웹 로봇의 성능 평가를 위한 방법론

김 광 현† · 이 준 호††

요 약

인터넷의 이용이 활발해짐에 따라 수 많은 정보들이 웹을 통하여 공개되고 있으며, 이용자는 웹 검색 서비스를 이용하여 이러한 정보들에 효과적으로 접근할 수 있다. 웹 검색 서비스의 구축을 위해서는 웹 로봇을 사용한 웹 문서 수집이 선행되어야 하며, 웹 문서들의 수가 급격히 증가하면서 양질의 웹 문서들을 효과적으로 수집할 수 있는 웹 로봇에 대한 필요성이 증가하고 있다. 본 연구에서는 웹 로봇들을 체계적으로 평가하기 위한 기준으로서 효율성, 지속성, 신선성, 포괄성, 정숙성, 유일성, 안전성을 제시하고, 이러한 평가 기준의 향상에 도움이 되는 기능들을 기술하였다. 또한, 본 연구에서는 네이버, 구글, 알타비스타 등에서 사용되고 있는 기존의 웹 로봇들에 구현된 기능들을 조사하였다. 본 연구의 결과는 보다 효과적인 웹 로봇의 개발에 기여할 것으로 기대된다.

A Methodology for Performance Evaluation of Web Robots

Kwang Hyun Kim† · Joon Ho Lee††

ABSTRACT

As the use of the Internet becomes more popular, a huge amount of information is published on the Web, and users can access the information effectively with Web search services. Since Web search services retrieve relevant documents from those collected by Web robots, we need to improve the crawling quality of Web robots. In this paper, we suggest evaluation criteria for Web robots such as efficiency, continuity, freshness, coverage, silence, uniqueness and safety, and present various functions to improve the performance of Web robots. We also investigate the functions implemented in the conventional Web robots of NAVER, Google, AltaVista etc. It is expected that this study could contribute the development of more effective Web robots.

키워드 : 정보 검색(Information Retrieval), 웹 로봇(Web Robot), 성능 평가(Performance Evaluation)

1. 서 론

인터넷의 이용이 활발해짐에 따라 수 많은 정보들이 웹 문서의 형태로 공개되고 있으며, 이러한 웹 문서들을 효과적으로 검색하기 위하여 웹 검색 서비스들이 이용되고 있다. 웹 로봇은 지정된 URL 리스트에서 시작하여 웹 문서를 수집하고, 수집된 웹 문서에 포함된 URL들의 추출 과정과 새롭게 발견된 URL에 대한 웹 문서 수집 과정을 반복하는 소프트웨어로서 웹 검색 서비스의 구축을 위해서는 웹 로봇을 이용한 웹 문서 수집이 선행되어야 한다. 웹 로봇의 웹 문서 수집 결과는 웹 검색 결과의 품질에 많은 영향을 미치며, 이는 웹 검색 서비스들이 수집된 웹 문서들만을 대상으로 검색을 수행하기 때문이다.

1990년대 중반의 웹 문서 수는 현재에 비하여 매우 적었기 때문에, 최초로 개발된 웹 로봇 Wanderer를 포함하여 이 당시 개발된 다수의 웹 로봇들은 대용량의 웹 문서들을 수집하도록 설계되지 않았다[1]. 그러나, 현재는 전세계적으로 30억 개 이상의 웹 문서들이 존재하며, 국내에도 5천만 개 이상의 웹 문서들이 존재하고 있다. 따라서 이처럼 많은 수의 웹 문서들을 효율적으로 수집할 수 있는, 즉 초당 수백 또는 수천개의 웹 문서들을 수집할 수 있는 웹 로봇의 필요성이 증가하고 있다.

한편, 웹 환경은 다음과 같은 특성들을 지니고 있으며, 웹 로봇은 이러한 특성들을 고려하여 개발되어야 한다. 첫째, 웹에 공개되는 문서들의 수가 매우 빠르게 증가하고 있다. 둘째, 웹 로봇이 문서들을 수집하고 있는 동안에도 이들에 대한 수정 및 삭제가 수행된다. 셋째, 웹에는 양질의 문서들뿐만 아니라 다수의 유해 또는 스팸 웹 문서들이 존재한다. 넷째, 웹에는 다수의 동일한 내용을 지닌 문서들이 존재한다. 다섯째, 로봇 배제 표준이 준수되어야 한다[2].

본 연구에서는 위에서 언급된 웹 환경의 특성들과 기존 웹 로봇들의 기능들을 조사하여, 웹 로봇들의 성능을 체계적으로 평가하기 위한 기준들을 제시하고, 각 평가 기준의

※ 본 연구는 숭실대학교 교내연구비 지원으로 이루어졌음.
† 준 회 원 : 숭실대학교 대학원 컴퓨터학과
†† 종신회원 : 숭실대학교 컴퓨터학부 교수
논문접수 : 2003년 7월 1일, 심사완료 : 2004년 2월 10일

향상에 도움이 되는 기능들을 기술한다. 또한, 본 연구에서는 네이버, 구글, 알타비스타에서 상업용으로 사용되고 있는 웹 로봇들과 폴리테크닉 대학교에서 연구용으로 개발된 웹 로봇에 구현된 기능들을 조사한다.

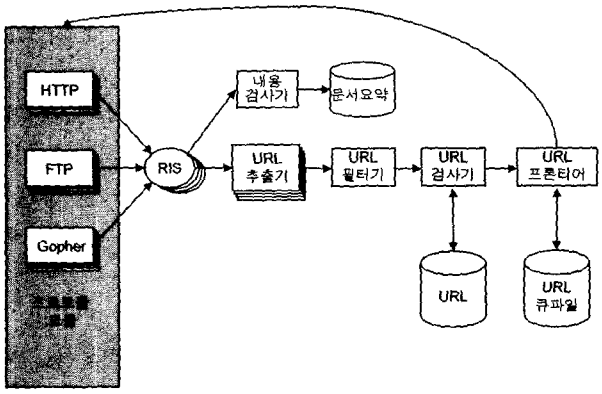
본 연구의 구성은 다음과 같다. 2장에서는 지금까지 개발된 웹 로봇들의 구조와 구성 요소들에 대하여 설명한다. 3장에서는 웹 로봇을 체계적으로 평가할 수 있는 기준들을 제시하고, 또한 각 평가 기준들을 향상시킬 수 있는 기능들을 기술한다. 그리고 4장에서는 기존의 웹 로봇들이 3장에서 기술한 기능들을 포함하고 있는가를 분석한다. 마지막으로 5장에서 결론을 맺는다.

2. 관련 연구

2.1 메르카토르

메르카토르(mercator)는 웹 검색 서비스인 알타비스타에서 이용하고 있는 웹 로봇으로서 DEC/Compaq에서 개발되었다[3, 4]. 메르카토르는 필요한 기능들을 플러그인 방식으로 추가함으로써 시스템을 쉽게 확장할 수 있도록 설계되었으며, 또한 자바 프로그래밍 언어로 개발되었기 때문에 자바 가상 기계가 설치된 모든 플랫폼에서 실행될 수 있다.

(그림 1)은 메르카토르 시스템의 구조를 보여준다. 메르카토르는 URL 프론티어를 호출하여 수집할 웹 문서의 URL을 획득하고, 이 URL을 HTTP, FTP, Gopher 중에서 적합한 프로토콜 모듈에게 전달한다. 프로토콜 모듈은 URL에 의해 지정된 웹 문서를 다운로드하며, 이 문서는 RIS에 의해 관리된다. 메르카토르는 RIS에 의해 관리되는 각각의 웹 문서에 대하여 내용 검사기를 호출함으로써 이미 수집된 웹 문서들과의 중복 여부를 확인한 후, 중복되지 않은 웹 문서로부터 URL들을 추출한다. 이 URL들 중의 일부는 URL 필터기에 의해 제거되며, 나머지 URL들로부터 URL 검사기는 지금까지 수집되지 않은 웹 문서들의 URL을 검출한 후, 이들을 URL 프론티어로 전달한다.

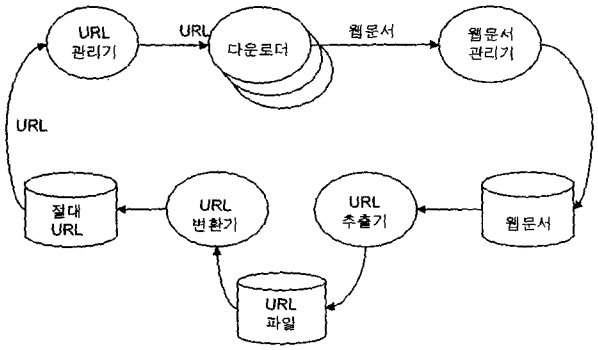


(그림 1) 메르카토르 시스템의 구조

2.2 구글봇

구글봇(googlebot)은 웹 검색 서비스를 제공하는 구글에서 사용하고 있는 웹 로봇으로, 스탠포드 대학의 학생이었던 Page & Brin에 의해 개발되었다[5]. 구글은 이러한 구글봇을 이용하여 전 세계를 대상으로 30억개 이상의 웹 문서를 수집하고 있다. 또한, 구글은 상업화된 이후에도 스탠포드 대학과 웹 문서 수집에 관련된 연구를 지속적으로 수행하고 있으며, 그 결과로는 웹 문서들의 병렬 수집[6], 중복된 문서들의 검출[7], 동적인 웹 문서들의 수집[8], 웹 문서들의 수정 주기 분석[9] 등이 있다.

(그림 2)는 구글봇 시스템의 구조를 보여 준다. 구글봇은 URL 관리기, 다운로더, 웹 문서 관리기, URL 추출기, URL 변환기로 구성되어 있으며, 각각의 구성 요소는 독립적인 프로세스로서 존재한다. URL 관리기는 수집할 웹 문서들의 URL들을 다수의 다운로더들에게 분배한다. 각각의 다운로더는 서로 다른 컴퓨터에서 실행되고, 웹 문서 관리기는 다운로드된 웹 문서들을 압축하여 디스크에 저장한다. URL 추출기는 디스크에 저장된 웹 문서들로부터 URL들을 추출하고, URL 변환기는 이 URL들을 절대 URL로 변환하여 디스크에 저장한다.



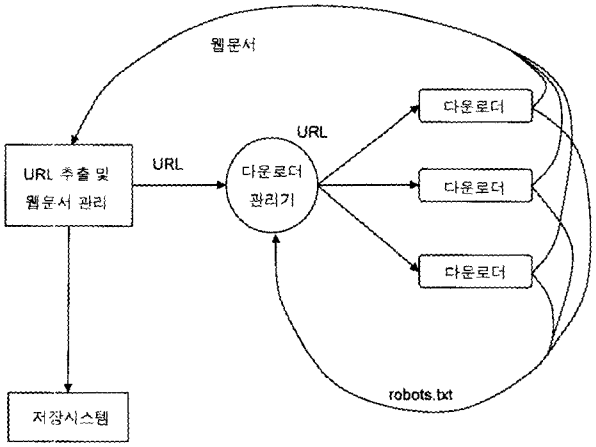
(그림 2) 구글봇 시스템의 구조

2.3 폴리봇

폴리봇(polybot)은 폴리테크닉 대학교에서 연구용으로 개발된 웹 로봇으로서, 폴리봇의 구성 요소들은 서로 다른 컴퓨터에서 독립적으로 실행될 수 있다[10]. 또한, 각각의 구성 요소에 컴퓨터를 추가함으로써 웹 문서 수집 성능을 향상 시킬 수 있다. 이러한 폴리봇은 실험을 통하여 18일 동안 약 5백만 개의 호스트에서 1억 2천만 개 이상의 웹 문서들을 수집하였다.

(그림 3)은 폴리봇 시스템의 구조를 보여준다. 웹 로봇이 특정 웹 서버로부터 많은 수의 웹 문서들을 단기간에 수집할 경우, 웹 서버에 과도한 부하를 유발시킬 수 있다. 이러

한 문제를 방지하기 하기 위해 다운로더 관리기는 URL들을 뒤섞음(shuffling)한 후, 이 URL들을 다운로더들에게 전달한다. 또한, 다운로더 관리기는 웹 서버의 robots.txt 파일을 분석하여 로봇 배제 표준을 준수한다. 폴리봇은 다운로드된 웹 문서들로부터 URL들을 추출하고, 지금까지 수집되지 않은 웹 문서들의 URL들을 다운로더 관리기로 전달하며, 웹 문서들을 저장 시스템에 전달한다.

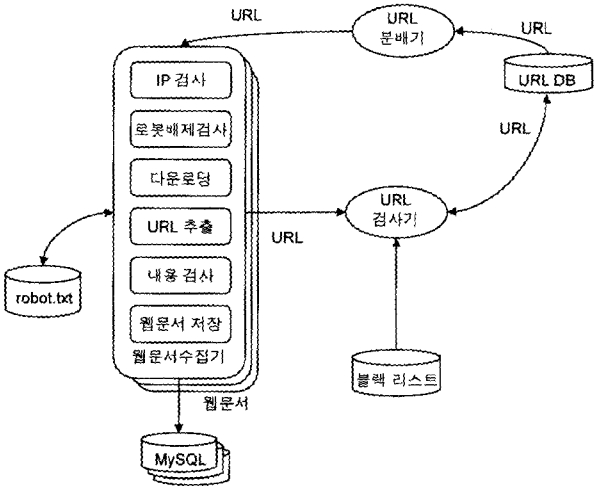


(그림 3) 폴리봇 시스템의 구조

2.4 네이봇

네이봇(nabot)은 웹 검색 포탈 네이버에서 사용하는 웹 로봇으로서 국내 및 일본의 웹 문서들을 수집한다. 네이봇은 데이터베이스 관리 시스템 MySQL을 사용하여 수집된 웹 문서들을 관리하며, 또한 과거에 수집된 웹 문서들을 지속적으로 수집하기 위하여 지금까지 수집된 전체 웹 문서들의 URL을 관리한다. 네이봇은 관리중인 URL들이 지시하는 웹 문서만을 수집하고, 수집된 웹 문서들로부터 발견된 새로운 URL들을 URL 데이터베이스에 추가한다. 따라서 새롭게 발견된 URL들이 지시하는 웹 문서들은 다음 번 네이봇 수행시에 수집된다.

(그림 4)는 네이봇 시스템의 구조를 보여준다. URL 분배기는 관리중인 URL들을 다수의 컴퓨터에 분산되어 있는 웹 문서 수집기들에게 분배하며, 웹 문서 수집기는 다음과 같은 작업들을 수행한다. 첫째, URL의 IP를 검사하여 국내 또는 일본의 웹 문서인지를 확인한다. 둘째, 로봇 배제 표준을 준수하기 위해서 웹 서버의 robots.txt 파일의 내용을 확인한다. 셋째, 웹 문서를 다운로드한다. 넷째, 다운로드된 문서로부터 URL들을 추출하여 URL 검사기로 전달한다. 다섯째, 웹 문서의 내용을 분석하여 유해 또는 스팸 문서인가를 검사한다. 마지막으로, 웹 문서를 압축하여 데이터베이스에 저장한다. 한편, URL 검사기는 전달된 URL들 중에서 블랙 리스트에 포함된 URL들과 기존의 URL들을 제거한 후, 나머지 URL들을 URL 데이터베이스에 추가한다.



(그림 4) 네이봇 시스템의 구조

3. 평가 기준

본 장에서는 웹 로봇의 성능을 체계적으로 평가하기 위한 기준으로서 효율성, 지속성, 신선성, 포괄성, 정숙성, 유일성, 안전성을 제시하고, 또한 이러한 평가 기준들의 향상에 도움이 되는 기능들을 기술한다. 웹 로봇 개발자는 각각의 평가 기준에 대하여 열거된 기능들을 웹 로봇에 추가함으로써, 해당 평가 기준을 향상시키고, 보다 높은 성능의 웹 로봇을 개발할 수 있다.

3.1 효율성

웹 문서들의 수가 급격히 증가하고 있는 현실을 고려할 때, 초당 수백 또는 수천 개의 웹 문서들을 수집할 수 있는 웹 로봇에 대한 필요성이 절실하다. 효율성(efficiency)은 주어진 시간 내에 수집할 수 있는 웹 문서들의 수를 의미하며, 보다 효율적인 웹 로봇의 개발을 위해서는 다음과 같은 사항들이 웹 로봇의 설계에 반영되어야 한다.

- **병렬 수집** : 웹 로봇은 수집된 웹 문서들을 디스크에 저장하기 위해 많은 시간을 소비하는 입출력 집중 프로그램(I/O intensive program)이다. 따라서 하나의 컴퓨터에서 다수의 웹 로봇 프로세스를 동시에 실행함으로써 주어진 시간에 보다 많은 웹 문서들을 수집할 수 있다.
- **분산 수집** : 현재 국내의 웹 문서들의 수는 5천만 건을 넘어서고 있으며, 이러한 웹 문서들을 하나의 컴퓨터를 이용하여 수집할 경우, 몇 주 정도의 시간이 소

비된다. 따라서 웹 문서 수집에 소비되는 시간을 보다 단축하기 위해서는 다수의 컴퓨터를 이용한 웹 문서들의 분산 수집이 절실하다.

- **분산 저장** : 일반적으로 웹 로봇 시스템은 서로 다른 기능을 수행하는 다수의 컴퓨터로 구성되며, 네트워크를 통하여 웹 문서들을 내려 받는 기능과 웹 문서들을 저장하는 기능은 별도의 장비에서 수행된다. 따라서 하나의 저장 서버에 모든 웹 문서들을 저장할 경우, 저장 서버에 입출력 병목 현상이 발생할 수 있기 때문에, 웹 문서들을 다수의 저장 서버에 분산 저장하는 것이 바람직하다.
- **DNS 캐시** : 하나의 컴퓨터에서 다수의 웹 로봇 프로세스들을 실행하고, 더불어 다수의 컴퓨터를 이용하여 웹 문서들을 수집할 경우, URL을 IP 주소로 변환하는 DNS 서버에 병목 현상이 발생할 수 있다. 이러한 병목 현상은 호스트 이름과 IP 주소를 자체적으로 관리함으로써 줄일 수 있다.

3.2 지속성

일반적으로 웹 검색 서비스들은 주기적으로 웹 로봇을 수행하며, 지속성(continuity)은 이전에 수집된 후 현재까지 삭제되지 않은 웹 문서들에 대한 재수집 된 웹 문서들의 비율로서 정의된다. 즉, 지속성은 이전에 수집된 후 현재까지 삭제되지 않은 웹 문서들이 어느 정도 재수집 되었는가를 나타낸다. 이러한 지속성을 향상시키기 위해서는 다음과 같은 사항들이 웹 로봇의 설계에 반영되어야 한다.

- **전체 URL 관리** : 많은 경우에 웹 로봇은 씨앗(seed) URL들에서 시작하여, 이들로부터 접근 가능한 모든 웹 문서들을 수집한다. 그러나, 네트워크의 속도 저하, DNS 서버의 장애, 특정 웹 사이트에서의 일시 점검 등으로 인하여, 동일한 씨앗 URL에 대하여 동일한 웹 로봇이 수행될지라도 웹 로봇의 수행 시점에 따라 서로 다른 웹 문서 집합들이 수집 될 수 있다. 따라서, 지속성을 높이기 위해서 과거에 수집된 모든 웹 문서들의 URL을 관리하는 것이 바람직하다.
- **웹 문서 추가** : 과거에 수집된 모든 웹 문서들에 대한 URL을 관리하고, 이후 이들에 대한 재수집을 항상 시도할지라도, 다양한 장애 요인들로 인하여 이전에 수집된 웹 문서들의 재수집에 실패할 가능성이 여전히 존재한다. 이 경우 재수집에 실패한 웹 문서들을 이전에 수집된 웹 문서 데이터베이스로부터 추출함으로써 지속성을 높일 수 있다.

3.3 신선성

신선성(freshness)은 이전에 수집된 웹 문서들에 대한 현재 수정 및 삭제되지 않은 웹 문서들의 비율로서 정의된다. 즉, 신선성은 이전에 수집된 웹 문서들이 현재 어느 정도 수정 및 삭제 되었는가를 나타낸다. 수집된 웹 문서들의 신선성이 저하될수록, 삭제 또는 수정된 웹 문서들이 검색 결과에 포함될 가능성이 증가한다. 따라서 신선성의 저하는 웹 검색 결과에 대한 이용자의 만족도를 저하시키기 때문에, 이를 향상시키기 위하여 다음과 같은 사항들이 웹 로봇의 설계에 반영되어야 한다.

- **수집 주기 관리** : 웹 문서는 그 내용에 따라 빈번하게 수정되거나, 또는 수개월이 지나도 수정되지 않을 수 있다. 이처럼 웹 문서들은 서로 다른 수정 주기를 지니기 때문에, 동일한 주기로 웹 문서들을 수집할 경우 신선성이 저하될 수 있다. 따라서, 웹 문서들의 수정 주기를 측정하고, 수정 주기가 짧은 웹 문서들을 보다 자주 수집함으로써 신선성을 향상시킬 수 있다.
- **URL 삭제** : 웹 로봇이 지속성의 개선을 위하여 수집 가능한 전체 URL을 관리하도록 설계된 경우, 웹 로봇은 보유중인 각각의 URL에 해당하는 웹 문서의 수집을 시도하며, 종종 데드링크 즉, 접근되지 않는 웹 문서에 직면한다. 데드링크가 발생하는 이유들 중의 하나는 해당 웹 문서가 삭제되어 더 이상 존재하지 않기 때문이다. 웹 로봇은 이러한 경우를 확인하여 삭제된 웹 문서의 URL을 관리 중인 URL 데이터베이스로부터 삭제함으로써 신선성을 향상시킬 수 있다.

3.4 포괄성

포괄성(coverage)은 전체 웹 문서에 대한 웹 로봇이 수행을 시작하여 종료할 때까지 수집한 웹 문서들의 비율로 정의된다. 웹 검색 서비스들은 웹 로봇이 수집한 웹 문서들에 대한 검색을 지원하기 때문에, 포괄성은 웹 검색 결과의 품질을 결정하는 중요한 요소들 중의 하나로 인식되고 있으며, 이를 향상시키기 위해서 웹 로봇은 다음과 같은 형태의 문서들을 수집할 수 있도록 설계되어야 한다.

- **정적 웹 문서** : 정적 웹 문서는 HTML로 작성된 가장 기본적인 형태의 웹 문서로서, URL에 의해 서로 연결되어 있다. 이러한 정적 웹 문서를 수집하기 위해서 웹 로봇은 지정된 URL 리스트에서 시작하여 웹 문서를 수집하고, 이후 수집된 웹 문서에 포함된 URL들의 추출 과정과 새롭게 발견된 URL에 대한 웹 문서 수집 과정을 반복한다.

- **동적 웹 문서** : 동적 웹 문서는 이용자가 원하는 정보를 얻기 위하여 웹 문서의 폼(form)에 값 또는 질의를 입력한 후 서버에 요청하여 생성되는 문서들이며, "Hidden Web", "Deep Web" 등으로 불리기도 한다. 현재 웹 상에 존재하는 동적 웹 문서들의 수는 정적 웹 문서 수의 수백 배를 넘는 것으로 조사되고 있다. 웹 로봇이 동적 웹 문서를 수집하기 위해서는 웹 문서 내의 폼을 분석하고, 적절한 값들을 이용한 URL 작성이 선행되어야 한다.
- **일반 문서** : 인터넷을 통한 정보 교류가 활발해 짐에 따라 학술지, 간행물, 논문집 등과 같은 많은 일반 문서들이 워드프로세서, PDF, PS 파일 형태로 인터넷에 공개되고 있다. 이러한 일반 문서들은 웹 문서들보다 양질의 정보들을 포함하고 있을 가능성이 높기 때문에, 현재 다수의 웹 검색 서비스들은 이러한 일반 문서들에 대한 검색을 제공하고 있다. 이러한 일반 문서를 수집하기 위해서는 웹 로봇은 정적 및 동적 웹 문서에 포함된 일반 문서 URL들의 추출이 선행되어야 한다.

3.5 정숙성

웹 로봇들은 인터넷에 연결되어 있는 웹 서버들에 접근하여 그 서버에 존재하는 웹 문서들을 수집한다. 따라서 웹 로봇이 특정 웹 서버로부터 단기간에 다수의 웹 문서들을 수집할 경우, 과도한 부하를 유발하여 웹 서버가 제공하는 서비스들에 피해를 줄 수 있다. 이러한 피해의 회피 및 보유하고 있는 정보들의 보호를 위하여 일부 웹 서버들은 로봇 배제 표준을 이용하여 웹 로봇들의 접근을 배제하고 있다. 정숙성은(silence)은 웹 문서 수집 기간 동안 피해를 주지 않은 웹 서버들의 수로 정의되며, 이러한 정숙성을 향상시키기 위해서는 다음과 같은 사항들이 웹 로봇의 설계에 반영되어야 한다.

- **로봇 배제 표준** : 로봇 배제 표준은 "robots.txt" 파일 또는 로봇 메타 태그를 이용하여 웹 로봇들의 웹 문서 수집을 제한한다. 즉, 웹 서버 관리자들은 "robots.txt" 파일에 웹 로봇들의 이름을 명시함으로써 사이트 내의 전체 웹 문서들에 대한 수집을 방지하거나, 또는 디렉토리 이름들을 명시함으로써 일부 웹 문서들에 대한 수집만을 방지할 수 있다. 또한, 웹 문서 작성자들은 로봇 메타 태그를 이용하여 작성된 웹 문서에 대한 수집을 제한할 수 있다. 따라서, 웹 로봇은 웹 문서를 수집하기 전에 "robots.txt" 파일과 로봇 메타 태그들을 확인하여, 지정된 내용을 준수해야 한다.
- **수집 속도 조절** : 웹 로봇이 효율성을 향상시키기 위하여 단기간 내에 많은 웹 문서들을 수집하도록 설계된 경우, 접근한 웹 서버의 부하를 급격히 증가시키는 경우가 종종 발생한다. 실제 웹 검색 서비스 업체의 웹 로봇 운영자들은 웹 로봇이 접근했던 웹 서버를 소유한 회사나 기관으로부터 항의성 메일이나 전화를 받은 경험들을 지니고 있다. 따라서, 웹 서버의 부하가 증가하지 않도록 웹 로봇의 웹 문서 수집 속도를 조절해야 한다.

3.6 유일성

인터넷에는 내용이 동일한 다수의 웹 문서들이 존재하며, 이러한 웹 문서들의 수집은 웹 로봇의 효율성을 저하시키고 저장 공간의 낭비를 초래한다. 유일성(uniqueness)은 수집된 전체 웹 문서들에 대한 중복되지 않은 유일 웹 문서들의 비율로 정의되고, 이러한 유일성을 향상시키기 위해서는 다음과 같은 사항들이 웹 로봇의 설계에 반영되어야 한다.

- **URL 정규화** : 인터넷에는 하나의 웹 문서에 대한 주소가 다양한 형태의 URL로 표현되고 있다. 예를 들어, "www.yahoo.co.kr", "www.yahoo.co.kr/", "www.yahoo.co.kr/index.html", "www.yahoo.co.kr:80"은 모두 동일한 웹 문서를 지시하는 URL들이다. 웹 로봇이 이러한 URL들 모두에 대하여 웹 문서를 수집할 경우, 동일한 내용의 웹 문서들이 중복 수집된다. 따라서, 이를 방지하기 위하여 하나의 웹 문서 주소에 대한 다양한 형태의 URL들을 하나의 URL로 정규화함이 바람직하다.
- **호스트 이름 그룹화** : URL 정규화 이후에도 서로 다른 URL들에 대해서 동일한 내용의 웹 문서들이 수집될 수 있으며, 그 이유는 다음과 같다. 첫째, 많은 경우에 하나의 호스트 이름은 다수의 별칭을 지니고 있으며, 이러한 별칭들은 URL 정규화 이후에도 서로 다른 URL로 표현된다. 둘째, 웹 사이트 전체를 다른 호스트 이름의 컴퓨터에 동일하게 복사한 미러(mirror) 사이트들이 존재한다. 따라서, 웹 로봇은 하나의 호스트 이름에 대한 별칭들 또는 미러 사이트 이름들을 그룹화하여 관리함이 바람직하다.
- **중복 문서 검출** : URL 정규화와 호스트 이름 그룹화를 통하여 수집된 웹 문서들 내에서도 여전히 동일한 내용의 문서들을 발견할 수 있다. 웹 로봇은 웹 문서들의 내용들을 비교하여 이러한 중복 문서들을 검출하는 것이 바람직하다. 일반적으로 웹 문서들의 내용을 비교하기 위해서 전체 문서 내용을 일정한 길이의 축

약된 메시지로 변환하는 MD5, SHA 등과 같은 알고리즘들이 이용된다.

3.7 안전성

인터넷에 공개된 웹 문서들 중에는 음란, 폭력 등의 내용을 포함하는 유해 웹 문서와 상업적인 목적으로 이용자들의 방문을 유도하는 스팸 웹 문서들이 존재하며, 웹 검색 서비스들의 검색 결과에도 이러한 웹 문서들이 포함되는 경우를 종종 볼 수 있다. 안전성(safety)은 수집된 전체 웹 문서들에 대한 유해 및 스팸이 아닌 웹 문서들의 비율로 정의되며, 이러한 안전성을 향상하기 위해서는 다음과 같은 사항들이 웹 로봇의 설계에 반영되어야 한다.

- **유해 웹 문서 검출** : 유해 웹 문서는 선정, 음란, 폭력적인 내용을 포함하고 있는 문서로서 성인들에게는 불쾌감을 유발하며, 특히 청소년들이나 어린이들에게는 악영향을 줄 수 있다. 따라서 웹 로봇은 이용자들이 안전하게 웹 검색 서비스를 이용할 수 있도록 수집된 웹 문서들의 내용을 분석하여 유해 웹 문서들을 검출한 후, 이들을 제거함으로써 안전성을 높일 수 있다.
- **스팸 웹 문서 검출** : 일반적으로 스팸 웹 문서는 웹 문서의 내용과 관련 없는 인기 키워드들을 웹 문서에 추가함으로써 생성된다. 스팸 웹 문서는 상업적인 목적으로 이용자들의 방문을 유도하기 위해 작성되며, 검색 결과의 품질을 저하시키는 요인이 된다. 웹 로봇은 수집된 웹 문서들의 내용을 분석하여 스팸 웹 문서들을 검출한 후, 이들을 제거하는 것이 바람직하다.
- **블랙 리스트 관리** : 블랙 리스트는 이전에 발견되었거나 웹 로봇의 수행 중에 발견된 유해 또는 스팸 사이트들의 목록이다. 웹 로봇은 이러한 블랙 리스트들을 이용하여 유해 또는 스팸 웹 문서의 수집을 사전에 차단할 수 있다.

4. 웹 로봇 분석

메르카토르, 구글봇, 네이봇은 각각 알타비스타, 구글, 네이버에서 사용되고 있으며, 폴리봇은 폴리테크닉 대학교에서 연구용으로 개발되었다. 본 장에서는 지금까지 발표된 논문들을 기반으로 메르카토르, 구글봇, 폴리봇, 네이봇이 3장에서 기술한 기능들을 포함하고 있는가를 분석하였으며, <표 1>은 이러한 분석 결과를 보여준다. 단, 네이봇에 대해서는 아직까지 발표된 논문이 없기 때문에, 이의 분석은 네이봇을 개발하고 있는 NHN㈜의 협조하에 수행되었다.

<표 1>에서 "○"와 "×"는 각각 웹 로봇에 구현된 기능과 구현되지 않은 기능을 의미하고, "•"는 발표된 논문들에서 확인할 수 없는 기능을 의미한다. <표 1>로부터 웹 로봇들은 효율성, 지속성, 포괄성, 정숙성, 유일성에 주안점을 두고 개발되었고, 상대적으로 신선성을 향상시키기 위한 기능들이 개발되지 않았으며, 또한 국외에서 개발된 웹 로봇들은 안전성을 위한 기능들의 개발에 많은 관심을 두지 않고 있음을 알 수 있다.

〈표 1〉 웹 로봇의 분석 결과

평가 기준	기 능	웹 로봇			
		메르카토르	구글봇	폴리봇	네이봇
효율성	병렬 수집	○	○	○	○
	분산 수집	○	○	○	○
	분산 저장	○	○	○	○
	DNS 캐시	○	○	○	×
지속성	전체 URL 관리	○	○	•	○
	웹 문서 추가	•	•	•	○
신선성	수집 주기 관리	×	○	×	×
	URL 삭제	•	•	•	○
포괄성	정적 웹 문서	○	○	○	○
	동적 웹 문서	○	○	•	○
	일반 문서	○	○	•	○
정숙성	로봇 배제 표준	○	○	○	○
	수집 속도 조절	○	○	○	○
유일성	URL 정규화	○	○	○	○
	호스트 이름 그룹화	○	○	•	○
	중복 문서 검출	○	○	○	○
안전성	유해 웹 문서 검출	•	•	•	○
	스팸 웹 문서 검출	•	•	•	○
	블랙 리스트 관리	•	•	•	○

다음에서는 각각의 평가 기준과 연관된 웹 로봇들의 기능에 대하여 기술한다.

- **효율성** : 웹 로봇들은 병렬 수집, 분산 수집, 분산 저장 기능을 공통적으로 사용하고 있다. 또한, 메르카토르, 구글봇, 폴리봇은 DNS 캐시 기능을 이용하여 DNS 접근시 발생할 수 있는 병목 현상을 완화하고 있다.
- **지속성** : 폴리봇을 제외한 웹 로봇들은 최상위 URL 뿐만 아니라 전체 URL들을 관리하고 있다. 또한, 네이봇은 재수집에 실패한 웹 문서들을 이전에 수집한 웹 문서 데이터베이스로부터 추출하여 추가함으로써 지속성을 향상시키고 있다.
- **신선성** : 구글봇은 웹 문서들의 수정 주기를 측정하여 그 주기에 따라 웹 문서들을 수집하며, 다른 웹 로봇들은 모든 웹 문서들에 동일한 수집 주기를 적용한다.

또한, 네이봇은 주기적으로 웹 문서들의 접근 가능 여부를 확인하여 삭제되었다고 판단된 문서들의 URL을 URL 데이터베이스로부터 제거한다.

- **포괄성** : 모든 웹 로봇들은 HTML로 작성된 정적인 웹 문서들과 더불어 일부의 동적인 웹 문서들을 수집하고 있으며, 폴리봇을 제외한 웹 로봇들은 워드프로세서, PDF, PS 파일 형태로 인터넷에 공개된 일반 문서들도 수집하고 있다.
- **정숙성** : 웹 로봇들은 로봇 배제 표준을 준수함으로써 네티켓을 지키고 있으며, 접근한 웹 서버에 과도한 부하를 주지 않기 위하여 수집 속도를 조절한다. 메르카토르는 일정한 시간 내에 수집하는 문서들의 수를 제한함으로써 직접적으로 수집 속도를 조절하고, 폴리봇과 네이봇은 수집할 전체 URL들을 뒤섞음하여 간접적으로 수집 속도를 조절한다.
- **유일성** : 모든 웹 로봇들은 URL 정규화와 중복 문서 검출을 수행하며, 폴리봇을 제외한 웹 로봇들은 호스트 이름 그룹화를 수행함으로써 웹 문서의 유일성을 향상시키고 있다. 한편, 중복 문서를 검출하기 위하여 구글봇은 웹 문서들 사이의 유사도를 계산하고, 네이봇과 메르카토르는 각각 MD5와 Finger Print 알고리즘을 사용하여 웹 문서에 대한 요약 정보를 생성한다.
- **안전성** : 네이봇은 웹 문서를 수집하는 과정에서 웹 문서들의 내용을 분석하여 유해 웹 문서와 스팸 웹 문서를 검출하여 제거한다. 또한 블랙 리스트를 구축하여 유해 또는 스팸 사이트들에 포함된 웹 문서들의 수집을 사전에 차단한다.

5. 결 론

인터넷 상에 공개되어 있는 문서들에 대한 접근을 지원하는 웹 검색 서비스는 현대 정보화 사회에서 필수적인 역할을 수행하고 있다. 일반적으로 웹 검색 서비스는 웹 로봇을 통하여 인터넷 상의 웹 문서들을 수집한 후, 이들에 대한 검색을 지원한다. 따라서 웹 로봇의 성능은 웹 검색 서비스의 질을 결정하는 매우 중요한 요소이다. 그러나 웹 로봇에 대한 연구 및 개발은 주로 산업계에서 수행되었기 때문에, 연구 및 개발의 결과에 대한 공개가 부족하며, 특히 웹 로봇을 평가하기 위한 기준들과 웹 로봇에 포함될 기능들에 대한 정립이 미흡한 실정이다.

본 연구에서는 웹 로봇들의 성능을 체계적으로 평가하기 위한 기준으로서 효율성, 지속성, 신선성, 포괄성, 정숙성, 유일성, 안전성을 제시하고, 각 평가 기준의 향상에 도움이 되는 기능들을 기술하였다. 또한, 본 연구에서는 네이버, 구글, 알타비스타에서 상업용으로 이용되고 있는 웹 로봇들과 폴리테크닉 대학교에서 연구용으로 개발된 웹 로봇을 분석하여 이들에 구현된 기능들을 조사하였다. 본 연구에서 제시한 평가 기준과 기능들은 보다 개선된 웹 로봇의 개발에 기여할 것으로 기대되며, 향후 연구 과제로는 해당 평가 기준들을 향상시킬 수는 있는 구체적인 방법들을 제시하는 것이 필요하다.

참 고 문 헌

[1] M. Gray, "Internet Growth and Statistics: Credits and Background," http://www.mit.edu/people/mkgray/net/background.html

[2] M. Koster, "A Method for Web Robots Control," Network Working Group, Internet Draft, Dec. 1996, http://www.robotstxt.org/wc/norobots-rfc.html

[3] A. Heydon and M. Najork, "Mercator : A Scalable, : Extensible Web Crawler," InRecordings of the 8th World Wide Web Conference, Toronto, Canada, 1999.

[4] M. Najork and A. Heydon, "High-Performance Web Crawling," SRC Research Report 173, Compaq Systems Research Center, 2001.

[5] S. Brin and L. Page, "The Anatomy of a Large-Scale Hypertextual Web Search Engine," In Proceedings of the 7th International World Wide Web Conference, Brisbane, Australia, 1998.

[6] J. Cho and H. Garcia-Molina, "Parallel Crawler," In Proceedings of the 11th International World Wide Web Conference, Hawaii, USA, 2002.

[7] J. Cho, N. Shivakumar and H. Garcia-Molina, "Finding Replicated Web Collections," In Proceedings of the ACM SIGMOD International Conference on Management of Data, Dallas, Texas, 2000.

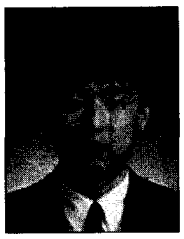
[8] S. Raghavan and H. Garcia-Molina, "Crawling the Hidden Web," Proceedings of the 27th International Conference on Very Large Databases, Rome, Italy, 2001.

[9] J. Cho and H. Garcia-Molina, "The Evolution of the Web and Implications for an Incremental Crawler," In Proceedings of the 26th International Conference on Very Large Databases, Cairo, Egypt, 2000.

[10] V. Shkapenyukn and T. Suel, "Design and Implementation of a High-performance Distributed Web Crawler," In Proceedings of the 18th International Conference on Data Engineering, San Jose, California, 2002.

김 광 현

e-mail : iamkkh@naver.com
1999년 숭실대학교 컴퓨터학부(학사)
2001년 숭실대학교 대학원 컴퓨터학과(석사)
2002년~현재 숭실대학교 대학원 컴퓨터학과 박사과정
관심분야 : 정보검색, 웹 로봇



이 준 호

e-mail : joonho@comp.ssu.ac.kr
1987년 서울대학교 컴퓨터공학과(학사)
1989년 한국과학기술원 전산학과(석사)
1993년 한국과학기술원 전산학과(박사)
1993년~1994년 한국과학기술원 인공지능 연구센터 연구원
1994년~1995년 코넬대학교 전산학과 방문연구원
1994년~1997년 연구개발정보센터, 선임연구원
1997년~현재 숭실대학교 컴퓨터학부 부교수
관심분야 : 정보검색