웹 접근성 향상을 위한 웹 서핑 도우미
(A Web Surfing Assistant for Improved Web Accessibility)

이 수 철 † 이 시 은 †† 황 인 준 †††
(SooCheol Lee) (Sieun Lee) (Eenjun Hwang)

요 약 정보의 급격한 증가로 인하여 웹에서 원하는 정보와 서비스를 찾고 이용하는 데 더 많은 시간이 소요되고 있다. 웹 상의 정보는 하이퍼링크(Hyperlink)를 통하여 여러 웹페이지에 걸쳐 표현되고 있으며 하나의 웹페이지는 여러 주제의 정보를 포함하고 있다. 그러나 대부분의 웹 탐색 도구들은 이러한 웹 저작경향을 반영하지 않고, 웹페이지를 독립적인 정보의 단위로 다루고 있다. 이러한 차이로 인해 사용자는 정보를 검색하고 브라우징하는 데 어려움을 겪는다. 본 논문에서는 웹 정보에 대한 접근성 향상을 위해 테이블 구조를 가지고 있는 여러 웹페이지에 흩어져있는 정보들을 모아 하나의 *컬렉션페이지*(*collection page*)에 새롭게 구성하여 제공하는 *LinkBroker* 시스템을 제안한다. 본 시스템은 논리적으로 연결된 정보를 담은 페이지들을 추출한 뒤 페이지셋으로 묶어 검색과 북마크의 단위로 다룬다. 그리고 페이지셋에 속한 페이지 내의 주요 컨텐츠 구역들을 이용해 컬렉션페이지를 생성함으로써 흩어져있는 정보를 하나의 페이지에 표현한다. 다양한 검색 목적에 부합되는 실험을 통해 페이지셋 단위의 검색과 북마크의 효율성 그리고 컬렉션페이지를 통한 정보 접근성의 향상을 확인할 수 있다.

키워드 : 의미구역, 웹 탐색, 접근성, 정보 검색, 북마크

Abstract Due to the exponential increase of information, search and access for the Web information or service takes much time. Web information is represented through several web pages using hyperlinks and each web page is contains several topics. However, most existing web tools don't reflect such web authoring tendencies and treat it as an independent information unit. This inconsistency yields inherent problems in web browsing and searching. In this paper, we propose a web surfing assistant called *LinkBroker* that provides *collection pages*. They are composed of relevant information extracted from several web pages that have table and frame structure in order to improve accessibility to web information. Especially, the system extracts a set of web pages that are logically connected and groups those pages using table and frame tags. Then, essential information blocks in each page of a group are extracted to construct an integrated summary page. It provides a comprehensive view to user and one cut way to access distributed information. Experimental results show the effectiveness and usefulness of LinkBroker system.

Key words : Semantic region, Web surfing, Accessibility, Information searching, Bookmark

1. 서 론

인터넷은 정보 전달과 서비스 이용의 수단으로써 큰 역할을 하고 있다. 그러나 기하급수적으로 증가하는 정보로 인해 웹에서 원하는 정보와 서비스를 찾아 이용하는 데 많은 시간과 노력이 소요되고 있다. 많은 웹 응용 프로그램들이 사용자가 필요한 정보들을 찾고 이용하는 것을 돕고 있지만, 복잡해진 웹 환경에서 효율적인 탐색 방법을 제공하기에는 충분하지 않다. 이는 HTML (Hyperlink Text Markup Language)의 불규칙적이고 비형식적인 특성으로 인해 정보 검색과 추출의 자동화에 많은 어려움이 있는 때문이기도 하지만, 웹사이트에서 다루어지는 컨텐츠의 양과 종류의 증가에 따라 웹페이지의 구성이 복잡해졌기 때문이기도 하다.

웹 접근성은 원하는 정보에 얼마나 빠르고 정확하게 접근할 수 있는지에 대한 정도를 일컫는다[1]. 현재의 웹 서핑 방식에는 다음과 같은 웹 접근성을 저하시키는 문제점이 있다.

· 본 연구는 과학기술부 국책연구개발 사업인 유전자원지원 활용사업단의 연구비(no. BDM0100211)의 지원에 의해 수행되었습니다.
† 학생회원 : 아주대학교 정보통신 전문대학원
juin@ajou.ac.kr
†† 비 회 원 : 삼성전자
sieun7.lee@samsung.com
††† 종신회원 : 고려대학교 전자공학부 교수
ehwang04@korea.ac.kr
논문접수 : 2003년 6월 25일
심사완료 : 2004년 7월 15일

첫째, 원하는 정보에 바로 접근하기 쉽지 않다. 이는 웹사이트에서 다루는 정보의 양의 증가로 인해 사용자가 사이트의 정보에 접근하기 위해 방문하는 첫 페이지에서 필요한 정보를 바로 수집하는 경우보다 여러 링크 경로를 거친 후에야 비로서 정보와 서비스에 접근할 수 있는 경우가 더 많기 때문이다. 또한 웹에서의 정보가 하나의 페이지에만 국한되는 것이 아니라 정보의 논리적인 흐름을 하이퍼링크와 프레임을 이용해 여러 웹페이지에 걸쳐 표현하고 있기 때문이다.

둘째, 하이퍼링크를 이용하는 페이지가 많기 때문에 사용자가 해당 링크에 연결된 페이지에서 다루는 정보에 대해 파악하는 것은 중요하나, 링크된 페이지의 컨텐츠를 미리 파악하는 것이 어렵고 부정확하다. 링크 텍스트나 이미지, 또는 <ALT> 태그를 통해 해당 페이지의 내용을 알 수 있지만, 전체 페이지의 내용을 다 표현하지 못하고 요약적인 수준의 정보만을 제공하기 때문에 의도하지 않은 정보를 담은 페이지로의 방문으로 시간을 소비하게 된다.

마지막으로, 구조를 알지 못하는 웹사이트 내에서 원하는 정보를 찾기가 쉽지 않다. 웹사이트에서 사이트 맵을 제공하지만 복잡한 사이트 내의 간략한 구조만을 담고 있어 원하는 정보를 찾기 어렵다. 사이트 내의 검색폼이나 사이트 검색 기능을 제공하는 검색엔진들은 단순히 검색된 페이지들의 링크와 몇 줄의 상응하는 문장을 보여주는 데 그치고 있어 페이지들의 관계를 알기 힘들며, 페이지들의 내용 또한 파악하기 어렵다.

또한 웹사이트 내의 구성은 대부분 사용자가 첫 페이지 즉 웹사이트의 메인 페이지로부터 방문하기 시작한다는 가정을 통해 작성되므로 검색엔진에서 반환하는 검색 결과의 링크를 통한 내부 페이지로의 직접 방문시 사이트 내의 해당 정보의 구성에 대해 짐작하기 어려울 수 있다.

이러한 문제점들은 실제 웹 저작환경과 웹 탐색도구들이 다루는 정보 단위의 차이로 인해 발생한다. 검색엔진과 브라우저 등의 웹 탐색도구들은 웹페이지를 정보의 단위로 다루고 있다. 그러나 실제 많은 웹 저자들은 정보를 체계적으로 저장, 관리하기 위해서 여러 페이지에 걸쳐 웹 정보를 표현하고 있다. 또한 각 웹페이지는 여러 종류의 정보를 담고 있기 때문에 정보의 논리적인 단위는 더 세부적이어야 한다.

본 논문에서는 이러한 웹 저작 경향을 반영해 웹 상의 정보 접근의 어려움을 해결하고자, 실제 정보의 단위를 정의하고 이를 검색과 북마크 그리고 브라우징의 단위로 활용하는 LinkBroker 시스템을 제안한다. 우리는 링크를 통해 여러 페이지에 걸쳐 표현되어 있으나 논리적으로는 하나의 문서로 볼 수 있는 정보 묶음을 페이지셋(pageset)이라 정의한다. 페이지셋을 검색의 단위로 이용함으로써 연관된 정보를 함께 검색할 수 있도록 하여 검색의 정확도를 높이고, 검색 결과에 대한 이해도를 향상시킨다. 그리고 페이지셋 단위의 북마크를 통해 자주 이용하는 정보에 대한 접근성을 높인다. 또한 페이지셋에 속한 여러 웹페이지들에 흩어져 있는 정보를 미리 가져와 하나의 컬렉션페이지를 구성해 보여줌으로써, 사용자가 여러 링크 경로를 거쳐야 하는 불편함을 줄이고 해당 정보 페이지들의 내용을 파악할 수 있도록 돕는다.

본 논문의 나머지 구성은 다음과 같다. 2장에서는 웹의 접근성 향상을 위한 관련 연구들에 대해 살펴본다. 3장에서는 페이지셋 단위의 검색 방법에 대해 설명하고, 4장에서는 페이지셋 단위의 북마크 기법에 대해 논한다. 5장에서는 전체 시스템의 구조와 기능들에 대해 기술한다. 6장에서는 시스템의 성능 평가를 위한 실험과 결과에 관해 살펴보고 끝으로 7장에서는 결론에 대해 서술한다.

2. 관련 연구

웹 접근성 향상을 목적으로 많은 연구가 진행되어 왔다. 특히 인공지능과 웹에서의 정보 통신 기술에 대해 다루고 있는 WI(Web Intelligence) 분야에서는 웹 정보에 대한 마이닝, 검색, 웹 에이전트 등의 다양한 연구가 이루어지고 있다[2]. 이들 연구는 기존의 웹 정보를 다루는 관점에서 벗어나 여러 관점에서 웹 접근성을 향상시키고자 하였다.

2.1 논리적인 정보 단위의 추출

Web Skimming[3]은 사용자 질의를 입력받아 관련있는 웹페이지들의 흐름을 지정하는 컨텍스트 패스(context path)를 찾아 정보의 흐름 순으로 해당 페이지들을 보여준다. 이를 이용해 사용자는 웹사이트 내에서 원하는 정보들의 구성을 쉽게 찾을 수 있다. Wen-syan L. 등은 관련된 웹페이지들의 묶음을 하나의 논리적인 정보의 단위로 보고, 이 연관된 정보를 담은 페이지들을 검색한다[4]. 이를 통해 논리적으로 연관된 정보들을 함께 볼 수 있는 장점이 있다. Xiaoli Li 등은 검색 시 하나의 웹페이지에 여러 주제의 정보가 함께 담길 수 있음을 고려해 페이지 내의 정보 단위를 검색에 활용하였다[5]. 이는 사용자의 질의어 분포가 하나의 구역에 밀집할수록 페이지에 포함된 질의어들이 같은 주제의 정보를 나타낼 가능성이 높음을 고려한 것으로, 프레임과 테이블 구조를 이용해 한 페이지 안에 많은 정보를 담는 경향을 반영한 효율적인 검색 방법을 제시하였다.

위의 연구들에서 지적하였듯이 웹페이지는 단말기의 화면에 보이기 위한 하나의 브라우징 단위이기 때문에 정보의 단위로 간주하기에는 여러 한계가 있다. 우리는

이러한 한계들을 극복하고자 한다는 점에서 위의 연구들과 그 목적을 같이한다. 다음은 웹페이지를 검색의 단위로 다룸으로써 발생하는 정보 검색의 한계를 크게 검색의 정확도 측면과 검색 결과에 대한 이해도 및 접근성 측면으로 보고, 이를 극복하고자 본 논문에서 제안하는 방법들이다.

첫째, 여러 웹페이지들에 나뉘어 기술되어 있으나 논리적으로는 하나의 문서로 볼 수 있는 정보들에 대한 정확한 검색을 위해, 정보 묶음을 이루는 페이지셋을 검색의 단위로 이용한다. 페이지셋 단위로 사용자 질의와의 유사도(similarity)를 계산하여 순위를 매겨 제공함으로써, 관련높은 정보를 상위에 랭크되도록 하여 검색의 정확도를 높인다.

둘째, 검색엔진의 검색 결과는 검색된 각 페이지의 URL로 이동할 수 있는 링크를 제공한다. 그러나 웹사이트 내의 구성은 대부분 첫 페이지 즉 웹사이트의 메인 페이지로부터 방문하기 시작한다는 가정을 바탕으로 작성되므로 검색 결과를 통해 웹사이트의 내부 페이지로 직접 방문시 원하는 정보로의 정확한 이동을 보장하지 않는다. 이 때문에, 방문 페이지 외에 연결된 다른 페이지들에 대한 정보를 쉽게 얻지 못하며 사이트 내의 해당 정보의 구성에 대해 짐작하기 어려울 수 있다. 우리는 페이지셋 추출을 통해 관련된 정보들을 함께 검색할 수 있도록 돕고, 페이지셋에 포함된 주요 컨텐츠들을 추출해 컬렉션페이지를 생성함으로써, 단편적인 부분들로 이루어진 기존 검색엔진의 결과 페이지의 정보 접근 한계를 극복하고자 하였다.

셋째, 하나의 웹페이지는 여러 종류의 정보를 포함하고 있어 사용자가 요구한 정보와 관련없는 정보들이 검색에 영향을 미친다. 페이지 내의 관련없는 여러 정보들을 동일 주제의 정보로 간주하면 검색의 정확도를 낮추고, 쓸모없는 정보들을 검색 결과의 상위에 랭크시킴으로써 검색 결과 내부에서의 웹 탐색시간을 소모시킨다. 우리는 페이지셋 내의 반복되는 정보와 광고 등의 불필요한 정보 구역을 찾아 제거하고 관련없는 정보구역에 의한 영향을 낮춤으로써 페이지셋을 정제하여 검색의 정확도를 높이고자 하였다.

2.2 재 방문 시의 웹 접근성

북마크 기능의 가장 큰 한계는 사용자가 직접 페이지를 북마크해야 한다는 점에서 수동적이라는 것이다. 또한 북마크 저장의 단위가 하나의 페이지 단위로 고정되어 있기 때문에 사용자의 관심 영역을 구체적으로 반영하지 못하는 문제가 있다. 이와 같은 문제점을 개선한 연구가 진행되어왔다.

Shaun K. 등은 "back", "history", 그리고 "bookmark" 기능을 통합해 사용자의 별도의 요청이 없이도 자동으로 사용자가 자주 방문하는 페이지를 북마크한다[6]. Tsuyoshi E. 등은 페이지 내의 관심있는 문장에 북마크할 수 있도록 함으로써 재 접근 시 해당 컨텐츠의 위치로 빠르게 이동할 수 있도록 한다[7]. 따라서 텍스트가 많은 페이지 내에서 정보의 재접근 시간을 단축시켜 준다. 이를 통해 새롭게 갱신된 부분에의 접근도 지원해 신문과 같이 내용 갱신이 잦은 페이지를 다시 읽어 내려가야 하는 불편함을 감소시키는 방법을 다루고 있다. Web VCR[8]은 웹에서의 정보가 정보의 논리적인 흐름을 따라 하이퍼링크와 프레임을 이용해 여러 웹페이지에 걸쳐 표현하고 있기 때문에 필요한 정보를 얻기위해 여러 페이지를 링크를 따라가며 일일이 열어보아야하는 불편함을 개선시키고자 하였다. 이를 위해 북마크 시 바로 서비스에 접근하지 않고 로그인 등의 과정을 거쳐야 도달할 수 있는 서비스에 바로 접근할 수 있도록 사용자의 접근 경로, 이동 경로와 로그인 폼, 검색 폼 등에 입력하는 정보 등을 저장해 재 접근 시 사용자의 경로 대로 반복하여 바로 접근할 수 있도록 한다.

본 논문에서는 사용자가 여러 페이지들을 링크를 통해 이동한 경로를 저장하여 해당 정보에 재 접근시 저장된 링크 페이지들을 페이지셋으로 구성하여 제공한다. 이는 기존 브라우저 북마크의 단점인 저장된 북마크 페이지를 북마크 리스트 중에서 다시 찾아내는 과정이 힘든 점과 페이지 단위의 북마크로 사용자의 관심정보를 정확하게 저장하지 못하는 점을 극복하기 위한 것이다. 페이지셋 단위의 북마크를 통해 사용자는 자주 접근하는 정보에 대해 빠르게 접근할 수 있고 일일이 북마크된 정보를 찾는 것이 아니라 해당 정보가 담긴 사이트에 방문하였을 때 쉽게 저장된 북마크를 이용할 수 있다. 또한 여러 페이지에 흩어진 관련 정보를 컬렉션페이지를 통해 함께 볼 수 있어 불필요한 이동 횟수를 줄여준다.

2.3 웹 탐색을 위한 WI

현재의 웹 서핑의 문제점과 불편함을 개선하기 위해 사용자의 웹 탐색을 도와주고 조언해주는 시스템에 대한 연구 또한 많이 이루어지고 있다[9]. 링크된 페이지의 내용을 미리 제공하여 사용자의 웹 탐색을 돕는 연구로서 MS WebScout[10]은 마우스 포인터가 링크 텍스트 위에 위치했을 때 링크 페이지를 이미지 형태로 미리 가져와 보여준다. 현 브라우저의 사용자가 접근했던 링크에 대해 색을 달리하여 구분케하는 방식에서 나아가 마우스를 링크에 대면 해당 링크에 대한 페이지 저자, 표현 언어, 그리고 접근 예상 시간 등의 정보를 보여주는 연구도 진행되었다[11].

본 논문에서는 페이지셋 내의 웹페이지들의 주요 정

보 구역들을 추출해 컬렉션페이지를 생성한다. 이를 통해 페이지들을 일일이 방문하지 않고도 페이지셋에 속한 페이지들의 내용에 대해 파악할 수 있게 돕는다.

2.4 웹 정보의 표현

좀 더 이해하기 쉬운 정보 형태를 제공하기 위해 웹페이지를 분석하여 편리한 인터페이스를 제공하고자 하는 연구 또한 많이 진행되었다. 예를 들어 Hunter Gatherer[12]는 여러 웹페이지로부터 문장, 이미지, 테이블 등의 정보 객체를 추출하여 하나의 페이지에 관리하는 기법을 제공한다. 웹페이지의 HTML코드 변환을 통해 원래의 구성, 형식과 다른 형태로 웹페이지를 변형하는 연구도 진행되어왔다. 대부분 HTML 코드 변환 과정은 여러 단으로 구성된 페이지를 하나의 단으로 만들고, 주석을 이용하여 중요도 높은 부분부터 위치하도록 하는 등 페이지의 구조를 바꾸는 데 이용된다. 또한 중복되는 부분을 제외하여 페이지의 복잡도를 줄여주는 방법으로 시각 장애인을 위해 간략한 형태의 문서로 만들어주거나 기기(device)의 차이를 고려하여 사용자의 기기에 알맞은 형태로 변환해 사용자 맞춤 변환 서비스를 제공할 수 있도록 한다[13].

본 논문에서는 페이지셋 내의 주요 정보 구역들을 이용해 변환된 페이지인 컬렉션페이지를 생성함으로써 정보 접근성을 향상시키고자 하였다.

3. 페이지셋 단위의 웹 정보 검색

페이지셋 단위의 검색은 웹페이지를 단위로 하는 기존 검색기법의 한계를 극복하고 검색엔진의 성능을 보완하기위해 논리적인 문서를 이루고 있은 페이지셋을 검색의 단위로 다룬다. 우리는 하나의 페이지로부터 일련의 링크된 페이지들을 탐색하여 사용자 질의와 관련된 페이지셋을 추출하고 페이지셋 단위로 질의와의 유사도를 측정하여 유사도 순위대로 결과를 제공한다. 이는 웹사이트 내에서 사용자가 원하는 정보에 쉽고 빠르게 접근할 수 있도록하고 복잡한 사이트 내에서 해당 정보와 서비스가 어떻게 관리되고 있는지 파악할 수 있도록 하여 구성을 알지 못하는 사이트 내에서의 검색의 효율을 높인다. 또한 검색엔진의 결과 페이지에 적용시, 검색의 정확도를 높이고, 검색 결과에 대한 이해도와 정보 접근성을 향상시킨다.

3.1 페이지셋 추출을 위한 웹 탐색 모델

페이지셋 추출을 위해 우리는 사용자 질의를 바탕으로 하나의 웹페이지로부터 시작되는 일련의 웹페이지들을 DFS(depth-first search)방식으로 탐색한다. 탐색 시작 페이지는 사용자가 지정한 웹페이지로써 웹사이트 내의 메인 페이지와 같이 사용자가 방문한 페이지 또는 검색엔진에서 반환한 결과 페이지 등이 될 수 있다.

웹페이지들을 연결하는 하이퍼링크는 웹 저자의 의도에 따라 위치한 것으로 페이지의 질을 결정하는 하나의 요소가 될 수 있다. 하이퍼링크의 쓰임은 크게 정보의 구조적, 순차적 구성을 위해 이용되는 경우, 관련있는 정보 및 참고자료를 위한 경우 그리고 광고와 같이 페이지의 주된 내용과 관계없는 페이지로의 이동을 위해 쓰인 경우 등으로 볼 수 있다.

하나의 페이지로부터 탐색할 수 있는 링크의 수는 페이지 당 수십개에 이르기도하기 때문에 이 링크들 중 질의와 관련된 중요한 링크와 그렇지 못한 링크를 구분해 방문할 링크들을 미리 결정하는 것은 중요하다. 하위 링크들에 대한 불필요한 탐색을 줄여 페이지셋 검색의 성능을 높이기위해, 우리는 페이지 내의 질의어 분포와 링크의 URL 디렉토리 구조, 그리고 페이지 내의 링크의 위치 등을 바탕으로 해당 링크에 대한 *방문가치*를 측정한다. 링크에 대한 방문가치는 곧 해당 링크로부터 연결되는 하위 페이지들로 이어지는 정보루트(information-route)의 가치를 의미한다. 표 1은 방문가치 결정에 영향을 미치는 요소들을 보여준다.

식 (1)은 표 1의 요소들을 이용하여 링크 i로의 방문가치 TV(i)을 계산하는 방법을 보여준다. 식 (1)에서 $Sim(P_1, L_i)$는 페이지 1과 페이지 1에 위치한 링크 i의 URL 디렉토리 구조의 유사도를 의미하고, $F(P_1, Q_j)$는 사용자 질의어 $Q_j=\{t_1, t_2, \cdots, t_n\}$가 페이지 1에 등장한

표 1 링크 방문가치 결정 요소

결정 요소	결정요소 설명	방문가치
링크 URL 디렉토리 구조	웹사이트 내에서 구조적으로 연관된 페이지들의 URL 주소값은 유사성을 보인다.	외부 사이트의 링크인 경우 광고 등 관련없는 페이지일 가능성이 높으므로, 방문가치를 줄인다.
사용자 질의어 분포	질의어를 포함한 페이지에 위치한 링크의 경우 관련정보를 담은 페이지로의 링크일 가능성이 있다.	관련있는 웹페이지가 관련된 링크를 가지고 있을 가능성이 높으므로, 발생한 질의어 수, 등장 빈도 등에 따라 페이지 내의 링크들의 방문가치가 결정된다.
링크 텍스트의 위치	페이지 내에서 질의어 가까이 위치한 링크들은 해당 질의어와 유사한 정보를 갖고 있을 가능성이 높다.	질의어가 등장한 의미구역과 같은 구역에 있는 링크의경우 같은 주제의 정보를 담고 있을 가능성이 높으므로 방문가치를 높여준다. 링크된 텍스트에 질의어가 등장한 경우는 질의와 매우 관련높은 페이지에 대한 링크이므로 방문가치를 높여준다.

각 질의어의 빈도수와 등장한 질의어 가짓수를 이용해 계산된 질의어 발생도이다. 질의어 발생도는 질의어의 빈도수보다는 가짓수에 더 비중을 둔다. 그리고 Dist(L_i, Q_j)은 링크 i가 페이지 1에 등장한 사용자 질의어의 페이지 내에서의 거리를 뜻한다. A, B, C는 각 요소의 중요도를 반영하기 위한 변수들이다.

$$TV(i) = A \times Sim(P_1, L_i) + B \times F(P_1, Q_j) + C \times Dist(L_i, Q_j) \quad (1)$$

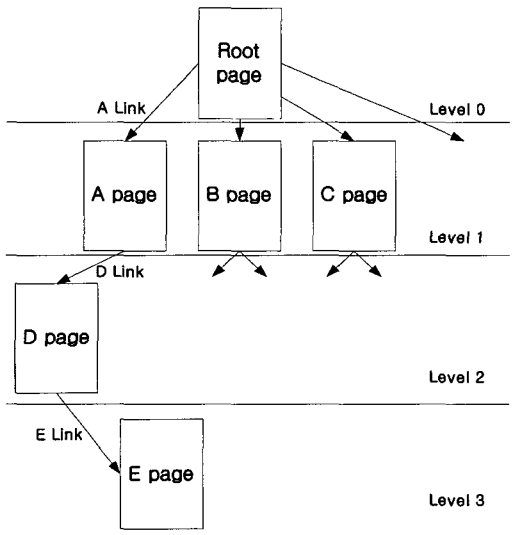
탐색 시작 페이지에서 계산된 링크에 대한 방문가치는 곧 해당 링크로부터 연결되는 하위 페이지들에 대한 정보루트의 가치를 뜻한다. 하위 정보루트의 가치가 높다고 판단되면, 정보루트에 속해있는 웹페이지들에 대한 탐색시 사용자 질의를 포함하지 않더라도 상위에서 판단했던 가치를 고려해 탐색을 중지하지 않는다. 또한 하위 정보루트의 가치가 낮더라도 정보루트 탐색시 웹페이지가 질의를 포함하고 있다면 이곳으로부터 이어지는 하위 정보루트의 가치를 높여준다. 즉, 하위 정보루트로의 탐색가치는 상위 노드 페이지에서 계산된 정보루트의 탐색가치를 바탕으로 하위 링크의 탐색가치를 동적으로 다시 계산하여 얻어진다.

그림 1은 정보루트의 탐색가치가 상위 링크로부터의 영향과 각 루트에 포함된 링크의 방문가치에 따라 결정되는 과정을 보여주고 있다. 그림 1의 레벨 0의 페이지가 포함하는 3개의 링크는 페이지의 질의어 분포와 링크 URL 디렉토리 구조에 따라 각각 다른 방문가치를 갖게 된다. A 페이지로의 이동을 위한 링크 A의 경우 세 링크들 중 가장 높은 방문가치를 갖고 있다고 가정하면, 이 링크로부터 연결되는 하위 페이지들에 대한 정보루트의 탐색가치가 높아지게 된다. A 페이지가 포함한 링크들은 상위 링크 즉 A링크가 갖는 하위 정보루트에 대한 탐색가치를 기반으로 하위 링크들에 대한 방문가치를 계산한다. D 페이지로의 링크인 링크 D의 경우, 상위의 탐색가치값을 물려받아 1/2배한다. 이는 A 페이지에서 D 페이지로 한 단계를 탐색하였으므로 상위에서 결정한 탐색가치를 줄이기 위한 것이다. 이 값에 D 페이지에서 표 2의 요소들을 바탕으로 계산한 하위 링크로의 방문가치값을 더하여 다음 링크로부터 시작되는 정보루트의 탐색가치가 결정된다. 표 1을 이용한 하위 링크의 방문가치값은 음수로 계산될 수 있어 정보루트의 탐색가치값은 음수가 될수 있다. 정보루트의 탐색가치가 기준치 이하일 경우 해당 루트로의 탐색을 중지한다.

3.2 페이지셋 추출 알고리즘

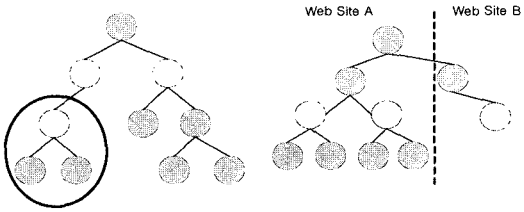
3.1장에서 기술한 조건들을 통해 탐색된 페이지들은 페이지셋에 속하게 될 후보 페이지들이다. 이들 중 정보단위로서 가치를 갖는 페이지 묶음을 분리하여 페이지셋을 추출한다. 즉 후보 페이지들을 페이지셋들로 나누기위한 *분리점(divide point)*이 필요하다. 분리점을 찾기위해 본 논문에서는 페이지셋의 크기 즉 페이지셋에 속한 페이지 수, 페이지셋의 깊이수(depth), 페이지의 URL, 페이지들간의 링크구조, 그리고 페이지에 질의어가 등장했는지의 여부를 고려한다. 그림 2는 탐색결과그래프에서 페이지셋을 추출하는 여러 경우의 분리점들의 예를 보여주고 있다.

그림 2에서 색칠된 노드는 질의어가 등장한 페이지를 뜻하며 색칠되지 않은 노드는 질의어를 포함하지 않은

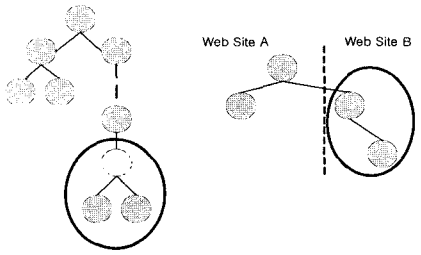


링크	링크로의 방문가치	링크로부터이어지는 하위정보루트의 탐색가치
A	2	2
D	- 0.5	2/2 - 0.5 = 0.5

그림 1 웹페이지 탐색



(a) 질의어 등장과 페이지 위치관계이용 (b) 질의어 등장과 URL 디렉토리구조이용



(c) 페이지셋, 사이즈, 깊이 이용 (d) 질의어등장과 URL 디렉토리구조이용

그림 2 페이지셋 추출을 위한 분리점의 예

노드이다. 질의어를 포함하지 않은 페이지의 경우, 질의어는 없더라도 사용자 검색 의도와 부합되는 정보를 가졌을 수 있고, 단지 하위의 관련된 페이지들을 위한 연결의 역할만을 위해 존재할 수도 있다. 이 두 가지 경우를 구분하기 위해 우리는 질의어를 포함하지 않은 페이지의 하위 페이지들을 살펴본다. 이때 하위 페이지또한 질의어를 포함하지 않는다면 이는 상위의 페이지와 하위의 페이지들 사이에 관련이 적다고 판단하여 그림 2의 (a)와 같이 분리한다. 그리고 페이지들의 URL을 바탕으로 탐색시작 페이지와 다른 사이트에 속한 페이지들의 경우 다른 페이지셋으로의 분리 여부를 고려한다. (b)의 경우, 다른 웹사이트에 속했으나 해당 페이지로부터 이어지는 페이지에 질의와 관련된 정보가 없을 때는 다른 웹사이트의 페이지이더라도 분리하지 않는다. 그러나 (d)에서와 같이 다른 웹사이트의 페이지들이 질의를 포함하는 페이지들을 하위에 둘 경우에는 분리한다. 이는 해당 정보들이 독립적으로 페이지셋을 이룰 수 있기 때문이다. 또한 (a), (b), (d)와 같은 조건이 발생하지 않더라도 페이지셋의 사이즈나 전체 깊이가 일정값이상이 되면 분리점이 될 수 있는 질의어가 등장하지 않은 페이지를 찾아 분리한다.

3.3 의미구역기반의 페이지셋 정제

페이지 상에서 불필요한 정보 중 하나는 페이지셋의 페이지들에서 계속 반복되는 정보구역이다. 즉 웹사이트의 이름 등을 나타내는 타이틀, 메뉴 링크들, 사이트 안내를 위한 글 등의 중복되는 부분에 사용자의 질의어가 존재한다면 해당 점수는 반복되어 계산되어 실제 관련성에 비해 높은 유사도를 갖게 된다. 따라서 이러한 중복되는 내용을 검색 대상에서 제외시킨다면 검색의 정확도를 높일 수 있다. 또한 외부 사이트로 이어지는 링크들을 담은 그림 및 텍스트로 이루어진 구역의 경우, 광고 페이지로의 링크일 가능성이 높으므로 제거하거나 낮은 점수를 부여하여 이들로인한 검색의 영향을 줄인다.

각 페이지의 주요 정보와 불필요한 정보의 구분을 위해 우리는 웹페이지를 여러 *의미구역(semantic region)*으로 나눈다. 이는 하나의 웹페이지는 여러 주제의 정보를 담고 있다는 가정에서 출발한다. 웹페이지의 저자는 사용자가 이러한 많은 주제의 정보들의 구성과 내용을 쉽게 파악할 수 있도록 프레임과 테이블 구조를 이용해 웹페이지를 구성하는 정보가 시각적으로 구역을 이룰 수 있도록 한다. 즉 같은 주제나 분류의 정보들을 하나의 구역 안에 묶어 정리해 놓음으로써 페이지 내에서 원하는 정보와 서비스를 찾기 쉽도록 한다.

의미구역은 HTML 페이지의 내용과 구조적인 정보를 이용하여 추출된다. 많은 웹페이지는 사용자에게 이해하기 쉽고 보기 좋은 구조를 제공하기 위해 <table> 태그를 이용해 여러 겹의 중첩 테이블 구조를 갖는다. 이 테이블 구조는 계층적 특성 때문에 트리 구조로 표현될 수 있고, <table>, </table> 태그로 구분되는 웹페이지의 컨텐츠는 이 트리의 노드들로 나누어진다. 그런데 많은 노드들의 컨텐츠는 독립적으로 정보를 전달하기에는 너무 적은 내용을 가졌고, 단지 페이지의 구조를 형성하기 위해 쓰이고 있는 노드 또한 많기 때문에 실제로 사용자가 웹페이지의 컨텐츠를 의미구역들로 구분하는 단

표 2 의미구역 형성을 위한 요소

태그(Tag) 구조	제목, 부제목 등과 그 하위 내용들의구분을 위해 이용하는 태그들인 <P>, <H1>, <HR> 등의 구조가 유사한 노드일수록 같은 주제의 정보일 가능성 높음
노드 위치	형제 노드보다 자식노드가 일관된 정보 표현을 위해 많이 이용되므로 병합노드 선택시 자식노드에 비중을 둠
컨텐츠 종류	컨텐츠의 배경색, 텍스트 크기, 글자체 등의 유사도. 링크 텍스트와 일반텍스트의 비, 그림 등의 멀티미디어 자료 바탕으로 함
컨텐츠 양	적은 양의 자료를 담은 노드는 주변노드와 병합

위는 노드 단위가 될 수 없다. 따라서 적은 양의 컨텐츠 정보를 갖고 있은 노드는 관련된 내용을 갖고 있은 주변 노드들과 병합하여 의미구역을 형성하도록 한다. 이때 주변 노드들 중 병합될 노드를 결정하기 위해 표 2의 요소들을 이용한다.

병합될 노드는 두 노드의 유사도를 계산하여 선택된다. 노드의 태그 구조가 유사할수록 그리고 컨텐츠 종류의 유사도가 클수록 연속된 주제의 내용이다. 또한 노드의 테이블 내에서의 위치를 고려하여 형제 노드와 자식 노드가 병합 될 노드와 일관된 정보를 담고 있다고 가정하고 주변 노드 중 형제 또는 자식 노드에 비중을 둔다. 노드의 컨텐츠 양과 전체 페이지의 컨텐츠 양의 비가 임계값 이상이 될 때까지 병합할 주변 노드를 탐색한다. 이렇게 병합된 노드들은 의미구역을 이룬다. 테이블 구조를 이용하지 않는 페이지의 경우에도, 제목과 내용과의 관계, 구성 관계등에 따라 의미구역을 나눌 수 있다. 본 논문에서는 테이블 구조를 가진 웹페이지들을 가정하였다. 그림 3은 웹페이지를 분석하여 전체 컨텐츠를 의미구역들로 나누는 알고리즘을 설명하고 있다.

우리는 앞장에서 소개한 링크 방문가치와 정보루트

```
Function SemanticRegion_Extraction

Variable
  node[] : node[i] contains the ith node's information  - depth of the node in the  table tree,
           amounts of the several kinds of contents, text style such as font, size in the node
  group[] : group[i] contains the ith group's information - nodes are consisting of the group

  C_NodeSim : the threshold of node similarity
  C_NodeAmount : the threshold of node amounts

(1) Parse the web page and construct the tree structure
   call NodeAnalyze(web page URL, node[], group[])

(2) Merge the adjacent nodes into a group
   While (every node)
    // the criteria for selecting node among the nighbor ones to group into the semantic region
              If (node[i].amount_of_contents < C_NodeAmount)
              then C_NodeSim <- low value
              else then C_NodeSim <- high value
              // larger the node size, getter the high posibility to be independent semantic region,
              // therefore, the criteria of similarity could be smaller value.

              foreach the nodes between the current node,
                  call SimOfNode()
         compare the i-1 th node, i+1 th node with the i th node
              if(similarity for node[i-1] or node[i] is larger than C_NodeSim)
              then  choose the most similar node
         call  Combine_Node()


Function NodeAnalyzer()
Input web page URL
 (1) parse the target web page to find the <body>,<table>,</table>,</body>
 (2) construct the table tree structure
    while()
     if(found the table tags)
                then   new node is created, node information is stored in node[i]
                     - node depth, amounts of contents(text, linking text, multimedia)
                     - tag structure(tag list in the node such as <P>, <H1>..)
    until found the </body> tag


Function SimOfNodes()
Input node[i], node[j]
  (1) compare the node information and compute the similarity for several factor
     Sim_Depth : the depth of node j is higher than the node i, these nodes are more similar
              Sim_Content : whether the ratio of contents (text, linking text, multimedia) is similar
              Sim_Tag : whether the kinds of the tags in the node is similar
              Sim_Text : whether the font, size, background color is similar
  (2) sum of the several similarity factors


Function NodeBinder()
Input node[i], node[j]
  the node i, node j would be combined and
  then the group information is stored in the group[k]
```

그림 3 페이지 분석을 통한 의미구역 추출

탐색가치의 계산을 위해서도 의미구역을 이용하였다. 이 뿐아니라 의미구역은 검색된 페이지셋과 질의와의 유사도 측정의 정확도를 높이기위해서도 이용된다. 하나의 웹페이지에는 여러 종류의 정보가 담겨있음에도 기존 검색기법에서는 페이지 상의 모든 컨텐츠들을 동일하게 다룬다. 그러나 하나의 주제의 정보컨텐츠에 질의어들이 등장한 경우와 다른 주제의 정보 컨텐츠들에 질의어가 등장한 경우는 다르게 취급되어야한다. 따라서 우리는 추출된 페이지셋에서 불필요한 정보를 담은 의미구역을 제거하고 관련성이 낮은 구역에 대해 낮은 점수를 부여하여 검색의 정확도를 높이고자 한다.

3.4 의미구역 기반의 페이지셋 유사도 계산 알고리즘

페이지셋과 질의와의 유사도는 크게 다음 세가지의 과정을 통해 계산된다. 첫째, 페이지셋에 속한 각 페이지를 여러 의미구역으로 나눈다. 그리고 페이지 내의 의미구역과 질의와의 유사도를 계산한다. 둘째, 의미구역의 유사도 점수를 기반으로 의미구역이 속한 페이지와 질의와의 유사도를 계산한다. 마지막으로, 페이지셋 유사도는 페이지셋에 속한 각 페이지와 질의와의 유사도를 바탕으로 계산된다.

그림 4는 페이지를 의미구역으로 나누어 유사도를 계산하는 이유를 보여주고있다. 그림 4는 질의 Q를 이루는 질의어 t1, t2, t3, t4의 웹페이지 내에서의 분포를 도식화 한 것으로, 두 페이지는 극단적인 경우를 보여주고 있다. 두 페이지 모두 사용자의 질의에 포함된 모든 질의어를 가지고 있고 각 질의어의 등장 빈도수 또한 같지만, 사용자의 질의어가 의미구역들에 분포된 위치는 상반된다. 질의어가 하나의 의미구역에 나타날수록 해당 의미구역이 사용자 질의와 밀접하다는 것은 직관적으로 알 수 있다. 또한 같은 구역에 위치할수록 각 단어들이 하나의 주제를 가지고 표현된 것이라 볼 수 있기 때문에 오른쪽 페이지의 경우처럼 질의어가 하나의 구역에 모여 있을수록 질의와 더 유사한 문서이고 따라서 결과에서 더 높은 순위에 랭크되어야 한다. 즉 질의어들이 여러 의미구역에 흩어져 있을수록 이는 다른 주제의 단어들일 가능성이 높기 때문에, 같은 수의 질의어가 같은 빈도로 나타나더라도 하나의 구역에 모여있는 경우가 더 관련성 높은 페이지라 할 수 있다. 본 논문에서는 이와 같은 가정을 바탕으로 웹페이지를 의미구역으로 나누어 사용자 질의와의 유사도를 측정한다.

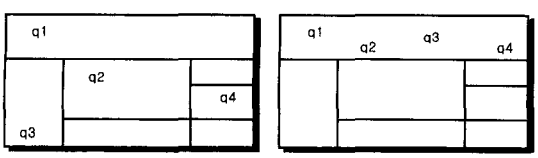


그림 4 페이지 상의 질의어 분포

식 2는 페이지 내의 각 의미구역과 질의와의 유사도를 계산하는 방법을 보여준다. R은 의미구역과 질의와의 유사도이며, N은 의미구역 내에 등장한 질의어의 개수 즉 가짓수이다. 그리고 F는 의미구역 안에서 각 질의어의 등장 빈도수 중 가장 작은 빈도를 보인 질의어의 빈도수 값이다. 웹 검색을 위한 사용자 질의는 비교적 짧기 때문에 사용자가 요구한 질의어가 모두 등장하는 것이 질의어의 빈도수보다 중요하므로[14] 이를 반영하여 각 요소들 중 N에 가중치를 주었다.

$$R = N*2 + F/2 \tag{2}$$

이때 해당 의미구역이 페이지의 메인 컨텐츠를 담은 구역이면 유사도 값에 가중치를 준다. 이는 페이지에서 해당 질의와 관련된 정보를 중점적으로 다룰 경우 더 가치있는 페이지이기때문이다. 메인 컨텐츠 구역은 페이지의 의미구역들 중 가장 많은 컨텐츠를 가지고, 가장 적은 수의 링크를 가진 구역을 추출하여 얻어진다.

페이지와 질의간의 유사도는 페이지 내의 의미구역들과 질의와의 유사도 R을 이용해 계산된다. 가장 높은 유사도를 보인 의미구역이 페이지 내에서 질의에 대한 가장 중요한 구역이므로 이 구역의 유사도와 나머지 의미구역들의 유사도의 평균을 더해 웹페이지와 질의와의 유사도를 얻게된다. 이는 페이지 내의 의미구역의 수에 의해 유사도값이 영향받지 않도록 하기 위함이다. 식 (3)에서 $Sim(P_i, Q)$는 페이지 P_i와 사용자 질의 Q와의 유사도를 뜻한다. R은 식 (2)에서 얻은 의미구역과 질의와의 유사도이다.

$$Sim(P_i, Q) = Max(R) + \frac{\sum R_i}{Num(R_i)} \tag{3}$$

또한 이 페이지로의 이동을 위해 상위 노드 페이지에 포함된 링크 텍스트 및 앵커 텍스트가 사용자의 질의어를 포함한다면, 이 페이지는 질의와 관련 가능성이 높다. 따라서 이러한 페이지의 경우 페이지의 유사도에 가중치를 준다. 그림 5는 링크 검색을 통해 의미구역과 질의 그리고 페이지와 질의의 유사도를 계산하는 알고리즘을 보여준다.

페이지셋과 질의와의 유사도는 페이지셋에 포함된 페이지들의 질의와의 유사도와 페이지들의 링크 구조를 이용해 계산된다. 페이지셋의 크기, 각 페이지의 유사도 점수 그리고 페이지셋 내에 질의를 포함한 페이지의 수와 그렇지 못한 연결고리로써의 페이지들의 수를 이용해 페이지셋의 유사도를 계산한다. 식 (4)의 $Sim(PS_i, Q)$는 질의 Q와 페이지셋 PS와의 유사도를 뜻하며, P_R은 페이지셋 중 질의어가 등장한 페이지의 질의와의 유사도 점수를 뜻한다. 우리는 각 페이지의 점수를 제곱하

```
Function MeasureSimilarityOfPageSet()
Input : start web page URL

(1) traverse the page set and calculate the similarity score of the pages
    foreach pages call SimOfPage ()
(2) calculate the Similarity of the pageset
    for( all page in a pageset)
        if(page have the query term in the page set)
            count_relevant_page ++
        Similarity of the pageset <- Similarity of the pageset + the score of page
    Similarity of the pageset <- Similarity of the pageset + (0.2*count)+0.1*(size of pagesize-count)

Function SimOfPage()
Input : web page URL

(1) parse the target page
    call SemanticRegion_Extraction()
(2) Look for the query term in each semantic region
    Find the minimum frequency among frequencies of query terms for each regioon
(3) Find the main semantic region

(4) measure the similarity of region
    for(i=0;i<totalRegionNum;i++)
        for(j=0;j<totalQueryNum;j++)
            if(Frequency for query j in region I is higher than 0)
                count the number of query term occurred in a region

    for(i=0;i<totalRegionNum;i++)
        calculate relevance score of each region
        // the number of occurring query term*2 + minimum frequency/2.0
        // if the query is composed only one query term,
        // the equation is occurring query terms + minimum frequency/2.0

    if (region is the main region)
        relevance score of region = relevance score of region * 1.2
        // give a weight the region that contains main region

(5) Find the semantic region whose relevance score is highes // most matched region
(6) calculate the page relevance
    the score for most matched region * sum of other region's score/(totalRegionNum-1)
    if (linking text of this page contains query term)
        relevance score of page = relevance score of page * 1.5
```

그림 5 페이지셋과 질의와의 유사도 계산 알고리즘

여 더함으로써 유사도 높은 페이지를 포함한 페이지셋 일수록 더 높은 순위에 랭크될 수 있도록하였다. 또한 페이지셋 의 크기로 나눔으로써 셋의 크기 차이에의한 영향을 줄이고자하였다. 질의어가 발생한 페이지수가 많을수록 이는 관련된 페이지셋일 가능성이 높으므로 가산점을 주었다. size(PS)는 페이지셋에 속한 페이지의 수이고 num(P_R)은 그중 질의어가 등장한 페이지의 수이다.

$$Sim(PS_i, Q) = \frac{\sum (P_R)^2}{size(PS)} + (0.2 \times num(P_R)) + (0.1 \times (size(PS) - num(P_R))) \quad (4)$$

그림 6은 식 4를 이용한 페이지셋과 질의와의 유사도 계산의 예를 보여주고 있다. 그림에서 보듯이 직관적으로 알 수 있는 페이지의 순위는 1, 2, 3, 5, 4이다. 식 4를 이용한 페이지셋의 유사도는 이와 같은 순서로 중요한 페이지들의 순위를 구분해내어 페이지셋을 이용해 일반 검색엔진을 통해 반영할 수 없는 정보의 가치를 잘 찾아주고 있다.

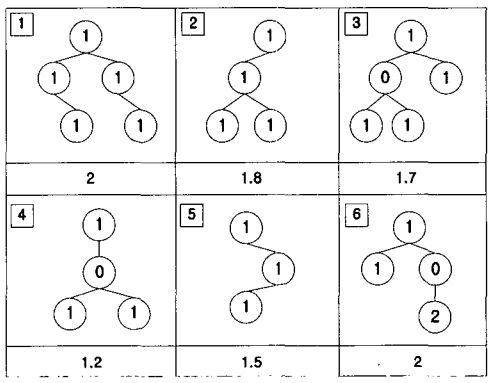


그림 6 페이지셋과 질의와의 유사도 예

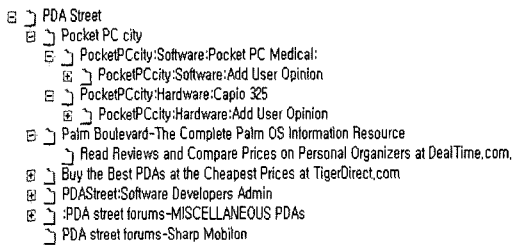
3.5 컬렉션페이지

기존의 검색엔진은 전체 웹을 대상으로 하기 때문에 많은 수의 결과 문서를 반환한다. 따라서 방대한 결과 중에서 원하는 정보를 찾기까지 많은 시간이 소요된다. 그러나 웹사이트와 같이 비교적 좁은 범위의 문서들은

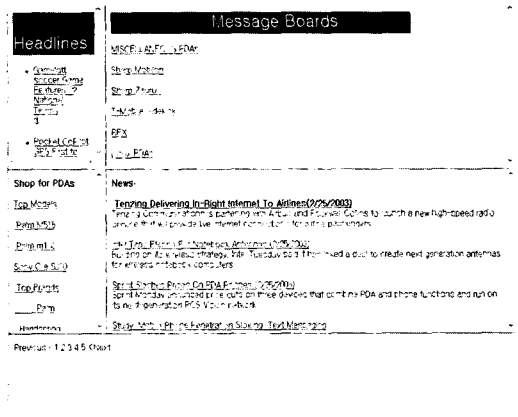
서로 관련된 범주의 페이지들일 가능성이 높기 때문에 이들을 검색의 대상으로 다루게 되면 광범위한 웹의 검색에 비해 정확도가 높고 검색 결과 중에서 원하는 정보를 빠르게 찾을 수 있는 장점이 있다. 이뿐만 아니라 해당 정보와 관련된 여러 정보가 검색된 페이지에 연결되어 있을 가능성이 높기 때문에 구체적인 정보를 검색할 때에는 좁은 범위의 문서들에 대한 검색도 필요하다.

예를 들어, 검색엔진을 통해 검색된 웹페이지가 사용자의 의도와 유사한 정보를 담고 있다면 이 페이지가 링크하고 있는 정보들이나 이 페이지가 위치한 웹사이트의 다른 문서들도 필요로하는 정보를 담고 있을 가능성이 크다. 따라서 이 페이지로부터 시작되는 일련의 문서들에 대해 검색할 수 있는 방법이 있다면 구체적으로 원하는 정보에 접근할 수 있다.

이와 같은 정보 접근을 지원하기 위해 컬렉션페이지는 페이지셋에 속한 페이지들의 의미구역들 중 질의와 가장 관련 깊은 의미구역들을 이용해 형성된다. 주요 의미구역들을 선정하기 위해 우리는 페이지 내 의미구역들과 질의와의 유사도를 계산하였다. 컬렉션페이지는 페이지셋의 정보에 대한 빠른 접근과 이해를 위해 페이지들의 링크관계를 추출하여 트리형태로 제공한다. 트리의 각 노드는 검색된 페이지로 이동할 수 있는 링크를 담고 있으며 페이지의 컨텐츠에 대한 이해를 돕기 위해 각 페이지의 제목, 앵커(Anchor) 텍스트 등의 정보를 담고 있다. 그림 7의 (a)는 페이지셋의 링크 구조를 보여준다. 또한 우리는 페이지들의 가장 관련 깊은 의미구역들을 이용해 컬렉션페이지를 생성한다. (b)는 주요의미구역을 이용한 컬렉션페이지를 보여준다. 이러한 컬렉션페이지를 통해 사용자는 보다 쉽게 웹사이트와 같은 페이지셋으로부터 원하는 정보를 찾을 수 있고, 하나의 페이지를 통해 여러 페이지의 내용을 동시에 볼 수 있다.



(a) 페이지셋들의 링크 구조

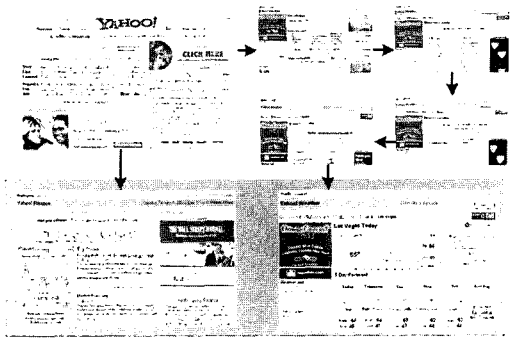


(b) 페이지셋에 속한 페이지들의 주요 의미구역을 이용한 컬렉션페이지

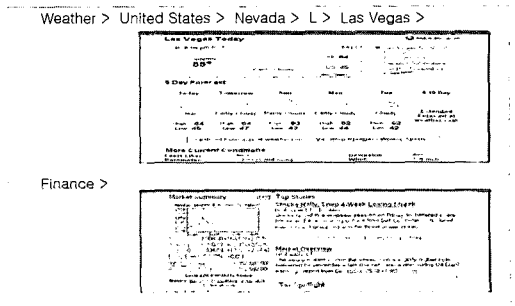
그림 7 컬렉션페이지의 예

4. 페이지 셋 단위의 북마크

본 장에서는 페이지셋 단위의 북마크 기법에 대해 설명한다. 자주 사용하는 페이지, 그 중에서도 특정 서비스를 제공하는 웹사이트의 경우 사용자는 자주 이용하는 구역, 링크들을 갖는다. 우리는 이러한 부분들을 하나의 객체로 보고 자동으로 추출하여 북마크한 뒤 사용자가 페이지를 재 방문했을 때 해당 페이지에서 관심을 보였던 의미구역과 이 페이지를 통해 접근한, 즉 이 페이지의 링크를 통해 접근하였던 서비스, 정보 페이지의 일부분을 미리 가져와 보여줄 수 있도록 컬렉션페이지를 생성한다.



(a) 시작 페이지로부터 진행되는 사용자의 정보 탐색 경로



(b) 컬렉션페이지

그림 8 사용자 접근 경로와 이를 반영한 컬렉션페이지

예를 들어 한 사용자의 웹사이트 내에서의 정보 탐색을 살펴보자. "www.yahoo.com"의 메인 페이지를 시작으로 사용자는 지역 날씨 정보와 증권 정보를 얻기 위해 여러 링크 경로를 따라간다. 그리고 필요한 정보를 얻은 후 다른 정보를 얻기 위해서 사용자는 시작 페이지로 가기위해 이전 페이지로 이동하거나 시작 페이지로 직접 갈 수 있는 링크를 이용한다. 사용자가 방문한 일련의 페이지들은 사용자가 관심을 가진 정보 묶음이다. Link Broker는 자주 방문하는 이러한 정보 묶음을 북마크하고, 사용자가 야후의 첫 페이지에 재 방문했을 때 사용자가 접근했던 링크들을 탐색해 링크된 페이지들에서 접근했던 의미구역 또는 메인 컨텐츠를 담고 있는 의미구역의 컨텐츠를 미리 가져와 컬렉션페이지에서 보여준다.

그림 8은 하나의 웹페이지로부터 시작되는 여러 웹페이지들에의 접근 경로와 이들에서의 사용자 접근 영역을 바탕으로 추출된 의미구역으로 이루어진 컬렉션페이지를 보여준다. 이 경로상의 페이지들이 북마크를 이루는 페이지셋이며 각 페이지에서 사용자가 접근한 의미구역들과 메인 컨텐츠 구역을 주요 구역으로 다룬다. 컬렉션페이지를 통해 사용자는 여러 링크 페이지에 흩어져 있는 정보를 하나씩 링크를 따라가며 방문하지 않고도 하나의 페이지에서 동시에 볼 수 있다.

접근한 의미구역 또는 메인 컨텐츠를 담고 있는 의미구역의 컨텐츠를 미리 가져오기 위해서, 웹페이지는 여러 의미구역들로 나뉘어지고 사용자가 링크에 클릭 등의 방법으로 접근하면 페이지에서 이 링크가 위치한 의미구역이 자동 북마크된다. 의미구역을 북마크 하기 위해서 시스템은 우선 해당 접근 링크의 URL과 링크된 텍스트 등의 정보를 추출하고 페이지를 분석하여 이 링크가 위치한 의미구역을 확인한다.

4.1 북마크 정보

자동 북마크 기능의 특징은 전체 페이지가 아닌 사용자가 접근한 의미구역 정보를 기록한다는 것이다. 사용자가 웹페이지 내의 링크에 접근하면, 이 링크가 위치하고 있는 의미구역이 자동적으로 북마크된다. 그림 9는 한 웹페이지의 의미구역들 중 사용자가 접근한 의미구역을 북마크 하기 위한 XML 스키마 구조를 보여준다.

페이지 전체의 <table>, </table> 태그의 순서 정보는 페이지를 의미구역으로 분할하는 과정에서 저장된다. 이때 이 태그 리스트는 각 의미구역의 Tag_Structure에 나뉘어 저장 된다. 태그 구조뿐 아니라 의미구역의 북마크는 또한 각 의미구역에 해당하는 컨텐츠의 첫번째 문장과 의미구역 내에서 접근한 링크에 대한 기록을 저장한다. 각 의미구역 북마크는 구역 내의 전체 링크의 수와 이 들 중 접근한 링크의 수의 비에 대한 정보를

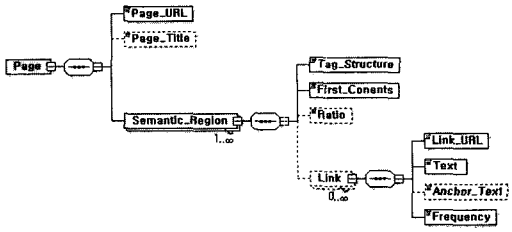


그림 9 북마크 스키마

가지고 있다. 이 비율에 따라 의미구역은 다르게 다루어진다. 해당 접근 링크 비율이 임계값 보다 작다면 접근한 링크들이 사용자의 관심 정보라고 간주하며 접근 링크의 비율이 임계값보다 크다면 접근했던 링크들뿐 아니라 의미구역의 나머지 컨텐츠들도 사용자의 관심 정보를 담고 있다고 판단한다. 북마크 정보는 또한 링크에 접근한 빈도수 정보를 담고 있다. 이 값은 사용자가 해당 링크 페이지에 재 접근할 때마다 갱신된다. 사용자가 북마크된 페이지에 일정 기간 이상 접근하지 않으면 이 페이지의 북마크된 의미구역 정보는 모두 삭제된다. 이는 일정 기간 동안 접근하지 않는다는 것은 사용자의 관심이 변경 되었음을 간접적으로 반영하는 것이고 또한 해당 기간 동안 페이지의 구조와 내용 구성 등이 바뀔 가능성이 커서 북마크된 의미구역 정보가 유용하지 않기 때문이다.

페이지에 재 방문 시, 시스템은 페이지의 URL을 이용하여 페이지의 북마크된 정보를 찾는다. 그리고 현재 페이지의 의미구역 정보와 북마크 되어 저장 되었던 정보들을 비교해 부합되는 의미구역들을 찾는다. 이는 페이지의 컨텐츠와 구조, 그리고 내용의 구성이 변경 되었을 가능성이 있기 때문이다. 구역의 컨텐츠와 구조가 일부 변경된 경우에는 북마크 정보를 수정하여 다음 방문을 대비한다. 페이지 내에서 북마크된 의미구역과 가장 유사한 구역을 선택하는 기준은 북마크된 의미구역의 컨텐츠, 포함된 링크들, 그리고 구역의 태그 구조이다. 가장 부합되는 구역을 찾은 뒤에 북마크 정보는 변경된 페이지의 내용에 맞추어 갱신된다.

4.2 북마크 정보를 이용한 컬렉션페이지 구성

자동 북마크기능은 사용자가 이전에 방문했던 페이지에 재 접근 시 북마크된 접근 영역을 바탕으로 사용자가 페이지 내에서 이용하게 될 정보와 서비스를 미리 가져와 보여준다. 또한 페이지 내에서 이용 빈도가 높았던 접근 객체의 링크를 통해 이동하는 페이지의 내용을 미리 사용자에게 보여줌으로써 사용자가 재 접근할 가능성이 높은 정보를 빠르게 접근할 수 있도록 하는 것이다. 즉 링크를 통해 연결된 여러 페이지 상에 흩어져 있는 여러 북마크된 의미구역들로 구성된 컬렉션페이지

를 제공하여 자주 이용하는 정보에 대한 접근성을 높인다. 사용자는 방문한 페이지와 이 페이지로부터 시작되는 경로에 존재하는 북마크된 구역으로 이루어진 컬렉션페이지를 요청할 수 있다.

컬렉션페이지는 요청된 페이지의 북마크된 의미구역 내의 컨텐츠와 접근 링크들을 방문 빈도수가 높은 순서대로 보여준다. 의미구역의 접근 링크 비율이 임계값보다 낮으면 구역내의 접근했던 링크들만이 컬렉션페이지에 포함되게 된다. 컬렉션페이지는 재 방문한 페이지의 구역 컨텐츠 뿐 아니라 이 페이지로부터 방문했던 링크 페이지의 컨텐츠도 보여준다. 접근했던 링크 정보와 함께 해당 링크를 통해 연결된 페이지의 정보도 컨텐츠 양에 따라 컬렉션페이지에 포함 될 수 있다.

링크된 페이지의 컨텐츠를 미리 가져오는 과정에서, 만약 링크된 페이지 내에 사용자가 접근한 의미구역이 있다면 해당 구역의 컨텐츠를 미리 가져와 컬렉션페이지에 위치하게 된다. 자주 접근한 링크 페이지임에도 이 페이지에서 자주 접근한 구역이 없다면 이는 해당 페이지가 다른 정보, 서비스에 접근하기 위한 경로가 아닌, 사용자가 원하는 정보를 제공하고 있는 정보 페이지이다. 따라서 이러한 페이지의 경우에는 메인 컨텐츠 구역을 대신 가져온다. 페이지의 메인 컨텐츠를 담은 의미구역은 구역의 컨텐츠 중 링크가 차지하는 비율이 적고 정보의 양이 제일 많은 의미구역을 선택하여 추출된다. 그림 10은 링크 페이지의 컨텐츠들을 미리 추출하여 가져오는 선 추출 알고리즘을 설명하고 있다. 선 추출 시에 컨텐츠는 페이지의 하나의 링크로부터 시작되는 일련의 경로에서 얻는 정보의 양이 임계값을 넘지 않을 때까지 접근 경로를 따라가며 추출된다. 이는 북마크된 웹페이지의 각각의 링크 당 추출되는 정보의 양을 제한하여 컬렉션페이지를 구성하기 위함이다.

5. 시스템 프로토타입

우리는 제안한 기법의 타당성과 효율성을 확인하기 위해 LinkBroker의 프로토타입을 설계하고 구축하였다. 그림 11은 LinkBroker 시스템의 전체 구조를 보여준다. LinkBroker는 웹 브라우저와 웹 서버 사이에서 가상의 중개자(intermediary) 역할을 한다. 프록시(Proxy) 또한 일종의 중개자이다. 이러한 중개자 모듈은 사용자 단, 프록시 서버, 웹 서버등에 모두 탑재될 수 있으나 우리는 프록시 서버를 이용하였다.

대표적인 가상 중개자로써 WBI(Web Intermediaries)를 들 수있다. WBI는 IBM에서 제공하는 개발툴킷으로 정보 스트림을 이용해 개인화, 트랜스코딩, 필터링 기능 등을 제공하는 모듈이다. WBI는 현재 IBM의 Websphere transcoding publisher 서버의 부분으로 포함되어 있다[15].

사용자 브라우저의 프록시로 중개자를 설정하게되면, 브라우저를 통해 사용자가 방문하는 링크 정보를 얻을

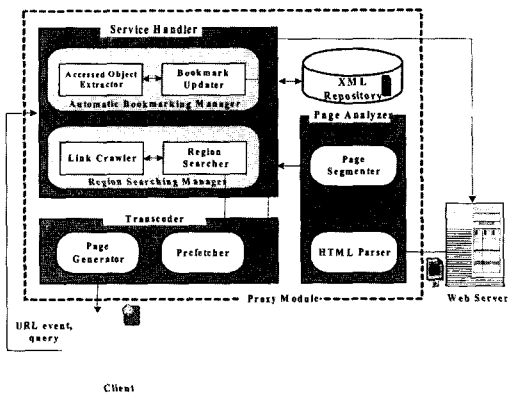


그림 11 LinkBroker 시스템의 구조

```
Algorithm Pre-fetching
Parameter : URL of accessing page, XML bookmark file for this page

X=threshold for choosing active link
Y=threshold to limit the size of pre-fetching from one active link

After call FindMatchingObject
Put all bookmarking accessed link object and information region finding

If (There is the linking object that the frequency is higher than  X)
 //if the link is active link
then
     While (the total byte of contents from one active link is lower than y and
            there is no linking info accessed in a page)       do
          FindMatchingObject in each linked page
          Fetch the relevant link or information unit in a linking page
          if (there is no accessed link in a page)      then
               fetch the most little linking region that have largest contents and
               least links
```

그림 10 선 추출(prefetching) 알고리즘

수 있다. 이렇게 저장한 정보를 이용해 자동 북마크 기능을 제공할 수 있고, 사용자가 웹 서버에게 요청한 페이지를 전송받아 사용자의 설정이나 목적에 맞도록 변환하여 제공할 수 있다. LinkBroker의 프로토타입은 WBI의 자바를위한 개발 API을 이용해 구축하였다.

전체 시스템은 크게 페이지 분석기(Page Analyzer), 서비스 처리기(Service Handler), 페이지 생성기(Transcoder)로 나뉜다. 페이지 분석기는 웹서버가 웹브라우저로 전달하는 사용자 요청 HTML 페이지의 태그를 분석한다. 이를 통해 웹페이지는 여러 의미구역들로 나뉘어진다. 사용자의 모든 방문 링크의 정보는 LinkBroker에 저장되며 저장된 정보를 분석해 각 페이지의 주요 의미구역 정보를 추출해 XML database인 eXcelon에 저장한다. eXcelon은 XML문서 내의 자료의 저장, 검색을 가능하게 해주어 사용자 데이타를 빠르게 추출할 수 있도록하기 위한 것이다. LinkBroker는 사용자의 요청을 수행하기위해 사용자 질의, 검색 모드 등을 입력받을 수 있는 웹페이지를 생성해 제공한다. 이 웹페이지에 질의와 검색 조건 그리고 북마크 요청등을 할 수 있다. 이는 JSP를 이용하여 처리된다. 서비스 처리기는 파악된 의미구역을 바탕으로 자동 북마크 기능과 페이지셋 검색 기능을 수행한다. 페이지 생성기는 페이지셋의 웹페이지들에서 관련있는 의미구역들을 미리 가져와 컬렉션페이지를 구성한다.

6. 실험 및 결과

제안하는 시스템의 효율성은 첫째, 물리적으로 여러 페이지 상에 흩어진 정보를 모아 의미적인 정보 단위로 정확하게 추출할 수 있는지의 여부와 둘째, 이를 검색, 북마크 등의 목적에 맞게 적용하여 정보에 대한 접근성을 향상하였는지의 여부에 달려있다. 따라서 우리는 정보 단위 추출의 정확성과 검색 정확도의 향상, 북마크에 의한 접근성의 향상을 여러 실험을 통해 살펴보았다.

6.1 의미구역 추출과 정보 중요도 파악의 정확성

LinkBroker는 각 질의에 대해 검색엔진에서 반환한 결과 페이지에 링크된 페이지들을 시작으로 일련의 하위 페이지들을 탐색하여 페이지셋들을 추출한다. 이때 추출된 페이지셋에서 중복되는 부분, 광고 등을 제외시켜 페이지셋을 정제한다. 페이지셋 정제의 효율성을 판단하기 위해 우리는 웹페이지를 의미구역으로 나누고 이들 중 페이지의 메인 컨텐츠를 갖는 의미구역 추출의 정확도를 측정하였다. 실험을 위해 우리는 페이지의 주제를 확연히 구분할 수 있는 즉, 메인 컨텐츠를 구별할 수 있는 페이지들을 선정하였다. 표 3은 실험 대상이된 페이지들의 주소와 각 페이지의 메인 컨텐츠를 보여준다. 이들 페이지들은 테이블 구조를 갖는 페이지들이다. 메인 컨텐츠 추출의 정확도는 사용자에의해 판단된 메인 컨텐츠 구역과 시스템에 의해 추출된 구역의 일치여부를 통하여 얻어진다.

실험을 통해 추출된 메인 컨텐츠 구역은 100% 모두 실제 메인 구역과 부분적으로 일치되고 있으나 해당 중심 컨텐츠가 아닌 다른 정보를 포함하거나 메인구역의 일부를 포함하지 않는 경우가 있었다. 그림 12는 추출된 메인 컨텐츠의 정확도를 보여준다. 평균 정확도 89%로 메인 컨텐츠 추출이 이루어졌다. 추출된 메인 컨텐츠 구역은 웹페이지의 내용을 대표하는 것이므로 검색에 영향이 큰 부분이고, 이를 이용해 컬렉션페이지를 생성하여 여러 웹페이지들의 내용을 동시에 볼 수 있도록 하여 링크들을 탐색하는 시간을 줄이게하므로 메인 컨텐츠 추출은 매우 중요하다. 질의에의해 의도한 정보를 담은 구역이 가장 중요한 구역이기 때문에, 검색 결과 페이지셋에 대한 컬렉션페이지는 페이지의 메인컨텐츠 구역을 주요구역으로 다루지는 않지만, 식 (2)에서와 같이 사용자의 질의를 포함한 구역들 중 메인 컨텐츠를 담은

표 3 실험 웹페이지들

Site	Main Contents	URL
Yahoo	**Weather** : Current condition, Local Forest	http://weather.yahoo.com/forecast/KSXX0037_f.html
	Stock : Market Summary	http://finance.yahoo.com/?u
CNN	**World** news : S. Korea to send envoy to Pyongyang	http://www.cnn.com/2003/WORLD/asiapcf/east/01/24/koreas.talks/index.html
	Tech. news : Nintendo to launch snazzier console	http://www.cnn.com/2003/TECH/fun.games/01/23/console.nintendo.reut/index.html
White House	**News** & Policies : A Column by Dr. Condoleezza Rice	http://www.whitehouse.gov/news/releases/2003/01/20030123-1.html
	Education : President Bush Celebrates First Anniversary of No Child Left Behind	http://www.whitehouse.gov/news/releases/2003/01/20030108-4.html

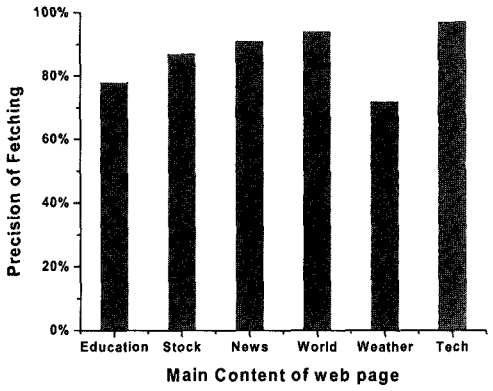


그림 12 메인 컨텐츠 추출의 정확도

구역이 있다면 가산점을 준다. 이는 페이지에서 중요하게 다루는 정보가 질의에 부합될 경우 더 중요한 정보일 가능성이 높기 때문이다.

6.2 페이지셋 검색의 정확성

페이지셋을 이용한 검색의 효율성 평가를 위해 우리는 Google 검색엔진에서 반환한 결과 리스트 페이지를 이용하였다. 검색엔진의 결과 리스트와 이 리스트에 링크된 페이지들을 탐색해 얻은 페이지셋을 비교하여 실제 질의와 관련된 정보를 높은 순위에 담고 있는지를 비교하고자 한다. 사용자가 검색하고자 의도한 정보의 내용과 이를 위해 이용한 각 질의는 표 4에 설명되었다.

표 4의 질의는 여러 개의 질의어로 이루어져 있으며 해당 질의어들을 모두 포함할수록 질의의 의도에 부합되는 결과일 가능성이 크다. 이 질의어들은 다른 구역에 존재할수록 의도한 주제와는 다른 정보일 가능성이 큰 것들이다. 예를 들어 질의 1의 경우, XML 전용 데이타베이스를 이용하고자 하는 사용자의 의도가 담겨있다. 그러나 실제 검색엔진에서 반환한 결과를 보면 대부분 shareware를 제공하는 사이트로의 링크이거나 database관련 링크 페이지들이 상위를 차지하고 있었다. 따라서 사용자는 원하는 정보에 구체적으로 접근할 수 있는 기회를 놓치고 있다. 질의 2의 경우에도 oil과 free가 함께 등장하는 경우들이 사용자의 의도에 부합되는 정보를 담고 있었으나 oil과 sample 또는 free와 sample 등이 서로 다른 구역에 퍼져있는 전혀 다른 내용의 페이지들이 상당수 상위에 랭크되어 있었다. 이는 질의어들이 하나의 주제를 이루어야 한다는 가정없이 질의어들의 빈도수들을 토대로 검색되어졌기 때문이다. 질의 3, 4 또한 각각 해당 질의어를 포함하고 있으나 질의의 내용과 관련없는 페이지들이 상위에 랭크되어 있었다.

우리는 이들 질의를 이용해 의미구역기반의 페이지셋 검색기법의 효율성을 증명하고자 한다. 페이지셋과 질의와의 유사도는 먼저 의미구역의 점수를 바탕으로 페이지 점수를 계산하고, 이 계산된 페이지 점수를 이용하여 계산된다. 우리는 의미구역기반의 페이지와 질의와의 유사도의 정확도를 측정하기 위해 Google에서 각 질의에 대해 반환한 페이지들과 이 페이지들을 의미구역으로 나눠 재계산한 순위를 비교하였다. 다음은 의미구역기반의 웹페이지 검색의 정확도 계산을 위한 지표이다.

- 결과셋(Relevance Set) - 검색엔진에서 반환한 페이지들을 검토하여 질의 의도에 부합되는 페이지들의 순위를 매겨 1-3위, 1-5위, 1-10위, 그리고 1-20위안에 드는 결과 페이지들을 각각 결과셋으로 만든다.
- 검색엔진의 결과셋(Result 1) - Google에서 반환한 결과 페이지들의 셋이다. 역시 사용자 판단에의해 형성된 결과셋과 마찬가지로 각 순위내의 페이지들을 이용해 각각 셋을 형성하도록 한다.
- 의미구역 기반의 검색 결과셋(Result 2) - 제안한 의미구역기반 페이지 검색기법으로 검색된 결과셋이다. 이는 Google에서 제공한 결과 페이지 내의 1-30위 내의 페이지들을 이용하여 실험되었다. 이들의 순위를 재 배열하여 결과셋의 경우와 같이 각 순위내의 페이지이 셋을 형성하도록 한다.

그림 13은 실제 사용자가 판단한 순위를 바탕으로 Google에서의 순위와 제안한 시스템에서 계산한 의미구역 기반의 페이지 검색 순위의 정확도를 비교한 그래프이다. 결과에서 보듯이 검색엔진에서 상위에 랭크되었던 페이지들 중 질의의 의도와 부합되지 않는 페이지들의 순위를 낮게 측정하고, 질의의 발생 빈도수는 적더라도 해당 질의어들을 다 포함하며 하나의 구역에 모여있는 페이지들은 상위로 랭크하여 검색의 정확도를 높임을 볼 수 있다. 따라서 관련없는 질의어들이 페이지 내에서 서로 떨어져있을 가능성이 높은 경우, 본 논문의 개념을 이용하여 높은 검색 정확도를 얻을 수 있음을 증명하였다.

표 4 실험 질의

Query	Query terms	Description
1	XML, database, freeware	XML 전용 데이타베이스를 무료로 제공하는 사이트 등의 정보
2	Oil, free, cleansing, sample	Oil을 함유하지 않은 클린징 화장품 샘플 정보
3	Java coffee sale	자바 원두 커피 세일, 판매 정보
4	Digital TV	디지털 TV의 기술 및 판매 정보

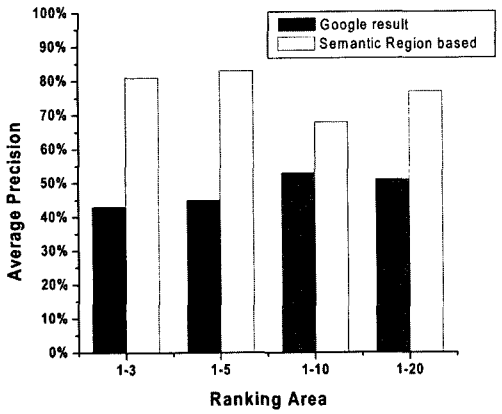


그림 13 의미구역 기반 기법과 기존 검색 기법의 비교

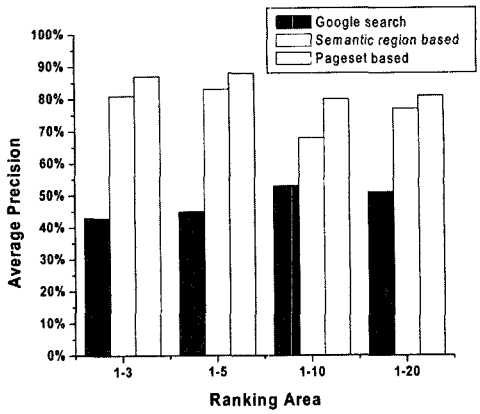


그림 14 페이지셋 기반 기법과 기존 검색 기법의 비교

실제 웹 검색엔진에서의 도입시 모든 페이지를 의미구역으로 나누고 이를 바탕으로 유사도를 계산하므로 검색 시간이 더 소요되나, 실시간으로 이루어지지 않고 인덱스 구성시에 이루어진다면 검색의 정확도를 높이므로 효율적이다. 본 시스템의 경우 웹에비해 매우 적은 양의 페이지들에 대해 탐색하고 의미구역으로 나누므로 그 오버헤드는 크지 않다.

우리는 또한 페이지셋 기반의 검색의 효율성을 측정하기위해 표 4의 질의를 이용해 반환된 Google 검색엔진의 결과리스트를 토대로 각 검색 결과 페이지들의 링크를 탐색하였다. 그리고 이들 탐색을통해 페이지셋을 추출하고 이 페이지셋 단위로 순위를 측정하였다. 각 질의를 통해 얻어진 결과 리스트 1위부터 30위까지의 페이지들로부터의 탐색을 통해 평균 52.7개의 페이지셋을 얻을 수 있었다. 페이지셋의 사이즈는 평균 3.8개였다. 추출된 페이지셋들중 검색엔진 결과리스트에 포함된 페이지를 담고 있은 페이지셋은 30개이고 나머지 22.7개의 페이지셋은 그이외의 페이지들 및 다른 사이트의 페이지들의 정보를 담은 페이지셋이었다.

결과 페이지의 리스트에 포함된 페이지를 담은 페이지셋에 대해 위에서 측정한 검색엔진의 결과 순위와, 의미구역기반의 검색과의 비교를 통해 페이지셋 기반 검색의 정확도와 효율성을 판단해보았다. 그림 14는 실제 정보의 가치 판단을 통해 얻은 순위와 검색엔진, 의미구역, 페이지셋 기반 검색의 순위를 비교하여 측정한 검색의 정확도를 보여준다. 의미구역기반 검색을 통해 계산된 페이지의 점수는 질의어의 분포를 고려해 검색 결과의 정확도를 높였고, 페이지셋 검색의 경우, 이를 기반으로 관련된 페이지셋 단위로 점수를 주어 더 좋은 성능을 보이는 것을 볼 수 있다. 이는 해당 페이지뿐 아니라 이어진 일련의 페이지들에 퍼져있는 정보를 하나로 다루어 순위를 매겼기 때문에 페이지의 정보가치를 더 정확하게 계산할 수 있었기때문이다.

또한 검색엔진의 결과 리스트에 포함되지 않았던 평균 22.7개의 페이지셋 또한 리스트에 포함된 페이지들과 순위를 측정 비교한 결과 질의의 의도와 관련된 페이지셋을 많이 찾을 수 있었다. 각 질의에 대해 평균 6.5개의 페이지셋의 점수는 30위내의 페이지셋에 포함 될 수 있는 점수를 보였다. 이를 통해 정보셋의 정확한 순위를 매기고 결과페이지에 링크된 페이지들로부터 일일이 탐색하지 않아도 유용한 페이지를 바로 접할 수 있음을 증명할 수 있었다.

웹사이트내에서 한 페이지로부터 시작하여 질의와 관련된 일련의 정보를 추출하는 경우의 검색 효율성을 판단하기 위해, 구글에서 제공하는 사이트 검색을 이용하여 결과를 비교해보았다. 추출된 페이지셋은 평균 5.5개이며, 평균 사이즈는 4.2이었다. 추출된 페이지셋에 포함된 여러 페이지들이 검색엔진에서 반환한 결과 리스트에서는 흩어져 나타나 있었기 때문에 순위 정확도의 차이가 그림 14에서 보여지는 웹 검색에서의 경우보다 더 크게 나타난다. 이는 또한 일반 웹 검색에 비해 사이트내의 검색은 페이지셋내의 중복되는 부분이 많아 본 시스템의 방식이 더 효율적일 수 있기 때문이다.

6.3 컬렉션페이지를 이용한 접근성 향상

컬렉션페이지는 여러 웹페이지에 흩어진 정보를 모아 페이지셋의 정보를 한눈에 볼 수 있도록 한다. 북마크를 위한 평균 페이지셋의 사이즈는 5.8이고 검색된 결과의 페이지셋 사이즈는 평균 3.9이다. 컬렉션페이지는 하나의 페이지셋을 의미구역으로 나누고 각 페이지셋의 점수를 계산하여 생성하며 생성에 소요되는 시간은 5600ms에 불과하다. 또한 북마크의 경우에도 구조의 업데이트가 없는 경우 각 링크를 따라가며 페이지의 북마크된 구역을 찾아 생성하는 데 매우 적은 시간 소요된다. 페이지셋을 통해 평균 5.5번의 링크 탐색 시간 횟수

를 줄일 수 있으므로 생성 시간에 의해 발생되는 오버헤드보다 서핑시간을 줄일 수 있어 컬렉션페이지가 효율적이다. 또한 구조가 복잡한 사이트의 경우나 링크 텍스트에 대한 이해가 어려운 경우에 발생할 수 있는 경우의 시간소모를 줄일 수 있으므로 유용하다.

7. 결 론

본 논문에서는 웹 접근성 향상을 위해 한 페이지로부터 시작 되는 일련의 링크 페이지들을 검색하여 관련된 정보들을 한 페이지에서 볼 수 있도록 컬렉션페이지를 생성하는 과정을 살펴보았다. 본 시스템은 기존의 웹 검색엔진과 웹 브라우저가 하나의 웹페이지 단위로 검색하고 정보를 제공하는 한계를 벗어나, 여러 페이지에 흩어져 있는 정보들을 하나의 페이지에 정리하여 표현해 줌으로써 보다 빠르고 쉽게 페이지들 내의 정보의 구성과 각 페이지의 내용을 이해할 수 있도록 하였다. 웹페이지의 메인 컨텐츠 구역을 추출하는 과정은 높은 정확성을 보였으며 이를 통해 사용자는 여러 링크 된 웹페이지들의 주요 내용을 한번에 볼 수 있게 한다. 또한 페이지셋 단위의 정보 검색의 정확성을 통해 웹 서핑의 효율성을 높일 수 있었다.

참 고 문 헌

[1] "Web Accessibility Initiative (WAI)," http://www.w3.org/WAI/.

[2] Ning Zhong, Jiming Liu, Yao, Y.Y., Ohsuga S., "Web Intelligence(WI)," Computer Software and Applications Conference, 2000. COMPSAC 2000. The 24th Annual International 2000. page 469-470.

[3] Kazutoshi S, et al., "Web Skimming: An Automatic Navigation Method along Context-path for Web Documents," 11th Int'l WWW Conference, 2002.

[4] Wen-Syan L., Quoc V., Divyakant A., "Retrieving and Organizing Web Pages by InformationUint," WWW10, May 1-5 2001, Hong Kong., ACM 1-58113-348-0/01/0005.

[5] Xiaoli Li, Bing L., Tong-Heng P., and Minqing H., "Web Search Based on Micro Information Units," 11th Int'l WWW Conference, 2002.

[6] Shaun K, et al, "Integrating Back, History and Bookmarks in Web Browsers," Proceedings of CHI'01, ACM Press, pp. 379-380, 2001.

[7] Tsuyoshi E, Seiji I, and Teruhisa M., "Fast Web by Using Updated Content Extraction and a Bookmark Facility," The 4th Int'l ACM conference on Assistive technologies 2000.

[8] Vinod A, et al., "Automating Web Navigation with the WebVCR," 9th Int'l WWW Conference 2000.

[9] Liebrman H., "An Agent that Assists Web Browsing," 14th Int'l joint conference on Artificial Intelligence IJCAI95, 1995.

[10] Natasa M., Ralph S., and Robert T., "MS WebScout: Web Navigation Aid and Personal Web History Explorer," 11th Int'l WWW Conference, 2002.

[11] Ramesh R. Sarukkai, "Link Prediction and Path Analysis Using Markov Chains," 9th Int'l WWW Conference 2000.

[12] Yuxiang Z, et al. "Hunter Gatherer: Interaction Support for the Creation and Management of Within-Web-Page Collections," The 11th Int'l WWW Conference, 2002.

[13] Hironobu T, and Chieko A., "Transcoding Proxy for Nonvisual Web Access," The 4th Int'l ACM conference on Assistive technologies 2000.

[14] Vo N. A., et al., "Impact Transformation: Effective and Efficient Web Retrieval," 25th Int'l ACM SIGIR Conference, 2002.

[15] http:// www.almaden.ibm.com/cs/wbi

이 수 철

1998년 한남대학교 컴퓨터 공학과(학사) 1998년~2000년 아주대학교 정보통신 전문대학원(석사). 2000년~2002년 아주대학교 정보통신 전문대학원 박사수료. 2003년~현재 아주대학교 정보통신 전문대학원 박사과정. 관심분야는 데이타베이스, 멀티미디어 시스템, 정보 통합, XML 응용, 유비쿼터스 컴퓨팅

이 시 은

2002년 아주대학교 정보 및 컴퓨터공학부(학사). 2002년~2004년 아주대학교 정보통신 전문대학원(석사). 2004년~현재 삼성전자. 관심분야는 데이타베이스, 멀티미디어 시스템, XML 응용, WWW

황 인 준

1988년 서울대학교 컴퓨터공학과(학사) 1990년 서울대학교 컴퓨터공학과(석사) 1998년 Univ. of Maryland at College Park 전산학과(박사). 1998년 6월~1998년 8월 Hughes Research Lab. 연구교수. 1998년 8월~1999년 8월 Bowie State Univ., Assistant Professor. 1999년 9월~2002년 아주대학교 정보통신전문대학원 조교수. 2003년~2004년 8월 아주대학교 정보통신 전문대학원 부교수. 2004년 9월~현재 고려대학교 전자공학부 조교수. 관심분야는 데이타베이스, 멀티미디어 시스템, 정보 통합, 전자 상거래, XML 응용, 유비쿼터스 컴퓨팅