COMP8547 Advanced Computing Concepts - Project Report

Web Search Engine

Prepared By: Group 12

Aayushee Dave (110023928)	Manan Parmar (110022360)	Rahul Pandya (110024678)	Richa Gupta (110013520)

Aim

The aim of this project is to develop a Web Search Engine for searching and indexing html documents implementing various concepts of computing using Java programming.

Concepts Implemented

We have implemented the following concepts for the development of the Search Engine:

Web Crawler

It is responsible for fetching relevant pages from the internet as per the user search query. The web crawler or spider will start from a reference webpage and then follow the hyperlinks to other pages, and so on.

BFS

In order to crawl through all the pages, we need to decide the approach to select the next web page to be visited. Hence, we have used the Breadth First Search algorithm to ensure that no page is visited more than once and hence obtain optimal results in least possible time.

JSoup

We have used JSoup for parsing HTML documents. JSoup will extract all the data like Page title, link, and description from the crawled web pages required displaying the results for the search query. JSoup is responsible for parsing HTML to DOM tree.

Tries

In this project we have used Tries (tree-based data structures) for text processing; representing a set of strings that is extracted from the HTML page. Trie is also used for word matching and finding the words with given prefix.

Autocomplete

While searching, the autocomplete feature predicts the following sequence of words. This feature is implemented using Tries.

Java Regex

We have used Java Regular Expressions while extracting text from HTML; to remove any special characters and insert strings into Tries and fetch URLs from HTML.

Sorting

Here we have used Quick Select, Quick Sort along with Insertion Sort for ranking the pages. The algorithm takes list of Tries as an input from the Web Crawler and sorts them based on the total occurrences of the key word searched by the user.

String Tokenizer

We have used String Tokenizer to tokenize the extracted text obtained from the HTML pages parsed by the Web Crawler.

How does it work?

The application first of all, takes user input and predicts the next sequence of letters to complete the following word sequence using autocomplete feature that is enabled by using Tries that is created from the 100 HTML files that were given in assignment.

Afterwards, the word entered in text box is passed to web crawler along with a starting URL in order to scan all the web pages using breadth first search algorithm. The web crawler crawls in the entire page and collects all the links within the web page. The web crawler also collects the URLs, their title and description of the web page and sends these results to process further.

Each web page is then forwarded to JSoup parser to convert the HTML data into text format. This converted text data is transformed to Tries data structure so that it becomes easy to find the occurrences of words in an efficient way.

All this information is then contained in object i.e.: URL, Title, Description and Tries. Then we are finding occurrence of keyword in the tries and based on the number of occurrences, we've applied the Quick Select, Quick Sort and Insertion Sorting algorithm to sort the results. This result set is then forwarded to UI.

Below is the pictorial representation of working the application:

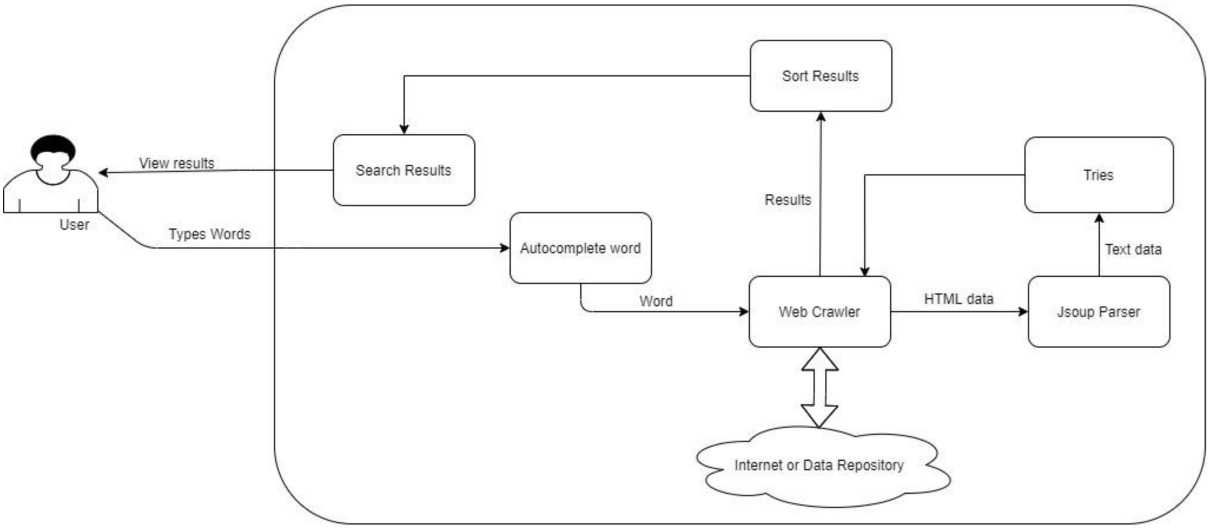

Fig: Block diagram of our Web Search Engine

References

- Class Slides
- Java Documentation: https://docs.oracle.com/javase/8/ docs/api/overview-summary.html
- Baeldung Documentation: https://www.baeldung.com/ spring-boot
- Algorithms: https://algs4.cs.princeton.edu/home/
- Web Crawler: http://www.netinstructions.com/how-to-make-a-simple-web-crawler-in-java/