웹사이트의 구조추출, 저장 및 가시화를 위한 구조분석 에이전트

정윤경 조성배
연세대학교 컴퓨터과학과
ygyoung@candy.yonsei.ac.kr, sbcho@csai.yonsei.ac.kr

A Structure Analysis Agent for Extraction, Storage and Visualization of Web Sites

Yun-Gyoung Chong and Sung-Bae Cho
Computer Science Department, Yonsei University

요 약

인터넷 비즈니스 사이트가 많아짐에 따라 사용자에게 편리한 비즈니스 사이트를 구축하기위해 웹관리자는 웹서버의 구조및 내용을 평가하고 재구성해야 하는데, 수작업에 의한 웹문서의 평가시 인적, 시간적 비용이 높고 시스템에 대한 평가값이 객관적이지 못하다. 이를 해결하기 위해 본 논문에서는 구조추출, 구조 저장, 구조가시화로 구성된 웹문서의 구조분석 에이전트를 제안한다. 구조추출 모듈은 URL로 웹문서를 받고 이를 잘정의된 XML형태로 변환한 후, 태그정보를 이용하여 웹문서의 구조를 구하고, 하이퍼링크데이터를 이용하여 웹문서간의 연결구조를 얻는다. 구조저장은 추출된 웹문서와 웹문서 연결관계를 웹서버와 같이 연결하여 저장하며, 구조 가시화에서 이를 계층적으로나 그래프형식으로 가시화한다. 제안한 시스템의 유용성을 보이기 위하여 웹문서의 평가문제에 적용한 결과, 많은 양의 데이터를 의 데이터의 기술적인 평가가 가능하고, 데이터를 수집하기 위한 인력자원, 시간과 비용을 줄일 수 있으며, 쉽게 사이트를 평가하여 서비스 수준을 향상 시킬 수 있음을 알 수 있었다.

1. 서론

인터넷과 네트워크 기술이 발전함에 따라 상거래 방식이 인터넷 비즈니스 사이트를 통한 전자적 거래 방식으로 변화되고 있다. 이에 따라 인터넷 비즈니스 사이트의 구성 및 정보운용 형태가 다양하게 이루어지고 있다. 웹서버 관리자는 사용자 편의와 양질의 사이트를 구성하기 위해서 웹서버의 구조, 내용 등을 모니터링하고 웹서버를 재구성하여 차별화된 사이트를 구성한다.

현재 평가사이트로 고메즈(http://www.gomez.com)와 포레스터(http://www.forrester.com)를 예로 들 수 있는데 이들은 수작업으로 사이트를 평가하여 인적, 시간적으로 비용이 높기 때문에 웹문서의 구조를 분석하여 웹문서의 구조를 파악할 수 있는 에이전트가 개발되었다[1]. 이를 이용하여 시스템에 의한 자동 평가할 경우 사용자를 대신함으로써 많은 양의 데이터를 처리할 수 있었다.

본 논문은 시스템의 구조 추출과 구조 가시화를 개선하고 웹평가 시스템에 적용함으로써 시스템의 유용성을 보인다. 또한 평가항목을 추가함으로써 다양한 웹평가를 시도한다. 이를 통해 기술적인 평가가 가능할 것이며 데이터를 수집하기 위한 인력자원, 시간과 비용을 줄일 수 있고 쉽게 사이트를 평가하고 서비스수준을 향상시킬 수 있다.

본 논문의 구성은 다음과 같다. 2절은 관련연구를 설명한다. 3절에서는 구조분석 에이전트의 구성과 각 모듈에 대한 기능을 설명한다. 4절에서는 구조를 이용하여 웹사이트를 평가한 분석결과를 제시하고, 5절에서 결론을 내린다.

2. 관련연구

비구조화된 문서인 웹문서에서 유용한 정보를 추출하는 방법에 대한 연구가 많이 진행되고 있는데, 구조를 이용한 웹에이전트로 Florid 시스템, Lore 시스템, Strudel시스템을 들 수 있다[2,3,4]. 다른 메타 검색으로 검색하는 다른 웹 검색 에이전트와는 달리 위 시스템은 웹문서의 구조정보를 검색/추출에 이용한다. Florid시스템은 특정 사이드에서 웹문서를 가져온 후 파서를 통해 문서구조를 획득한 다음 링크 정보를 추출하고 이들의 구조를 저장하는 방식을 사용하였다. 스탠포드에서 개발된 Lore시스템은 XML의 DTD정보를 이용하여 다목적으로 반구조화된 데이터를 관리하는데 적용하였다. AT&T 연구실에서 개발된 Studel시스템은 패턴매칭 방법을 이용하여 웹사이트의 내용을 추출하는 시스템이다. 이들 시스템을 포함한 대부분의 시스템이 단순한 패턴매칭방법을 사용했으며 대부분 정보검색에 편중되어 있으며 구조정보를 이용한 내용추출에 대한 패턴매칭방법을 이용하고 있으므로 연구가 미비하다.

3. 구조분석 에이전트

구조분석에이전트는 웹문서내의 태그에 대한 속성값, 태그에 대한 내용정보를 추출하여 웹문서의 구조를 추출하고 하이퍼링크데이터를 이용하여 웹문서간의 구조를 추출한다. 이를 위해 링크된 문서에 대해서 방문하면서 연결성을 검사한다. 또한 추출한 하이퍼링크 연결관

계를 이용하여 계층적 또는 그래프형식으로 가시화한다. 그림 1은 구조분석 에이전트의 기능을 보여준다.

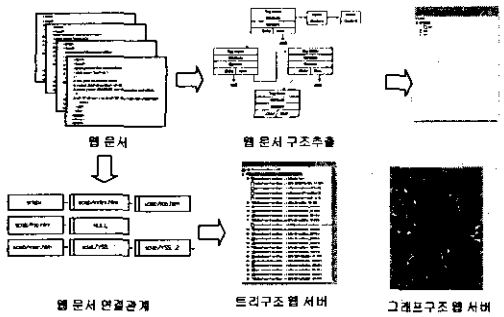

그림 1. 구조분석 에이전트 기능.

시스템은 크게 구조추출, 구조 저장, 구조가시화로 정의할 수 있는데, 시스템 구성도는 그림 2와 같다. CGI, XML, HTML 등의 URL이 들어오면 시스템은 잘정의된 XML형태로 변환하여 웹문서와 웹문서간의 구조를 추출하고 이를 저장한다. 또한 저장된 데이터로 계층적으로나 그래프형식으로 가시화한다.

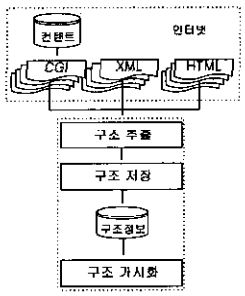

그림 2. 시스템 구성도.

3.1 구조추출

구조추출은 URL의 반구조화된 데이터를 정형화된 형식의 문서로 변환하고 문서변환으로 얻은 태그정보와 링크정보로 계층적 구조를 생성한다. 이를 위해서 HTML문서를 잘 정형화된 XML문서로 변환하는데 XML형식으로 변환 후 태그정보를 이용하여 웹문서 내 구조를 추출한다. 웹문서의 구조를 얻기 위해서 웹문서의 태그간의 상하관계는 HTML 4.01스펙의 상하관계를 이용한다[5,6]. 태그의 계층구조는 이진트리로 표현하며 태그 방문시 URL임시 저장공간으로 스택을 이용한다. 하이퍼링크 구조는 웹문서의 태그정보 중 링크정보를 포함할 수 있는 태그를 이용하여 문서들을 연결한다. 이때 사용되는 태그는 'a'태그, 'area'태그, 'frame'태그, 'select'태그이다. 'a'태그와 'area'태그의 경우 'href'속성에 연결된 웹문서를 이용하며 'frame'태그의 경우 'src'속성에 연결된 웹문서를 이용하게 된다. 때때로 'select'태그에 대해서도 웹문서를 연결시키기도 하는데 이때 'option'속성을 사용하여 이 속성값이 웹문서인 경우에 대해서만 문서들을 연결시킨다. 하이퍼링크 추출방법은 BFS(Breadth First Search)방식으로 웹문서들을 방문하여 연결하게 된다. 즉 서버의 첫 웹문서를 기준으로 링크정보를 추출하고 하이퍼링크구조에 링크정보를 연결한다. 링크정보를 연결할 때는 이전에 방문했던 링크는 다시 방문하지 않도록 표시한다. BFS방식의 임시 주소 기억공간으로 큐(Queue)를 사용한다.

3.2 구조저장

추출된 구조를 저장하기 위해서 구조정보는 계층적으로 표현되므로 시스템은 DFS(Depth First Search)방식으로 방문하여 저장하고 저장된 데이터를 읽어온다. 웹문서 내 태그들의 구조만 저장한 경우와 속성도 같이 포함한 경우로 나누어 저장하며 태그 구조를 저장한 경우에는 웹문서의 구조정보만 필요한 경우에 사용하며 태그구조와 속성정보를 저장한 경우에는 태그의 속성정보를 이용할 때 사용하게 된다. 웹사이트의 연결관계는 계층적이면서도 하위URL이 상위URL을 가리키므로 그래프형식으로 저장한다. 또한 웹문서의 하이퍼링크 추출시 방문순서가 BFS방식이므로 이 순서대로 인접리스트로 구성하여 저장한다. 시스템은 서버주소로 검색했던 다른 서버들의 웹문서 구조와 하이퍼링크 구조를 구분하여 저장한다. 그림 3과 4는 구조저장에 사용된 저장형식이다.

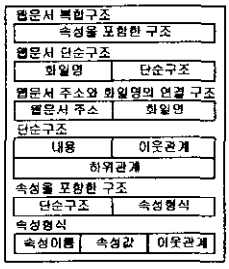

그림 3. 웹문서 구조의 저장형식.

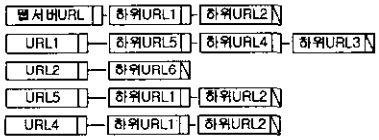

그림 4. 하이퍼링크 구조의 저장형식.

3.3 구조 가시화

인접리스트형식으로 저장된 하이퍼링크 구조는계층적으러로 그래프적으로 가시화된다. 계층구조적인 가시화방법은 인접리스트형식으로 저장된 링크정보를 BFS방식으로 읽어들여 트리를 구성하여 가시화한다. 이를 위해서 큐에 주소들을 임시로 저장하고 순서대로 연결시킨다.

그래프 형식의 가시화방법은 인접리스트의 하이퍼링크구조를 큐에 주소를 임시로 저장하고 동심원 그래프 표현방식을 적용하여 표현한다. 이는 웹서버 주소는 가시화 공간의 중심에 위치시키며 기준노드의 인접리스트를 통해 얻은 하위관계로 웹문서간의 상하 연결관계를 표시한다. 중심노드$_{(x_{00}, y_{00})}$로 두면 중심노드 값은 다음과 같다.

$$x_{00} = \frac{Scalewidth}{2}, \quad y_{00} = \frac{Scaleheigh}{2} \qquad (1)$$

t시간에 i번째 노드에 대해서 $t-1$ 시간과의 변화값을 dx_{it}, dy_{it}라 하면 연결된 노드들 간의 노드 배치는 중심노드를 기준으로 다음과 같이 정의한다.

$$dx_{it} = \frac{\sqrt{(x_{i+1t} - x_{it})^2 + (y_{i+1t} - y_{it})^2} - \sqrt{(x_{i+1t-1} - x_{it-1})^2 + (y_{i+1t-1} - y_{it-1})^2}}{\sqrt{(x_{i+1t} - x_{it})^2 + (y_{i+1t} - y_{it})^2}} \times (x_{i+1t} - x_{it}) \quad (2)$$

$$dy_{it} = \frac{\sqrt{(x_{i+1t} - x_{it})^2 + (y_{i+1t} - y_{it})^2} - \sqrt{(x_{i+1t-1} - x_{it-1})^2 + (y_{i+1t-1} - y_{it-1})^2}}{\sqrt{(x_{i+1t} - x_{it})^2 + (y_{i+1t} - y_{it})^2}} \times (y_{i+1t} - y_{it}) \quad (3)$$

중심노드에 연결된 노드에 대해서 다음과 같이 노드위치를 변경시킨다.

$$x_{i+1t+1} = \sum_{k=1}^{i-1} (dx_{it} - dx_{i+1k}) - dx_{i+1t} + x_{00}, \quad y_{i+1t+1} = \sum_{k=1}^{i-1} (dy_{ik} - dy_{i+1k}) - dy_{i+1t} + y_{00} \quad (4)$$

연결되지 않은 노드들간의 노드 배치(중심노드 외 노드간)에 대해서는 중심노드에 변화값을 더한다.

$$x_{i+1t+1} = \sum_{k=1}^{i-1} dx_{ik} + x_{00}, \quad y_{i+1t+1} = \sum_{k=1}^{i-1} dy_{ik} + y_{00} \quad (5)$$

그림 5는 인접리스트로 구조를 가시화한 순서를 나타낸다.

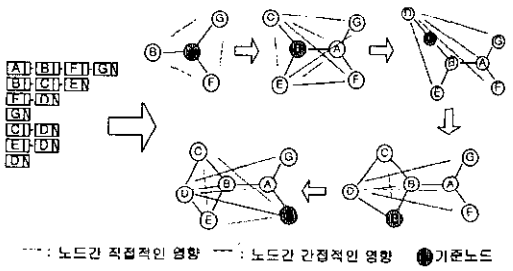

그림 5. 그래프가시화 연결 순서.

4. 구조분석 에이전트의 웹평가 적용

구조분석을 이용하여 웹평가 시스템에 적용해 보았다. 이를 위해 안정성의 경우 웹기술 남용여부를 위한 CGI, HTML문서 수, 링크와 관련하여 사이트 전체 링크 개수, 하이퍼링크가 웹문서와 링크되었는지 확인하기 위한 브로큰 링크수를 두었다. 접근성의 경우 웹사이트의 평균깊이, 평균너비, 최대너비, 바로가기 제공여부를 평가항목으로 두었다.

4.1 평가환경

구조분석 에이전트로 웹사이트를 평가시 평가대상으로 CD,교육, 경매사이트의 21개 사이트를 사용했으며 T1급의 네트워크에 펜티엄급, 128M메모리의 컴퓨터, JAVA 1.2.2환경에서 테스트하였다. 이때 읽어들일 웹페이지 수는 5000페이지로 웹페이지 요청시 반응시간은 3초로 두고 1주일동안 실험했다.

4.2 평가결과

문서 종류와 구조에 따른 평가결과 1000개 미만의 웹문서를 읽어들인 웹사이트는 9개였고, 2000개 미만인 웹사이트는 2개, 3000개 미만인 사이트는 0개, 4000개미만인 사이트는 1개, 5000개 미만인 사이트는 2개, 5000개의 문서를 읽어들인 웹사이트는 7개였다. 이 중 한 사이트에 대해서만 5000개의 HTML문서로 구성되었고 8개의 사이트에 대해서 90%이상이 CGI로 구성되었다. 100개이하의 웹문서를 추출한 사이트는 웹문서는 4개였는데 원인은 시스템이 분석할 수 없는 자바스크립트로 구성되거나 사용자 ID입력에 따른 웹문서 접근이 불가능하기 때문이다.

구조에 따른 평가항목의 경우 평균 1.39의 평균깊이와11.190의 평균너비, 3706의 최대너비를 보였다. 브로큰 링크는 3개의 사이트에 대해서 0의 값을 보였고 최대 718개의 연결되지 않은 문서를 갖는 사이트도 존재했다. 프레임 수는 대부분 1개였다.

HTML에러에 따른 평가결과는 0~10개 에러에 대해서는 13개의 사이트, 10~20개의 에러의 경우 4개 사이트, 20~30개의 에러에 대해서 1개, 30~40개에 대해서 1개, 80개이상의 에러에 대해서 2개의 사이트값을 보였다. HTML자체 내 오류가 발생하지 않는 웹사이트는 9개이며 21개 사이트중 익스플로러 버전에러값이 큰 경우 82001개였다.

5. 결론

본 논문은 웹문서의 구조분석 에이전트를 개선하였다. 이를 이용하여 21개의 웹문서를 평가했으며 평가항목으로 웹문서의 종류, 평균깊이, 평균너비, 최대너비, 브로큰링크수, 메인페이지의 프레임 수와 HTML문서 에러값을 사용했다.

향후 연구로 현재 시스템은 자바스크립트에 대한 일부만의 처리를 하고있기 때문에 정확한 사이트맵을 구성하기 위해서는 자바스크립트에 대한 정확한 처리가 필요하다. 또한 본 시스템은 테스트 서버에 여러 개의 URL의 포함시 외부 URL과 내부 URL의 구분이 불가능하기 때문에 이를 위한 처리로 시스템이 여러 개의 URL이 있는 사이트에 대한 지식을 갖고 구조추출시 내부 URL인지 판별하는 방법을 생각해 볼 수 있다. 그 외 사용자 ID와 연관된 문서 추출에 관한 연구도 필요하다.

참고문헌

[1] 서연규, 김경중, 정윤경, 조성배 "웹사이트의 구조분석을 위한 소프트웨어 에이전트," 정보과학회 학술발표 논문집 27권 2호, pp. 21~23, 2000.

[2] W. May, et. al, "A Unified Framework for Wrapping, Mediating and Restructuring Information from the Web," *WWWCM'99*, pp. 307~320, 1999.

[3] J. McHugh, S. Abiteboul, R. Goldman, D. Quass, and J. Widom.," Lore: A Database Management System for Semistructured Data," *SIGMOD*, Vol 26(3) pp.54~66, 1997.

[4] M. Fernandez, D. Florescu , J. Kang , A. Levy , D. Suciu, "STRUDEL: a Web-site management system," *SIGMOD*, 1997.

[5] http://www.ncdesign.org/html/list.htm

[6] H. Ouahid, A. Karmouch, "Converting Web Pages into Well-formed XML Documents," *IEEE Int'l. Conf. on Communications*, Vol.1, pp. 676~680, 1999.