Web Information Retrieval and Text Mining
Project 1: A small search engine

Chi-Hsuan, Huang
Student ID: F84004022

Dept. of Computer Science and Information Engineering
National Cheng Kung University
Tainan, Taiwan, ROC

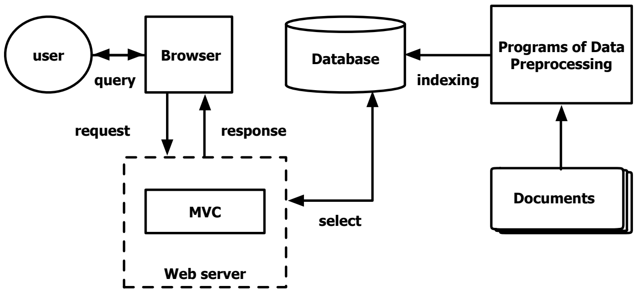


Figure 1: The overview of the system architecture

1. INTRODUCTION

Information retrieval(IR) is very important and closely related to our lives. For example, we use Google to search the information for solving the problem or learning something new. Moreover, looking for books in library is also a IR problem. The goal of IR is to obtain the information resources which is relevant to use information need from a collection of information resources.

A web search engine is a system that is designed to search for information on World Wide Web and it is the most important application in IR field. As mentioned above, the rise of Google Inc. is relied on its search service since 1998.

The goal of this project is to designed a small search engine that search for more than thirty thousand articles documents. The following I will illustrate the method how I implement a small search engine and demonstrate the results.

2. FRAMEWORK

In this section, I introduce the proposed system architecture and describe the flow of each stage.

2.1 System architecture

The overview of system architecture is shown in Figure 1. There are three major components are included in my system architecture, namely (1) Programs of Data Preprocessing (2) Database (3) Web programs.

The goal of programs of data preprocessing is to transform documents to tokens, do the text processing and then, insert processed data to database. This component do a lot of things, including text tokenizer, stoplist, stemming, and etc. Therefore, it play an important role in search engine.

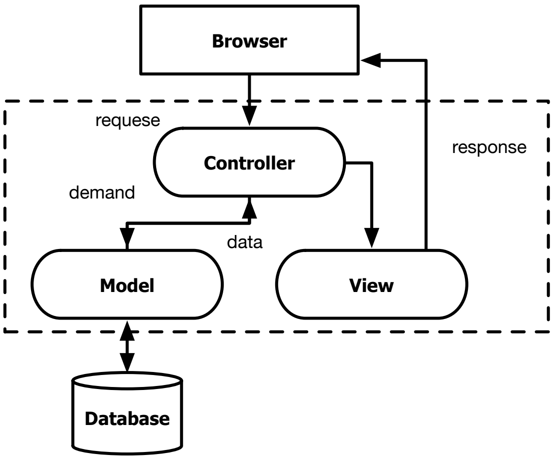


Figure 2: The overview of the MVC framwork

The database is an organized collection of data. Due to the large amounts of documents, we need a database to deal with the data storage and indexing for quickly retrieve.

In this project, I create a web site for user querying and the back-end is adopted to MVC framework. The overview of MVC framework is shown in 2.

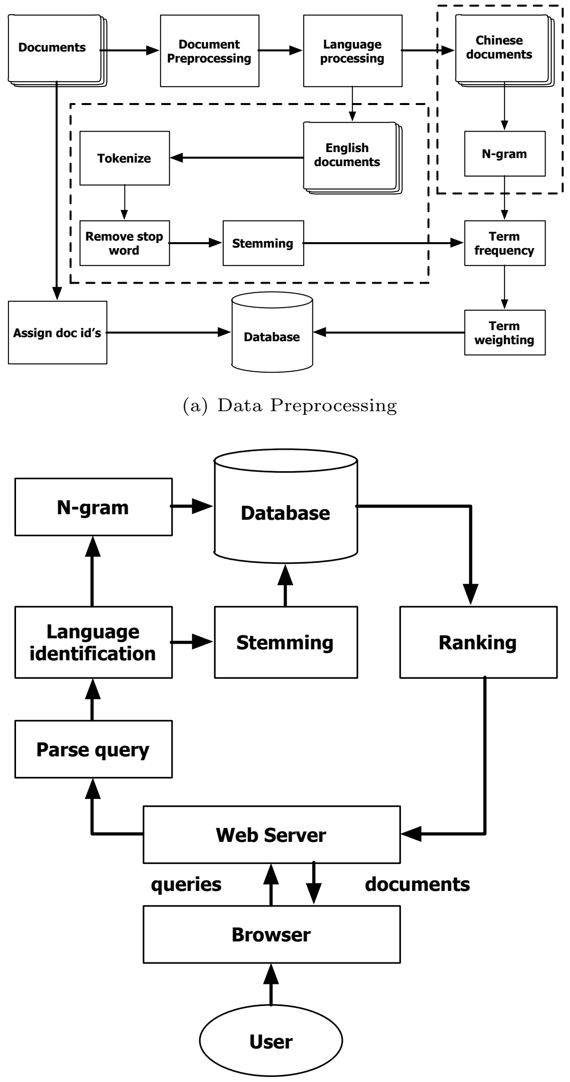
2.2 Flow

In the figure 2.2, it is demonstrated the flow of this project. There is two main stage of this project, *data preprocessing* and *user query.*

Figure 3(a) shows the *data preprocessing* stage. We can clearly see that it is the stage about the documents to database. The first thing I do is to assign each document with it's id and insert to database for showing the content of document after the user click document link.

Because user seldom queries English and Chinese terms together, it is clearly to see that I deal with Chinese and English separately. Besides, there are a lot of different properties between Chinese and English, for example, English has the space between words and Chinese does not. Therefore, after the stage of *Language processing*, the English text is processed by *tokenize*, *remove stop word* and *stemming.* However the Chinese text is only handle by the *n-gram.*

After the text processing of each subflow, I calculate the



(a) Data Preprocessing

(b) User query

Figure 3: The overview of the flow

term frequency and *term weighting* separately and then, insert to database.

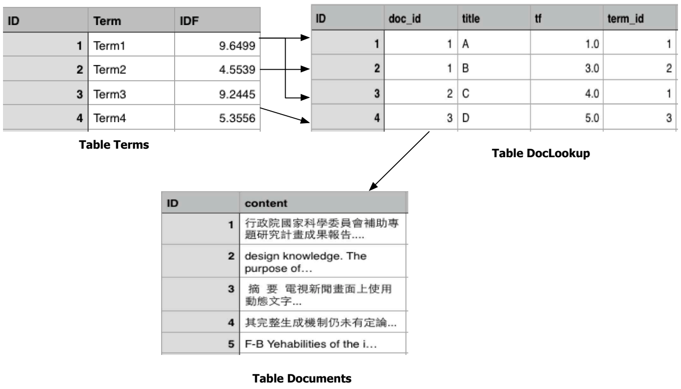
The stage of user query is shown in figure 3(b). Similarly, when the user queries, I do the different process by different language input and select the related documents from the database. By executing the ranking apporaches, the documents are responesed to browser.

3. TECHNIQUES

In this section, I introduce the techniques I used in this project, including *normalization*, *removing stopwords*, *stemming*, *n-gram* and *TF-IDF*. Moreover, I will show the deign of database table.

3.1 Techniques introduction

1.Normalization: The normalization is to modify text to make it consistent. Firstly, I normalize the documents by transform each document to only one line and convert words to the lowercase. The reason transforming to one line is because the newline is useless and may affect the following stages.



ID	Term	IDF
1	Term1	9.6499
2	Term2	4.5539
3	Term3	9.2445
4	Term4	5.3556

Table Terms

ID	doc_id	title	tf	term_id
1	1	A	1.0	1
2	1	B	3.0	2
3	2	C	4.0	1
4	3	D	5.0	3

Table DocLookup

ID	content
1	行政院國家科學委員會補助專題研究計畫成果報告....
2	design knowledge. The purpose of…
3	摘 要 電視新聞畫面上使用動態文字...
4	其完整生成機制仍未有定論...
5	F-B Yehabilities of the i…

Table Documents

Figure 4: Design of the database tables

2.Removing Stopwords: Stop words are the noise words, such as ”a” and ”the” which may meaningless in documents, thus, we remove then. [1]The stop word list, I used, was built by Gerard Salton and Chris Buckley for the experimental SMART information retrieval system at Cornell University.

3.Stemming: Stemming is a process to reducing the derived words to their base form. For example, if the word ends in ”ed”, remove the ”ed”. The algorithm I used is [2]*Porter Stemming Algorithm*. And if the input token is digital, this stage will be skipped.

4.N-gram: N-gram is a famous tokenization method in field of IR, which is a contiguous sequence of n items from a given sequence of text. In this project, I tokenize the Chinese text by N-gram because the Chinese tokenization is a very challenging problem in text processing.

5.TF-IDF: Term frequency inverse document frequency(TF-IDF) is a common method that can reflect how important a word is to a document. In this project I used it to rank the result of user querying The following are the formula of TF-IDF.

$tf_{i,j} = \frac{freq_{i,j}}{freq_{max,j}}$

$idf_i = \log \frac{N}{n_i}$

where $freqi, j$ = occurrence of k_i in d_j where N = total number of documents, n_i = number of documents that contains index term k_i.

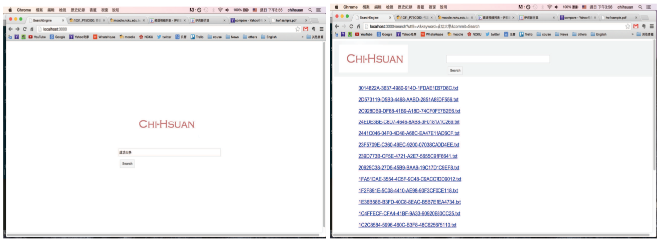
The weight of a word is obtain by the following formula. $W_{i,j} = tf_{i,j} * idf_i$

3.2 Database

The design of the database tables is illustrated in figure 4. It can be seen that I use three tables in the database for user querying. Take td-idf method for example, when the user query, the term's idf in ”Term table” are selected by input terms first. Then, we select the tf of top-k related documents by ”DocLookup table”. Therefore, we can rank them and return the documents. Finally, if the user click the link, we can return the document content by ”Document table”.

[1] http://www.lextek.com/manuals/onix/stopwords2.html

[2] http://tartarus.org/martin/PorterStemmer/



localhost:3000/search?utf8=✓&keyword=成功大學&commit=Search

3014822A-3637-4980-914D-1FDAE1D67D8C.txt
2D573119-D5B3-4468-AABD-2851A89DF556.txt
2C928DB9-DF88-41B9-A18D-74CF0FE7B2E6.txt
24EDE3BE-C8D7-4646-8AB8-3F0181A1C269.txt
2441C046-04F0-4D48-A68C-EA47E11AD6CF.txt
23F5709E-C360-49EC-9200-07038CADD4EE.txt
239D773B-CF5E-4721-A2E7-5655C91F6641.txt
20925C38-27D5-45B9-BAA9-19C17D1C9EF8.txt
1FA51DAE-3554-4C5F-9C48-C9ACC7DD9012.txt
1F2F891E-5C08-4410-AE98-90F3CF0CE118.txt
1E36B58B-B3FD-40C8-8EAC-B5B7E1EA4734.txt
1C4FFECF-CFA4-41BF-9A33-90920B80CC25.txt
1C2C8584-5996-460C-B3F8-48C625675110.txt

Figure 5: User interface

4. PPROJECT RESULTS

In this section, I will present the project results, evaluate performance and show the top-k precision vs. k graph.

4.1 Results Description

The data preprocessing stage is implemented in python and the database I used is postgresql. Some of text processing packages is provided by [3]NLTK. The web back end framework is constructed by ruby on rails. There are two ranking approaches, which are TF-IDF and with TF only. All the code can be found in my [4]github.

Figure 5 is the screen captures of the user interface. The result of project can be demo by created a small web site. And if the query terms cannot be found in database, we will redirect to the document-no-found page.

4.2 Performance evaluation

To evaluate the performance of the search engine in this project. I manually query 6 keywords that meet the requirements of this project. The response time of each query is shown in table 1. As can be seen, the response time is very short in our case even the input is more than two words and it can still be improved by the distributed Systems.

The accuracy, in this report, is defined as the correlation between the query keyword and documents. The result of each query and it's accuracy is tabulated in Table 2. With different query terms, the accuracy is various. In the first case, "coming soon...", most documents only match "coming" in database so the result is not good. However, the keyword, "google yahoo", matches all case. The reason of that is probably due to the fact that the documents are academic and if the term is more relevant to this academic documents, the more accuracy it will be.

I compare the two ranking approaches between TF-IDF and TF only to see what would be different. The result is different in case 2 and 6. The the ranking is different between the 5 to 20 documents. And it is interesting that keyword "yahoo" is more significant than google in our case.

4.3 Top-k precision vs. k graph

Figure 4.3 is the Top-k precision vs. k graph. As mentioned above, the reuslt of keyword "coming soon" is terrible and the keyword "google yahoo" is very good. In particular, the precision of digit queruy is unstable because the digit is not meaningful in most cases.

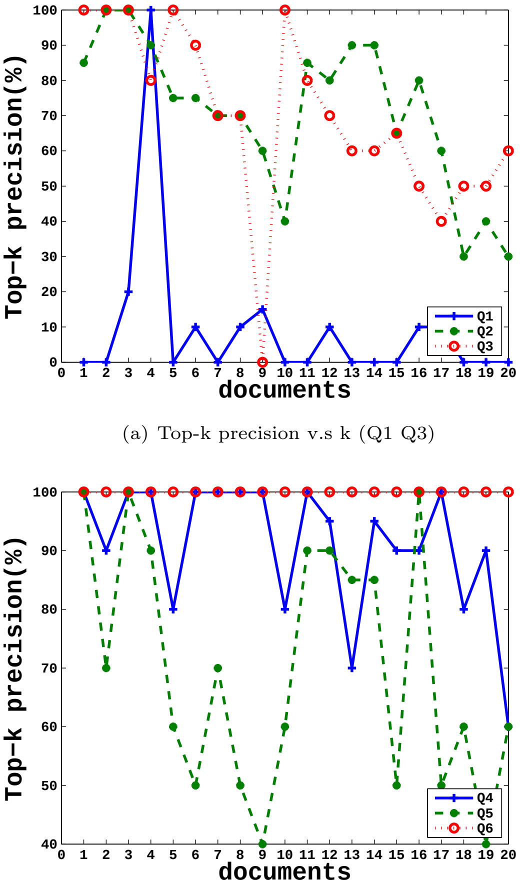
the following are the additional query terms after the project was announced. In figure 7, it is clear to see that

Table 1: Response time of each query

Case	Query term	Response time
1	coming soon...	48ms
2	101 102	55ms
3	2.4	53ms
4	100	52ms
5	40	54ms
6	google yahoo	5ms

Table 2: Accuracy of each query

Case	Query term	TF-IDF/Accuracy	TF only
1	coming soon...	10%	10%
2	101 102	75%	75%
3	2.4	85%	80%
4	100	95%	95%
5	40	85%	85%
6	google yahoo	100%	100%

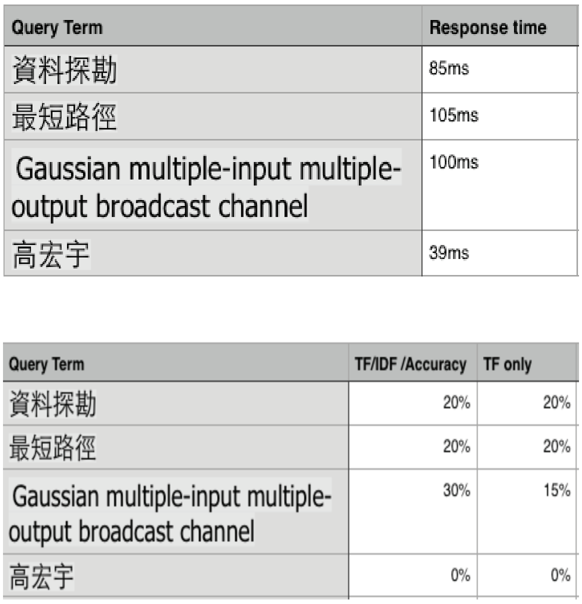


(a) Top-k precision v.s k (Q1 Q3)

(b) Top-k precision v.s k (Q4 Q6)

Figure 6: Top-k precision v.s k

[3] http://www.nltk.org/
[4] https://github.com/chihsuan



Query Term	Response time
資料探勘	85ms
最短路徑	105ms
Gaussian multiple-input multiple-output broadcast channel	100ms
高宏宇	39ms

Query Term	TF/IDF /Accuracy	TF only
資料探勘	20%	20%
最短路徑	20%	20%
Gaussian multiple-input multiple-output broadcast channel	30%	15%
高宏宇	0%	0%

Figure 7: Performance accuracy evaluation

the result Chinese keyword is very poor. Therefore it seem that there is still much room for improvement.

5. DISCUSSION & CONCLUSIONS

The search engine is a interesting application. In this project, I learn a lots about text processing techniques and also know how to to identify between Chinese and English characters. Besides, the design of database table is a very challenging problem. I think it can be improved. For example, we can divide the ”Term” table into two subtable by different language to speed up the selecting time of the database. And I found that the columns in table indexed or not will lead to a very different performance, which indexing column will reduce the a lot of query time.

In conclusion, even this project is to implement a small search engine, it let me know how the search engine operate and the framework of the search engine. Moreover, this project make me understand how powerful and amazing thing Google does and want to know more detail of search service.