비교쇼핑 에이전트를 위한 Wrapper의 자동생성에 관한 연구

(A Study on Automatic Wrapper Generation for a Comparison Shopping Agent)

이승아*, 김종완**, 김병만***, 권영직**

(Seung-A Lee, Jong-Wan Kim, Byeong Man Kim, Young-Jik Kwon)

요약 WWW의 확산과 함께 온라인 쇼핑몰 사용자들에게 상품 정보를 수집하고 제공하는 비교 쇼핑 에이전트들의 필요성도 증가하고 있다. 그러나, 웹사이트들은 대부분 그들 자신의 데이터 표현 포맷을 가지므로, 각 웹사이트별로 다른 wrapper가 작성되어야 한다. Wrapper는 특정한 포맷으로 쓰여진 웹 페이지들로부터 정보를 추출하는 특수 목적의 프로그램이다. 본 논문에서는 효율적인 wrapper 작성을 위해서 주어진 URL로부터 자동적으로 wrapper를 생성하는데 사용되는 핵심 정보를 추출하는 에이전트를 제안한다.

1. 서론

인터넷의 발달과 함께 빠르게 성장하고 있는 각 온라인 쇼핑몰들에서 제공하는 상품 검색 서비스를 이용해서 원하는 상품정보를 얻기 위해서는 쇼핑몰 검색 서비스의 규칙에 적합한 질의어를 생성해야 하고, 결과로 되돌려지는 문서의 형태를 분석하여, 가격 비교와 같은 특정한 목적에 맞게 상품 정보를 자동으로 추출할 수 있도록 규칙을 생성해야 한다. 이와 같이 특정한 포맷으로 작성된 웹 문서들로부터 정보를 추출해주는 특별한 프로그램을 Wrapper라고 한다. 그러나 쇼핑몰 사이트들은 자신의 사이트만을 위한 데이터 포맷을 사용하므로 각 사이트별로 다른 Wrapper가 작성되어야 한다.

본 논문에서는 주어진 쇼핑몰에서 검색 서비스를 제공하는 URL로부터 상품 정보를 추출하기 위해 경험적 요소인 테이블 헤더 정보와 테이블 내에서 나타나는 패턴의 규칙성을 이용하여, 비교 쇼핑 에이전트들을 위한 Wrapper 자동 생성 프로그램을 제안한다.

* 대구대학교 대학원 컴퓨터정보공학부 박사과정

** 대구대학교 정보통신공학부 교수

*** 금오공과대학 컴퓨터공학부 교수

2. 비교 쇼핑 에이전트와 Wrapper

2.1 에이전트의 정의와 특징

에이전트(agent)는 '사용자를 대신하여 사용자가 원하는 어떤 일을 수행해 주는 프로그램'으로 정의할 수 있다[1]. 에이전트는 개인화(personalization), 자율성(autonomy), 적응성(adaptivity), 사교성(sociality) 등의 특징을 가지며 적용 분야에 따라 적절한 특징이 적용될 수 있다[2,3].

대표적인 에이전트의 예로 사용자가 원하는 상품에 대해서 최저가의 판매자를 탐색하고 추천해주는 비교쇼핑 에이전트(Comparison Shopping Agent)를 들 수 있다[4]. 비교쇼핑 에이전트는 사용자가 어느 특정 제품에 대한 정보를 온라인 쇼핑몰의 웹사이트에서 일일이 수동적으로 확인하지 않고도 획득할 수 있도록 도와주는 에이전트이다. 이를 위해 여러 온라인 쇼핑몰의 정보제공 포맷을 분석하여 원하는 상품 정보를 추출하고 통합하여 보여주어야 한다. 가장 널리 알려진 비교쇼핑 에이전트로는 국내의 경우 Yavis[5]와 Omi[6], Shopbinder[7] 등이 있고, 국외의 경우 ShopBot[8] 등이 있다.

2.2 Wrapper 생성

정보추출(Information Extraction)은 입력으로 눈

에 보이지 않는 텍스트를 가지고 MUC(Message Understanding Conference) 평가에 의해 전형이 된 것처럼 출력으로 정해진 형태의 명확한 자료를 생산하는 과정이다[9]. 정보추출 시스템의 운용대상인 웹 페이지는 형태에 따라 structured texts(표로 만든 정보를 사용하는 웹 페이지 등), semi-structured texts(온라인 동정 기사 등), free texts(뉴스 기사 등)의 3가지로 분류될 수 있다.

웹 페이지에서 원하는 정보만을 추출하고 통합하기 위해 나타난 연구분야가 Wrapper 생성이고, Wrapper 자동생성의 문제점인 웹 페이지들의 이형질성을 극복하기 위해 제안된 방법이 Wrapper Induction이다[8]. Wrapper Induction System들로는 Kushmerick등이 제안한 WIEN[8], Hsu와 Dung이 제안한 SoftMealy[10], Minton 등이 제안한 STALKER[11] 등이 있다.

3. Wrapper 자동생성기의 설계 및 구현

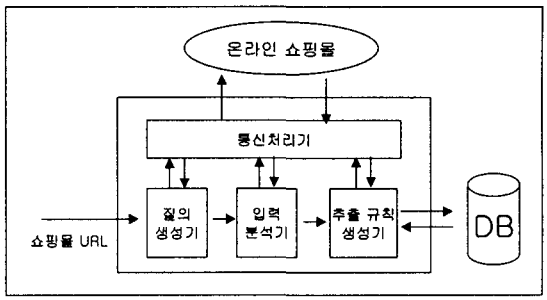
사용자의 개입 없이 주어진 쇼핑몰의 주소를 이용하여 자동으로 Wrapper를 생성하기 위해서는 먼저 해당 쇼핑몰에서 제공하는 상품정보를 얻어 이를 질의어로 생성한다. 각 쇼핑몰에서 상품에 관한 검색을 지원하는 문서의 위치와 검색 지원 방법이 서로 다르므로 이를 해결하기 위해 입력으로 받은 쇼핑몰의 주소를 이용하여 주어진 쇼핑몰에서 상품에 관한 검색을 지원하는 문서를 찾아내고 검색에 필요한 정보인 Action, Method, Parameter들을 추출하는 일을 수행한다. 이때 생성된 질의어를 이용하여 상품검색을 실시하고 결과로 되돌려지는 응답문서를 분석하여 찾아낸 검색 부분이 실제 상품 검색인지를 판단하게 된다.

본 논문에서 구현한 Wrapper 자동생성기의 전체 구성은 <그림 1>과 같으며 해당 온라인 쇼핑몰에서 상품에 대한 검색 지원이 있다는 것을 전제로 설계 및 구현하였다. 시스템의 구현언어는 Java를 사용하였고, JDK 1.2를 이용하였다[12].

각 모듈의 기능과 구조를 살펴보면 다음과 같다.

① 질의 생성기

상품 검색을 하고 그 결과로 응답 페이지를 얻어 상품 정보 추출 규칙을 생성하기 위해 가격 정보를 이용하여 질의어를 생성한다. 본 논문에서는 해당 쇼핑몰에서 이러한 상품 정보를 제공하고 있고, 제공하는 상품에 대한 검색이 가능하다고 가정하며 검색에 사용된 질의에 대한 정보는 데이터베이스에 저장된다. 이 정보는 추후에 쇼핑몰이 변경되었을 경우 이용될 수 있다.



<그림 2> Wrapper 자동 생성기의 시스템 구성도

② 입력 분석기

입력 분석기는 질의 생성기로부터 전달된 쇼핑몰의 주소를 이용하여 주어진 쇼핑몰에서 상품에 관한 검색을 지원하는 문서를 찾아내고 검색에 필요한 정보들을 추출하는 일을 수행하며, 같이 전달되는 질의어 후보들을 이용하여 검색 부분이 실제 상품 검색에 관한 부분인지를 실제 질의를 통해 판단하며, 원하는 형태의 결과인 경우는 응답 결과와 검색 정보를 추출 규칙 생성기로 전달한다.

본 논문에서는 이러한 상품 검색 서비스를 이용하여 wrapper를 생성하기 위해서 먼저 쇼핑몰에서 제공하는 검색 서비스의 검색 주소 등 검색 서비스를 위한 각종 정보가 필요하다. 필요한 정보는 상품 검색을 실제로 서비스해주는 검색 주소, 질의어를 전달하는 방법, 실제 질의어를 전달하는 형식이다.

③ 추출 규칙 생성기

추출 규칙 생성기는 입력 분석기를 통해 전달된 응답 문서를 입력으로 받아 추출하고자 하는 정보의 규칙성을 파악하고, 파악된 정보를 이용하여 데이터베이스에 저장한다.

④ 통신 처리기

본 논문에서 기술되는 각 처리 모듈은 주어진 쇼핑몰에 접속하여 원하는 정보를 얻고 이 정보를 바탕으로 각각의 기능을 수행하고 있다[13,14]. 이러한 일을 수행하기 위해서 주어진 쇼핑몰 주소 정보를 이용하여 접속하고 결과로 문서를 받아서 되돌려주는 부분이 통신 처리기이다. 우리는 Java의 URL class를 이용하여 구현하였다.

⑤ 데이터베이스

본 논문에서는 최종결과와 상태 정보 등 자료의 저장과 검색을 위해 데이터베이스를 이용하고 있으며 실제 구현에서는 데이터베이스 엔진으로 Mysql을

이용하였다.

4. 실험 및 결과 분석

본 논문에서 구현한 Wrapper를 이용해서 쇼핑몰에서 제공하는 상품정보를 받아오는 방법은 다음과 같다. 예를 들어, 교보문고 사이트로 이동하기 위해 'www.kyobobook.co.kr'이라는 주소를 입력하면 교보문고 사이트에서 지원하는 검색 URL인 'http://www.kyobobook.co.kr/intershoproot/eCS/Store/en/Home/home.htm'의 주소로 이동한다. 이동한 URL에서 검색창을 찾아 다시 주어진 키워드로 입력하여 찾아낸다.

검색결과 URL과 검색된 도서명을 저장한다. 그리고 가격 정보를 찾기 위해서 HTML Tag에서 "00"을 찾거나 "원"을 찾아 앞과 뒤의 tag를 제거한 후, 가격정보만 남는다. 결과적으로, link된 웹페이지 URL과 도서명, 가격 등 세 가지를 검색결과로 제시한다.

그러나, 자바 스크립트로 URL을 넘기는 사이트나 검색기능 자체를 지원하지 않는 사이트에 대해서는 Wrapper를 실행시킬 수 없었다.

5. 결론 및 향후 연구방향

최근 인터넷 기술의 발전과 더불어 온라인 쇼핑몰의 증가로 사용자가 어떤 특정한 제품에 대해 구매를 하고자 할 때, 여러 온라인 쇼핑몰의 웹사이트를 일일이 수동적으로 확인하지 않고도 벤더가 제공하는 상품에 대해 비교 할 수 있도록 도와주는 비교쇼핑 에이전트가 등장하게 되었으며 이러한 에이전트를 구성할 때 필수 적인 Wrapper를 자동으로 생성해주는 방법이 필요하게 되었다.

이에 본 논문에서는 비교쇼핑 에이전트를 위한 Wrapper의 자동 생성에 목적을 두고, 테이블 헤더 정보로부터 가격 정보 추출을 위해 휴리스틱(heuristic)한 방법으로 온라인 쇼핑몰에서 제공하는 상품에 관한 가격정보를 추출하였다.

본 논문에서 구현한 Wrapper의 자동 생성으로, 수동적으로 생성했을 때에 비해 시간적, 경제적 효과를 기대할 수 있으며 이는 쇼핑몰의 수가 증가하면 할수록 더욱 효과는 증대될 것이다. 또한 XML을 사용하여 Wrapper를 생성함으로써 비교쇼핑 에이전트간 더욱 다양한 정보 교환이 가능하고 이를 통해 더욱 정확한 Wrapper 생성이 가능하게 될 것이다.

본 논문에서는 문서가 일정한 형식으로 구성된 경우에 한하여 Wrapper를 생성하고 실험을 통하여 결과를 보였으며 실패 원인에 대해 알아보았다. 향후의 연구방향으로는 입력 분석부분을 강화하고 일정한 형식이 아닌 일반 텍스트로 구성된 웹 문서에 대한 Wrapper의 생성에 관한 연구와 가격정보 외 벤더가 제공하는 상품에 관련된 다른 정보들을 효과적으로 추출하는 방법에 관한 연구가 필요하며, 검색주소를 찾기 위한 광고 배제 방법과 직관적 판단이 불가능한 경우 상품명 추출 방법에 대한 더욱 정확한 추출 방법에 대한 연구도 요구된다.

참 고 문 헌

[1] Nicholas Kushmerick, Daniel S. Weld, Robert Doorenbos, "Wrapper Induction for Information Extraction", IJCAI-97 (Nagoya), 1997.

[2] M. Wooldridge, N. Jennings, "Intelligent Agents: Theory and Practice", The Knowledge Engineering Review, Vol. 10, No. 2, pp.115-152, 1995.

[3] Nicholas R. Jennings, Michael J. Wooldridge(Eds.), Agent Technology: Foundations, Applications, and Markets, Springer, 1998.

[4] Robert B. Doorenbos, Oren Etzion, Daniel S. Weld, "A Scalable Comparison-Shopping Agent for the World-Wide Web", ACM Autonomous Agents97, 1997.

[5] Yavis, http://www.yavis.com

[6] Omi, http://www.omi.co.kr

[7] Shobinder, http://www.shopbinder.com

[8] Nicholas Kushmerick, Wrapper Induction for Information Extraction, Proceedings of 15th International Conference on Artificial Intelligence(IJCAI-95), pp.729-735, 1995.

[9] Oren Glickman, Rosie Jones, "Examining Machine Learning for Daptable End-to-End Information Extraction Systems", The AAAI-99 Workshop on Machine Learning for Information Extraction, 1999.

[10] Hsu, Dung, "Generating Finite-State Transducers For Semi-Structured Data Extraction for the Web, Information Systems, 1998.

[11] Muslea, Minton, Knoblock, "STALKER:

Learning Extraction Rules for Semistructured, Web-based Information Sources", 1998.

[12] 이현우 외, Java Programming Bible, 영진출판사, 1999.

[13] KQML Advisory Group, "An Overview of KQML:. A Knowledge Query and Manipulation Languages", 1992.

[14] T. Finin, R. Fritzson, "KQML - A Language and Protocol for Knowledge and Information Exchange", Proceedings of the 13th International Distributed Artificial Intelligence Workshop, pp.127-136, 1994.