Documentation

Overview

We used the NOSQL database mongodb to represent the database of the search engine for the crawler,indexer,and other collection that we will mention

Crawler

We use multi threads in crawlers for the sake of speed . We represent links as unvisited,CurrentVisiting,and Visited.

- First of all we insert the seeds that we put in the unvisited collection
- Second we put all links in the link that is in the top of the link in the unvisited collection after finishing the first layer of seeds we then dig in the links in unvisited using breadth first
- When the link is currently visiting the link goes to the currvisiting collection and removed from the unvisited collection when interrupting the currvisiting goes to the unvisited to be visited again
- The crawler checks on if the link is allowed to be crawled or not through the robot.txt of the link and check if there is disallowed in the file
- The crawler checks if the link is url normalized
- Also it checks if the link has the same page and if so it does not insert it again
- In the Crawler we insert the pageRank to save the link that the links are pointing to

Indexer

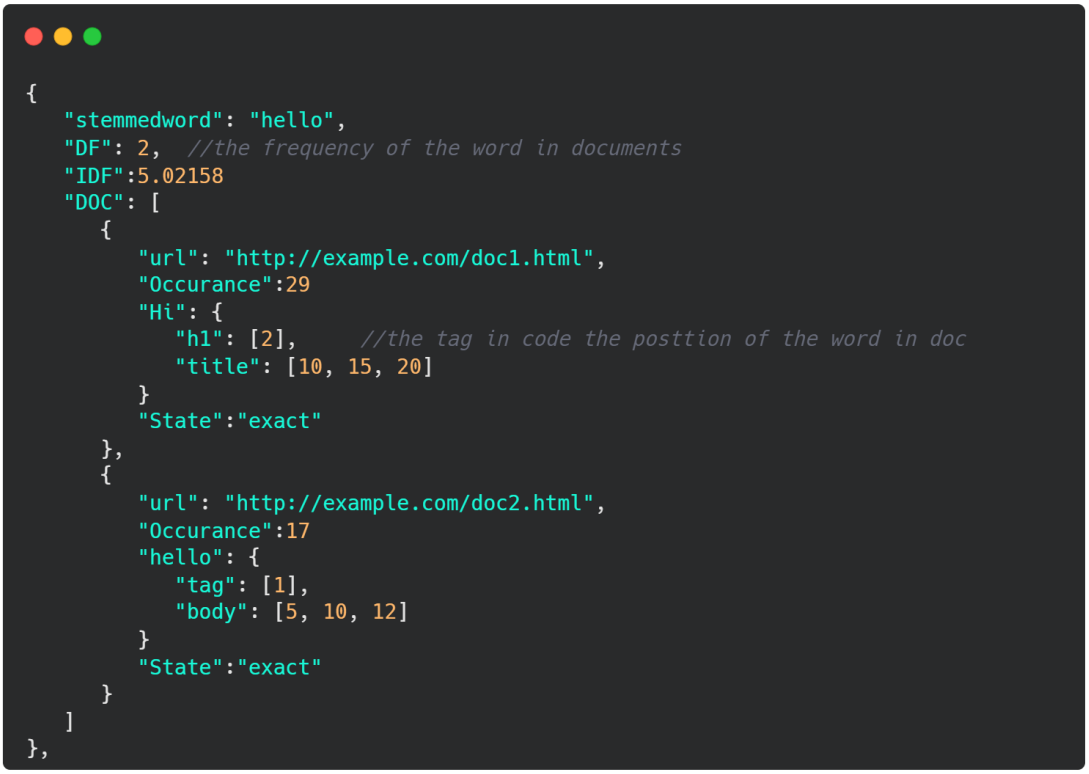
The indexer is divided into two part

Index

- We get the source code of the page which is in the database which was stored in the crawler we used the library jsouep
- We then divide the source code into elements of HTML tags and then enter them in the database in the TagContent collection for the sake of phrase searching
- After that we have to put the word to the database

AddWordtoDB

- We then add the word in db like in the following shape



```
{
   "stemmedword": "hello",
   "DF": 2,  //the frequency of the word in documents
   "IDF":5.02158
   "DOC": [
      {
         "url": "http://example.com/doc1.html",
         "Occurance":29
         "Hi": {
            "h1": [2],     //the tag in code the posttion of the word in doc
            "title": [10, 15, 20]
         }
         "State":"exact"
      },
      {
         "url": "http://example.com/doc2.html",
         "Occurance":17
         "hello": {
            "tag": [1],
            "body": [5, 10, 12]
         }
         "State":"exact"
      }
   ]
},
```

Query Processor

In the query processor we get the documents that is applied to one of the following

- Match the exact words
- Match the stemmed word
- Has the same meaning using wordNet dictionary

The return of the Query words is list of documents that is matched one of the previous conditions

Phrase Searching

In the phrase searching,we can get if it is a phrase in the link if

- All of the words are in the same links through get the list of documents when the state of the word is exact
- When the words are in the same type of the content through checking the content of the element of HTML through the collection TagsCollection that we have mentioned before in the indexer as we save the content of the url by the tag type

- We then check using regex if every two words in the query are at most 5 words apart of each other and that is within the same content of the tag
- The operations of AND ,OR ,and NOT are possible in phrase searching through AND_Expression,OR_Experssion ,and NOT_Experssion

Web interface

We use Servlets and tomcat to link between the front end and back end and some other tools

- HTML
- CSS
- Javascript

Appendix - Features

- Voice Recognition
- Operation of and ,or,and not
- Phrase Searching using Single quotes as a phrase
- Searching include to get the meanings of the words of the Query not just the The words itself