Java를 이용한 정보 검색 최적화 알고리즘에 관한 연구

김용호* · 정종근** · 이윤배***

A Study on Optimized Information Search Algorithm Using Java

Yong-Ho Kim · Jong-Geun Jeong · Yun-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. 따라서, 인터넷이 보편화되기 이전에는 정보의 습득이 중요한 문제가 되었지만 인터넷의 사용이 일반화되고 있는 현대의 사회에서는 정확한 정보를 신속하게 취득하는 것이 중요한 문제로 대두되고 있다. 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 정보 사이트의 양이 적은 대신 검색결과의 배드 링크율은 개선됨을 보였다.

ABSTRACT

As internet use is being generalized central of WWW(World Wide Web) service of multimedia based recently, we could acquire many informations that exist to all over the world's computer network. Therefore, picking up of information became important problem before that internet is generalized, but it is risen to important problem to acquire correct information rapidly on modern society that use of internet is generalized. This paper designed internet search engine and understand structure of internet search engine drawing URL that is optimized, and secure embodiment technology using Java that is language of object base. Search engine that proposed in this paper maintained user's the convenience by offer keyword search, and simplify user interface. And although quantity of searched information site is few, search engine show that the bad link rate of searched result is improved compare with existent domestic manufacture search engines.

Ⅰ. 서론

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 것이 중요한 문제로 대두되고 있다. 인터넷을 통해 필요한 정보를 신속하게 취득할 수 있는 환경을 갖게 되었지만 웹서비스의 보편화로 정보제공자들이 많아지면서 인터넷에서의 정보항해(information navigation)에 소요되는 시간이 점차 늘어나고 있다[2,6,7]. 따라

* 조선대학교 전자계산과 대학원
** 동강대학 전자정보과
*** 조선대학교 컴퓨터공학과
접수일자 : 2002. 10. 15

서, 전 세계에 퍼져있는 인터넷 정보제공자들로부터 자신에게 유효한 정보만을 선별적으로 취득할 수 있도록 함으로써 인터넷의 정보항해 시간을 줄이는 문제가 중요하게 대두되고 있다. 즉, 인터넷의 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 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 링크의 자동 제거하며, 중복된 URL을 제거하는 알고리즘을 연구하였다.

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

인터넷 검색엔진들은 검색엔진의 검색목적, 검색방법, 검색대상 등의 기준에 따라 분류할 수 있다. 각 기준에 따른 검색엔진의 분류는 다음과 같다.

(1) 검색목적에 의한 분류

인터넷 검색엔진의 검색목적에 의한 분류는 검색엔진의 결과로 얻어지는 결과정보에 이용목적을 기준으로 하는 것으로, 크게 단순검색(hit search) 엔진과 전략검색(strategy search) 엔진으로 구분할 수 있다. 단순 검색엔진은 사용자의 요구에 부합하는 정확한 한 정보를 검색하는 것을 목적하며, 전략 검색엔진은 사용자가 검색엔진의 결과로 얻어지는 정보의 획득과 이의 가공을 통해 다른 경쟁자들과 비교해서 정보우위를 확보할 수 있도록 사용자의 요구에 부합하는 유용한 정보들을 검색한다.

(2) 검색방법에 의한 분류

인터넷 검색엔진의 검색방법에 의한 분류는 검색엔진이 사용자의 검색요구를 수용하는 방법을 기준으로 하는 것으로, 크게 주제(subject) 검색엔진과 키워드(keyword) 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 사용자는 주제 검색엔진이 단계적으로 제공하는 주제영역 분류 화면에서 자신이 검색하고자 하는 주제를 선택하면 최종적으로 검색결과를 얻을 수 있으므로, 특별히 검색결과에 대한 사전지식이 없이 단지 검색주제의 영역만 구분하면 검색이 가능하게 된다. 그러나 주제 검색엔진은 일반적으로 여러 단계에 걸쳐 검색 주제영역을 선택해야 하며, 한 번 검색영역을 잘못 선택하게 되면 결과적으로 부적합한 검색결과를 초래하게 되는 단점이 있다.

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 문서들을 추출을 순환적으로 수행하는 프로그램으로[6], 달리 완더러(wanderer), 크라

우러(crawler), 스파이더(spider)라고도 한다.

HTTP는 웹 공간의 중심핵이라 할 수 있는 프로토콜이며, 웹 공간에서 웹 브라우저, 검색엔진의 로봇 등의 에이전트가 활동할 수 있도록 해준다[3]. 최근에는 웹 에이전트의 붐과 지능형 에이전트의 요구와 더불어 상호 연동 가능한 여러 에이전트 시스템 간의 공통된 기술의 사양을 표준화하고 구체적 협력을 도모하려는 움직임이 나타나고 있다[3].

(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 비트맵 색인기법(bitmap indexing), 역파일기법(inverted file indexing), 요약파일 기법(signature file indexing)이 있다.

III. 정보 검색 최적화 알고리즘

3.1 개요

본 논문에서는 검색엔진을 통해 얻어온 정보들에서 중복된 링크와 잘못된 링크에 대한 정보를 제거하여 질의에 최적화된 정보를 추출하는 알고리즘을 제안한다. 이는 로봇 에이전트가 하이퍼텍스트 구조를 따라 다니며 문서를 추출하고, 그 문서에 포함된 참조되는 다른 문서들을 추출하는 점에 착안하였다. 일반적으로 각 홈페이지는 관련 링크를 포함하고 있으며, 이 링크들은 해당 홈페이지와 전혀 다른 내용이 아니라 서로 연관성이 있는 내용을 지니고 있는 경우가 크기 때문이다. 중복된 링크와 잘못된 링크를 제거한 정보들은 사용자의 의사 결정에 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 알고리즘은 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.

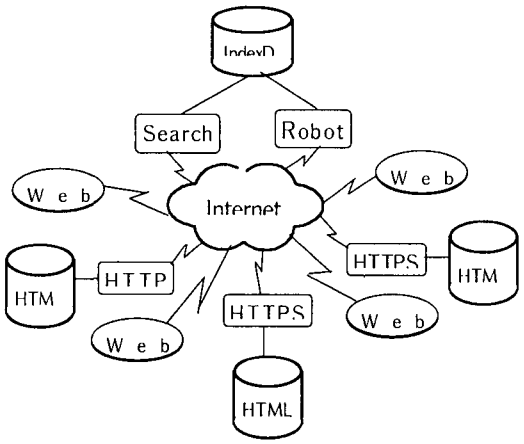

그림 1. 로봇/에이전트의 네트워크 환경

Fig 1. Network environment of robot/agent

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.

검색 에이전트(Search Agent)는 ASP(Active Server Page)를 통해 사용자 인터페이스를 제공한다. 사용자가 웹 브라우저를 통해 검색 요청을 하면 검색 에이전트는 ASP를 실행하여 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. 이를 그림으로 표현하면 [그림 2]와 같다.

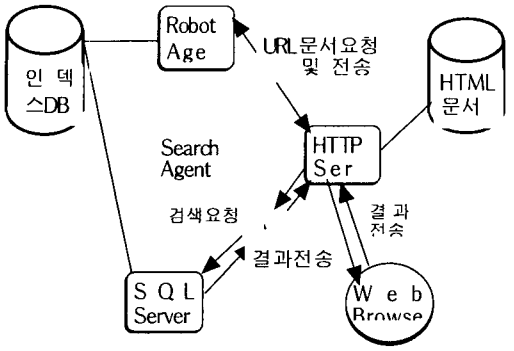

그림 2. 검색 과정
Fig 2. Searching process

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 구성되며 이들 테이블 각각에 대한 구조는 다음과 같다.

① 키워드 테이블
- KeyWord 필드와 URL 테이블을 참조하기 위한 KeyWord_ID를 포함한다.

② URL 테이블
- KeyWord 테이블과 같은 KeyWord_ID를 포함하여 해당 키워드의 URL과 URL의 빈도수를 포함한다.

③ 방문한 URL 정보 테이블
- 로봇 에이전트가 이미 방문하여 파싱이 끝난 URL 정보를 포함한다.

④ 아직 방문하지 않은 URL 정보 테이블
- 로봇 에이전트가 URL은 추출해 놓았으나 아직 방문하지 않은 URL정보를 포함한다.

⑤ 조사 테이블
- 띄어쓰기 단위의 끝에 오는 조사와 어미 사전정보를 포함한다.

⑥ 불용어 테이블
- 동사와 불용어 사전을 포함한다.

(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. 이러한 로봇 에이전트의 URL문서의 문서파싱(parsing) 과정은 [그림 3]과 같다.

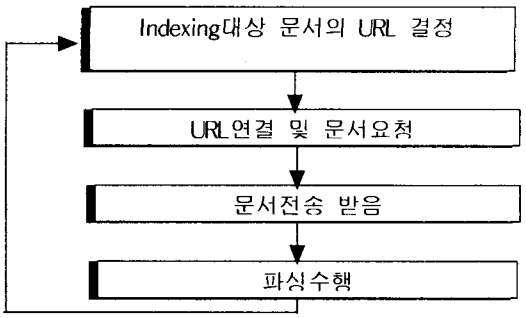

그림 3. 파싱
Fig 3. Parsing

(3) 로봇 에이전트와 데이터베이스의 연동 구조

본 논문의 구현에서 로봇 에이전트와 데이터베이스의 연동은 JDBC(Java DataBase Connectivity)를 이용한다.

3.4 구현환경 및 검색 예

(1) 검색엔진 RONY의 구현환경

본 논문에서 설계, 구현한 검색엔진은 IBM PC (Pentium-IV1.5GHz)의 Windows 2000 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.

ASP 파일은 HTML 형식과 유사한 형태를 띄며 ASP 파일 자체가 실행가능 파일이기 때문에 데이터

베이스 연결이나 SQL 명령문 수행을 위한 별도의 파일이 필요 없고, 결과를 보여주기 위한 파일 역시 ASP 파일로 작성 가능하다.

(2) 검색엔진의 항해

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

[그림 4]와 [그림 5]는 구현한 로봇 에이전트의 작업 화면으로, 로봇 에이전트는 로봇 관리자가 제공한 시트 사이트(seed site)의 HTML 문서원본과 이를 통해 URL 및 키워드를 추출한다. [그림 4]는 로봇 에이전트의 초기화면이다. 그리고 [그림 5]는 로봇 에이전트 초기화면의 주소부분에 임의의 URL을 부여했을 때 파싱을 수행중인 로봇 에이전트의 화면이다.

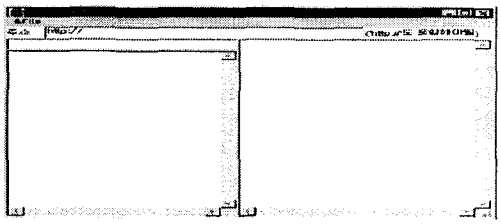

그림 4. 에이전트 초기화면
Fig 4. First Display of agent

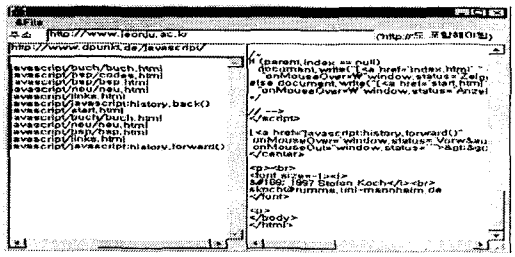

그림 5. 파싱 에이전트
Fig 5. parsing agent

[그림 6]은 로봇 에이전트의 검색 에이전트의 사용자 인터페이스 화면으로, 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 6]과 같이 키워드 검색방법만을 제공하는 것으로 단순화하였다.

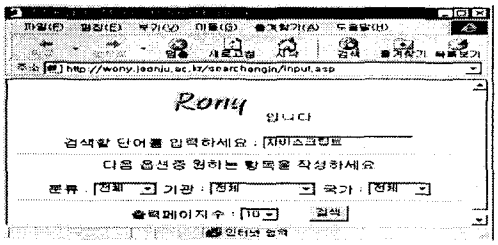

그림 6. 검색에이전트 초기화면
Fig 6. First display of search agent

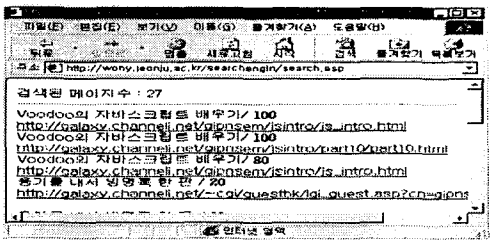

그림 7. 검색 결과 화면
Fig 7. Search result

[그림 7]은 사용자가 [그림 6]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.

IV. 성능 비교 · 분석

이 장에서는 본 논문에서 제안한 검색엔진과 최적화 알고리즘의 특성을 대표적인 국내제작 검색엔진

과 비교 분석한다.

4.1. 검색엔진의 특성 비교

검색엔진의 특성은 사용자 인터페이스, 운영환경, 검색할 때의 검색 우선순위, 검색결과 페이지의 요약 특성, 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	S	K
사용자 인터페이스	0.5	2	1
운영체제	Windows 2000	Solalis 2.5	-
우선순위	Title	Reviewed Ranking	title
페이지 요약	Title	동일개념 단어의 출현빈도가 높은 문장	검색어의 출현빈도가 높은 문장
페이지당 출력 항목수	10 (선택가능)	10 (선택가능)	10

4.2. 검색결과의 불량링크율 비교

본 논문에서 구현한 프로토타입의 검색엔진의 검색성능을 대표적인 국내제작 검색엔진들과 하면, [표 2]와 같다. [표 4]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 링크 개수가 검색엔진마다 다르게 나타나기 때문에 300개의 링크를 기준으로 조사하여 계산하였으며, 검색결과의 링크개수가 300개가 넘을 경우에는 310개만을 조사하여 계산하였다. 표 4에서 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 4] 검색 결과 비교(검색결과의 불량 링크율 비교)

(단위: %)

검색 키워드	제안한 검색엔진	심마니	까치네
"자바스크립트"	0	16.23	41
"내각제"	0	23.23	18.5
"기업인수"	0	1.8	-
"명예퇴직"	0	52.26	33.5
"박찬호"	0	7.42	58

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 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 개발

하는 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 배드 링크율은 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 검색엔진의 성능을 개선하기 위해서는 효율적인 색인 추출 및 색인 데이터베이스의 구축, 그리고 불량링크의 자동 제거방법, 중복된 URL의 제거 방법 등에 대한 연구가 필요하다.

참고문헌

1. 김수동, "Java기반 인터넷 어플리케이션 아키텍처 및 설계 기법", 정보과학회지 제16권 4호 pp.9-15, 1998. 4.
2. 신봉기, 김영환, "인터넷 정보검색 서비스 동향," 정보과학회지, 정보과학회, 16권 8호, pp.16~20, 1998. 8.
3. 신봉기, 김영환, "웹 에이전트", 정보과학회지 제15권 제3호 pp.61-68, 1997. 3.
4. 오종인 외, "사용자 중심의 웹 정보검색 시스템 설계," 정보과학회지, 정보과학회, 24권 1호, pp. 425~428, 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(공학사)

1993년 경남대학교 전자계산과 대학원(공학석사)

1999년 - 2002년 현재 조선대학교 대학원 전자계산학과 박사수료

※관심분야: 멀티미디어, 전자상거래, 검색엔진

정종근(Jong-Geun Jeong)

1995년 조선대학교 전자계산학과 졸업(이학사)

1997년 조선대학교 대학원 전자계산학과 졸업(이학석사), 2002년 8월 조선대학교 대학원 전자계산학과 대학원 이학박사

1999년 3월-2002년 현재 동강대학 전자정보과 겸임교수

※관심분야: 인공지능, 검색엔진, 데이터베이스, 정보보안, 전자상거래, 바이러스

이윤배(Yun-Bae Lee)

1980년 광운대학교 전자계산학과 졸업(이학사)

1983년 광운대학교 대학원 전자계산학과 졸업(이학석사), 1993년 숭실대학교 대학원 전자계산학과 졸업(공학박사)

1988년 4월-현재 조선대학교 컴퓨터공학부 교수

1999년 7월-2000년 현재 광주광역시 시정정책자문회의 위원

※관심분야: 인공지능, 전문가시스템, 멀티미디어, 데이터베이스, 정보보안, 바이러스