OOPL을 이용한 정보 검색 최적화 시스템에 관한 연구

김용호* · 오근탁* · 이윤배*

A Study on Information Search Optimization System Using OOPL

Yong-Ho Kim* · Geun-Tac Oh* · Yeun-Bae Lee*

요 약

최근 멀티미디어 기반의 WWW(World Wide Web) 서비스를 중심으로 하는 인터넷의 사용이 일반화되면서 전 세계의 컴퓨터망에 존재하는 수많은 정보들을 취득할 수 있게 되었다. 인터넷의 사용이 일반화되고 있는 현대의 사회에서는 정확한 정보를 신속하게 취득하는 것이 중요한 문제로 대두되고 있다.

본 논문에서는 OOPL(Object-Oriented Programming Language)인 JAVA를 이용하여 검색엔진을 설계하고 최적화된 URL을 추출하여 이용자에게 제공함으로써 더욱 정확한 정보를 획득할 수 있음을 보였다. 또한 기존의 국내 제작 검색엔진들과 비교하여 제안한 시스템에서는 배드 링크율이 개선됨을 보였다.

ABSTRACT

As use of internet generalized laying stress on WWW(World Wide Web) service of multimedia based recently, we could acquire many informations that exist to all over the world's computer network. It is risen to important problem that use of internet acquires correct information rapidly on modern society which is generalized.

This paper designed internet search engine and understand structure of that drawing URL which is optimized, and secure embodiment technology using OOPL(Object-Oriented Programming Language). Also, compare with existent domestic manufacture search engines and system that propose showed that the bad link rate is improved in this paper.

키워드

검색엔진, 정보검색 최적화, 색인 데이터베이스, URL

Ⅰ. 서 론

최근 멀티미디어 기반의 WWW(World Wide Web) 서비스를 중심으로 하는 인터넷의 사용이 일반화되면서 전 세계의 컴퓨터망에 존재하는 수많은 정보들을 취득할 수 있게 되었다. 따라서, 인터넷이 보편화되기 이전에는 정보의 습득이 중요한 문제가 되었지만 인터넷의 사용이 일반화되고 있는 현대의 사회에서는 정확한 정보를 신속하게 취득하는 것 중요한 문제로 대두되고 있다[2,6,7]. 인터넷의 활용의 핵심은 인터넷을 통해 자신이 필요로 하는 적절한 정보를 신속하게 검색하는 것이며, 이를 위해서는 인터넷의 수많은 정보들로부터 자

*조선대학교 전자계산과

접수일자 : 2004. 4. 1

신이 필요로 하는 적절한 정보들을 선택적으로 탐색해주는 수단이 필요한 데, 이것이 바로 인터넷 정보검색 엔진(internet information search engine)이다[2,6].

그러나 이러한 인터넷 정보검색 엔진들 또한 질의에 대한 잘못된 링크, 중복된 URL 등의 문제점을 가지고 있다.

이에 본 논문에서는 분산환경에 적합한 객체기반 언어(Object-based Language)인 Java를 이용하여 검색 엔진을 통해 얻어온 정보에 대한 효율적인 색인 추출 및 색인 데이터베이스를 구축하고 잘못된 링크를 자동 제거하며, 중복된 URL을 제거하는 알고리즘을 이용하여 정보검색을 최적화하고 이용자의 요구에 근접한 정보를 제공하는 시스템을 구현하였다.

본 논문은 2장에서 기존의 인터넷 검색엔진들의 특성을 고찰하여 분류하고, 제안한 알고리즘의 구현에 필요한 핵심 기술들을 고찰한다. 그리고 이를 토대로 3장에서 최적화 알고리즘을 제안하고, 4장에서 제안한 시스템과 기존 검색엔진의 추출된 정보를 비교·분석하며, 5장에서는 본 논문의 결론과 향후의 연구과제에 대하여 논의한다.

II. 관련 연구

2.1. 인터넷 검색엔진의 분류

(1) 검색방법에 의한 분류

인터넷 검색엔진의 검색방법에 의한 분류는 검색엔진이 사용자의 검색요구를 수용하는 방법을 기준으로 하는 것으로, 크게 주제 검색엔진과 키워드 검색엔진으로 구분할 수 있다.

주제 검색엔진은 검색 영역을 주제별로 분할하고, 필요하면 각 주제 영역을 다시 세부주제 영역으로 분할하여 각 영역별로 검색결과를 제공하는 것으로, 달리 메뉴(menu) 검색엔진 또는 디렉토리(directory) 검색엔진이라고도 한다. 주제 검색엔진의 예로 야후(Yahoo, http://www.yahoo.com) 검색엔진과 WWW Virtual Library(http://vlib.org) 검색엔진이 있다.

키워드 검색은 사용자가 원하는 키워드 입력을 통하여 결과를 얻는 검색법이다. 이를 위한 데이터베이스를 구축하는 방법에는 매뉴얼 색인(manual index) 기법과 에이전트 색인(agent index) 기법이 있다. 매뉴얼 색인 기법은 웹서버를 구축한 사람이 직접 검색엔진에 자신이 구축한 서버나 HTML (HyperText Markup Language) 문서의 URL (Uniform Resource Locator)을 등록해 주어야만 데이터베이스가 갱신되는 기법이다. 에이전트 색인 기법은 웹 로봇이 인터넷상의 웹서버들을 돌아다니며 자신이 방문한 서버들의 URL을 자신의 데이터베이스에 자동으로 등록해 준다. 키워드 검색은 키워드를 통하여 원하는 정보를 신속하게 찾을 수 있다.

(2) 검색대상에 의한 분류

검색대상에 의해 분류했을 때 위의 웹을 대상으로 하는 검색엔진 외에 유즈넷검색과 전문 데이터베이스, 그리고 지역에 국한된 검색으로 분류된다.

2.2. 검색엔진의 구현기술

키워드 검색 방법을 제공하는 인터넷 검색엔진의 구현기술은 크게 인터넷상의 사이트 정보들을 구성하여 색인 데이터베이스를 구성하는 로봇 에이전트 구현기술과 이러한 색인 데이터베이스를 이용하여 사용자의 검색요구를 처리해주는 검색에이전트 기술로 구분할 수 있다.

(1) 로봇 에이전트

키워드 검색을 제공하는 검색엔진에서 색인데이터베이스 구성을 위해 에이전트 색인기법을 사용한다면 반드시 로봇 에이전트의 사용이 필요하다. 로봇 에이전트는 자동으로 웹의 하이퍼텍스트 구조를 따라 다니며 문서를 추출하고, 다시 그 HTML 문서에서 참조되는 다른 HTML 문서들의 추출을 순환적으로 수행하는 프로그램으로[6], 달리 wanderer, crawler, spider라고도 한다.

(2) 색인 데이터베이스 관리

정보엔진의 색인구성은 한 단어(색인어)가 어떤 문서에 출현했는지를 신속하게 알 수 있도록 구조화하는 작업으로, 이러한 과정을 색인화(indexing)라고 한다. 대표적인 검색엔진의 색인화 방법에는 bitmap indexing, inverted file indexing, signature file indexing이 있다.

III. OOPL을 이용한 정보 검색 최적화 시스템

3.1 개요

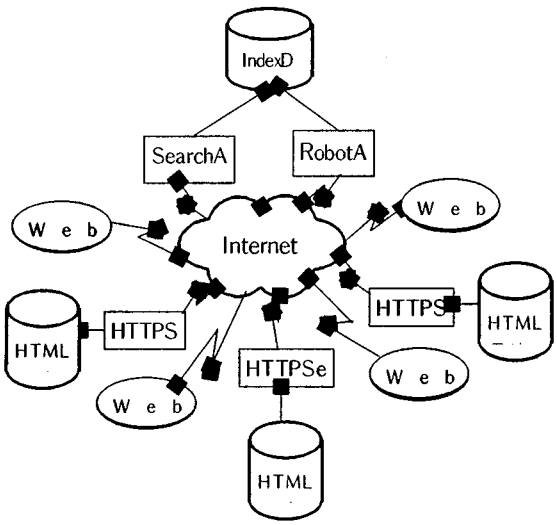
본 논문에서는 검색엔진을 통해 얻어온 정보들에서 중복된 링크와 잘못된 링크에 대한 정보를 제

거하여 질의에 최적화된 정보를 추출하는 알고리즘을 이용하여 이용자의 요구에 근접한 정제된 정보를 제공하는 시스템을 제안하였다. 중복된 링크와 잘못된 링크를 제거한 정보들은 사용자의 의사결정에 있어 지대한 영향을 미치게 된다.

3.2 로봇/검색 에이전트의 네트워크 환경

본 연구에서 사용된 검색엔진의 네트워크 환경은 기존의 인터넷 검색엔진과 유사하다. [그림 1]과 같이 각 HTTP 서버상의 HTTP 문서, 인덱스 데이터베이스, 로봇 에이전트, Web 브라우저 등으로 구성되어 있으며 인터넷이 이들을 연결시켜 준다. 로봇 에이전트는 HTTP 서버의 URL을 중심으로 해당 디렉토리 및 하위 디렉토리의 각 HTML 문서를 대상으로 인덱스 정보를 파싱하고, SQL 서버를 경유하여 Index DB로 저장된다.

제안한 시스템은 특정 URL을 부여하게 되면 그 URL 문서를 비롯한 하위 문서들을 자동으로 인덱스 데이터베이스 사이트로 가져와서 문서에 첨부된 링크(link)를 추출하고 이를 URL 테이블에 저장한다. 태그(tag)를 분리하고 태그가 제거된 문서의 내용을 인덱스 데이터베이스에 저장하게 된다. 이렇게 저장된 인덱스 데이터베이스에서 1차로 중복된 링크를 제거하고 그 다음 잘못된 링크를 산출하여 그 정보를 인덱스 데이터베이스에서 제거하여 질의에 보다 신뢰도와 유사도가 높은 문서 정보를 제공하는 알고리즘을 제안하였다.



[그림 1] 로봇/에이전트의 네트워크 환경
Fig 1. Network environment of robot/agent

3.3 인덱스 데이터베이스와 로봇 에이전트의 구조

(1) 인덱스 데이터베이스 구조

로봇 에이전트가 HTML 문서로부터 키워드 및 관련 정보를 추출하여 인덱스 데이터베이스로 저장하게 되는데, 인덱스 데이터베이스가 포함하는 일반적인 릴레이션 테이블은 키워드 테이블(KeyWord), URL 테이블(url), 방문한 URL 테이블(Visited_URL), 아직 방문하지 않은 URL 테이블(Non_Visited_URL), 조사 테이블(Auxil), 불용어 테이블(stopword)로 구성된다.

(2) 로봇 에이전트 구조

인터넷 검색엔진의 로봇 에이전트는 로봇 에이전트 관리자로부터 시드사이트의 URL을 받아 URL의 HTML 문서를 획득하면, 이 문서를 분석하여 이의 결과를 키워드 색인 데이터베이스에 추가하고, 다시 한 URL을 선택하여 같은 과정을 반복해야 한다. 로봇 에이전트 프로그램의 파싱알고리즘은 [알고리즘 1]과 같다.

(3) 로봇 에이전트와 데이터베이스의연동 구조

본 논문의 구현에서 로봇 에이전트와 데이터베이스의 연동은 JDBC(Java DataBase Connectivity)를 이용한다. JDBC는 자바 프로그램에서 데이터베이스 질의어인 SQL (Structured Query Language) 명령문을 실행하기 위한 자바 API(Java Application Program Interface)이다.

JDBC는 데이터베이스와 연결하고, SQL문을 전송하며, 결과를 처리하는 등의 세 가지 기능을 수행할 수 있다[5].

3.4 구현환경 및 검색 예

(1) 검색엔진 구현환경

본 논문에서 설계, 구현한 검색엔진은 IBM PC(Pentium-IV1.5GHz)의 Windows XP 환경에서 구현되었다. 웹 서버는 Windows 2000에서 제공하는 IIS(Internet Information Server) 4.0을 사용하였고, 키워드 인덱스 데이터베이스의 구성은 관계형 데이터베이스 시스템인 MySQL을 사용하였으며, 웹서버인 IIS와 데이터베이스 시스템인 MySQL의 연동은 JDBC를 통해 구현하였다. 그리고 로봇 에이전트 프로그램은 객체기반 언어인

Java(JDK1.3.1)를 사용하여 구현하였다. 그리고 검색 에이전트 프로그램은 IIS의 ASP를 사용하여 구현하였다.

```
Read URL;
If (URL is exist in VU)    URL reset;
While (NVU is not empty) {
  Read the file in URL;
  Abstract title;
  While (file is not empty) {
    If (character is "<")
      While (character is not ">") {
        If (next character is "a") {
          Abstract URL;
            If (URL is non-exist in VU)
              Add URL to NVU;}
        Else
          Skip character is ">";
        }
    Else {
      Abstract the word;
      If (word is stopword)
        Skip;
      Else If (word is exist in KeyWord table)
              Increase the Rate in URL table;
           Else
              Add word to KeyWord table and URL
table;
     }
  Pop and Delete URL from NVU;
}
```

[알고리즘 1] 로봇 에이전트의 pseudo code

① inputquery.asp 파일: 입력처리

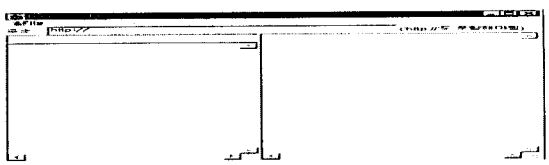
```
<form method="post" action="searchonweb.asp">
검색할 단어를 입력하세요 :
<input type="text" name="input_keyword"
size="20"><br><hr>
다음 옵션중 원하는 항목을 작성하세요
<br><br>
분류 : <select name="clas">
<option selected value="tot">전체</option>
<option value="person">개인홈</option>
<option value="site">사이트
</option></select>
 기관 : <select name="organ">
<option selected value="tot">전체</option>
<option value="corp">기업</option>
        :
<option selected value="tot">전체</option>
<option value="korea">한국</option>
        :
<option value="ukd">영국</option></select>
<br><hr>
<input type="submit" value="검색"
width="200"> </p></form></center>
</body></html>
```

② searchonurl.asp 파일: 검색 키워드 탐색 화일

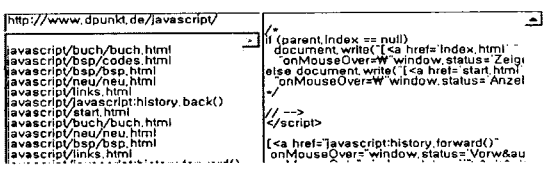
```
<%@ LANGUAGE="VBScript" %>
<%
keywd = Request.form("input_keyword")
keywd = "'" & "%" & keywd & "%" & "'"
clas = Request.form("clas")
org = Request.form("organ")
nation = Request.form("national")
if clas = "tot" and org = "tot" and nation = "tot"
then schSql = "SELECT * FROM keyword,url WHERE
keyword=" & keywd
end if
Set searchConn =
Server.CreateObject("ADODB.Connection")
searchConn.Open ("DSN=search;uid=sa;pwd=")
Set searchRec =
Server.CreateObject("ADODB.RecordSet")
searchRec.CursorType = 1
searchRec.Open schSql,searchConn
searchRec.MoveLast
%>
<% if postcount = 0 then %>
검색된 결과가 없습니다.
<% else %>
<% Do While searchRec != empty %>
<%=searchRec("title")%><br>
<A href =
"<%=searchRec("url")%>"><%=searchRec("url")%></a
>
<%=searchRec("rate")%>
<%Loop%>
```

(2) 검색엔진의 항해

[그림 2]와 [그림 3]은 구현한 로봇 에이전트의 작업 화면으로, 로봇 에이전트는 로봇 관리자가 제공한 시드사이트(seed site)의 HTML 문서원본과 이를 통해 URL 및 키워드를 추출한다.

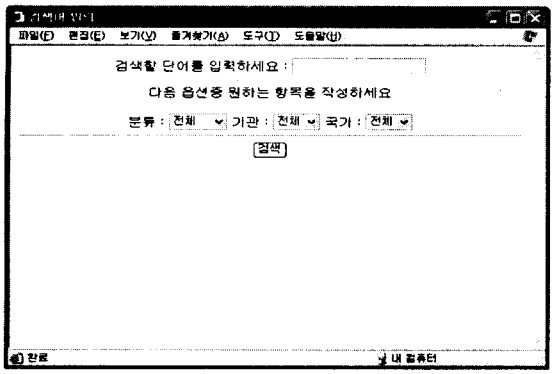


[그림 2] 로봇 에이전트의 실행 초기화면
Fig 2. ExecuteFirst Display of robot agent



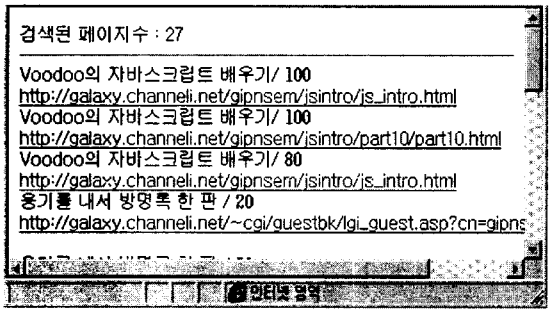
[그림 3] 파싱중인 로봇 에이전트
Fig 3. Robot agent in Parsing

[그림 4]는 로봇 에이전트의 검색 에이전트의 사용자 인터페이스 화면으로, 기본적으로 키워드 검색만을 제공한다.



[그림 4] 검색에이전트 초기화면
Fig 4. First Display of search agent

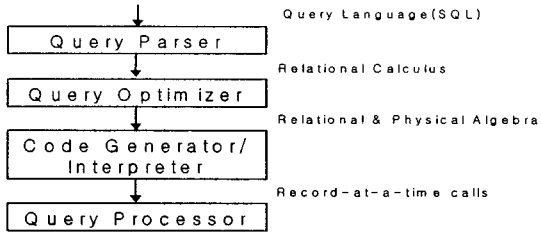
[그림 5]는 사용자가 [그림 4]의 화면에서 "자바스크립트"를 검색 키워드로 검색했을 때의 검색결과를 나타낸다. 검색된 페이지 수는 검색된 항목수를 나타낸다. 초기화면에서 출력 페이지당 출력 항목 수를 10으로 주었기 때문에 결과화면에서 27개의 항목을 3 페이지로 나누어 보여주게 된다.



[그림 5] "자바스크립트" 검색 결과 화면
Fig 5. search result display "javascript"

3.5 검색 최적화 알고리즘

현재의 클라이언트 컴퓨터의 성능은 예전에 비해 매우 향상되었고 상상을 초월하는 속도와 연산을 할 수 있다. 또한 JAVA 플랫폼의 특성은 플랫폼의 독립성으로 인해 어느 운영체제에서도 운영되는 환경을 VM(Virtual Machine)으로 제공한다. 이를 이용하여 3.5절에서 로봇 에이전트가 항해를 통하여 가지고온 URL들의 정보를 클라이언트에서 다시 한번 최적화하여 질의어에 신뢰도가 높고 유사도가 높은 URL을 추출하였다. 질의 최적화는 [그림 6]과 같이 DBMS를 통하여 수행할 수 있다.



[그림 6] DBMS를 통한 Query 흐름
Fig 6. Query flow of DBMS

Ⅳ. 성능 비교 · 분석

4.1. 검색엔진의 특성 비교

검색엔진의 특성은 사용자 인터페이스, 운영환경, 검색할 때의 검색 우선순위, 검색결과 페이지의 요약특성, 검색결과의 페이지 출력수 등으로 파악할 수 있다. 이러한 특성들을 기준으로 본 논문에서 제안한 검색엔진과 상용의 국산 검색엔진들을 비교하면 [표 1]과 같다.

[표 1] 기존검색엔진과의 비교표

검색엔진 / 비교기준	제안한 검색엔진	심마니	까치네	정보탐정
사용자 인터페이스	0.5	2	1	1.5
운영체제	Windows 2000	Solalis 2.5	-	Solalis 2.5 Linux
우선순위	Title	Reviewed Ranking	title	title, heading
페이지 요약	Title	동일개념 단어의 출현빈도가 높은 문장	검색어의 출현빈도가 높은 문장	페이지 상위 160byte
페이지당 출력 항목수	10 (선택가능)	10 (선택가능)	10	10 (선택가능)

[표 1]에서 보면, 각 검색엔진의 사용자 인터페이스는 출력 윈도우의 크기를 최대로 했을 때 초기화면에 나타낼 수 있는 크기이다. 페이지 당 출력

항목 수는 검색엔진이 검색결과를 화면에 출력할 때 출력항목의 개수를 나타낸 것으로, 까치네를 제외한 나머지 검색엔진들은 사용자가 선택적으로 구성 가능하다.

4.2. 검색결과의 불량링크율 비교

본 논문에서 구현한 프로토타입의 검색엔진의 검색성능을 대표적인 국내제작 검색엔진들과 하면, [표 2]와 같다. [표 2]에서, 각 검색엔진의 검색결과는 검색 키워드를 각 검색엔진에 요청해서 얻어지는 검색결과들 중에서 부적절한 링크의 비율이다. 부적절한 링크란 검색엔진이 검색의 결과로 제공한 정보사이트이지만 현재 인터넷 서비스를 제공하지 않는 사이트를 의미하며, 이를 간단히 불량링크(bad link)라고 한다. 불량링크는 검색엔진이 관리하는 인터넷 색인 데이터베이스의 비현실성을 의미한다. 검색엔진의 불량링크율은 다음과 같이 계산된다.

$$\text{검색링크의 불량링크율(\%)} = \frac{\text{검색결과에서불량링크의개수}}{\text{검색결과의총링크개수}} \times 100$$

[표 2]에서, 불량링크율 계산에서는 각 검색엔진의 검색결과 링크들중에서 해당 문서가 이동되거나 없어진 경우만을 포함하였다. 검색결과의 링크개수가 검색엔진마다 다르게 나타나기 때문에 300개의 링크를 기준으로 조사하여 계산하였으며, 검색결과의 링크개수가 300개가 넘을 경우에는 300개만을 조사하여 계산하였다. [표 2]에서 보는 바와 같이, 심마니, 까치네, 정보탐정 등 상용의 검색엔진들과 비교해서 본 논문에서 제안한 검색엔진의 불량링크율이 매우 낮았다. 이것은 제안한 검색엔진의 색인DB 구성이 가장 최신으로 구성된 이유가 있다는 점을 고려하더라도 다른 검색엔진과 비교해서 상대적으로 낮은 불량링크율을 갖는다.

[표 2] 검색결과의 불량 링크율 비교 (단위:%)

검색 키워드	제안한 검색엔진	심마니	까치네	정보탐정
"자바 스크립트"	0	13.23	31	30.27
"헌법재판"	0	23.23	18.5	31.76
"쇼핑인구"	0	1.8	22	32.74
"우리나라"	0	52.26	33.5	31.03
"고로쇠"	0	7.42	58	18.39

V. 결론 및 향후 연구

인터넷 검색엔진은 인터넷 사용자가 정보를 검색하기 위해 인터넷을 항해할 때 도움을 주는 수단으로, 사용자는 검색엔진이 제공하는 검색결과들을 토대로 인터넷의 정보사이트들을 항해함으로써 인터넷의 정보항해 시간을 줄이고, 필요한 정보를 신속하게 획득할 수 있도록 해준다. 따라서, 인터넷에서의 정보검색은 검색엔진의 성능에 의해 영향을 받는다.

인터넷 검색엔진의 성능은 사용자의 검색요구에 대해 검색엔진이 제공하는 정보사이트의 양과 검색속도에 의해 평가되며, 특히 검색한 정보사이트의 유효성은 중요하다. 검색엔진이 제공하는 불량사이트(bad link)는 불필요한 인터넷 항해를 초래한다. 이것은 결과적으로 사용자의 불필요한 인터넷 사용시간을 증대시킨다. 따라서, 인터넷의 이용효율을 높이기 위해서는 낮은 배드링크율을 제공하는 검색엔진을 개발하는 것이 중요하며, 이를 위해서는 검색엔진이 관리하는 색인 데이터베이스가 인터넷 상의 정보사이트 변화를 지속적으로 반영하여 관리해야 한다.

본 논문에서는 객체 기반의 언어인 Java를 사용하여 인터넷 검색엔진을 설계하고 최적화된 URL을 추출함으로써 인터넷 검색엔진의 구조를 이해하고, 구현 기술을 확보하였다. 논문에서 제안한 검색엔진은 키워드 검색을 제공하며, 사용자 인터페이스를 단순화함으로써 사용자의 편의성을 도모하였다. 그리고 기존의 국내 제작 검색엔진들과 비교해서 검색된 정보사이트의 양이 적은 대신 검색결과의 배드링크율은 개선됨을 보였다. 따라서, 본 논문에서 제안한 검색엔진은 범용의 검색엔진으로 사용하는 것보다는 특정 인트라넷 내에서 또는 특정 개인의 검색특성을 고려하는 개인용 검색엔진으로 사용하는 것이 유용하다.

향후 연구로는 제안한 시스템의 성능을 더욱 개선 시키기 위해 정제된 자료의 불량 링크율을 더욱 최소화 시키는 알고리즘과 연산의 수행을 더욱 단축할 수 있는 알고리즘을 연구하여 이에 합당한 시스템을 구현해야할 필요가 있다고 사료된다.

참고문헌

[1] 김수동, "Java기반 인터넷 어플리케이션 아키

텍쳐 및 설계 기법", 정보과학회지 제 16권 4호 pp.9-15, 1998.4.
[2] 신봉기, 김영환, "인터넷 정보검색 서비스 동향," 정보과학회지, 정보과학회, 16권 8호, pp16~20, 1998.8.
[3] 신봉기, 김영환, "웹 에이전트", 정보과학회지 제 15권 제 3 호 pp.61-68, 1997.3.
[4] 오종인 외, "사용자 중심의 웹 정보검색 시스템 설계," 정보과학회지, 정보과학회, 24권 1호, pp425~428, 1997.
[5] Graham Hamilton, Rick Cattell, Maydene Fisher, "JDBC Database Access with Java:A Tutorial and Annotated Reference", Addison Wesley pub, 1997.
[6] Kartijn Koster, NEXOR "Robots in the web:threat or treat?" connxions, volume9. No4, April 1995
[7] Kathleen webster, kathryn paul, "Beyond surfing: Tools and Techniques for searching the web" http://magi.com/~mmelick/it96jan.htm
[8] Mark Watson, "Intelligent Java Applications for the Internet and Intranets" Morgan Kaufman Publishers, 1997.
[9] Ronan Sorensen, "Inside Microsoft Windows NT Internet Development", Microsoft Press, 1998.
[10] Scot Hillier, Paniel Mezick, Dan Mezick, "Programming Active Server Pages", Microsoft Press, 1997.

저자소개

김용호(Yong-Ho Kim)

1989년 광주대학교 전자계산과 졸업
1993년 경남대학교 전자계산과 공학석사
2002년 조선대학교 전자계산과 박사수료
2003년 멀티미디어 기술사
2004년 동강대학 컴퓨터정보계열 프로그래밍 전임강사
※관심분야 : 멀티미디어, 유비쿼터스, 모바일 컨텐츠

오근탁(Geun-Tack Oh)

조선대학교 전자계산과 이학석사
현재 조선대학교 전자계산과 박사과정
현 (주)서림정보시스템 대표이사

이윤배(Yun-Bae Lee)

1980년 광운대학교 전자계산학과
1983년 광운대학교 대학원 전산학과 이학석사
1993년 숭실대학교 대학원 전산학과 공학박사
1988년 4월~조선대학교 컴퓨터공학부 교수
2003년 한국정보처리학회 부회장
2001년 한국정보처리학회 호남 제주지부 부회장
2002년 2월~과학기술홍보대사
2001년 1월~한국해양정보통신학회 학술이사
2001년 1월~한국정보과학회 논문심사위원
※관심분야 : 인공지능, 전문가시스템, 멀티미디어, 데이터베이스, 정보보안