퍼지 및 지능시스템학회 논문지 2007, Vol. 17, No. 6, pp. 849-854

URL 패턴 스크립트를 이용한 효율적인 웹문서 수집 방안

A Method of Efficient Web Crawling Using URL Pattern Scripts

장문수 · 정준영
Moon-Soo Chang and June-Young Jung

서경대학교 소프트웨어학과

요 약

수많은 웹문서 중에서 원하는 문서만을 수집하는 것은 쉽지 않다. 이것을 해결하는 한 방법은 원하는 분야의 정보를 많이 제공하는 사이트에서 원하는 부분만 골라서 수집하는 것이다. 본 논문에서는 웹사이트의 URL 패턴을 XML 기반의 스크립트로 정의하여, 필요한 웹문서만을 지능적으로 수집하는 방안을 제안한다. 제안하는 수집 방안은 데이터베이스와 같은 구조화된 자료를 정보로 제공하는 사이트에 대해서 매우 빠르고 효율적으로 적용될 수 있다. 본 논문에서는 제안하는 방법을 적용하여 5만개 이상의 웹 문서를 수집하였다.

키워드 : 웹 크롤링, URL, 패턴 스크립트, URL 필터링

Abstract

It is difficult that we collect only target documents from the innumerable Web documents. One of solution to the problem is that we select target documents on the Web site which services many documents of target domain. In this paper, we will propose an intelligent crawling method collecting needed documents based on URL pattern script defined by XML. Proposed crawling method will efficiently apply to the sites which service structuralized information of a piece with database. In this paper, we collected 50 thousand Web documents using our crawling method.

Key Words : Web Crawling, URL, Pattern Script, URL Filtering

1. 서 론

인터넷 사용이 일상화되면서 우리가 접하는 대부분의 정보가 웹에서 제공되고 정보를 찾기 위해 가장 먼저 찾는 것이 웹 검색이 되었다. 이로 인해 웹문서가 엄청난 양으로 늘어나게 되어 검색엔진으로 모든 문서를 검색할 수 없을 만큼 웹이 방대해지고 있다. 검색엔진에서 문서를 검색할 수 있는 것은 웹 크롤러(web crawler)가 웹 상의 모든 문서를 수집하여 검색엔진을 위해 제공하기 때문이다. 그러나 이제는 하나의 웹 크롤러가 모든 웹문서를 수집할 수 없게 되었다.

그리고 일상생활에서 웹을 이용할 뿐만 아니라 여러 분야의 연구에서 사용하는 정보도 웹에서 검색되고 있다. 따라서 특정한 목적에 맞는 정보를 가진 문서를 웹에서 대량으로 수집하는 사례가 늘어나고 있다. 그 한 예로 시맨틱 웹(Semantic Web)[1]의 지능을 부여하는 온톨로지를 구축하는데 있어서도 웹정보를 수집하여 온톨로지에 활용하는 연구도 진행되고 있다.

본 논문에서는 특정 분야의 웹문서를 수집하기 위한 새로운 웹 크롤링(web crawling)기법을 제안한다. 기존 크롤링 기법들은 문서를 모으는 것 자체가 목표였기 때문에 웹의 모든 링크를 추적하여 문서를 수집하였다[2]. 본 논문에서는 필요한 분야의 정보가 많이 있는 사이트를 조사하여 그 사이트에서 효율적으로 정보를 수집하는 크롤링 기법을 개발하고자 한다. 제안하는 방법은 대용량의 데이터베이스를 가지고 웹서비스를 제공하는 웹사이트들을 대상으로 사이트의 컨텐츠를 가리키는 주소인 URL(Universal Resource Locator)의 패턴을 이용하여 스크립트를 작성하고 스크립트에 의해서 필터링되는 문서만을 수집한다.

그리고 웹 크롤링은 일반적으로 접속 제한을 하지 않으면 과도한 접속 시도로 자료를 제공하는 웹서버의 트래픽을 증가시키고 불필요한 정보로의 액세스로 인해 해당 서버에 피해를 줄 수 있다. 그러나 스크립트를 이용하여 수집하면 필요 없는 링크로의 액세스가 줄어들기 때문에 일반적인 사용자가 웹 서핑을 하는 수준의 트래픽으로 필요한 문서를 신속하게 수집할 수 있다.

이후 2장에서는 기존의 웹 크롤러와 웹 로봇에 관해 설명한다. 3장에서는 URL 패턴을 이용한 문서 수집 방법과 이를 적용한 수집 시스템을 제안한다. 4장에서는 제안된 방법으로 문서를 수집한 결과를 나타내고, 마지막으로 5장에서는 결론과 향후 계획을 기술한다.

2. 연구 배경

웹 크롤러는 시작 URL로 지정된 인터넷 상의 웹서버를

접수일자 : 2007년 10월 15일
완료일자 : 2007년 11월 24일

접근하여 HTML문서를 읽어와 참조되지 않은 하이퍼링크(HyperLink)를 자동으로 추적하여 원하는 정보를 수집하고 자신의 데이터베이스에 내용을 저장하는 것을 목적으로 한다. 기본적인 웹 크롤러는 그림 1의 알고리즘을 사용하고 있다[3]. 이때 중복하여 수집하지 않도록 참조된 모든 URL을 저장하여 비교한다. 이 과정에서 웹문서 수집기는 웹 페이지의 모든 링크를 접근하면서 불필요한 문서를 수집하기도 한다. 이런 문제를 해결하기 위해 URL의 Ordering을 통한 웹 크롤링 방법[4]과 같이 링크에 가중치를 주어서 중요한 문서를 먼저 수집하는 알고리즘도 개발되었다.

```
Initialize:
  UrlsDone = {}
  UrlsTodo = {"naver.com", ..}

Repeat:
  url = UrlsTodo.getNext()

  ip = DNSLlookup( url.getHostnamer() )
  html = DownloadPage( ip, url.getath() )

  UrlsDone.insert( url )

  newUrls = parseForLinks( html )
  For each newUrl
    If not UrlsDone.contains( newUrl )
    then UrlsTodo.insert( newUrl )
```

그림 1. 일반적인 웹 크롤러 알고리즘.
Fig. 1. The algorithm of general Web crawler.

초기의 웹 크롤러는 정보 검색엔진의 색인 데이터로 사용하기 위하여 최대한 문서를 많이 수집하는 웹로봇으로 발전해왔다. 그 후, 인터넷 정보의 활용도가 커짐에 따라 정보 수집을 위한 웹 크롤링 기술이 요구되어 한 사이트의 모든 문서를 수집하는 기술이 개발되기도 하고, 필요한 분야의 문서를 수집하는 기술이 연구되었다.

웹 크롤러는 일반적으로 문서를 분류할 수 없기 때문에 웹사이트의 모든 페이지를 수집해서 필요한 부분을 추출한다. 그러나 웹 트래픽 양이 증가하고 HTML의 기능이 다양화됨에 따라 모든 웹문서를 수집하는 것은 불가능한 일이 되어 갔다. 본 논문에서는 웹사이트의 모든 링크를 조사하지 않고 필요한 문서의 URL 패턴을 분석하여 필요한 링크만을 크롤링하여 수집하는 방법을 제시한다.

3. URL패턴을 이용한 문서 수집

전체 웹을 수집하는 것이 불가능해질 정도로 엄청나게 많은 양의 웹문서가 인터넷 상에 존재한다. 지금은 모든 웹 문서 중에 원하는 문서를 찾는 것이 아니라 원하는 영역의 문서 중에서 꼭 맞는 문서를 찾는 시대가 되었다. 본 논문에서는 문서 수집을 위하여 원하는 분야에 가장 적합한 웹사이트들을 선정하고, 이 사이트의 문서를 수집한다. 하나의 웹문서 안에는 다른 문서로 연결되는 많은 링크들이 포함되어 있고 이 링크 중에는 불필요한 링크들, 즉 광고 링크, 중복 링크, 주제와 관련 없는 링크들이 존재한다. 본 논문에서는 필요한 문서들의 URL 패턴을 찾아 패턴 스크립트로 기술함으로써 필요한 문서만을 필터링하여 수집한다.

3.1 기본 URL 패턴

국내 대부분의 대용량 데이터베이스를 가진 웹사이트는 카테고리별로 데이터베이스 내용을 분류한다. 데이터베이스의 내용은 웹서버 측의 서버 사이드 페이지(ASP, ASP.NET, JSP, PHP 등)를 이용하여 보여준다. 서버 사이드 페이지는 문서를 동적으로 생성하기 때문에 정적인 웹페이지와 다른 URL 패턴을 가지고 있다.

```
http://www.web.com/detail.aspx?prod_id=100320&vie
w=contents
http://웹사이트주소/서버사이드페이지?변수전달용 쿼리
```

그림 2. 서버 사이드 페이지 URL 패턴.
Fig. 2. URL pattern of server-side-page.

그림 2는 웹사이트에서 사용하는 일반적인 서버 사이드 페이지 URL의 형식과 한 예를 나타낸 것이다. 이 URL의 뒷부분은 변수 전달 쿼리로 구성되는데 여기서 데이터베이스와 연결되는 ID값으로 데이터베이스에 저장된 각각의 웹페이지 정보와 연결된다. 따라서 이 ID값을 이용하면 사이트 내의 문서에 접근할 수 있다. 쿼리에는 상세정보의 ID값 외에도 카테고리의 ID와 한 페이지 내의 특정 정보만을 보여주도록 변수를 전달하는 쿼리가 있다.

상세정보 혹은 카테고리의 ID가 동일하게 된다면 해당 URL에 별도의 쿼리 변수가 존재하더라도 크롤러는 같은 페이지로 인식해 중복하여 수집하지 않도록 해야 한다. 본 논문에서는 이러한 URL 패턴 정보를 패턴 스크립트로 처리함으로써 각 문서에 대한 직접적인 내용 분석을 하지 않고 중복 판별을 가능하게 한다. 그림 3의 두 개의 URL을 수집하여 보면 그림 4의 동일한 페이지가 수집되는 것을 알 수 있다. 서로 다른 링크이지만 실제로 이 링크의 내용을 보면 서로 같은 페이지의 URL인 것이다.

```
http://shopping.naver.com/detail/detail.nhn?cat_id=0
0040102&nv_mid=4045429642

http://shopping.naver.com/detail/detail.nhn?cat_id=0
0040102&nv_mid=4045429642&ani=0&tc=2
```

그림 3. 동일 상세정보에 대해 서로 다른 쿼리.
Fig. 3. The different queries point to the same page.

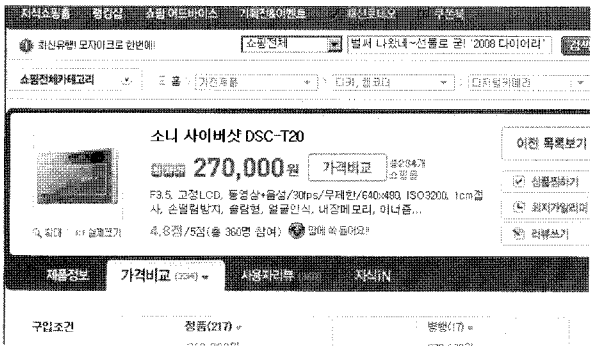


그림 4. 다른 URL을 가진 동일 페이지.
Fig. 4. The page linked two different URL.

3.2 웹페이지 유형

상품 정보나 뉴스와 같이 많은 양의 데이터베이스화된 페이지를 가진 사이트들의 사이트 구성을 분석해 보면 몇 가지 페이지 유형으로 분류할 수 있는데, 일반 페이지, 리스트 페이지, 상세내용 페이지로 구분을 할 수 있다. 일반 페이지는 웹사이트 내에서 다른 메뉴로 가기 위한 중간 단계, 즉 정적인 자료를 서비스하기 위한 페이지들이다. 리스트 페이지는 특정 카테고리의 상세내용 페이지들을 나열하여 보여준다. 그리고 상세내용 페이지는 웹사이트에서 제공하는 주 컨텐츠로, 상품 정보의 경우 표를 이용한 구조문서와 특정한 문자열 패턴으로 표시되는 반 구조 문서로 되어 있다[5]. 본 논문에서 제안하는 웹문서 수집 방법은 이러한 페이지 유형에 따라 다른 URL 패턴 정보를 이용한다.

3.3 URL패턴 스크립트

하나의 웹문서에는 여러 종류의 링크 정보가 있다. 수집에 필요한 페이지로 연결되는 링크뿐만 아니라 사이트 내의 일반 페이지로의 링크나 전혀 다른 사이트로 연결되는 광고 페이지 링크도 존재한다. 또한 자바 스크립트나 웹사이트에서 제공하는 리다이렉션(Redirection) 링크가 있다.

수집 목적에 맞지 않는 불필요한 링크들은 적절한 과정을 통하여 다음 수집 작업에서 제외할 필요가 있다. 또한, 최근에 데이터베이스화된 많은 웹사이트들은 웹과 데이터베이스를 연결하기 위하여 리다이렉션 링크를 많이 사용한다. 이 링크 자체만으로는 수집 판단이 어렵기 때문에 수집 가능한 형태의 URL로 변환할 필요가 있다.

본 논문에서는 필요한 URL만 골라내는 URL 필터링 과정에 필요한 여러 가지 URL 패턴을 URL 패턴 스크립트로 정의하여 사용한다. 이 스크립트는 XML 형식으로 작성된다. 그림 6은 URL 패턴 스크립트의 예를 나타내고 있다. URL 패턴 스크립트는 복잡한 URL 구조를 분석하여 수작업으로 기술하기 때문에 오류를 유발할 가능성이 많다. XML로 작성된 스크립트는 향후 개선에 따른 확장성이 좋고, 스크립트 작성을 위한 어플리케이션 개발이 용이한 장점이 있다.

```
<crawlscript>
<site title="사이트 이름">
<identity value="www.web.com" />
<url type="list" class="인물" >
<originalAddr type='url'>http://www.web.com/list.
aspx?cate_id={0}&page={1}</originalAddr>
<targetAddr>http://www.web.com/list.aspx?cate_id
={0}&page={1}</targetAddr>
</url>
<url type="detail" class="인물" >
<originalAddr type='javascript'>
javaScript:show_detail({0})</originalAddr>
<targetAddr>http://www.web.com/detail.aspx?per_i
d={0}</targetAddr>
</url>
</site>
</crawlscript>
```

그림 6. URL 패턴 스크립트.
Fig. 6. URL pattern script.

URL 패턴은 수집 대상 사이트의 링크 정보를 분석하여 패턴화가 가능한 부분을 패턴처리를 하여 스크립트에 추가한다. 그림 7은 URL의 패턴화 과정을 보여준다. 이러한 패턴을 이용하게 되면 기존의 웹 크롤러가 무작위로 모든 문서를 수집하여 목적에 적합하지 않은 문서를 모두 수집하는 문제점을 해결하여 필요한 문서의 URL만 남김으로써 문서 수집 속도와 수집된 문서의 질을 향상시킨다.

URL의 쿼리 변수에는 실제 페이지로 이동하는데 필요한 변수 이외에도 수집 목적과 관계없는 변수도 존재한다. 경우에 따라서는 이러한 변수로 인하여 동일한 문서를 가리키는 링크임에도 불구하고 서로 다른 URL로 인식되어 중복 수집을 유발한다. 본 논문에서는 문서 위치를 나타내는 URL 패턴만을 스크립트로 사용하여 이러한 문제점을 해결한다.

```
http://people.naver.com/directory/list.nhn?query=연
구관련단체인&where=job&dirid=467
                    ↓
http://people.naver.com/directory/list.nhn?query={0}
&where={1}&dirid={2}
{0} = 카테고리 명(필수)
{1} = 카테고리 분류 기준(필수)
{2} = 카테고리 고유ID(필수)
```

그림 7. URL 패턴 생성.
Fig. 7. URL pattern generation.

기본적인 URL을 이용한 링크 외에도 그림 8에서처럼 자바스크립트를 이용한 링크도 자주 사용된다. 이는 기존의 웹 로봇의 접근을 막아 트래픽 요인을 줄이는 효과를 위해 사용되기도 하고, 웹사이트를 리뉴얼할 경우 해당 링크들을 일일이 수정하는 웹페이지 수정 없이 자바스크립트만을 수정하여 리뉴얼이 가능하게 하기 위해서 사용되기도 한다. 이런 자바스크립트를 이용한 링크도 스크립트를 작성하게 되면 해당페이지로의 URL로 변환할 수 있다.

```
pBlog('541293/C/860/869/10599/0');
<!-- 함수내용 -->
function pBlog(param)
{
var sParam = " ";
                ---- 중략 ----
var screenHeight = screen.height-20;
var url = "http://blog.danawa.com/prod/"+sParam;
WindowOpen(url, '_blank', 0, 0, 936, screenHeight,
false, false, false, false, true);
}
```

그림 8. 자바스크립트 링크.
Fig. 8. Javascript Link.

3.4 웹문서 수집 시스템

본 논문에서는 제안한 웹문서 수집 알고리즘을 바탕으로 웹문서 수집 시스템을 개발한다. 웹문서 수집 시스템은 URL 추출 모듈, URL 필터링 모듈, 수집목록 관리 모듈, 문서수집 모듈로 구성된다. 그림 9는 제안하는 시스템의 구성도를 나타내고 있다.

URL 추출 모듈은 입력된 문서의 링크 정보로부터 추출이 가능한 URL을 전부 추출한다. URL 필터링 모듈은 추출된 URL 중에서 수집대상이 되는 URL을 패턴 정보를 이용하여 걸러낸다. 수집목록 관리 모듈은 수집된 URL의 목록을 정해진 우선순위에 맞춰 순서를 조정하고 중복된 항목을 제거하

는 등 수집목록을 관리한다. 문서수집 모듈은 수집 대상 URL 목록으로부터 URL을 읽어 HTML 문서를 수집하고 수집한 문서의 정보를 XML 형태의 정보파일에 기록한다.

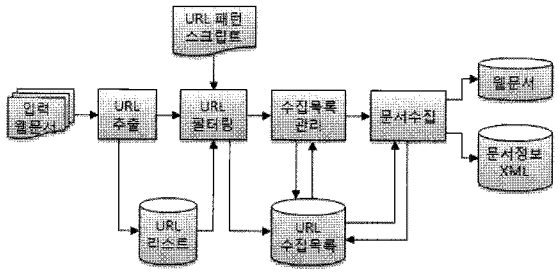


그림 9. 웹문서 수집 시스템의 구성도.
Fig. 9. Overview of Web crawling system.

3.4.1 URL 추출과 필터링

시작 URL로 주어지는 입력 웹문서로부터 URL 링크들을 추출한다. URL 추출 모듈에서는 필요없는 태그들을 제외하고 다른 문서와 연결되는 링크들을 추출한다. 추출된 링크들은 3.2절에서 설명한 URL 패턴 스크립트를 이용하여 필요한 URL만 골라내는 URL 필터링 모듈을 통하여 관심있는 URL만 URL수집 목록으로 보내진다. URL 수집 목록의 URL을 이용하여 수집목록 관리 모듈과 문서 수집 모듈을 통하여 실제로 문서 수집이 이루어진다.

3.4.2 수집목록 관리 모듈

URL 수집과정에서는 여러 문서들의 링크를 통하여 URL을 수집하기 때문에 중복되는 링크들이 많이 존재한다. 수집목록 관리 모듈에서는 우선적으로 목록에서 중복된 URL들을 제거한다.

수집 목록에 등록된 URL들은 현재 페이지가 속한 카테고리와 관련된 리스트 페이지와 상세정보 페이지를 가리키는 링크들이다. 수집목록 관리 모듈에서는 URL 수집목록에 등록된 URL들을 우선순위에 따라 순서를 조정한다. 상세정보 페이지는 목표한 수집 문서이므로 HTML 문서 수집을 위해 최우선 순위로 그 URL을 목록의 상위에 올린다.

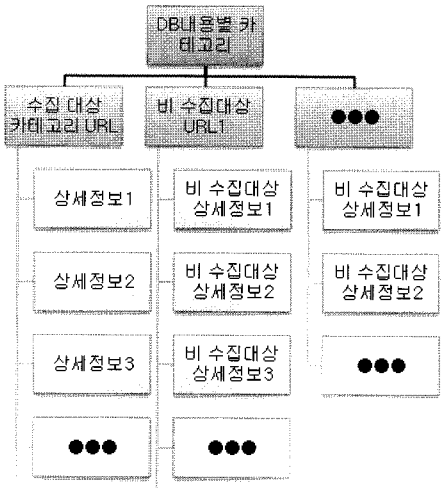


그림 10. 수집목록 관리 모듈을 이용한 분류.
Fig. 10. Grouping by crawling list management module.

리스트 페이지는 수집이 요구된 카테고리 소속인지 판단해야 한다. 리스트 페이지의 URL에는 시스템의 입력으로 들어온 시작 URL에 있는 것과 같은 종류의 링크가 있다. 이 URL의 패턴에는 수집하고자 하는 카테고리의 ID가 있는데, 수집목록에 있는 리스트 페이지 URL 중에 카테고리 ID가 같은 페이지들은 동일 카테고리 페이지의 링크들이다. 이 URL들은 수집 목록에서 상세정보 URL 다음의 우선순위를 부여하여 목록의 순서를 조정한다. 나머지 타 카테고리 리스트 페이지의 URL들은 우선순위의 수집목록의 하단에 등록된다. 페이지 유형에 따라 분류된 URL을 이용해 현재 페이지를 중심으로 그림 10과 같이 다른 카테고리와의 자료를 분리한다.

3.4.3 문서수집 모듈

문서수집 모듈은 HTML 문서 수집과 수집정보의 저장, 수집목록의 업데이트로 구성된다. HTML 문서는 자체적인 파일명 생성을 거쳐서 로컬 하드디스크에 저장된다. 이때 기 수집된 문서의 중복 수집을 피하기 위하여 URL 수집목록에 수집완료에 따른 정보 업데이트를 동시에 수행한다.

파일 저장이 완료된 후 수집과정에 사용된 정보를 XML 형식의 수집문서 정보파일로 저장한다. 이것은 수집된 웹문서를 이용하는 다른 어플리케이션에서 문서 수집 상황을 지식으로 활용할 수 있도록 하기 위함이다. 그림 11은 수집된 하나의 HTML 문서에 대한 저장 정보이다.

```
<doc type="struct">
<source>
<url>http://newprice.empas.com/pd/pd_list.php?cid=
010370010</url>
<file>./Data/제품_노트북_엠파스_/00000000.html
</file>
<site>엠파스</site>
<domain>제품</domain>
<category>노트북</category>
<date>Sun Aug 12 19:55:32 KST 2007</date>
</source>
</doc>
```

그림 11. 수집 정보 XML.
Fig. 11. XML output of crawling information.

4. 실험 및 결과

웹문서 수집 시스템은 J2SDK 1.5.0_03-b07과 공개 소스인 JTidy R7 개발자용 버전을 이용하여 구현하였다. 본 논문에서는 구현된 시스템을 이용하여 IT 분야 온톨로지 구축을 위한 웹문서를 수집하였다. 수집을 위한 사이트 선정은 주요 포털, 가격비교 사이트, 분야별 전문 웹사이트를 대상으로 하였다.

4.1 동일 카테고리 문서 수집 예

HTML 페이지의 링크 태그에는 각 카테고리에 대한 정보를 담은 텍스트와 이미지 등을 포함하고 있다. 이런 정보를 이용하면 수집하고자하는 정보들의 카테고리를 알 수 있다. 문서 수집을 위하여 그림 12의 웹문서에 대해 링크들을 필터링하면 그림 13과 같은 링크 태그를 얻을 수 있다. 이 링크

태그 중 현재 수집한 페이지의 링크 주소와 같은 URL을 포함한 링크 태그에서 현재 문서의 카테고리를 얻을 수 있다. 현재 문서의 카테고리 정보를 얻고 나면 이후로는 이 카테고리 정보를 이용하여 같은 카테고리의 문서를 수집할 수 있다. 해당 카테고리 이외의 대부분의 링크는 무시하거나 다른 카테고리의 문서 수집을 위해 저장하게 된다.

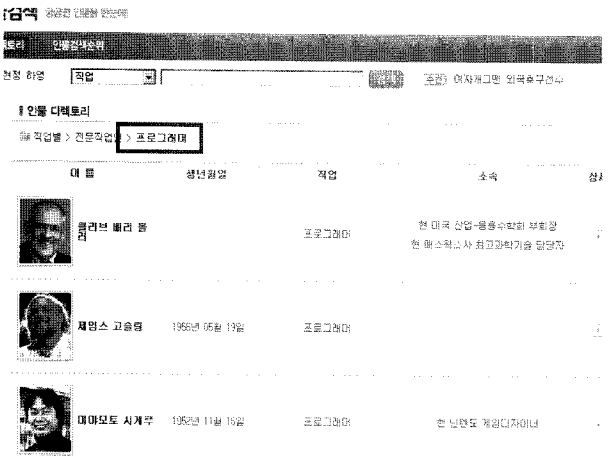


그림 12. 수집 대상 페이지 예제.
Fig. 12. An example of crawling target page.

```
<!-- 카테고리 -->
<table width="100%" border="0" cellpadding="0" cellspacing="0">
                ---- 중략 ----
<a href="DirectoryList.nhn?query=%EC%A0%84%EB%AC%B8%EC%A7%81%EC%97%85%EC%9D%B8&where=job&dirid=471&depth=1" class=' '>전문직업인</a>
&gt;
<a href="DirectoryList.nhn?query=%ED%94%84%EB%A1%9C%EA%B7%B8%EB%9E%98%EB%A8%B8&where=job&dirid=494&depth=2" class='b'>프로그래머</a>
<br />
<img height="10" width="1" /><br />
<table width="100%" border="0" cellpadding="0" cellspacing="0">
</table>
</td>
</tr>
</table>
```

그림 13. 카테고리 링크의 예제.
Fig. 13. An example of category Link.

또한 리스트 페이지에는 그림 14의 하단과 같은 일련의 숫자로 이루어거나 특정단어로 표시된 링크들을 볼 수 있다. 대부분의 웹 사이트는 한 카테고리마다 많은 정보를 포함하기 때문에 이런 링크들은 동일 카테고리로 이동하는 링크일 가능성이 매우 높다. 따라서, 이러한 링크들은 동일한 카테고리 정보로 판단하여 수집 대상이 된다.

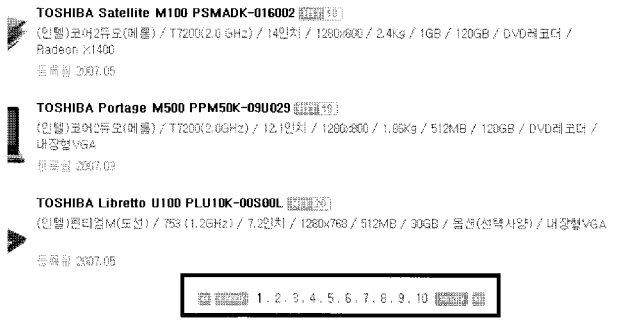


그림 14. 동일 카테고리로의 링크의 예제.
Fig. 14. An example of Links to same category.

4.2 웹문서 수집 실험

표 1은 9개 웹사이트에 대해서 문서를 수집한 결과를 나타낸 것이다. 전체 문서량은 추정치로서 경우에 따라서는 개수를 알 수 없는 사이트도 있다. 전문 사이트의 경우에는 수집한 문서 수가 전체 문서와 비슷하고, 일반 가격비교 사이트는 IT 외에도 많은 상품 문서가 있기 때문에 전체 문서 수보다 상당히 적은 양이 수집된 것을 알 수 있다.

표 1. IT분야 웹문서 수집 결과.
Table 1. The result of Web crawling on IT domain.

웹사이트	전체 문서(추정)	수집 문서
인물정보1	약 5만개	7332개
인물정보2	약 30만개	9803개
가격비교1	약 55만개	13939개
가격비교2	약 1천개	3358개
가격비교3	알 수 없음	859개
가격비교4	약 8천개	7004개
도서정보1	알 수 없음	968개
기업정보1	1077개	976개
기업정보2	약 1만 3천개	152개

본 논문에서 제안하는 URL 패턴을 이용한 필터링 효과를 확인하기 위하여 하나의 웹문서에서 추출되는 URL의 수를 확인하였다. 표 2는 실험 대상 9개 사이트의 리스트 페이지를 하나씩 선정하여 URL 패턴 필터링 전후의 수집 대상 URL의 개수를 비교한 것이다. 사이트의 페이지의 구성 정책에 따라 리스트 내용과 관련 없는 정보가 많은 사이트는 필터링 전의 많은 URL이 필터링 후에 제거되었다. 전체적으로 88%의 링크가 제거되고 12%만 다음 문서 수집에 사용되어 모든 링크를 조사하는 일반적인 웹 크롤러보다 효율적으로 문서를 수집하는 것으로 나타났다.

표 2. URL 패턴 필터링 결과.
Table 2. The result of URL pattern filtering.

웹사이트	필터링 전	필터링 후
인물정보1	222개	29개
인물정보2	145개	39개
가격비교1	125개	10개
가격비교2	81개	6개
가격비교3	832개	21개
가격비교4	2835개	417개
도서정보1	359개	21개
기업정보1	30개	15개
기업정보2	67개	13개

5. 결론 및 향후 과제

인터넷이 방대해짐에 따라 필요한 정보를 찾는 일이 점점 어려워지고 있고, 인터넷에 있는 자료를 정보로 사용하기 위해 필요한 문서만 수집하는 일도 어려워졌다. 여기에는 다양해진 링크 방식과 동적인 문서의 비중이 높아진 것도 큰 원인이 된다. 본 논문에서는 이러한 웹 상황에서 필요한 문서를 수집하기 위하여 URL 패턴을 이용하여 문서를 수집하는 방법을 제시하고 시스템을 구성하였다.

본 논문에서 제시한 웹 문서 수집기는 불필요한 링크를 수집목록에서 제외시킴으로써 필요한 문서만을 수집할 뿐만 아니라 웹서버의 트래픽을 줄이고, 수집 속도를 빠르게 하였다. 또한 기존의 다른 수집기에서는 구별할 수 없거나 구별을 위하여 별도의 내용 비교가 필요한 중복링크에 대한 문제를 해결하였다.

향후에는 URL 패턴으로 구분되지 않는 링크들은 제안하는 시스템으로는 처리가 불가능하므로 이에 대한 보완 연구가 필요하다. 그리고 URL 패턴 스크립트의 작성을 수작업으로 작성하기 때문에, 웹사이트의 구조에 대한 지식이 없는 사람은 작성하기가 어렵다. 향후에는 이러한 스크립트를 따로 작성하지 않고 웹사이트 내의 링크들을 자동으로 분석하여 스크립트를 작성할 수 있는 모듈 개발을 위한 연구가 보완되어야 한다.

참 고 문 헌

[1] Tim Berners-Lee, "Enabling Standards & Technologies," (http://www.w3.org/2002/Talks/04-sweb/slide12-0.html).

[2] 김성진, 이상호, "웹 로봇 구현 및 한국 웹 통계보고," *한국정보처리학회논문지C*, 제10권, 4호, pp. 509-518, 2003.

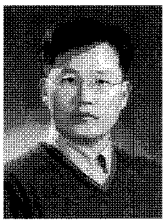
[3] 강문수, 최영식, "대용량 분산 웹 크롤러", *한국인터넷정보학회 학술발표대회 논문집*, 제6권 1호, pp. 185-188, 2005.

[4] J. Cho, "Efficient Crawling through URL ordering," *Computer Networks and ISDN Systems*, Vol.30, pp. 161-172, 1998.

[5] 장문수, 강선미, "도메인지식의 계층화를 통한 온톨로지 인스턴스의 속성정보 추출", *퍼지및지능시스템학회 논문지*, 17권 3호, pp. 291-296, 2007.6.

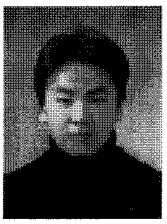
[6] "The Web Robots FAQ", http://www.robotstxt.org/faq.html.

저 자 소 개



장문수(Moon-soo Chang)
1992년 : 고려대학교 전자전산공학과 학사.
1994년 : 동 대학원 전자공학과 석사
2001년 : 동경공업대학 지능시스템전공 박사
2000년~2003년 : 한국전자통신연구원 선임연구원
2003년~현재 : 서경대학교 소프트웨어학과 전임강사

관심분야 : 언어이해, 대화처리, 지능시스템, 정보검색, 온톨로지
E-mail : cosmos@skuniv.ac.kr



정준영(June-young Jung)
2004년~현재 : 서경대학교 소프트웨어학과 재학중.

관심분야 : 웹 크롤러, 온톨로지, 웹 2.0, 웹프로그래밍
E-mail : grounder@naver.com