

ASURITE : psuryawa

ASU ID : 1213231238

Collection of Data: Used BeautifulSoup in python to crawl data and recursively collect all the links and content from the main page from both java wikibooks and oracle javaTM tutorials. data.json file has all data collected.

Content indexing: I have used elastic search to index data which internally uses Apache Lucene.

Data is indexed based on content and titles collected using bulk method in ES. While performing search, the query data has been processed to remove stopwords and stemming using python nltk lib.

Those tokens are later used for string matching. Query data along with tokens is in questions.js. Mapping used to store index is present in esLib.js which has all CRUD functions used to bulk add data and perform search.

Web App: I have created web app to get requests from UI and perform necessary actions on elastic search. When page loads jquery sends request to add and index data, once that is done it will call searchAllDocs method to get all recommendations and dynamically load them on UI.

Originality:

I have used node js based express app to handle AJAX requests from UI. It uses 'elasticsearch' module as client to actual running instance. I have also used bootstrap and materialize for UI.

I have indexed data in Elasticsearch using similarity algorithm BM25, I have also applied the english analyzer in index mapping while building of indices.

In information retrieval, Okapi BM25 (BM stands for Best Matching) is a ranking function used by search engines to rank matching documents according to their relevance to a given search query.

Steps to run:

- Make sure Elastic search is running with default port 9200.
- Do 'npm install' to download dependencies and set project.
- start local node server by command 'node index.js' which will start server on localhost:3000
- Go to localhost:3000, when page is loaded after few seconds recommendations will be shown per query. Click on query button to show and hide recommendations.