Methodology

System Architecture

The search engine implements a distributed BM25 ranking algorithm using a MapReduce pipeline with the following components:

Data Processing Layer:

Document preprocessing with NLTK (tokenization, stopword removal, stemming)

Term frequency calculation

Document length statistics collection

Storage Layer:

Cassandra database for efficient term-document indexing

Schema optimized for search operations:
Search Layer:

BM25 ranking algorithm implementation

Spark-based parallel scoring

Query preprocessing pipeline
BM25 Implementation:

Used standard parameters (k1=1.2, b=0.75) based on research

Implemented proper IDF calculation: log((N - df + 0.5)/(df + 0.5))

TF normalization accounting for document length

Performance Optimizations:

Batch processing in Cassandra (1000 records/fetch)

Spark parallelization for scoring

Prepared statements for Cassandra queries

Error Handling:

Comprehensive logging at all levels

Graceful degradation for malformed documents

Resource cleanup guarantees

Text Processing:

Porter stemming for term normalization

Aggressive punctuation removal

Length limits on processed tokens

Demonstration

just run the system using
docker compose build
docker compose up

No screenshots provided because my mapreduce algorithms does not work