Journal of the Institute for Engineering and Technology
Jeonju Univ. Vol. 4, No. 2, pp.23~31, 1998.

Java를 이용한 인터넷 정보검색 엔진의 설계 및 구현

조 동 영*, 이 원 희**

Design and Implementation of the Internet Information Search Engine Using Java

Dong-Young Cho, Won-Hee Lee

Abstract

An information search engine is a useful internet tool that helps to retrieve information sites quickly and it decreases unnecessary information navigations. Performance of an search engine can be evaluated by measuring speed of information search and effectiveness of retrieved results. Particularly, since the information sites in the internet are frequently changed through time, it is important that a search engine should consider this factor. Because a bad-link in a search engine means unnecessary navigations through the internet, it is necessary to develop an efficient search engine with a low bad-link rate.

In this paper, we design and implement a prototype search engine, RONY, using Java, an object-based internet language. We compare RONY with some other existing search engines that have been currently developed. The engine implemented in our project uses a keyword search method and provides a simple user interface for user's convenience. The research shows that RONY's search results have a comparatively low bad-link rate.

1. 서 론

최근 멀티미디어 기반의 Web 서비스를 중심으로 하는 인터넷의 사용이 일반화되면서 우리는 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 것이 중요한 문제로 대두되고 있다. 즉, 인터넷 활용의 핵심은 인터넷을 통해 자신이 필요로 하는 적절한 정보를 신속하게 검색하는 것이며, 이를 위해서는 인터넷의 수많은 정보들로부터 자신이 필요로 하는 적절한 정보들을 선택적으로 탐색해주는 수단이 필요한데, 이것이 바로 인터넷 정보검색 엔진이다

* 전주대학교 컴퓨터공학과 조교수
** 전주대학교 컴퓨터공학과 석사과정

[2].

정보검색에 대한 연구는 인터넷의 출현 이전에도 이미 데이터베이스, 문헌정보 관리 등의 분야에서 있어 왔으며, 최근에는 인터넷상에서의 필요 정보사이트를 검색해주는 검색엔진이 많이 출현하고 있다[2]. 인터넷 검색엔진의 성능은 검색엔진 출현 초기에는 검색된 데이터 양과 검색속도에 의해 평가되었지만 인터넷상의 정보량이 방대해지고 변화 사이클이 짧아진 최근에는 검색결과의 유효성과 신뢰성으로 평가된다. 따라서, 최근에는 이러한 문제들을 부분적으로 해결하기 위해 인터넷 검색의 범위를 개인 또는 특정 전문분야로 한정하는 검색엔진들의 필요성이 대두되고 있으며, 이의 연구가 활발히 진행되고 있다[2].

본 논문에서는 분산환경에 적합한 객체기반 언어인 Java를 이용하여 개인 또는 소규모 인트라넷 범위에서 유용하게 사용될 수 있는 프로토타입 수준의 인터넷 검색엔진을 설계하고 이를 구현한다.

2. 인터넷 검색엔진의 고찰

2.1 인터넷 검색엔진의 분류

인터넷 검색엔진들은 검색엔진의 검색목적, 검색방법, 검색대상 등의 기준에 따라 분류할 수 있다.

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

주제 검색엔진은 검색 영역을 주제별로 분할하고, 필요하면 각 주제 영역을 다시 세부 주제 영역으로 분할하여 각 영역별로 검색결과를 제공하는 것으로, 달리 메뉴 검색엔진 또는 디렉토리 검색엔진이라고도 한다.

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 문서의 URL을 등록해 주어야만 데이터베이스가 갱신되는 기법이다. 에이전트 색인 기법은 웹 로봇이 인터넷상의 웹서버들을 돌아다니며 자신이 방문한 서버들의 URL을 자신의 데이터베이스에 자동으로 등록해 준다.

2.2 검색엔진의 구현기술

(1) 로봇 에이전트

키워드 검색을 제공하는 검색엔진에서 색인데이터베이스 구성을 위해 에이전트 색인 기법을 사용한다면 반드시 로봇 에이전트의 사용이 필요하다. 로봇 에이전트는 자동으로 웹의 하이퍼텍스트 구조를 따라 다니며 문서를 추출하고, 다시 그 HTML 문서에서 참조되는 다른 HTML 문서들을 추출을 순환적으로 수행하는 프로그램으로[13], 달리 완더러(wanderer), 크라우러(crawler), 스파이더(spider)라고도 한다.

로봇은 통계분석, 유지보수, 밀러링, 자원발견, 복합적인 사용 등에 이용된다. 또한 로봇은 네트워크 자원과 서버에 대해 과부하를 발생시킬 수 있으며, 로봇에 의해 생성된 데이터베이스의 갱신에 대한 효과적인 변환 제어 메카니즘의 부재, 로봇의 잘못된 구현으로 인한 네트워크와 호스트의 손상 등의 문제를 발생시킨다.

(2) 색인 데이터베이스 관리

정보엔진의 색인구성은 한 단어(색인어)가 어떤 문서에 출현했는지를 신속하게 알수

있도록 구조화하는 작업으로, 이러한 과정을 색인화(indexing)라고 한다. 대표적인 검색엔진의 색인화 방법에는 비트맵 색인기법(bitmap/Bit Vector indexing), 역파일기법(inverted file indexing), 요약파일 기법(signature file indexing)이 있다.

① 비트맵 색인 : 문서로부터 단어들을 추출해 내면 그 문서는 단어들의 열로 표현할 수가 있다. 단어와 문서로 구성된 배열을 축변환하면 표 1과 같은 배열을 얻을 수 있다.

표 1 축 변환 후 배열(비트벡터)

문서 \ 단어	D_1	D_2	D_3	…	D_n
W_1	0	1	0	…	1
W_2	0	0	1	…	0
W_3	0	1	1	…	0
…	…	…	…	…	…
W_m	1	0	0	…	0

축 변환된 배열을 저장한 파일을 넓은 의미에서 역파일이라고 부른다. 여기에서 사용되는 이진배열을 비트벡터라고 한다.

② 역파일 색인기법 : 역파일은 다음과 같이 비트벡터에서 표현된 것에서 값이 1인 항목만을 리스트 구조로 표현한 것이다.

$W1=(\cdots,D_n)$
$W2=(D_1,D_3,\cdots,D_n)$
$W3=(D_2,D_3,\cdots)$
$Wm=(D_1,\cdots)$

여기서 영어문서의 'a', 'The', 'of'와 같은 불용어(stopword)를 제거하면 색인의 크기가 급격하게 줄어들게 된다. 다음과 같이 각 단어가 각 문서에서 몇 번씩 나타났는지를 같이 표현하고, 여기에 단어들이 나타나는 위치정보까지 표현하면 보통 상용으로 나오는 검색기에서 사용되는 정보표현을 얻을 수 있다.

$W_1=(\cdots,\{<D_n,1>105\})$
$W_2=(\{<D_1,2>37,56\},$
$\{<D_3,7>10,29,36,89,103,304,356\},\cdots,$
$\{<D_n,4>21,34,71,106\})$
$W_3=(\{<D_2,2>38,280\},$
$\{<D_3,5>29,30,45,50,77\},\cdots)$
$W_m=(\{<D_1,4>15,83,91,123\},\cdots)$

색인구성에서 어려운 문제는 단어추출 문제이다. 여기서 단어는 문서에서 조사나 어미 등을 제외한 순수한 '명사'만을 의미하는데, 명사, 조사, 동사, 어미 등으로 분석하는 형태소 분석은 자연어 처리의 가장 기본적인 작업이면서도 가장 어려운 문제이다.

③ 요약파일 색인기법 : 요약파일 색인기법은 역파일 색인기법의 형태소 분석의 어려움을 해결하기 위해 입력 문서로부터 단어를 찾는 것이 아니라 해싱(hashing) 함수를 통해 고정 길이의 비트열로 변환한다.

- 문장 : 정보검색만을 위해서라면…
- 4 바이트 패턴 : 정보/보검/검색/색만/만을/을…

각 4 바이트 패턴을 해싱함수를 두 번씩 사용해 표현하면 다음과 같다.

정보 : 0010000000…000000000010
보검 : 0001000010…000000000000
검색 : 0000010000…000000000000
색만 : 0000000000…000101000000
만율 : 0100000000…000000010000
위해 : 0000000000…100000000100
…

문서 : 0111010010…100101010110

각 비트패턴은 모두 길이가 일정한 데, 이 일정길이의 비트 패턴들을 패턴의 요약이라고 한다. 문서 안에 들어 있는 모든 패턴의 요약을 모두 중첩시켜(비트별로 OR 연산을 수행함) 같은 길이의 종합적인 패턴을 만들면 그 결과로 나오는 것이 바로 문서의 요약이 된다. 요약파일 색인기법은 자연어에 대한 깊은 이해 없이도 색인을 할 수 있다는 장점이 있다. 그러나 요약파일을 검색할 때에는 있는 것처럼 보이다가 실제로는 없는(false drop) 경우가 발생될 수 있고, 문서에 나타나는 단어들의 분포나 연관관계, 위치 관계를 사용하여 어느 문서가 설정된 질의어 단어에 가장 적합한지를 판단하는 등의 처리가 어렵다는 단점이 있다.

본 논문에서는 이들 기술 중 검색종류는 키워드 검색을 수행하는 형태로 하고, 이를 위해 에이전트 인덱싱 기법을 이용하여 인덱스 데이터베이스를 관리한다. 색인기법은 조사와 어미, 그리고 동사와 불용어 사전을 가지고 비트맵 색인기법을 이용하여 색인어를 추출한다.

3. 검색엔진의 설계 및 구현

3.1 개요

본 연구에서 개발된 로봇 에이전트는 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.

로봇 에이전트는 HTTP 서버의 URL을 중심으로 해당 디렉토리 및 하위 디렉토리의 각 HTML 문서를 대상으로 인덱스 정보를 파싱하고, SQL 서버를 경유하여 Index DB로 저장된다.

검색 에이전트는 ASP(Active Server Page)를 통해 사용자 인터페이스를 제공한다. 사용자가 웹 브라우저를 통해 검색 요청을 하면 검색 에이전트는 ASP를 실행하

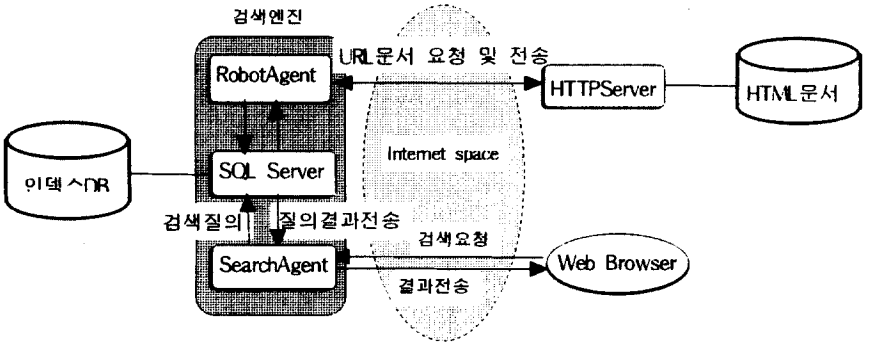

그림 1 로봇/검색 에이전트의 네트워크 환경

여 SQL 질의를 SQL 서버에게 전달한다. SQL 서버는 검색 에이전트가 요청한 질의를 수행하게 되고 수행결과를 다시 검색 에이전트에게 넘겨주게 된다. 검색 에이전트는 넘겨받은 결과를 출력형식에 따라 웹 브라우저로 전달되고 궁극적으로 사용자에게 제시된다(그림 1).

3.2 인덱스 DB와 로봇 에이전트 구조

(1) 인덱스 DB 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 구성되며, 각 테이블의 구조는 다음과 같다.

- KeyWord = (keyWord_ID, keyword)
- URL=(keyWord_ID, url, title, rate)
- Visited_URL=(url)
- Non_Visited_URL=(url)
- Auxil =(auxil)
- StopWord =(stopword)

(2) 로봇 에이전트 구조

인터넷 검색엔진의 로봇 에이전트는 로봇 에이전트 관리자로부터 시드 사이트(seed site)의 URL을 받아 URL의 HTML 문서를 획득하면, 이 문서를 분석하여 이의 결과를 키워드 색인 데이터베이스에 추가하고, 다시 한 URL을 선택하여 같은 과정을 반복해야 한다. 이러한 로봇 에이전트의 URL문서의 문서파싱 과정은 알고리즘 1과 같다.

```
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 에이전트 알고리즘

3.3 구현환경 및 검색사례

(1) 검색엔진의 구현환경

본 논문에서 설계, 구현한 프로토타입 인터넷 검색엔진인 RONY(Robot wONY)는 IBM-PC의 Windows NT Server 4.0 환경에서 구현하였다.

웹 서버는 Windows NT server 4.0에서 제공하는 IIS(Internet Information Server) 3.0를 사용하였고, 키워드 인덱스 데이터베이스의 구성은 관계형 데이터베이스 시스템인 MS-SQL Server 6.5를 사용하였으며, 웹 서버인 IIS와 데이터베이스 시스템인 MS-SQL Server 6.5의 연동은 JDBC를 통

해 구현하였다. 그리고 로봇 에이전트 프로그램은 객체기반 언어인 Java(JDK1.1.2)를 사용하여 구현하였다. 그리고 검색 에이전트 프로그램은 IIS의 ASP를 사용하여 구현하였다.

ASP 파일은 HTML 형식과 유사한 형태를 띄며 ASP 파일 자체가 실행가능 파일이기 때문에 데이터베이스 연결이나 SQL 명령문 수행을 위한 별도의 파일이 필요 없고, 결과를 보여주기 위한 파일 역시 ASP 파일로 작성이 가능하다.

(2) 검색엔진 RONY의 항해

검색엔진 RONY의 구현은 현재 프로토타입 수준으로 구현되었고, 키워드 색인 DB의 구성을 현재 구성중이다. 따라서, 상용의 검색엔진들과 비교해서, RONY의 검색결과는 열악할 수 있는데, 이것은 현재 계속적으로 수행중인 로봇 에이전트의 결과로 확장중인 키워드 색인 데이터베이스가 일정 수준에 이르면 극복될 수 있다.

그림 2과 그림 3은 RONY의 로봇 에이전트의 작업 화면으로, 로봇 에이전트는 로봇 관리자가 제공한 시드 사이트의 HTML 문서원본과 이를 통해 URL 및 키워드를 추출한다. 그림 2은 로봇 에이전트의 초기화면이다. 그리고 그림 3은 로봇 에이전트 초기화면의 주소부분에 임의의 URL을 부여했을 때 파싱을 수행중인 로봇에이전트 화면이다.

그림 4는 RONY의 검색 에이전트의 사용자 인터페이스 화면으로, RONY는 기본적으로 키워드 검색만을 제공한다. 대부분의 검색엔진들의 사용자 인터페이스를 살펴보면, 대부분이 키워드를 입력부분과 주제검색을 위한 하위 분류들이 초기 화면에 같이 제공된다. 이러한 것은 검색방법의 편의성을 제공하지만 키워드 검색만을 위해 검색엔진을 호출할 경우 검색엔진의 불필요한 로딩시간을 소비한다. 따라서 검색엔진 RONY의 사용자 인터페이스는 그림 4과 같이 키워드 검색방법만을 제공하는 것으로 단순화하였다.

그림 5는 사용자가 그림 4의 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 3페이지로 나누어 보여주게 된다.

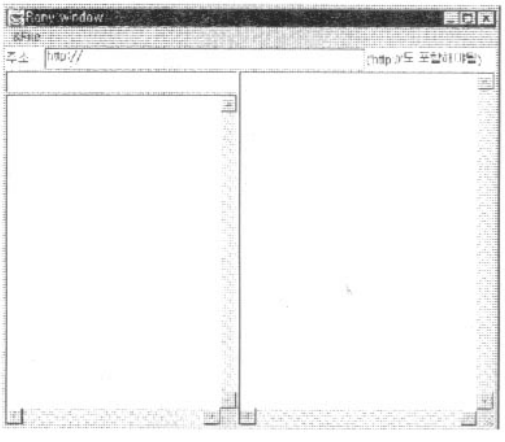

그림 2 로봇 에이전트의 실행 초기화면

그림 3 파싱중인 로봇 에이전트

출력수 등으로 파악할 수 있다. 이러한 특성들을 기준으로 본 논문에서 구현한 검색엔진인 RONY와 상용의 국산 검색엔진들을 비교하면 표 2과 같다.

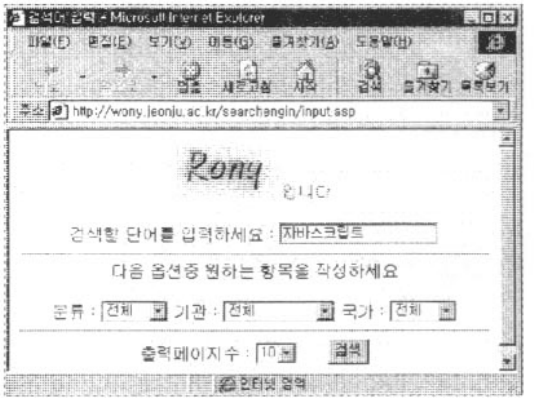

그림 4 검색에이전트 초기화면

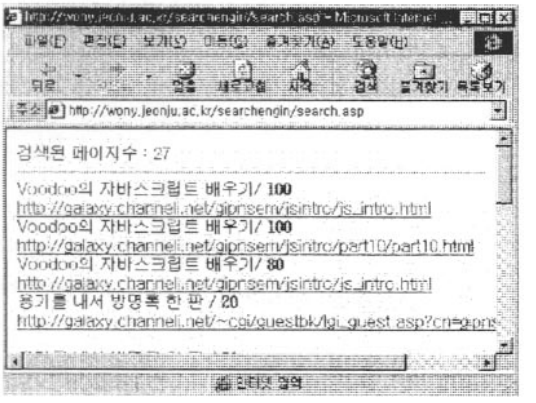

그림 5 "자바스크립트" 검색 결과 화면

4. 검색엔진의 성능 비교

4.1 검색엔진의 특성 비교

검색엔진의 특성은 사용자 인터페이스, 운영환경, 검색할 때의 검색 우선순위, 검색결과 페이지의 요약특성, 검색결과의 페이지

표 2에서 보면, 각 검색엔진의 사용자 인터페이스는 화면 해상도를 "800×600"으로 설정하고, 출력 윈도우의 크기를 최대로 했을 때 초기화면에 나타낼 수 있는 크기이다. 페이지당 출력 항목 수는 검색엔진이 검색

표 2 기존검색엔진과의 비교표

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

결과를 화면에 출력할 때 출력항목의 개수를 나타낸 것으로, 까치네를 제외한 나머지 검색엔진들은 사용자가 선택적으로 구성 가능하다.

4.2 검색결과의 불량링크율 비교

본 논문에서 구현한 프로토타입의 검색엔진인 RONY의 검색성능을 대표적인 국내제작 검색엔진들과 하면, 표 3과 같다. 표 3에서, 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.

검색링크의 불량링크율(%)
= (검색결과에서 불량링크의 개수
÷검색결과의 총 링크 개수)×100

표 3에서, 불량링크율 계산은 각 검색엔진의 검색결과 링크들중에서 해당 문서가 이동되거나 없어진 경우만을 포함하였다. 검색결과의 링크 개수가 검색엔진마다 다르게 나타나기 때문에 300개의 링크를 기준으로 조사하여 계산하였으며, 검색결과의 링크갯수가 300개가 넘을 경우에는 310개만을 조사하여 계산하였다. 표 3에서 보는 바와 같이, 심마니, 까치네, 정보탐정 등 상용의 검색엔진들과 비교해서 본 논문에서 구현한 검색엔진 RONY의 불량 링크율이 매우 낮았다. 이것은 RONY의 색인 DB 구성이 가장 최신으로 구성된 이유가 있다는 점을 고려하더라도 다른 검색엔진과 비교해서 상대적으로 낮은 불량링크율을 갖는다.

5. 결 론

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

표 3 검색 결과 비교 (검색결과의 불량 링크율 비교) (단위:%)

검색 키워드	Rony	심마니	까치네	정보탐정
"자바스크립트"	0	16.23	41	30.97
"내각제"	0	23.23	18.5	31.76
"기업인수"	0	1.8	-	32.74
"명예퇴직"	0	52.26	33.5	31.03
"박찬호"	0	7.42	58	18.39

언어인 Java를 사용하여 인터넷 검색엔진을 설계·구현함으로써 인터넷 검색엔진의 구조를 이해하고, 구현 기술을 확보하였다. 본 논문에서 구현한 검색엔진은 키워드 검색을 제공하며, 사용자 인터페이스를 단순화함으로써 사용자의 편의성을 도모하였다. 그리고 기존의 국내 제작 검색엔진들과 비교해서 검색된 정보사이트의 양이 적은 대신 검색결과의 배드 링크율은 개선됨을 보였다.

향후 검색엔진의 성능을 개선하기 위해서는 효율적인 색인 추출 및 색인 데이터베이스의 구축, 그리고 불량링크의 자동 제거방법, 중복된 URL의 제거방법 등에 대한 연구가 필요하다.

참고문헌

[1] Graham Hamilton, Rick Cattell, Maydene Fisher, "JDBC Database Access with Java:A Tutorial and Annotated Reference", Addison Wesley pub, 1997.

[2] Kartijn Koster, NEXOR "Robots in the web:threat or treat?" connxions, volume9. No4, April 1995

[3] Martijn Koster "The Web Robots pages" http://info.webcrawler.com/mak/projects/robots/

[4] Mark Watson, "Intelligent Java Applications for the Internet and Intranets" Morgan Kaufman Publishers, 1997.

[5] Matthew D. Siple, McGraw Hill, "The Complete Guide to Java Database Programming", Computing McGraw-Hill, 1997.

[6] Nothan Wallace, "Active Server Page HOW-TO," Waite Group Press, 1997.

[7] Ronan Sorensen, "Inside Microsoft Windows NT Internet Development", Microsoft Press, 1998.

[8] Scot Hillier, Paniel Mezick, Dan Mezick, "Programming Active Server Pages", Microsoft Press, 1997.

[9] 고형일 외, "Web 상에서의 정보검색을 위한 에이전트의 설계 및 구현에 관한 연구", 정보과학회 학술발표논문집(II) 제 24권 2호 pp.285-288, 1997.10.

[10] 심해청 외, "효율적인 웹 로봇의 설계 및 구현에 관한 연구", 정보과학회 학술발표논문집(III) 제 24권 2호 pp.465-468, 1998.10.

[9] 백승구 외, "웹과 DB 연동시 CGI모델과 Java모델의 성능평가", 정보화학회 학술발표논문집 제 25권 1호 pp.328-330, 1998.4.

[10] 백광진, 김태윤, "자바에 기반한 웹 정보검색 에이전트의 설계", 정보과학회 학술발표논문집 제 25권 1호 pp.455-457, 1998.4