**COMP4321 - Search Engine for Web data**

**Final Report**

Spring 2019 - Group 25

AU, Ho Leong ([hlauac@connect.ust.hk](mailto:hlauac@connect.ust.hk)) 20260060

KIM, Hyun Gyu ([hgkim@connect.ust.hk](mailto:hgkim@connect.ust.hk)) 20375138

CHEUNG, Ka Ho ([khcheungap@connect.ust.hk](mailto:khcheungap@connect.ust.hk)) 20465294

## Overall design

### CRWALER and indexer

The crawler and the indexer (/main/SpiderMain) fetch the pages in [www.cse.ust.hk](http://www.cse.ust.hk) recursively. 8 RocksDBs are created after running the program. For more information, please refer to file structure.

### search engine

After getting the indexed pages from database, Term weighting and document similarity is calculated.

### Web Interface

There is a webpage written in html which serves as a user interface. It allows user to input their queries.

We use Spring Boot and WebJars. Sprint Boot provides a good platform for Java/Kotlin developers to develop a stand-alone and production-grade spring application that you can just run. Many of the steps found on the [Spring Guides](https://spring.io/guides) for creating a RESTful service can be followed verbatim for Kotlin.

WebJars are client-side web libraries packaged into JAR files. It can explicitly and easily manage the client-side dependencies in JVM-based web applications

## File structure

The database includes 8 different indexes in different RocksDB.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| DB | Key | datatype | Content | datatype |
| URL\_DB\_NAME | Web url | String | Web id | Int |
| URL\_INFO\_DB\_NAME | Web id | Int | Triple(Title, Date-Modified,  Size) | (String, Long, Int) |
| URL\_CHILD\_DB\_NAME | Web id | Int | Child Web id | List(Int) |
| WORD\_DB\_NAME | Word | String | Word id | Int |
| SPIDER\_DB\_NAME | Word id | Int | List(Web id, Word Location) | List(Int, Int) |
| URL\_WORDS\_DB\_NAME | Web id | Int | List(WordID) | List(Int) |
| PAGE\_RANK\_DB\_NAME |  |  |  |  |
| URL\_PARENT\_DB\_NAME) |  |  |  |  |

## Algorithms used

Term weighting and Document similarity are based on tf\*idf/max(tf) and Cosine Similarity respectively.

## Installation procedure

(it could be as simple as “Type make in the project directory”)

## Fatures beyond the required specification

## Testing

screenshots

## Conclusion

### Strengths and weaknesses

### Improvement

### Interesting features to add