WayDa Search Engine

Project Proposal for Database and Information Systems Course

Project Guide : Prof. S Sudarshan

Submitted by :

Neelesh Verma(160050062)

Shreyash Meena(160050058)

Maitrey Gramopadhye(160050049)

Suseendran Bhaskaran(160050105)

Overview and Specification

We have a designed a Search Engine - WayDa

The Input will be a Query from the user side , the output will be the most relevant URLs along with a Title.

We are drawing results from a Database that contains approximately 10,000 of URLs and show the most relevant ones based on a method described later.

Features implemented :

● We are building a Web Based Application.

A. Our Application supports Multiple Users across different Sessions. Each user will have a different session and this will be wiped off completely when they log out.
B. We will then parse the Query to extract meaningful keywords from it and proceed towards searching in our Database.

C. Our database contains a single table “search_engine” that contains an id (auto increment), “title” of the web-pages, “heading” (h1 heading), and “imp_words”. Imp_words column contains some important words (we have set the limit to 5 now but it is configurable) from the remaining headings and meta-data. We determine the importance of a word by calculating its frequency count and sorting the words in descending order by their frequency.

D. We have divided the user-query into 2 parts.

   a. AND part :- First of all, the user query is parsed to remove the special characters and some stopping words (like “and”, “or”, “to”, etc). The parsed string is tokenized into words (splitting of sentence into words). Then we search for urls that contains all of these words. For ranking, first priority is given to title, then to heading, then to imp_words. Imp_words also contains frequency count of words as well, so for all these urls, we do ranking based on frequency count.

   b. OR Part :- Similar to the above part, except here we search for urls that contains either of the words from parsed string.

E. It is possible that a particular link contains hundreds of links but all these links belong to same organization. In order to have diversity in the links, we have set a limit to extract the number of links from a particular page. It is configurable.

F. We have done the **Generalized Inverted Index** on title, heading and imp_words attributes. The reasons we didn’t choose **GiST** index is because GIN indexing is 3 times faster then GiST. Although GIN takes more space but our database isn’t that much large to worry about the space.

Interface

1. ***“You never get a second chance to make a great first impression”.*** We firmly believe in this saying and strive to provide a beautiful User Interface.
2. The Login Window screen will then have the Search Engine Logo along with the Search Field and the Go! Button. The user will enter their query in the Search Field and click on the Go! Button. This screen will also showcase some flash based content.

3. This screen will then display the most relevant results according to the query entered by the user. It might happen that the number of results are less than the webpage’s threshold if there aren't many relevant URLs.
4. The above screen shall also have the Back Button, the Back button will take us to the Search Engine Page.

E-R Diagram

Search_Engine
ID
URL
Title
Heading
Important words

Table Design

Search_Engine
ID
URL
Titlet
Heading
Imp_words

Testing and Evaluation

1. The testing phase will focus on finding the most relevant URLs for the given query. The user shall determine whether the URLs they were looking for were given preference or not.
2. The user can also twist the query so that the resulting keywords by Natural Language Processing are the same.
3. Finally , the User can also give Corner Cases and Test certain queries that they feel would be challenging enough for our implementation.