Group: 8

Li Shing Fung

Daniel Schlaug

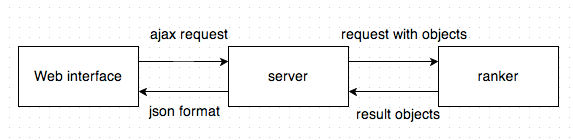
Lui Tsz Ching

Gopher

Comp4321 project documentation

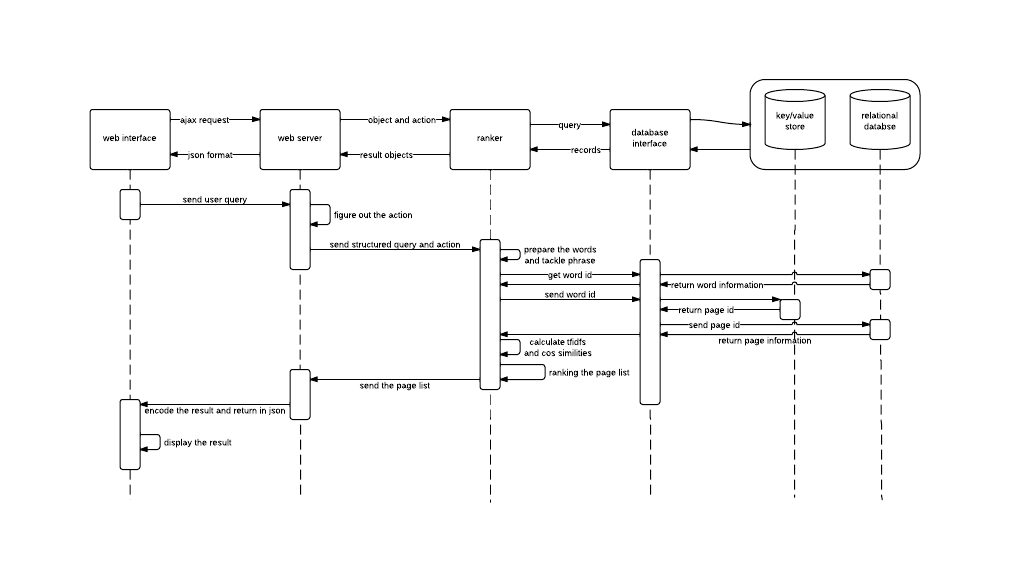
Overall design of the system

The system can be divided into three parts: web interface, server and ranker.

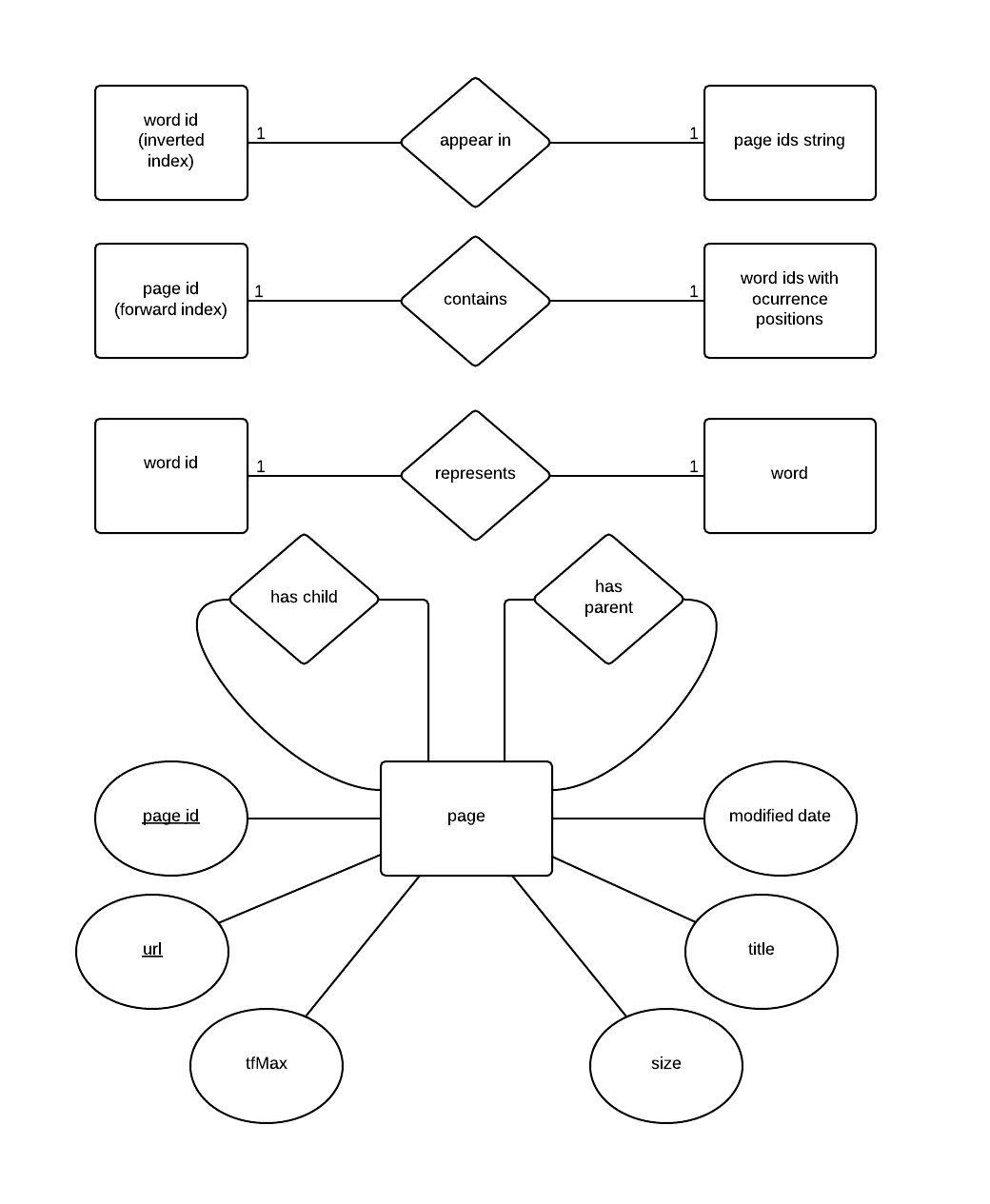
The web interface receive user’s query or index retrieval enquiry.

The server will act as the port to receive the queries and call ranker’s function to search on the database and retrieve lists of relevant websites. It will finally return results in json format to the user interface for display.

The ranker get query from the server, process them and calculate corresponding scores to find out the most related websites. The list of websites would be packed and sent back to the server.



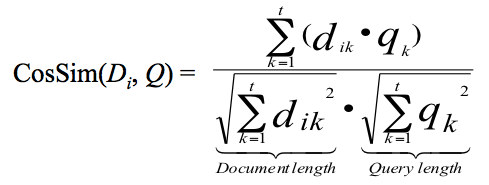
File structures used in the index database



Algorithm used

Title matching

Cosine Similarities



Tfidf

*wij =* (*tfij /maxl*{*tflj*}) *· idfj*

Installation procedure

crawler

search engine

browser the url: http://localhost:8081

Highlights of features

Autocompletion

Autocompletion is implemented in the search bar. It will suggest the words from the database index, which give a hints to the user what other website contains.

Similar Pages searching

The similar pages links would extract the top 5 keywords from the web page and use them to recontruct the query. Webpages shared high cosine similiarities with the query would be considered as similar pages and sent back to the user.

Query History

User’s input query would be stored in html5’s local storage so that user could take a look on them. This allows users to track back on their past queries and make use of them to recontruct queries.

Database Index exposure

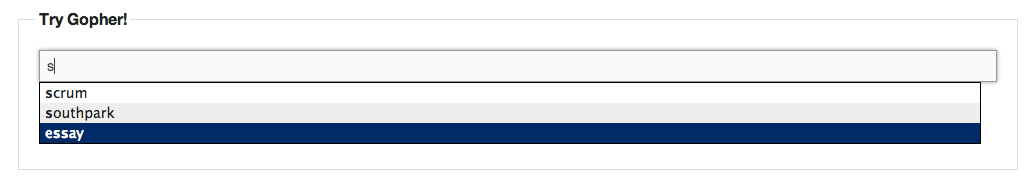
User could look at the existing indexes inside the database so that they could get more ideas about what words should they use and what do websites contains.

User Friendly Interface

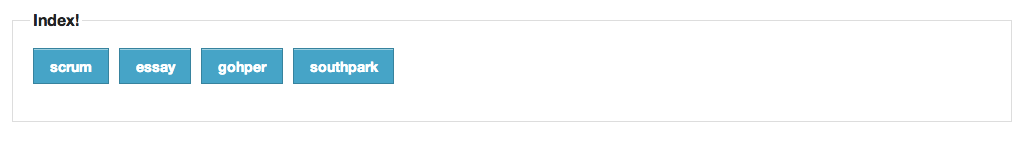
The interface is well designed with pominent buttons and style. User could use the website pretty well without any practise. Friendly features such as “Clear History” is provided in case the history become too messy.

Testing of the functions

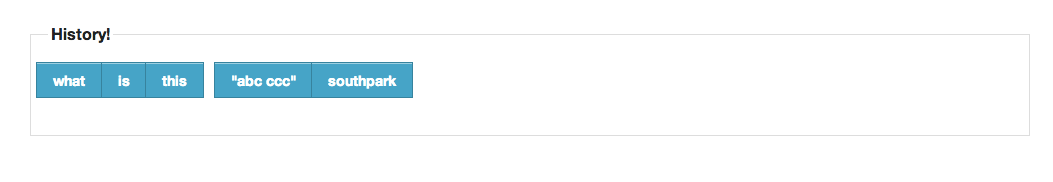
Autocompletion



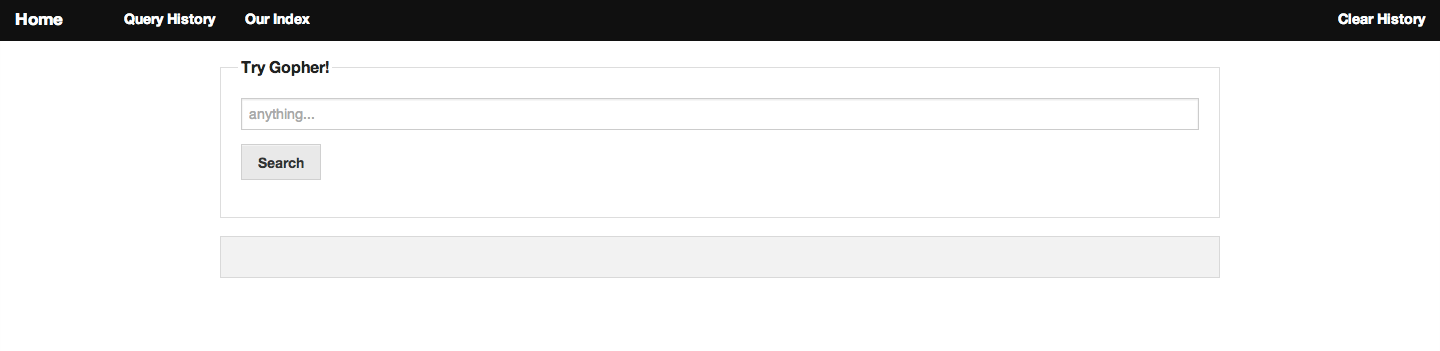
Database Index



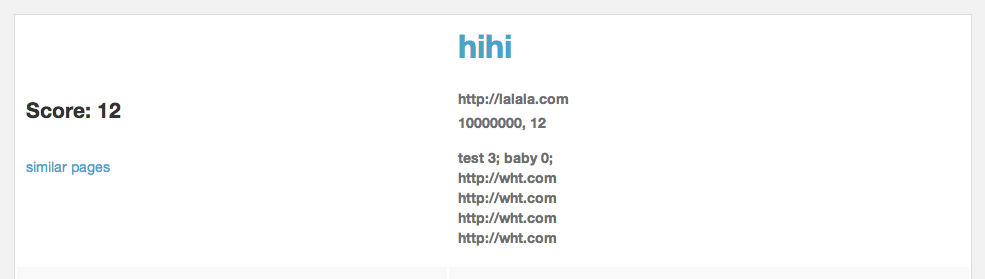
Query History



User Friendly Interface



Similar Pages searching



Conclusion

strengths

The system provides searching service mainly based on term frequency-inverse document frequency, cosine similarities and vector space model. With the implementation of query history, index requerying, user could try different combinations of keywords to get a better result.   
  
We provided a handy interface which is clear and allow user to

weakness

Our system have not scheduled regular crawling to update the database of the websites. It does not handle the dynamic web page generated by the server which is potentially useful.  
  
Image searching is not supported due to its complex procedures such as image resizing, image shearing, image merging, image thresholding, eliminating noise and image correlation calculation. These processing steps would cost a huge amount of waiting time which is not user friendly.

We have not placed any heavy weighting on meta-tag which would be useful for website to specify their features.  
  
We have not implement advanced searching based on user’s location, preferred domain nor personalization on search engine.