OTTO VON GUERICKE UNIVERSITÄT MAGDEBURG

Information Retrieval

Use case: Search Personalization

Topics can be marked as interesting by the user and causes a re-ranking of the results if the user is searching for the same query or if he/she performs a new search.

Objective :

Design a web search engine with search index which shows a representative graphical image (i.e a thumbnail) from each topic that is listed in the results. Topics can be marked as interesting by the user and causes a re-ranking of the results if the user is searching for the same query or if he/she performs a new search.

Input	Output
• Corpus Directory (Containing folder and sub-folders) • User's Query	• 10 most relevant topics • Rank of returned object • Image Path • Relevance Score • Title • File path • Sorting in desc order

**Please note that we have considered documents to be in the English language.

Graphical User Interface:

WEB DEVELOPMENT FRAMEWORK/TOOLS USED:
- Angular 8.3
- Node JS 14.15
- IDE Used : Visual Studio Code
- Request Validation tool : Postman
- SearchInterface is the extract file that contains angular project

PACKAGES USED:

1) @angular-devkit/architect
2) @angular-devkit/core
3) @angular-devkit/schematics
4) @schematics/angular
5) @schematics/update
6) @angular-devkit/build-angular
7) @angular/cdk
8) @angular/cli
9) @angular/material
10) typescript (4.1.3)

IMPORTANT INFORMATION ON GUI:

- The user interface is designed to mainly perform two main functions ie Searching and Displaying Results.
- The user will be initially asked to provide the name on the welcome and then the user can proceed with the query search.
- The user will be redirected to the search engine page after entering the name where the user will provide the query and hit the search.
- Top 10 results as per the query will be displayed on the results page. These top 10 result set will be displayed by doing some backend calculation based on the relevance score, which will be discussed in the later sections.
- A thumbnail will also be displayed with each result to provide a better visualisation of every result for the topic.
- Based on user's interest, he/she can personalise the result by providing the feedback in terms of star rating.
- We have used Angular ngb rating to implement the mechanism of Star Ratings.
- User is allowed to rate 1-5 star or can also opt out to provide any feedback.
- These rating will be personalised and the next time user makes the same topic search, he/she will get a re-ranked result based on the rating provided (user's preferred doc based on higher rating will be given the higher priority in the result set)
- The above functionality is fulfilled a Re-rank mechanism - which allows user to click and send the ratings to backend and re-calculate the ranking to be retrieved to the frontend.
- We have used RestAPI to make a connection between the front end and backend thus connecting Angular and Spring Boot application.
- In total there are 3 URLs which are used to establish the front-end to back-end connection, which are listed as below
- We have used two way dynamic binding which binds the changes in the UI in the variables which flows to the backend.
  - GUI - Page 1: Welcome Page
  - GUI - Page 2: Search Page
  - GUI - Page 3: Results Page

- In each component there is a .html and .ts file which will give the UI and the actions in the UI.

Backend Implementation:

- **Score Calculation** : Initial search method will expect a query string which will be provided by the user, on the basis of the user query we will be performing a search on the created indexes of the search file.
- Lucene search method will be called which will provide the total number of hits based on the query.
- We will be calculating highest and lowest score from the total hits which intern will be used later for re-ranking.
- Below are the integral features which are used in our application for returning the results as per the search query.
    - 10 most relevant topics
    - Rank of returned object
    - Image Path
    - Total number of relevant docs
    - Relevance Score
    - Title
    - File path
    - Re-calculation of relevance score
    - Rating given by user feedback
    - Sorting in desc order
    - Max of 10 and min of null doc

- **SCORE CALCULATION AFTER USER FEEDBACK:**

```
IF (RATING > 0 && RATING < 3) {
        RETRIEVALRESULTS[I][3] =
FLOAT.TOSTRING(FLOAT.PARSEFLOAT(RETRIEVALRESULTS[I][3]) - LOWESTSCORE - RATING);
    } ELSE IF (RATING > 3) {
        RETRIEVALRESULTS[I][3] =
FLOAT.TOSTRING(HIGHESTSCORE + RATING);
    }
```

EXAMPLE:

- Suppose a user enters a query of "Germany". Initially lucene top docs will return the total number of relevant document hits.
- For Example: 12 documents were returned with a rating of 1 to 12.
- We will just show the top 10 descending documents on the basis of relevance score.
- Now the user will give his feedback form 1 to 5 based on his preference for all the 10 documents.
- For the documents with a rating of 1 and 2, we will penalise those documents and decrease their score.
- For the documents with a rating of 4 and 5, we will increase the final score of those documents.
- So if the user gives 1 rating for the first 2 documents then after he submits his feedback, the next result set of 10 documents will contain the 11th and 12th documents and so the 1st and 2nd document will not be shown since their score were decreased based on users feedback.

- **SaveFeedbackRating**: After this a folder will be created with the name “Feedback” in the root folder of the project directory which will be .json file and will contain Titles and User ratings for each query. Thus if the user will fire the same query then the re-ranked list as per the ratings will be updated if the title will be present in the file and for new titles, new entries will be made.
- The displayed results will be calculated on the basis of the rating feedback file which will be added to the relevance score weightage.

- **Reset Functionality** : When the user will decide to logout from the system, then his preferences will be cleared out for all the queries : ie the feedback files will be cleared.

How to run the application:

- **Creating Dataset:**
  Please use the attached python script which will create the dataset. We have made some changes in the already provided script to function better.

Example : Please pass the arguments as the below mentioned syntax

ExtractTopics(r"C\Users\user\IRUpdated\laiqueqq",r"C\Users\user\IRUpdated\Wikipedia_topics\Wikipedia_topics",100)

- **Pre-requisities for GUI:**
  - Download visual studio code - https://code.visualstudio.com/download
  - Download Node Js from the link- https://nodejs.org/en/#download
    NPM also will also be installed with this.
  - Download Angular Cli - use this command to download angular cli in cmd prompt
    ```
    npm install -g @angular/cli
    ng --version (To check the version if installed properly)
    ```
  - Download type script package using the below command
    ```
    npm install –g typescript
    tsc -v (will give the version of installed tsc)
    ```
  - Go visual studio code and click File—> Add folder to Workspace —> select AngularUI from our zip—> go to terminal—> new terminal—> run the code
  - Use the below commands to install angular packages if required.
    ```
    ng add @angular/material to install angular material
    npm install ng-mat-search-bar --save
    ng add @ng-bootstrap/ng-bootstrap
    npm i ng4-loading-spinner --save
    npm install ngx-spinner --save —force
    ```
  - Use this command to run the angular project in angular CLI
    ```
    ng serve -o —port 4200
    ```
  - Add node module folder from below google drive link to AngularUI before running the UI, this is done since the file was not manageable.
    https://drive.google.com/drive/folders/1X-NdIIE3KCYJd9KFJqeeXDsvKaP-YBU0

-**Running the angular App:**
  - http://localhost:4200

- The jar file is expecting a dataset directly ie Extract file directory.
- Run the jar file using command
  - java -jar IRP02.jar "Extract file Path"

Points to remember:

- Please note that the image (thumbnail) can be accessed by clicking the Image link in the result set.
- We are displaying top 10 documents for every query search.
- The logic is designed to process English alphabets and if there are some non-english

alphabets in the URL/search than the results will be altered and no thumbnail image will be displayed for the same.

- If you are facing any issue(any security issue or ng not recognised) while running program in visual studio code(after going in the SearchInterface folder) then please remove ng file in powershellscript which will be at this kind of path C:\Users\malla\AppData\Roaming\np
- The dump which we used for our testing is placed on Google drive since it is very huge in size.
  https://drive.google.com/drive/folders/1uH3cbyJ5ONzOznGRwwDp5xKk0hMerRec

Code Structure:-

Files Used-

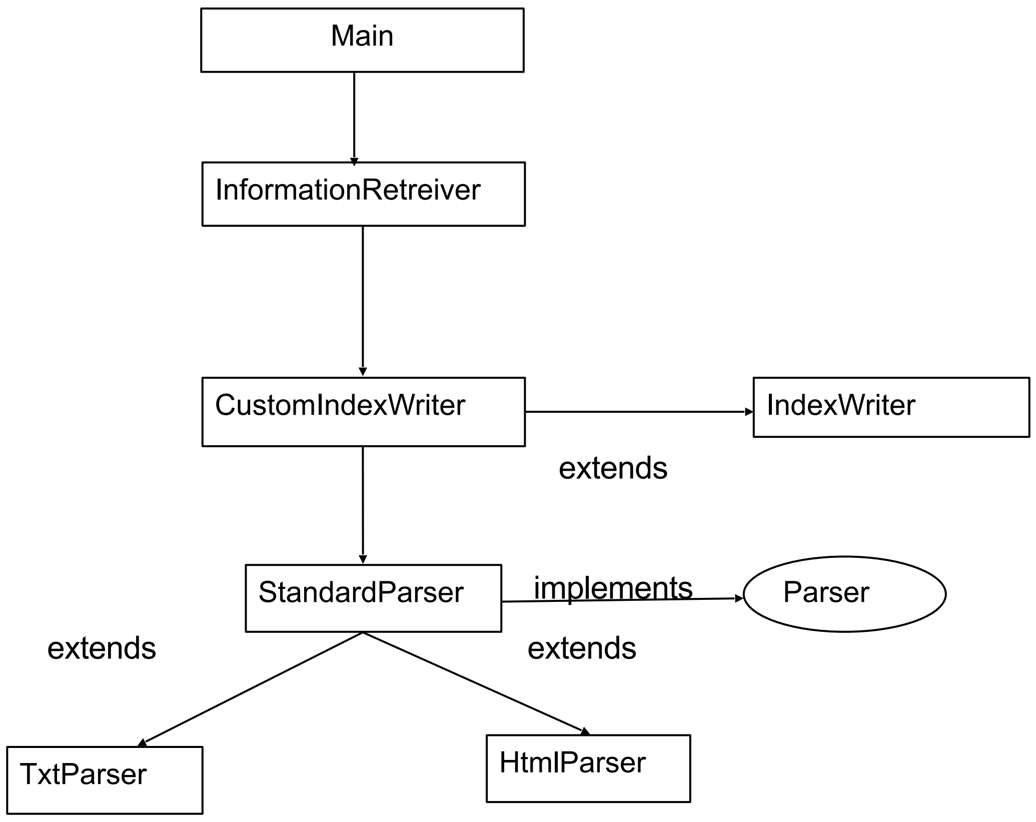
- Parser.java- To parse filename and field type.
- TxtParser.java- To read and store the text fields from the files and store them. It inherits the StandardParser class to access all its features.
- StandardParser.java- To obtain filename, file path along with the last modified date in a sorted manner for both plain text and HTML files.
- CustomIndexWriter.java-To select an available search index and creating a new index. Here the text files and HTML files are parsed as objects.CustomIndexWriter class inherits the IndexWriter class to search and create a new index.
- InformationRetreiver.java-To parse the document and creating index, stemming of the document using an analyzer, fetching or retrieving the top most relevant document along with their last modified date and the score of relevance. The class InformationRetriever sums up all the functionalities here together.
- Other:
- DocumentIndexWriter.java
- FieldNames.java
- Helpers.java
- HtmlParser.java
- QueryData.java
- RerankData.java
- SimpleCORSFilter.java

Important Classes Used-

- CustomIndexWriter
- FieldNames
- Helpers
- Parser
- StandardParser
- TxtParser

- InformationRetriever

Class Diagram:-



Important Methods/Constructor Used:-

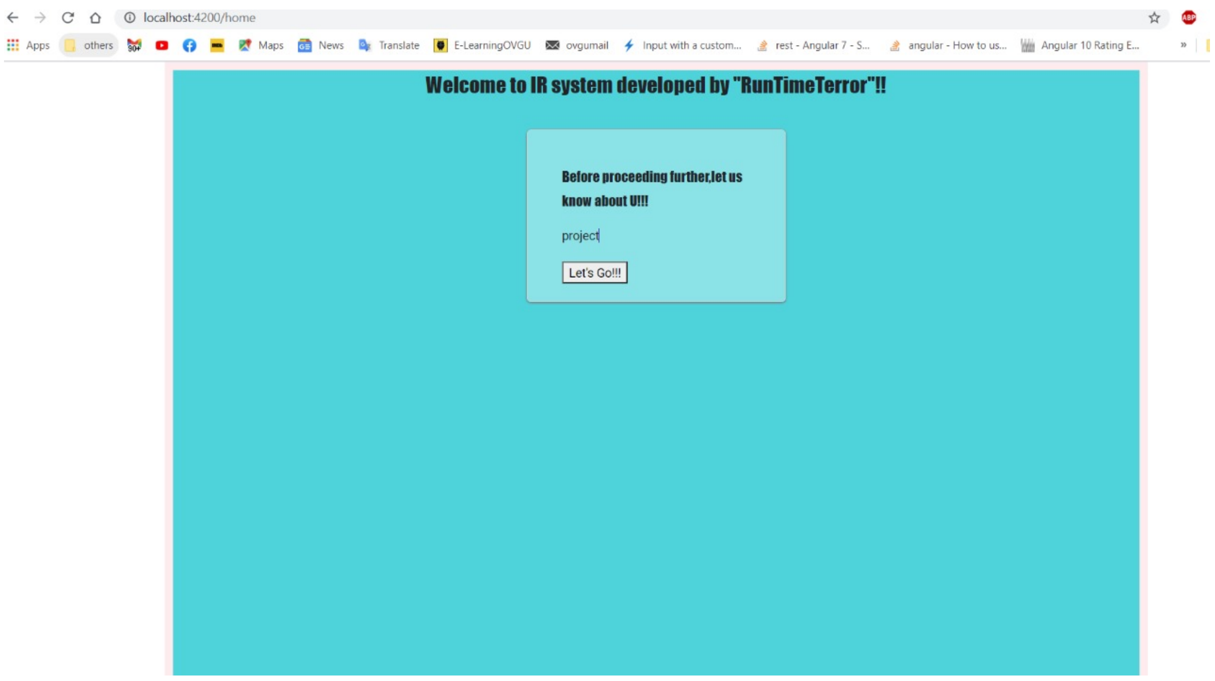
- **ParsingAndIndexing()**- Within the InformationRetreival class in this method, the file parsing and index creation is done.The index is stored in a directory for future retrieval purpose.
- **RelevanceRetrieval()**- Here the stemming using the analyzer is done using the MultiFieldQueryParser. Also the ranked list of a most relevant article given a search query is also determined. The output ,i.e, the retrievalResults contains the most relevant document along with its rank, path, last modification time and the hits obtained.
- **CreateIndexFromFilesDirectory()-**Within CustomIndexWriter class using CreateIndexFromFilesDirectory objects of HtmlParser and TxtParser is created and the files are stored in the respective directory in order to obtain the index.
- **ResetSystem()**
- **SaveFeedbackRating()**

Libraries Used:

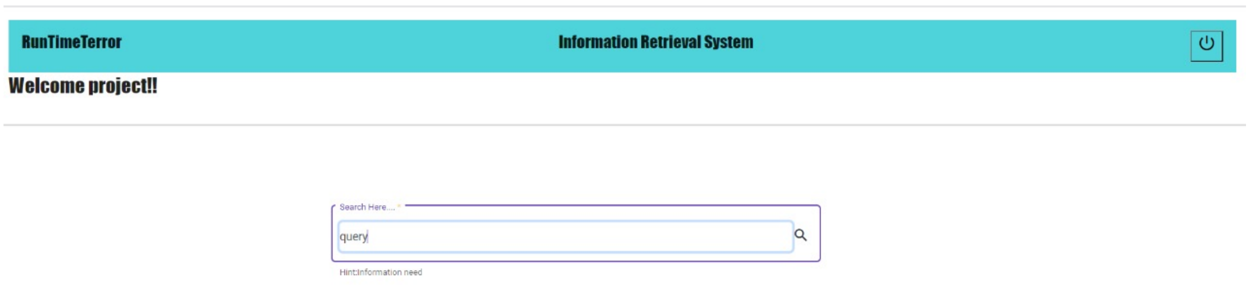
- commons-io-2.6.jar
- jsoup-1.12.1.jar
- lucene-analyzers-common-8.3.0.jar
- lucene-core-8.3.0.jar
- lucene-queryparser-8.3.0.jar
- import flexjson.JSONDeserializer;
- import flexjson.JSONSerializer;

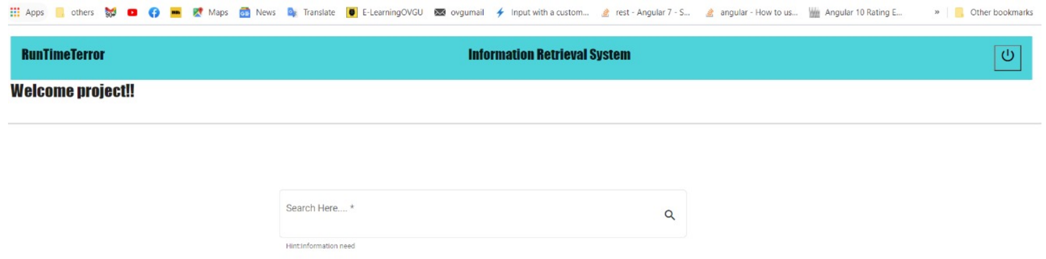
Screenshots:

Welcome Page


localhost:4200/home
Welcome to IR system developed by "RunTimeTerror"!!
Before proceeding further,let us know about U!!!
project
Let's Go!!!

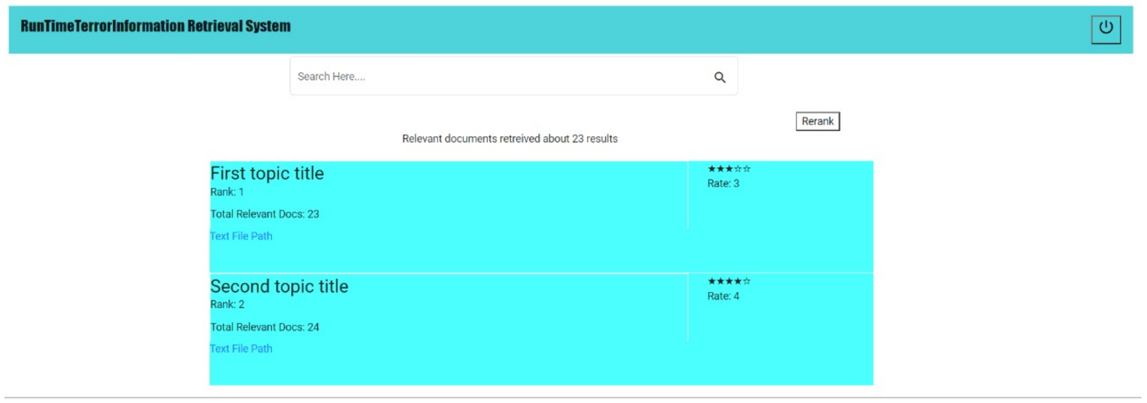
Search Page:





Welcome project!!

Results Page:



Relevant documents retreived about 23 results

First topic title
Rank: 1
Total Relevant Docs: 23
★★★☆☆
Rate: 3

Second topic title
Rank: 2
Total Relevant Docs: 24
★★★★☆
Rate: 4

References:

https://lucene.apache.org/core/4_1_0/analyzers-common/index.html

https://lucene.apache.org/core/2_9_4/queryparsersyntax.html

https://lucene.apache.org/core/2_9_4/api/all/index.html

https://code.visualstudio.com/docs/nodejs/angular-tutorial

https://stackoverflow.com/questions/59909589/angular-2-accessing-reactive-forms-independently-with-validation-in-ngfor-loop

https://stackoverflow.com/questions/57881532/angular-clear-array

https://www.c-sharpcorner.com/article/how-to-setup-angular-development-environment-in-visual-studio-code/

https://angular.io/tutorial/toh-pt0

https://cli.angular.io/

https://stackoverflow.com/questions/38593515/warning-sanitizing-unsafe-style-value-url