The integration of the Porter Stemmer significantly improves recall by grouping inflected forms, although it may occasionally reduce precision when different words share the same stem. The StopWords filter is effective in reducing noise, yet its static list could be enhanced by adapting it to term frequency statistics from the corpus.

The most relevant optimization is the introduction of skip pointers, which reduce the time complexity of AND queries from linear to sub-linear by allowing jumps in posting lists. This demonstrates an understanding of algorithmic efficiency, although the heuristic used for skip distance could be dynamically tuned.

While the implementation is functional and well-structured, it remains a Boolean retrieval model.

4. Conclusions

The project successfully implements a functional Information Retrieval system combining theoretical principles with practical design. The modular structure, stemming, and skip list optimizations show a strong understanding of IR fundamentals. Although it currently supports only Boolean queries, it provides a solid foundation for advanced retrieval models.

Future developments could include ranked retrieval, phrase searching, and index compression. Overall, the project achieves its educational goal: bridging the gap between IR theory and the implementation of real-world search systems.