로컬 웹사이트의 탐색전략과 웹사이트 유형분석에 관한 연구

황 인 수*

A Study on the Crawling and Classification Strategy for Local Website

Insoo Hwang*

Abstract

Since the World-Wide Web (WWW) has become a major channel for information delivery, information overload also has become a serious problem to the Internet users. Therefore, effective information searching is critical to the success of Internet services. We present an integrated search engine for searching relevant web pages on the WWW in a certain Internet domain. It supports a local search on the web sites. The spider obtains all of the web pages from the web sites through web links. It operates autonomously without any human supervision. We developed state transition diagram to control navigation and analyze link structure of each web site. We have implemented an integrated local search engine and it shows that a higher satisfaction is obtained. From the user evaluation, we also find that higher precision is obtained.

Keywords : Internet, Information Search, Search Engine, Information Retrieval

논문접수일 : 2006년 1월 19일 논문게재확정일 : 2006년 4월 19일

* 전주대학교 정보기술공학부 부교수, (560-759)전라북도 전주시 완산구 효자동 3가 1200, Tel : 063-220-2757, Fax : 063-224-9920, e-mail : insoo@jj.ac.kr

1. 서 론

인터넷을 위한 인프라의 구축이 확대되고 인터넷의 활용이 생활화됨에 따라 사이버 공간의 정보는 기하급수적으로 증가하고 있으며, 이는 정보검색(information retrieval)에 새로운 과제를 안겨주고 있다. 무어의 법칙에 따르면, 인터넷의 정보량은 18개월마다 2배로 증가하고 있으나, 홈페이지의 개수는 이를 훨씬 더 초과하여 매 6개월 혹은 그보다 짧은 기간에 2배로 증가하고 있다[Chen, 1998]. 이에 따라, 인터넷 사용자에게 있어서 자신이 필요로 하는 홈페이지 혹은 정보를 검색하는 것은 새로운 도전이 되고 있다.

인터넷 사용자가 웹의 정보를 검색할 경우 일반적으로 링크 그래프(link graph)를 이용한다. 이 때, 특별한 목적을 갖고 의도적으로 구성한 디렉터리 목록이나 검색엔진의 검색결과로부터 출발한다. 수작업으로 관리하는 디렉터리 목록은 보편적인 주제를 효과적으로 포괄할 수 있다는 장점이 있지만, 주관적이기 쉬우며, 구축/유지비용이 높고, 환경의 변화에 따른 개선속도가 느릴 뿐만 아니라, 다양한 주제를 모두 포괄하지 못하는 단점이 있다.

인터넷에 존재하는 엄청난 분량의 웹페이지를 수집하여 일일이 색인(index)하는 것은 거의 불가능한 일이다. 이에 따라, 인터넷 정보검색에서는 에이전트를 개발하여 웹페이지의 색인을 자동적으로 작성하고 있으며, 색인의 대상이 되는 웹페이지는 로봇에 의해 자동적으로 수집된다. 로봇(robot)은 크롤러(crawler), 스파이더(spider), 웜(worm), 혹은 워커(walker) 등으로 불리기도 하는데, 방문한 웹페이지로부터 다른 웹페이지로 연결되는 링크를 추출함으로써 사전에 알려지지 않은 웹페이지도 자동적으로 수집하게 된다.

그러나 웹페이지를 효과적으로 검색하는 검색엔진을 만들기 위해서는 많은 과제를 해결해야 한다. 첫째로, 웹페이지를 수집하고 최신 상태로 유지하기 위해서는 빠른 속도의 크롤링 기술(crawling technology)이 필요하다. 둘째로, 색인(index)과 내용(content)을 저장하기 위해서는 많은 공간이 필요하다. 색인 시스템은 수백 기가바이트의 데이터를 효율적으로 처리할 수 있어야만 하고 초당 수백에서 수천 개 이상의 많은 질의를 처리해야 한다. 이러한 과제는 웹이 계속 성장해 감에 따라 점점 더 어려운 문제가 되고 있다.

정보검색에 있어서 보다 많은 정보를 수집하여 색인할수록 정보검색의 효과는 향상되지만 정보수집 및 검색에 소요되는 시간은 이에 비례하여 증가된다. 즉, 정보검색의 정확성과 효율성은 상충관계(trade-off)에 있다. 이에 따라, 상용 서비스되는 검색엔진들은 일정수준의 절충점에서 정보를 수집할 수밖에 없기 때문에, 정보검색에서 누락되는 웹사이트 혹은 웹페이지를 흔히 찾아볼 수 있다.

이에 대한 보완책으로서 특정 홈페이지내의 HTML(Hypertext Markup Language) 파일과 텍스트 파일 등의 검색을 지원하는 로컬검색엔진이 개발되고 있다. Arnold[2005]는 "통계적 기법을 이용하는 구글과 야후는 정보검색에서 매우 좋은 성과를 제시하고 있으나, 조직내부에 존재하는 정보를 검색하는 것을 별개의 문제이다."라고 주장하였다. 김성진, 이상호[2002], 김영자 외 2인[2004], Rolleke et al.[2006], Spink et al.[2006], Vechtomova and Karamuftuoglu [2006] 등 정보검색에 관한 최근의 논문들은 주로 글로벌 인터넷에서 정보를 효과적/효율적으로 검색하는 방법을 제시하고 있으나, 조직내의 정보를 효과적으로 검색하기 위한 연구는 미진한 상태이다.

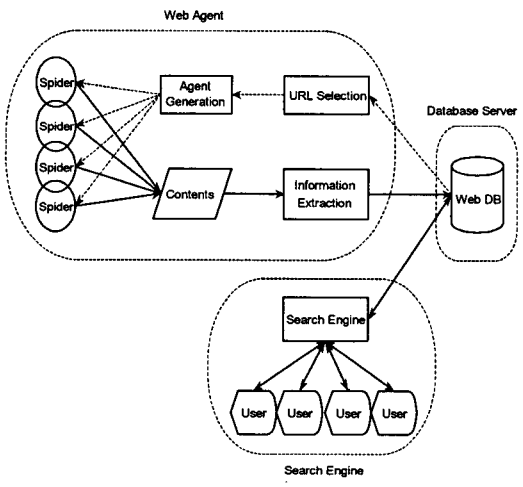
특히 대학이나 대기업 등과 같이 특정 도메인(domain)에서 운영하는 여러 개의 웹사이트 및 웹페이지를 통합 검색하는 방법에 대한 연구개발은 미진한 상태이다. 이에 따라, 본 연구에서는 동일한 도메인을 갖는 다수의 웹사이트를 통합 검색하기 위한 전략과 검색결과를 이용한 웹사이트의 분류방안에 대해 연구하였다.

본 논문의 구성은 다음과 같다. 제2장에서는 정보수집을 위한 탐색전략과 정보수집 에이전트의 설계에 대해 기술하며, 제3장에서는 시스템의 구현 및 적용결과를 기술하고, 제4장에서는 결론 및 향후 연구방향을 기술한다.

2. 정보검색 에이전트의 설계

2.1 정보검색 에이전트의 구성

정보검색 에이전트는 인터넷의 각 웹페이지를 방문하여 정보를 수집하는 웹 에이전트, 수집된 정보를 색인하여 관리하는 데이터베이스, 그리고 사용자의 질의에 대한 응답을 제공하는 검색엔진 등으로 구성된다. 이를 그림으로 나타내면 <그림 1>과 같다.



〈그림 1〉 정보검색 에이전트의 구성

정보검색 에이전트의 URL Selection 모듈은 사용자가 지정하는 시작 웹페이지 혹은 데이터베이스(database)에 저장되어 있는 링크 중에서 우선순위가 높은 링크를 선택한다. 여기서 많은 웹페이지를 순차적으로 탐색할 경우, 상당히 많은 시간이 소요되어 현실적으로 사용하기 어렵다. 따라서 본 연구에서는 자바의 쓰레드(thread) 프로그래밍 기법으로 생성한 수십에서 수백 개의 로봇(spider)이 웹페이지를 동시에 탐색하도록 하여 정보수집의 속도를 현저히 향상시켰다. 로봇에 의해 수집된 정보는 전처리 과정을 거쳐 데이터베이스에 저장되며, 탐색을 계속하기 위해 웹페이지에 포함되어 있는 링크를 추출한다. 데이터베이스에 저장된 정보는 검색엔진에서 키워드 혹은 질의어 검색에 활용된다.

```
public void webCrawling(URL_DB db) {
 While ((url=db.getURL()) != null) {
   Create InputStream to the url
   if (InputStream==null) {
     Update db Set URL_NOT_FOUND where URL=url
   } else {
     Read contents on the url using InputStream
     Extract <TITLE> and <BODY> from contents
     Remove HTML tags and scripts from <BODY>
     Insert and/or Update CONTENT_DB
     Extract urls from contents using information
           extraction wrappers
     Insert urls into db
   }
   Update db Set lastVisit=NOW, nextVisit=NOW+τ
         where URL=url
   }
}
```

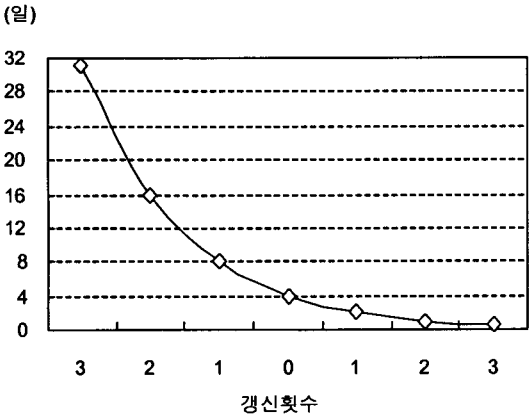
〈그림 2〉 정보수집 에이전트의 동작원리

<그림 2>는 정보를 수집하는 각 로봇의 동작과정을 자바 프로그래밍언어의 문법에 따라 기술한 것이다. 먼저 URL(Uniform Resource Locator)

데이터베이스로부터 방문할 웹페이지의 주소를 획득하여 입력 스트림(stream)을 생성한다. 이 때 입력 스트림이 Null 이면 해당 URL이 존재하지 않음을 나타내며, 입력 스트림이 존재할 경우에는 URL에 존재하는 정보를 문자열로 읽어서 웹페이지의 <TITLE>과 <BODY>, 그리고 다른 웹페이지를 연결하는 링크(link)를 추출한다.

정보수집이 끝나면 재방문을 위해 다음 방문 예정 시점을 설정한다. 이 때, 신규 방문의 경우에는 τ일 후에 재방문하도록 하며, 재방문의 경우에는 웹페이지가 갱신되었으면 차기 방문 예정기간을 1/2로 감소시키고, 갱신되지 않았으면 2배로 증가시킨다. 이것은 웹페이지의 갱신 빈도에 따라 방문빈도를 자동적으로 조정하는 역할을 한다.

본 연구에서는 방문대상 웹페이지의 개수와 웹서버의 부하 등을 고려하여 τ의 초기값을 4일로 설정하였으며, 웹페이지의 갱신여부에 따라 <그림 3>과 같이 최소 12시간으로부터 최대 31일까지 방문주기를 변화시킨다. <그림 3>의 X축에서 0을 중심으로 우측은 재방문시 갱신이 이루어진 횟수를 나타내며 좌측은 갱신이 이루어지지 않은 횟수를 나타낸다.



〈그림 3〉 재방문주기의 변화

2.2 웹페이지 탐색전략

인터넷 검색엔진에서 인터넷에 존재하는 모든 웹페이지를 수집하여 데이터베이스화 하는 것은 물리적으로 불가능한 일이기 때문에, 짧은 시간 내에 보다 유용한 많은 웹페이지를 수집하는 것이 중요하다. 이에 따라, 정보수집의 효율성을 제고하기 위해 멀티 에이전트를 활용하며, 상대적으로 중요성이 낮은 웹사이트 혹은 홈페이지의 방문을 억제하는 등의 기법을 사용하고 있다.

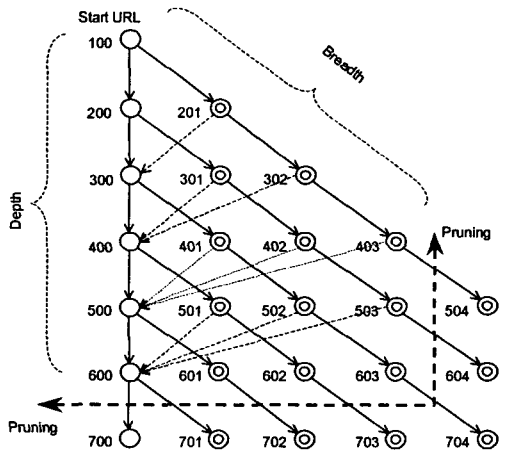
최근에는 인터넷의 속도와 검색엔진의 기능이 향상됨에 따라 정보검색의 성과가 크게 향상되었으나, 존재하지 않는 웹페이지를 검색결과로 제시하는 경우를 흔히 경험할 수 있다. 이것은 정보수집 에이전트가 웹페이지의 정보를 읽은 후 재방문을 통해 홈페이지 혹은 정보의 변화를 지속적으로 감시하지 못하기 때문이다.

사전에 알고 있는 웹사이트의 정보를 검색하기 위해 검색엔진을 사용하는 경우는 거의 없으며, 해당 웹사이트의 홈페이지로부터 링크를 따라 차례대로 검색을 하는 것이 일반적이다. 좋은 웹사이트는 잘 구축된 내비게이션 체제를 제공하거나 로컬검색기능을 통해 정보검색을 지원한다. 그러나 이것은 하나의 웹사이트에 국한될 뿐만 아니라, 데이터베이스화되어 있는 게시판 검색 등에 제한적으로 적용되고 있다. 대학이나 대기업 등과 같이 하나의 도메인에 다수의 웹사이트가 운영되고 있을 때, 각 웹사이트를 지속적으로 모니터링 하여 최신의 정보를 제공하는 검색엔진의 구조 및 고려사항에 대해서는 그 필요성에 비해 연구 및 개발이 미진한 상태이다.

특정 도메인내의 정보를 검색하는 것은 검색해야할 웹페이지의 개수가 상대적으로 작다는 점에서 상용 검색엔진과의 차이가 있으며, 하나

이상의 웹사이트를 포함하고 있다는 점에서 로컬검색엔진과 차이가 있다. 또한, 로컬검색엔진은 파일을 검색함에 있어서 운영체제의 도움을 받을 수 있으나, 일반적인 검색엔진은 전적으로 인터넷 프로토콜에 의존해야 한다. 이에 따라, 도메인내의 웹페이지를 탐색하는 전략은 모든 웹페이지를 탐색대상으로 하지만, 일정관리나 게시판 등과 같이 웹페이지를 무한히 생성할 수 있는 링크에 대해서는 통제된 탐색이 요구된다. 이를 위해 본 연구에서는 <그림 4>에서 보는 것과 같은 웹페이지의 상태천이도를 개발하였다.

<그림 4>에서 ○ 표시는 HTML로 작성되는 정적 웹페이지(static web pages)를 나타내며, ◎ 표시는 ASP, JSP, PHP 등의 인터넷 프로그래밍 언어로 작성되어 매개변수에 따라 웹페이지를 생성하는 동적 웹페이지(dynamic web pages)를 나타낸다. 탐색을 시작하는 웹페이지의 인덱스를 100으로 설정한 후, 클릭(click) 혹은 홉(hop)이 증가할 때마다 100을 더하며, 동적웹페이지인 경우에는 다시 1을 더한다.



〈그림 4〉 웹페이지의 상태천이도

<그림 4>에서 노드 100에서부터 노드 700으로 이어지는 경로(path)는 정적웹페이지로 연결되고 있음을 나타내며, 노드 100에서 노드 504, 604, 704 등으로 연결되는 경로는 동적웹페이지로 연결되고 있음을 나타낸다. 노드 201에서 노드 300으로 연결되는 경로는 게시판 등의 동적웹페이지가 정적웹페이지를 포함하고 있음을 의미한다.

그림에서 깊이(depth)는 최초 URL로부터 특정 웹페이지에 이르기까지의 클릭수로서, 위의 예는 600단위의 노드까지만 탐색한다. 또한 너비(width)는 게시판 등에서 다음 페이지로 이동하는 등의 동작을 지속하는 것으로서 한 단계를 더 내려갈수록 탐색할 웹페이지의 개수는 기하급수적으로 증가하기 때문에 적정한 수준에서의 통제가 필요하다.

본 연구에서는 상태천이도를 기초로 한 제한된 너비우선탐색(restricted Breadth First Search, rBFS) 방법을 사용하였다. 즉, 동적웹페이지의 런(run)의 크기를 일정수준으로 제한하면서 너비우선탐색을 수행하는 것으로써, 동적웹페이지를 무한히 검색하는 것을 방지할 뿐만 아니라, 시작페이지로부터 클릭수가 작은 웹페이지를 먼저 탐색하는 효과를 얻을 수 있었다.

기타로, 웹사이트에서 각 노드에 존재하는 웹페이지의 개수를 상태천이도에 매핑하면, 웹사이트를 링크구조에 따라 몇 가지로 분류할 수 있다. 즉, 깊이가 깊은 웹사이트는 전통적인 계층형의 메뉴구조로서 상세한 정보에 도달하기까지 많은 클릭을 요구하지만, 깊이가 얕은 웹사이트는 포털구조로서 첫 화면을 비롯하여 각 웹페이지가 많은 링크를 갖는 특징을 갖는다. 또한 동적웹페이지가 정적웹페이지보다 현저히 많은 경우에는 정보를 제공하기 보다는 공지사항, 질의응답, 자료실 등 커뮤니티 중심의 웹사이트임을 알 수 있다.

2.3 정보추출규칙

HTML로 이루어진 웹페이지로부터 정보를

추출하기 위해서는 정보가 존재하는 양식을 파악한 후, 이에 적합한 정보추출규칙(wrapper)을 구성해야 한다[최중민, 2000]. 예를 들어, 웹페이지로부터 전자우편주소를 추출할 경우에는, 전자우편주소가 '@'를 중심으로 사용자의 아이디와 메일서버의 주소로 구성되어 있으며, HTML의 <A> 태그와 함께 나타난다는 점을 고려해야 한다.

본 연구는 웹페이지의 <BODY> 부분에 존재하는 텍스트(text) 정보와 함께 다음 웹페이지로 연결되는 링크를 추출하는 것을 목적으로 한다. 여기서, 텍스트 정보는 모든 HTML 태그를 제거함으로써 쉽게 얻을 수 있으나, 링크정보는 <BODY>외에도 웹페이지의 어느 곳에든지 다양한 형태로 존재할 수 있기 때문에 이를 포괄하는 일관성 있는 형식에 따라 정보추출규칙을 구성해야 한다. 본 연구에서는 링크가 존재하는 <A> 등의 HTML 태그와 href 등의 옵션에 따라 <표 1>에 기반을 둔 정보추출규칙을 구성하였다.

<표 1> 링크를 갖는 HTML 태그의 예

태그	옵션	예 제
<A>	href	<A href="/bbs/view.jsp?id=1234">
<AREA>	href	<AREA href="profile.html">
<FRAME>	src	<FRAME src="top.html">
<IFRAME>	src	<IFRAME src="main.html">
<META>	url	<META url="/main/first.asp">

2.4 정보검색 전략

검색엔진에서 정보를 검색하는 방법은 키워드 혹은 질의어에 의한 방법과 디렉터리에 의한 방법이 있다. 디렉터리는 사용자 혹은 관리자가 웹사이트를 등록할 때 사전에 계층구조의 디렉터리를 지정하는 것으로서 주제별로 검색할 경우 유용하게 사용될 수 있다. 그러나 이는 본 연구에서 의도하는 로컬 웹사이트의 통합검색과는 거리가 멀기 때문에, 본 연구에서는 사용자가 입력한 키워드 혹은 질의어에 적합한 웹페이지를 검색하는 것을 목표로 한다.

다수의 웹사이트를 갖는 대학이나 대기업은 공식적으로 운영되는 웹사이트 외에도 각 부서 혹은 동호회 등에서 운영되는 웹사이트가 존재하기 때문에 이를 구분하여 검색하는 기능이 요구된다. 또한, 웹사이트에서 제공하는 정적인 정보와 커뮤니티를 통해 제공되는 동적인 정보를 구분하여 검색하는 기능이 필요하다.

로컬 웹사이트는 일반적으로 수천에서 수만 개의 웹페이지를 갖고 있으므로 별도의 색인을 구성하지 않고 전문(full text)검색을 수행하며, 색인에 기초하여 단어 및 웹페이지의 빈도에 따라 적합도를 계산하는 Jaccard's Score를 적용할 수 있다[Chen et al., 1998]. 사용자가 k개의 질의어를 입력한 경우 웹페이지 i의 적합도 d_i는 다음과 같이 계산된다.

$$d_i = \sum_{j=1}^{k} tf_{ij} \times \log\left(\frac{N}{df_j}\right) \div k$$

여기서, tf_{ij}는 웹페이지 i에서 용어 j가 나타나는 빈도(term frequency)이며, N은 데이터베이스에 저장된 웹페이지의 총 개수, df_j는 용어 j를 포함하고 있는 웹페이지의 개수이다.

3. 시스템 구현

3.1 시스템 구현 환경

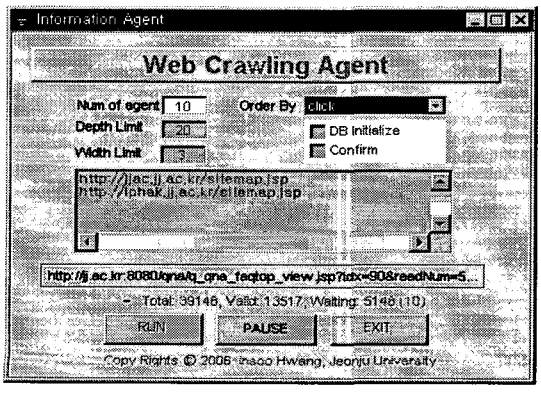
본 연구에서는 <그림 1>에서 제시한 정보검색 에이전트를 개발하기 위해 웹사이트와 데이터베이스 서버를 독립적으로 운영하는 3계층(tiers) 방식을 사용하였다. 웹사이트는 후지쯔의 Primergy 230 Dual Processor에서 마이크로

소프트 윈도우즈 서버를 운영체제로 하고, 웹사이트 프로그램은 Apache Tomcat 4.1을 사용하였다. 데이터베이스는 유닉스를 운영체제로 하는 HP 서버에서 MySQL 5.0을 사용하였다. 또한 웹사이트와 데이터베이스 서버는 JDBC를 이용하여 연동하였다.

로컬 웹사이트의 각 웹사이트에 존재하는 웹페이지를 탐색하여 정보를 추출한 후 다음 탐색을 위해 새로운 링크를 발견하기 위한 웹 프로그램은 기종간의 호환성이 최대한 보장될 뿐만 아니라, 자바빈즈(Javabeans)를 이용하여 컴포넌트 기반의 프로그래밍이 가능한 JSP로 작성하였다. 이것은 자바를 이용한 프로그래밍의 MVC(Model-View-Control)모델을 준수한 것이다.

3.2 정보수집 및 검색

인터넷 웹페이지를 방문하여 정보를 수집하는 정보수집 에이전트는 자바의 스윙(swing)을 이용하여 윈도우즈 애플리케이션으로 개발하였다. <그림 5>에서 보는 바와 같이, 최대로 생성할 에이전트의 개수, 클릭횟수를 제한하는 깊이(depth), 그리고 동적 웹페이지의 런(run) 크기를 제한하는 너비(width) 등을 설정할 수 있다.

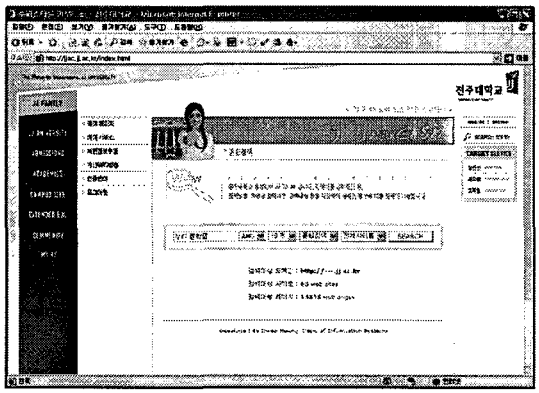


〈그림 5〉 정보수집을 위한 에이전트

방문할 웹페이지가 여러 개인 경우 방문할 순서를 지정할 수 있는데, 클릭횟수를 디폴트로 하고 있다. 가운데 부분의 텍스트 영역에는 방문을 시작하는 URL을 지정하는데, 여러 개의 URL을 동시에 지정하는 것도 가능하다.

윈도우의 하단부에는 방문한 총 웹페이지의 개수, 검색에 유효하게 사용될 수 있는 웹페이지의 개수, 그리고 방문을 기다리는 웹페이지의 개수가 나타난다. 여기서, 유효한 웹페이지의 개수가 총 웹페이지보다 작은 것은 존재하지 않는 링크, 내용이 없이 제어목적으로 사용하는 웹페이지, 검색에 사용할만한 내용을 갖지 않은 웹페이지, 그리고 검색의 효율성을 제고하기 위해 중복된 내용을 갖는 웹페이지 등을 제외시켰기 때문이다.

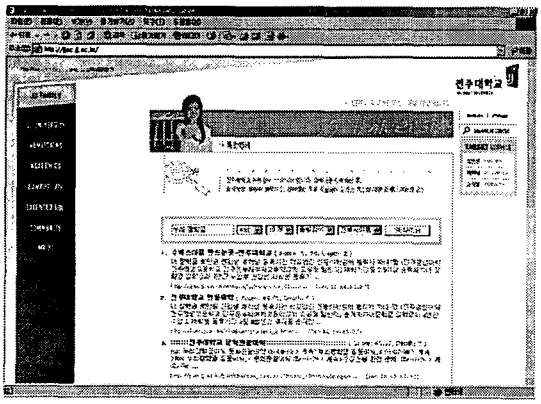
<그림 6>은 정보검색을 위해 질의어를 입력하며 관련 옵션을 설정하는 화면의 예로서 총 63개의 웹사이트에 존재하는 13,555개의 웹페이지를 검색한다.



〈그림 6〉 정보검색을 위한 검색화면

<그림 6>에서 검색어로 "누리 장학금"을 입력한 경우 검색결과를 Jaccard's Score에 따라 정렬하면 <그림 7>과 같다. 첫 번째 결과는 대학 웹사이트에 수록되어 있는 교내장학금에 대한 정보이고, 두 번째 결과는 단과

대학 웹사이트에서 제공하는 유사한 정보이며, 세 번째 결과는 단과대학의 학사 Q/A의 결과이다.



〈그림 7〉 사이트 통합검색의 결과

이와 같이, 로컬 웹사이트의 통합검색은 한 번의 검색으로 관련 사이트를 모두 검색할 수 있다. 뿐만 아니라, 앞에서 기술한 바와 같이, 검색조건에서 대학사이트 혹은 입시사이트만을 지정하거나, 게시판을 제외한 웹페이지만을 지정하여 검색할 수도 있다.

3.3 정보검색의 성과 평가

정보검색의 성과를 측정하는 방법으로 가장 많이 사용되고 있는 것은 사용자가 제시한 질의어에 대해 웹문서에 나타난 용어상의 일치도에 따라 계산되는 재현율(recall)과 정확도(precision)를 사용한다. 그러나 본 연구는 로컬에 존재하는 모든 문서를 수집하여 검색하는 것을 목적으로 하기 때문에 재현율은 성과지표로서의 의미를 갖지 못하게 된다.

이에 따라, 본 연구에서는 정확도만을 성과평가지표로 사용하였으며, 정확도를 평가하는 한 가지 기준으로서 검색된 결과에 대한 클릭률을 사용하였다. 본 연구의 대상이 대학의 웹사이트로서 대학의 웹사이트는 요일별로 접속률이 현저히 변화될 뿐만 아니라, 아직까지 통합검색엔진의 사용이 활성화되지 않았기 때문에, 2006년 4월 1일부터 7일까지 1주일동안에 발생한 검색엔진의 사용빈도 및 클릭율을 조사하여 분석하였다.

조사기간동안 발생한 총 검색횟수는 173회로서 검색결과중에서 하나 이상의 문서를 클릭한 횟수는 156회로서 클릭률은 약 90.2%로 나타났다. 클릭률이 높을수록 검색하고자 하는 정보를 담고 있을 가능성이 높은 것은 사실이지만, 이 수치가 검색의 정확도를 의미하기에는 부족함이 있는 것으로 판단된다. 이에 따라, 향후 연구에서는 새로 구축된 웹사이트에 대한 설문조사에 검색엔진의 성과를 평가하기 위한 설문을 추가하여 사용자들의 주관적인 평가를 실시할 예정이다.

3.4 웹사이트의 유형분석

정보검색 에이전트가 수집한 웹페이지의 링크정보를 <그림 4>의 상태천이도에 적용하면, 웹사이트를 몇 가지 유형을 분류할 수 있다. 즉, 정적페이지와 동적페이지로 결정되는 웹사이트의 깊이와 너비에 따라 정보제공형 웹사이트와 커뮤니티형 웹사이트 등으로 구분할 수 있다.

(1) 정보제공형

강의 콘텐츠를 제공할 목적으로 운영되고 있는 A 웹사이트의 상태천이도는 <표 2>와 같다. 이 사이트는 계층적 구조를 갖는 메뉴 아래 정적인 웹페이지들이 링크를 따라 연결되어 있는 구조로서, 콘텐츠에 이르기까지 클릭수가 많으며, 상호작용이 없이 단방향으로 정보를 제공하는 특징을 갖는다.

〈표 2〉 정보제공형 웹사이트의 상태천이 예

깊이 \ 너비	0	1	2	3
100	1	-	-	-
200	3	0	-	-
300	14	0	0	-
400	216	0	0	0
500	13	0	0	0
600	53	0	0	0
700	14	0	0	0

(2) 커뮤니티형

대학에서 누리(NURI)사업을 수행하면서 학생들에 대한 공지사항과 질의응답 등 의사소통을 위해 운영되고 있는 B 웹사이트의 상태천이도는 <표 3>과 같다. 이 사이트는 정적인 웹페이지보다는 동적인 웹페이지를 중심으로 구성되어 있는 구조로서, 쌍방향의 의사소통이 이루어지는 커뮤니티 형 구조이다.

〈표 3〉 커뮤니티형 웹사이트의 상태천이 예

깊이 \ 너비	0	1	2	3
100	1	-	-	-
200	7	26	-	-
300	14	70	68	-
400	1	51	642	358
500	1	0	297	791
600	0	0	0	126

(3) 혼합형 또는 포털형

포털(portal)형으로 설계된 C 웹사이트의 상태천이도는 <표 4>와 같다. 이 사이트는 하나의 웹페이지가 다수의 링크를 갖는 특징이 있으며, 최종 정보에 접근하기까지 클릭의 수를 최소화하는 구조를 갖는다. 또한, 상태천이도의 각 셀이 모두 채워진 밀집된 형태를 갖는 특징이 있다.

〈표 4〉 혼합형 웹사이트의 상태천이 예

깊이 \ 너비	0	1	2	3
100	1	-	-	-
200	151	51	-	-
300	353	358	884	-
400	441	388	1410	1907

4. 결론 및 향후 연구계획

인터넷이 전 세계적으로 누구나 공유하는 거대한 데이터베이스로 등장하면서, 원하는 정보를 효율적으로 검색하는 방법에 대한 관심이 증대되어 왔다. 많은 검색엔진이 상용 서비스되고 있으나, 대학이나 대기업 등에서 운영하는 다수의 웹사이트를 통합 검색하는 방안에 대한 연구는 미진한 상태이다.

이에 따라, 본 연구에서는 로컬 웹사이트를 통합 검색하기 위해 요구되는 탐색전략, 정보추출규칙, 그리고 정보검색전략에 대해 기술하였다. 본 연구의 결과로 개발된 검색엔진은 J 대학의 웹사이트에서 서비스되고 있다. 또한 본 연구에서 제안한 웹페이지의 상태천이도를 이용하여 웹사이트를 분류한 결과, 깊이와 너비에 따라 정보제공형, 커뮤니티형, 그리고 혼합형 등으로 구분할 수 있었다.

본 연구의 한계점으로는 조직내부에 존재하는 다양한 형태의 파일을 모두 고려하지 못하고 있다는 점과 로컬검색엔진의 성과를 평가함에 있어서 사용자들의 주관적인 평가를 포함하지 못했다는 점이다. 이에 따라, 향후의 연구에서는 콘텐츠관리시스템의 개념을 도입하여 보다 양한 정보를 수집하여 관리하는 방법과 함께 수집된 웹문서들을 자동으로 분류함으로써 디렉터리 검색을 가능하게 하는 연구를 수행하고자 한다.

참 고 문 헌

[1] 김성진, 이상호, "웹 로봇 구현 및 한국 웹 통계보고", *정보처리학회논문지C*, 제10권 4호, 2003.

[2] 김성진, 이상호, 방지환, "페이지랭크 알고리즘 적용을 위한 구현 기술", *정보처리학회논문지D*, 제9권 5호, 2002.

[3] 김영자, 김현주, 배종민, "구조기반검색을 위한 색인구조에 대한 분석", *한국멀티미디어학회 논문지*, 제7권 5호, 2004, pp. 601-616.

[4] 김은정, 배종민, "XLinks를 이용한 하이퍼텍스트 검색시스템", *정보처리학회논문지D*, 제8권 5호, 2001.

[5] 박상위, 오정석, 이상호, "메타 검색엔진을 위한 HTML 문서변경탐지기의 설계 및 구현", *정보처리학회논문지D*, 제9권 3호, 2002.

[6] 이동원, 현순주, "Hyperlink 구조와 Hypertext 분류방법을 이용한 Web Crawler", *한국정보처리학회 춘계학술발표논문집*, 제9권 1호, 2000.

[7] 이말례, "인터넷에서 정보서비스를 위한 검색시스템", *한국멀티미디어학회 논문지*, 제4권 1호, 2000, pp. 29-36.

[8] 이분녀, 김동규, "압축된 인덱스 자료구조를 위한 구축 및 검색알고리즘의 성능 분석", *한국멀티미디어학회 춘계학술발표대회논문집*, 2004, pp. 640-643.

[9] 최중민, "에이전트의 개요와 연구방향", *정보과학회지*, 제15권 3호, 1997, pp. 7-16.

[10] 최중민, "인터넷 정보추출 에이전트", *정보과학회지*, 제18권 5호, 2000, pp. 48-53.

[11] 황인수, "적응적 탐색기법을 이용한 인터넷 정보추출 에이전트의 설계 및 구현", *산경논총*, 제21권 2호, 2003, pp. 241-251.

[12] 황인수a, "정보검색에서 웹마이닝을 이용한 동적인 질의확장에 관한 연구", *Journal of Information Technology Applications and Management*, 제11권 2호, 2004, pp. 227-237

[13] 황인수b, "웹의 연결구조와 웹페이지의 적합도를 이용한 효율적인 인터넷 정보추출", *Journal of Information Technology Applications and Management*, 제11권 4호, 2004, pp. 49-60.

[14] Arnold, S., "Recent Trends in Enterprise Search", http://www.cmswatch.com/Feature/130-Search-Marketplace, 2005.

[15] Chen, H., Chung, Y., Ramsey, M., and Yang, C., "An Intelligent Personal Spider Agent for Dynamic Internet/Intranet Searching", *Decision Support Systems*, Vol. 23, No. 1, 1998, pp. 41-58.

[16] Cho, J., Garcia-Molina, H., and Page, L., "Efficient Crawling Through URL Ordering", *Proceedings of the Seventh International Web Conference*, 1998.

[17] Cho, J. and Gracia-Molina, H., "Parallel Crawlers", *26th Conference on VLDB*, 2002, pp. 124-135.

[18] George C., Healey, J., McHugh, M., and Wang, L., *Mining the World Wide Web : An Information Search Approach*, Kluwer Academic Publishers, 2000.

[19] Jennings, N., Sycara, K., and Wooldridge, M., "A Roadmap of Agent Research and Development", *Autonomous Agents and Multi-Agent Systems*, Vol. 1, 1998, pp. 7-38.

[20] Page, L., Brin, S., Motwani, R., and Winograd, T., "The PageRank Citation Ranking : Bring Order to the Web", *Technical Report*, Stanford University,

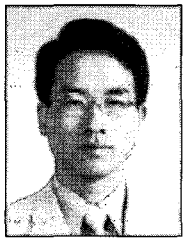
Stanford, CA, 1998.

[21] Rolleke, T., Tsikrika, T., and Kazai, G., "A General Matrix Framework for Modelling Information Retrieval", *Information Processing and Management*, Vol. 42, 2006, pp. 4-30.

[22] Spink, A., Park, M., Jansen, B., and Pedersen, J., "Multitasking During Web Search Sessions", *Information Processing and Management*, Vol. 42, 2006, pp. 264-275.

[23] Vechtomova, O. and Karamuftuoglu, M., "Elicitation and Use of Relevance Feedback Information", *Information Processing and Management*, Vol. 42, 2006, pp. 191-206.

□ 저자소개



황 인 수

전주대학교 정보기술공학부 정보시스템 전공의 부교수로 재직 중이다. 고려대학교 경영학과를 졸업하고 동 대학원에서 경영정보시스템을 전공하여 석사 및 박사학위를 취득하였으며, 산업연구원(KIET) 물류·유통연구센터의 연구원을 역임하였다. 주요 관심분야는 e-Business, CRM, 데이터마이닝, 웹 에이전트 등이다.

◈ 이 논문은 2006년 1월 19일 접수하여 1차 수정을 거쳐 2006년 4월 19일 게재확정되었습니다.