Report on BM25 Document Search Engine Implementation

1 Methodology

2 Design Overview

The search engine implemented in this repository follows a classic document indexing and retrieval architecture based on the BM25 ranking algorithm. The system is designed to work in a distributed environment using Hadoop MapReduce for indexing and PySpark for retrieval operations. Cassandra is used as the underlying database to store index information.

3 System Components and Architecture

3.1 Data Preparation Pipeline

- Downloads a Parquet file containing document data (id, title, text)
- Extracts at least 1000 documents with the required fields
- Creates individual document files following the naming convention `<doc_id>_<doc_title>.txt`
- Uses PySpark to normalize and transform the data
- Stores the processed documents in HDFS under the `/data` directory
- Prepares the data for indexing by creating an RDD with appropriate format

3.2 Indexing Pipeline (Hadoop MapReduce)

Mapper (mapper1.py):

- Reads document data in the format: `<doc_id><doc_title><doc_text>`
- Tokenizes and normalizes the text using NLTK for:
  - Tokenization
  - Stopword removal

- Counts term frequencies for each document
- Outputs data in the format: `<doc_id>\t<term>\t<term_frequency>\t<doc_length>`

Reducer (reducer1.py):

- Groups data by doc_id
- Calculates document frequency (df) for each term
- Computes inverse document frequency (IDF) for terms
- Calculates corpus statistics
- Stores results in Cassandra tables

3.3 Cassandra Schema Design

- **terms** table: Stores vocabulary information
  - `term (text, primary key)`
  - `doc_frequency (int)`
- **documents** table: Stores document metadata
  - `doc_id (text, primary key)`
  - `doc_length (int)`
- **term_docs** table: Stores term-document relationships with BM25 scores
  - `term (text, part of composite primary key)`
  - `doc_id (text, part of composite primary key)`
  - `term_frequency (int)`
- **global_stats** table: Stores corpus-wide statistics
  - `docs_num (int, part of composite primary key)`
  - `total_doc_len (int, part of composite primary key)`

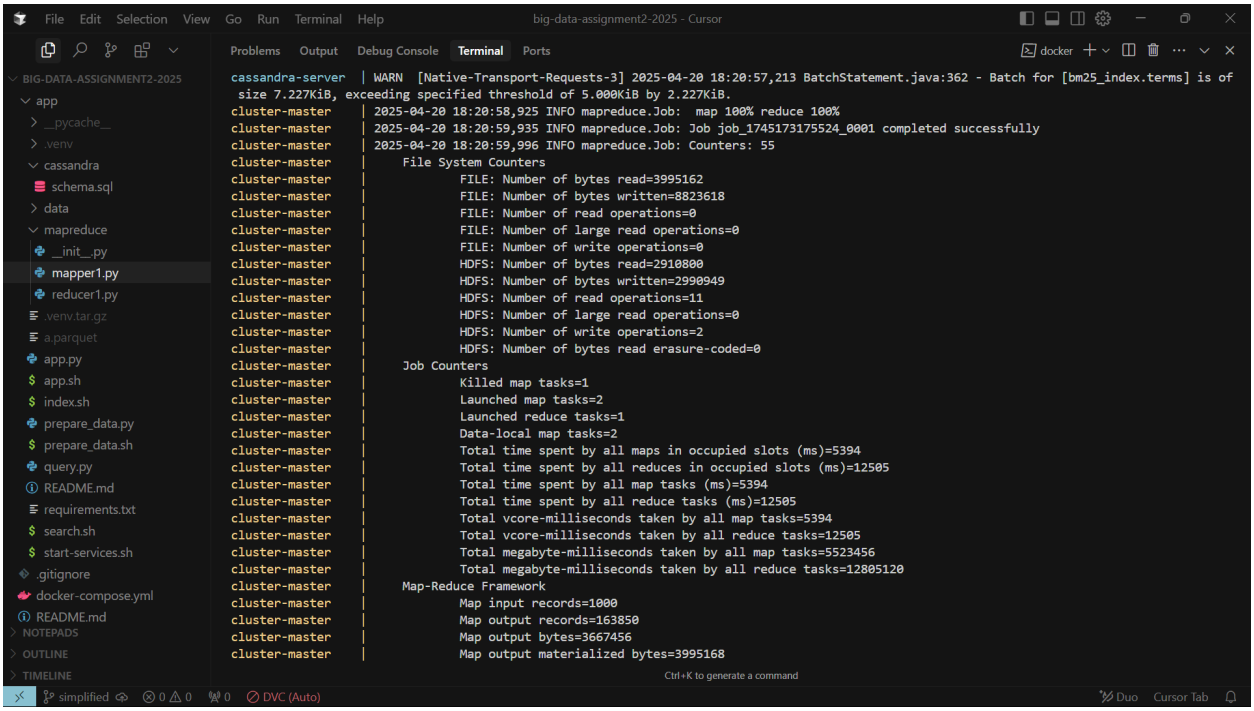

Figure 1: Enter Caption