A Experiment Report about a Web Information Retrieval System for 3rd NTCIR Web Task

Iwao NAGASHIRO
Department of Information and Network, Tokai University
2-3-23, Takanawa, Minato-ku, Tokyo 108-8619, Japan
nagasiro@keyaki.cc.u-tokai.ac.jp
Dafeng CAO
Beijing Center for Japanese Studies
2, Xi Shan Huan Bei Lu, Beijing, China
cdfeng@163.net

Abstract

We joined 3rd NTCIR web task from October 2001. For this task, we constructed a small web information retrieval system. By this system, we completed "dry run" and "formal run" retrieval topics of the task. In this report we will give a brief description about our basic method for web information retrieval, our web information retrieval system and some retrieval experiment results.

Keywords: *Information Retrieval, Search Engine, Site Rank.*

1 Introduction

Internet web sites have increased very rapidly and spread over all of world in resent year. It is possible for every one to access internet from his office or home. But because web sites are developed in great disorder, it becomes difficult to find necessary information precisely from year to year. However information retrieval technology gives some solution for this problem. We have constructed a web information retrieval system to get experiment results and verify our new method for retrieval precision. By the system we joined 3rd NTCIR web task from October 2001, and this paper is a brief report of our work, it includes basic idea for web information retrieval, web information retrieval system description and some retrieval experiment results.

2 Basic Method for Web Retrieval

Here we will show some basic method for our web retrieval system. The diagram of our system is given in Figure 1. All of the computers are x86-based with Windows operating system.

2.1 Procedure of our web search system

We used the test collection resources provided by NII (National Informatics Institute) as original web documents. For implementing a web retrieval system, we processed these documents by following procedures.

(1) Extract important parts (words, sentences or short paragraphs) with some discriminating tags for making index data file.

© 2003 National Institute of Informatics

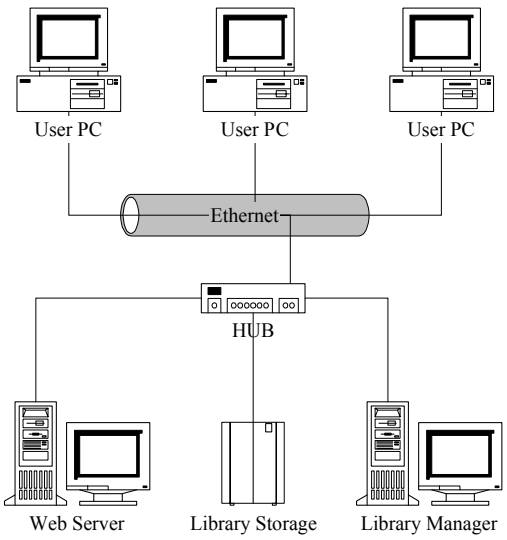

Figure 1. System diagram

(2) Extract all hyper linked web site from original documents, and calculate a site rank parameter for each site. Also sort and save these data to site rank data file.

(3) When received a search request, web server will search the keywords from index data file at first.

(4) If hit, then find the short passage including each keyword from hit original documents.

(5) Return all search results with URL address, short passage, relevancy, site rank to request user.

2.2 Making Index Data File

Because a web site is consisted by a lot of homepage, and every homepage includes many hyperlink to other homepage generally, we think that the information of a target homepage should be is gathered not only from the target page itself but also from all of these pages linked from the target page. As the link page from a target page can be divided into two type, the first type of these link pages have same URL address with the target page, and second type of these link pages have different URL with the target page. We think the target page and all of link pages with same URL address should be considered as one web article. Specific information properly is included in the unit of web article. So we make index data file by the unit of web article. From these pages we extract following sentences and save these to index data file.

(1) Sentence between <title> and </title> tag.

(2) Sentence of first level heading between <h1> and </h1> tags.

(3) Sentence of second level heading between <h2> and </h2> tags.

(4) Sentence of third level heading between <h3> and </h3> tags.

(5) Sentence of description of hyperlink between and tags.

(6) Sentence in big font between and tags.

(7) Sentence in big font between and tags.

(8) Sentence in big font between and tags.

(9) Sentence in big font between and tags.

(10) Sentence in big font between and tags.

(11) Sentence in big font between and tags.

(12) Sentence in big font between <big> and </big> tags.

(13) Sentence in emphatic font between and tags.

(14) Sentence in list after <li> tag.

2.3 Ranking of Web Site

In order to provide a parameter for the quality of web site, we used a site rank parameter which can be calculated by count the appearance times of it is linked from other pages with different URL address. For the real Internet it is difficult to show the site rank. By using NII test collection, because it is a collection of offline web documents, we try to find the site rank of all web site appeared in the collection documents. The flow chart for calculating site rank is shown in Figure 2.

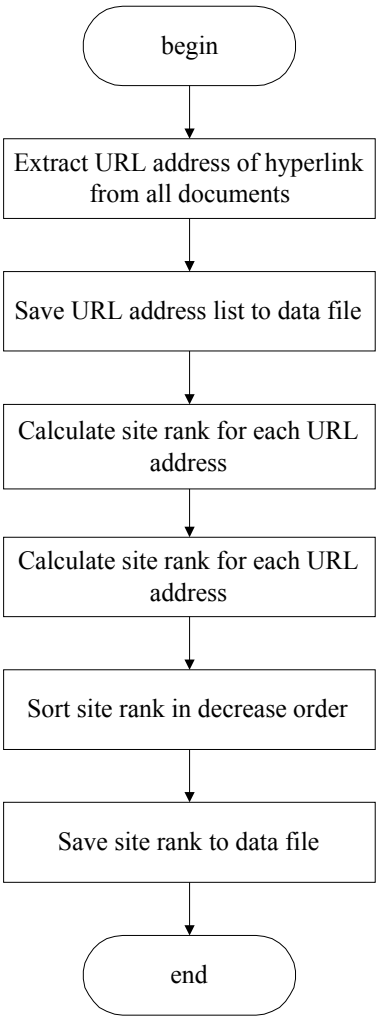

Figure 2. Flow chart for calculating site rank

2.4 Search Information by Keywords

For the use of information retrieval by internet browser, a web server with search engine must be constructed. Our search engine provides a three keyword input fields, and also provides a compound search condition by a logical expression of three keywords. Generally, the search engine looks for each keyword from index data file, and then provides results of logical operation specified by user's request. For the retrieval precision, we search the keywords from original documents when the keywords is found in index data file, and also we show the relevance parameter by search results from original documents. The flow chart of our search engine is shown in Figure 3.

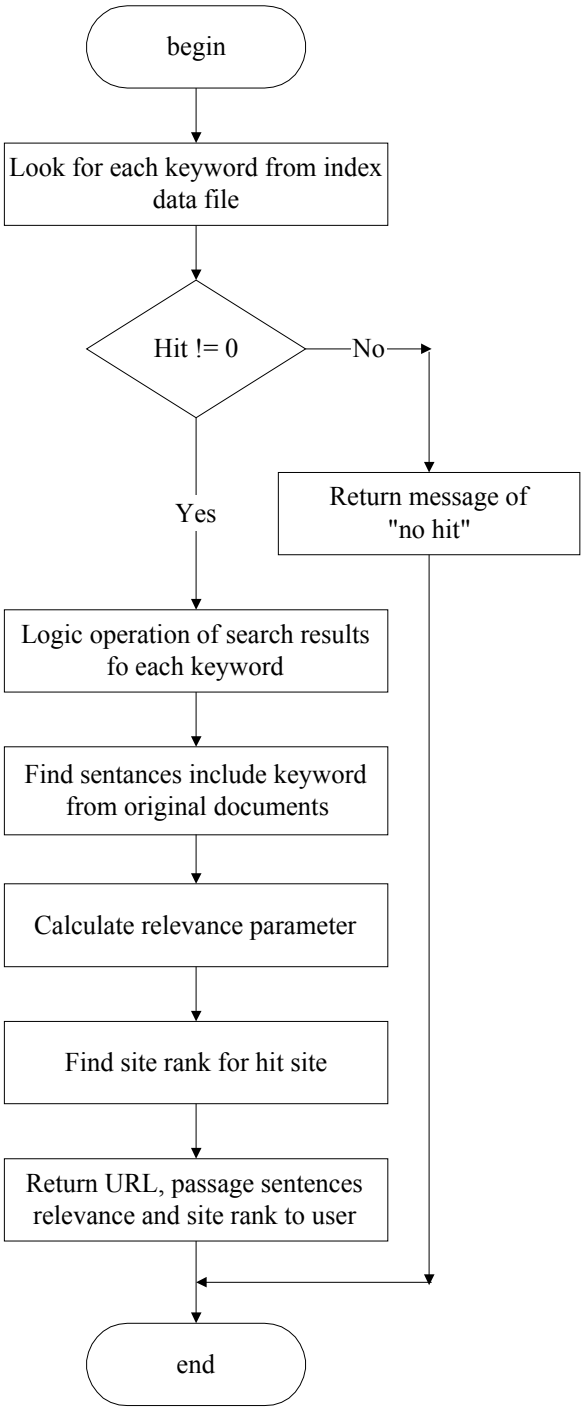

Figure 3. Flow chart of search engine

3 System Description

The diagram of our web information retrieval test system has been shown in Figure 1. The library storage is a hard disk for original web documents, index data file and site rank data file. The library manager is a computer for making index data file,

and calculating site rank. And the web server is computer that provides web services, especially a search engine service. Some detail description is given as following.

3.1 Server machine

The server machine is an x86-based personal computer, its CPU clock is 1000MHz, and its physical memory is 2096KB, and also a 500GB raid is included. The operating system of this machine is Microsoft Windows 2000 Server.

3.2 WEB Server

For realizing a web search engine, we used a free web server of Apache for Win32. Also we configured it to execute CGI program. The programming language we used for CGI is Perl. And the Perl interpreter we used for our system is ActivePerl 5.6.1.

3.3 CGI Program

In order to construct a web information retrieval system described in section 2, we prepared three CGI program.

(1) makeindex.cgi: This program makes a index data file from original web site documents.
(2) siterank.cgi: This program gives a rank parameter of a web site by count the appearance times of it is linked from other pages.
(3) searchft.cgi: This program accepts user's request and provides searching results including some evaluation parameters.

4 Retrieval Experiment

Here we will give a short description of our web information retrieval experiment by PC screen of search engine program.

4.1 Input Field

The PC screen of user's request input is shown in Figure 4. Three keywords can be specified in our system, and user's retrieval request can be given by a logical expression of the three keywords.

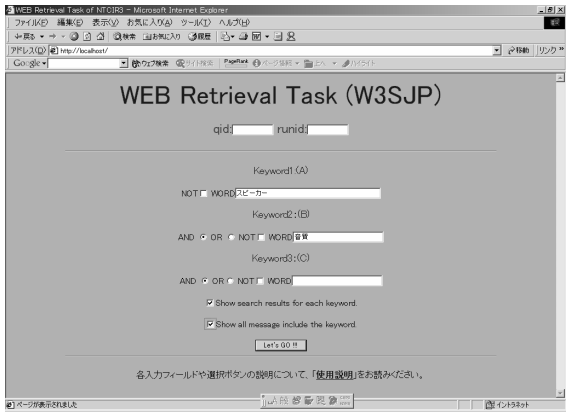

Figure 4. User's request input field

4.2 Search Results

The PC screen of messages for each keyword is shown in Figure 5. The system can show some detail messages if the keyword is hit in a web document, which includes a document number, passages, and name of HTML tag.

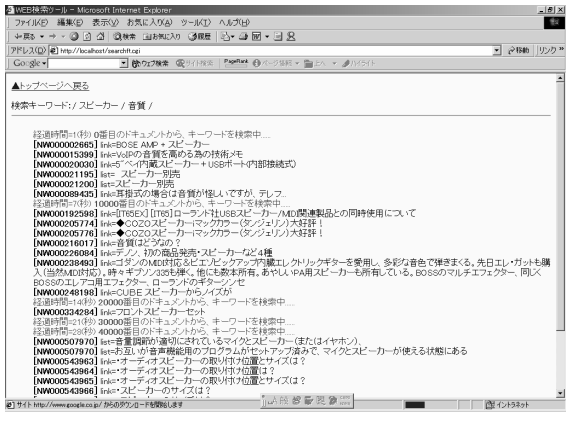

Figure 5. Messages for each keyword

The PC screen of total search results for user's request is shown in Figure 6. This table is listed in document ID, title of the page, relevance to user's request, URL address and rank of the site. The table

is sorted in decrease order of relevance.

Figure 6. Total results for user's request

5 Conclusions

By the information retrieval experiments of 3rd NTCIR web task, we have found some problems that should be improved.

(1) Because a web site includes many pages for related contents, we should search the user's request from all page of the site, so that we should give a method to determine that which hyperlink is belong to the site and which one is not.

(2) Our system needs to read the index file from hard disk every time, so that if the web server can preload the index file, the speed of this system will be improved obviously.

(3) Generally a Perl program needs more run time, we should improve the system performance by updating a high speed Perl interpreter (for example, PerlEx of Active State)

We will continue our work to give a more precise method, and improve the web retrieval system.

Acknowledgment

The authors would like to thank web task co-chair of 3rd NTCIR for their support and suggestions in this experiment. The authors also would like to thank NII (National Institute of Informatics) for the permission of the Open Laboratory.

References

[1] The Clever Project. Hypersearching the Web. *Scientific American*, June, 1999.

[2] Web site: SEARCH ENGINE WATCH, http://www.searchenginewatch.com