Report

Abstract

This report details the implementation and optimization of several similarity models using Apache Lucene to improve the search accuracy of the Cranfield Collection. A key part of the project was the use of EnglishAnalyzer for document processing and extensive testing of models such as BM25Similarity, ClassicSimilarity and LMDirichletSimilarity. A Python script was developed for automated parameter tuning and performance was evaluated by TREC Eval. The final optimal parameters for BM25 were set to `k1 = 3.3` and `b = 0.8`, yielding the best MAP scores.

Introduction

In this project, we explored a variety of Lucene similarity models to enhance the retrieval of relevant documents.Apache Lucene's ability to process and search large collections of text makes it an ideal tool for this task, and we tested models such as BM25, TF-IDF, and Dirichlet smoothing.The Cranfield Collection is a standard dataset containing short summaries, which we indexed and processed using **EnglishAnalyzer**, which was selected for its ability to process deactivated words and stems. Our goal is to find the best performing similarity model and parameters to maximize retrieval accuracy, which is measured by MAP (mean average precision).

Project Structure

The project directory is structured as follows:

```
.
├── change_minus_one.py
├── data
│   ├── cran.all.1400
│   ├── cran.qry
│   └── cranqrel
├── find_best_bm25_args.py
├── index
│   ├── _5.cfe
│   ├── _5.cfs
│   ├── _5.si
│   ├── segments_6
│   └── write.lock
├── pom.xml
├── results
│   ├── query_results_bm25_3.3_0.8.txt
│   ├── query_results_classic.txt
│   ├── query_results_lmdirichlet_1500.0.txt
│   └── query_results_lmdirichlet_2000.0.txt
├── src
│   ├── main
│   │   ├── java
│   │   └── resources
│   └── test
│       └── java
└── trec_eval-9.0.7
```

Key files:

- **data/**: Contains the Cranfield Collection (`cran.all.1400`), queries (`cran.qry`), and relevance judgments (`cranqrel`).
- **src/main/java/**: Contains the Java classes used for indexing (`CreateIndex.java`) and querying (`QueryIndex.java`).
- **results/**: Stores the query results for each similarity model.
- **find_best_bm25_args.py**: Python script used for finding the optimal BM25 parameters.
- **trec_eval-9.0.7/**: TREC Eval tool used for performance evaluation.

Methodology

Analyzer Selection

We used the **EnglishAnalyzer** to preprocess the Cranfield documents. EnglishAnalyzer was chosen because it is well-suited for English text, offering efficient stemming and stop-word removal. Given the short abstracts in Cranfield, stemming and the removal of frequent stop words (e.g., "the," "is," "and") were crucial to improve precision in document retrieval.

```
EnglishAnalyzer analyzer = new EnglishAnalyzer();
```

Indexing the Documents

The dataset was parsed and indexed using Lucene. Each document from `cran.all.1400` was treated as a Lucene document, with the text content stored in the "content" field. Only the `.W` section of each Cranfield entry was indexed, as it contained the main abstract.

```
Document doc = new Document();
doc.add(new StringField("filename", line.substring(3).trim(), Field.Store.YES));
doc.add(new TextField("content", content.toString(), Field.Store.YES));
```

Similarity Models

Three different similarity models were evaluated:

1. **BM25Similarity** (best parameters: `k1=3.3`, `b=0.8`).
2. **ClassicSimilarity** (traditional TF-IDF).

3. **LMDirichletSimilarity** (best `mu=1500`).

For each model, the following steps were carried out:

- The Java file `QueryIndex.java` was modified to set the appropriate similarity model.
- The query results were saved to files (`query_results_bm25_3.3_0.8.txt`, `query_results_classic.txt`, etc.).
- TREC Eval was used to assess the results.

Command Line Usage

The following commands were used to compile and run the project:

1. **Compile the Java project**:

```
mvn clean package
```

2. **Run the create indexer**:

```
java -jar target/create-index.jar
```

3. **Run the query indexer**:

```
java -jar target/query-index.jar
```

4. **Evaluate results using TREC Eval**:

```
./trec_eval-9.0.7/trec_eval ./data/cranqrel ./results/query_results_bm25_3.3_0.8.txt
```

Parameter Tuning for BM25

A **Python script** (`find_best_bm25_args.py`) was written to automate the process of testing different `k1` and `b` values in BM25Similarity. The script iterates through a series of `k1` and `b` values, compiles the project, runs the query indexer, and evaluates the MAP score using TREC Eval. The best parameters found were `k1=3.3` and `b=0.8`.

```
k1_values = np.arange(0.5, 3.5, 0.1)
b_values = np.arange(0.0, 1.1, 0.05)
```

The script writes the results to a file and tracks the MAP score for each combination of `k1` and `b`.

Results and Evaluation

The results for each similarity model were evaluated using TREC Eval. Here are the MAP scores for each model:

Similarity Model	MAP Score
BM25Similarity (`k1=3.3`, `b=0.8`)	0.4070
ClassicSimilarity	0.3877
LMDirichletSimilarity (`mu=1500`)	0.3100
LMDirichletSimilarity (`mu=2000`)	0.3030

BM25Similarity performed the best with a MAP score of **0.4070**, closely followed by ClassicSimilarity and LMDirichletSimilarity.For BM25, the fine-tuned parameters (`k1=3.3` and `b=0.8`) proved to lead to the highest accuracy.

Conclusion

By using Apache Lucene and TREC Eval tools, we successfully tested several similarity models and optimized their parameters for retrieving documents from the Cranfield Collection. The parameter-tuned **BM25Similarity** model obtained the highest MAP score. This approach demonstrates the importance of model selection and parameter tuning in achieving high retrieval accuracy.

References

- Apache Lucene Documentation
- Cranfield Collection Dataset
- TREC Eval Documentation