비교쇼핑을 위한 쇼핑몰 학습 에이전트

구남숙*, 양재영, 서희경, 최중민
한양대학교 전자계산학과

A Shoppingmall Learning Agent for Comparisonshopping

Namsuk Koo, Jaeyoung Yang, Heekyoung Seo, Joongmin Choi
Dept. of Computer Science and Engineering, Hanyang University

요 약

전자상거래는 서비스 제공자마다의 특정 인터페이스를 가진다. 그렇기 때문에 사용자는 원하는 결과를 얻고자 검색에 많은 시간과 노력을 투자해야 한다. 그래서 여러 쇼핑몰을 통합하여 사용자에게 결과를 제공하는 쇼핑 에이전트가 현재 여러 분야에서 연구되고 있다. 그러나 현재 개발된 쇼핑 에이전트들은 대부분 새로운 도메인이 추가되면 쇼핑몰에 대한 규칙을 수동작성 해야 한다는 문제점을 갖고 있다.

본 논문에서는 기존 쇼핑 에이전트의 이러한 한계를 극복하기 위한 쇼핑몰 학습을 위한 패턴생성 알고리즘을 제안하고, 이 알고리즘을 이용한 시스템을 구현하였다.

1. 서 론

최근 인터넷이 발달하면서 우리의 생활 전반에 그 영향을 미치고 있다. 그 중 전자상거래는 다양한 서비스 제공과 이용의 편리함 때문에 사용자가 날로 증가하고 있다. 그러나 전자상거래는 서비스를 제공하기 위한 표준이 존재하지 않기 때문에 상점이 증가할수록 선택의 폭은 넓어지지만, 사용자는 혼란을 느낄 수 있다. 그 이유는 각 사이트들은 각기 다른 사용자 인터페이스를 통하여 서비스를 제공하기 때문인데, 이런 사이트의 이질성은 사용자에게는 자신이 원하는 조건에 맞는 상품을 검색하기 위해서 많은 시간과 노력을 소비하게 하고, 사용자 자신도 어떤 상품을 검색하고 선택했을 때 과연 자신이 원하는 최상의 상품인지 확신할 수 없는 경우가 발생하기도 한다.

그래서 요즘은 온라인 쇼핑몰을 이용하는 사용자들의 편의를 위하여 여러 사이트를 통합하여 하나의 인터페이스를 제공하기 위한 쇼핑 에이전트에 대한 연구가 진행되고 있다. 그러나 현재 개발되어 있거나 개발중인 쇼핑 에이전트는 구성 측면에서 여러 가지 제약을 갖고 있는데, 주로 문제가 되는 것이 새로운 도메인이 추가되었을 때 새로운 쇼핑 몰에서 해당 정보를 추출해내기 위해 필요한 규칙(rule)을 수동으로 만들어야 한다는 점이다. 이 경우 쇼핑 에이전트의 성능저하를 가져오고, 도메인 추가시 문제가 발생하는 쇼핑몰은 에이전트에 추가될 수 없으므로 사용자에게는 제한된 상품정보만을 제공하게 된다.

본 논문에서는 이런 기존 쇼핑 에이전트에서 도메인 추가시 발생하는 문제점을 해결하기 위해 각 온라인 쇼핑몰의 정보를 추출해 내는 규칙을 자동으로 생성하는 패턴생성 알고리즘을 제시하고, 이를 이용하여 쇼핑몰 학습 에이전트를 구현하였다.

2. 관련연구

웹 상에서 원하는 정보를 얻기 위해서는 정보의 저장 형태를 파악해야 한다. 예를 들면, 정보들이 내용에 따라 장이나 단락으로 구별되어 있거나, 특정 tag로 묶여있는 경우 - 예를 들면, table에 관련된 tag들을 이용하는 경우 - 그리고 특정 폰트를 이용하는 경우 등 다양한 형태가 존재한다. 이런 정보들을 이용하여 wrapper를 만들게 되면 wrapper는 사용자 질의를 통해 가져온 결과를 사용자에게 보여주거나, DB에 정보를 저장하는 역할을 한다. 이것이 semi-structured 문서의 형태가 된다 [1][2][4][5].

BargainFinder는 음악 CD 상점 10개를 통합하여 사용자가 원하는 앨범에 대한 가격만을 보여준다. 이 경우 상점은 고정되어 있고, 한번 상점 정보에 대해 지식 베이스에 저장하면 문제없이 사용할 수 있다. 그러나 지식 베이스는 개발자가 구축하기 때문에 도메인 확장시 문제가 되고, 쇼핑몰 학습에 대해서 고려하지 않는다.

ShopBot은 비교쇼핑 에이전트로 정보제공을 위해 복잡한 자연어 처리를 이용하는 것이 아니라 학습을 통해 얻어진 쇼핑몰 구성의 규칙성을 이용한다. 그래서 이런 점은 본 논문에서 제안한 시스템과 유사성을 가지지만, ShopBot에서는 'regularity'라는 일종의 규칙성을 전제로 하기 때문에 이 규칙에서 벗어나는 경우는 에이전트의 도메인에 포함되기 어렵다 [3].

3. 쇼핑몰 학습 에이전트

3.1 시스템 개요

쇼핑 에이전트는 새로운 도메인이 추가될 경우 쇼핑몰에 맞는 규칙을 작성해야한다. 그러나 수동으로 규칙을 작성할 경우 규칙은 정확히 작성할 수 있지만, 시스템 효율은 저하된다.

그림1은 본 논문에서 제안하는 패턴생성 알고리즘을 이용한 자동규칙생성기의 구성도이다.

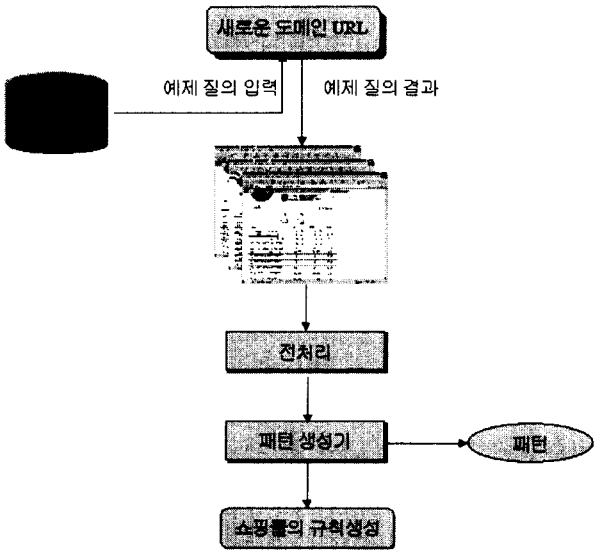

그림 1 자동규칙 생성기

쇼핑몰들은 각각의 인터페이스로 사용자에게 결과를 제공한다. 그러나 대부분의 쇼핑몰은 상품에 대한 정보를 데이터베이스를 이용해 저장하고 있다. 그러므로 저장된 정보를 다시 웹 문서로 나타내기 위해서는 일정한 형식을 이용해 보여주게 된다. 이것을 이용하면 각 쇼핑몰의 규칙을 만들기 위한 패턴을 찾아낼 수 있다. 패턴을 찾기 위해 이용하는 것은 예제 질의를 통해 얻어진 결과 파일이다. 예제 질의는 실험을 통해 패턴 찾기에 가장 알맞다고 판단된 질의를 이용한다.

그림 2는 서점 아마존에서 'Java' 라는 예제질의를 전송했을 때 제공하는 도서정보의 HTML 형태이다.

```
<a href = CGI address> Abstract Data Type in Java </a>
~ <nobr><font color=#990033>Usually ships in 24 hours</font></nobr>
<dd> --> <blank tag>
Michael S. Jenkins / Paperback / Published1997
<br> --> <blank tag>
Our Price: $44.95
<br> --> <blank tag>
```

그림 2 아마존 검색결과

예제 질의에 대한 결과는 헤더와 테일 정보를 제외하면 이와 같은 형식을 기본으로 반복적으로 나타난다. 이런 특징을 다른 쇼핑몰에도 적용하여 그 쇼핑몰의 패턴을 찾아내는데 이용한다.

3.2 전처리

쇼핑몰에서 패턴은 상품에 대한 정보를 어떤 내용과 어떤 순서로 보여주는지를 의미한다. 그러므로 똑같은 내용이라도 어떻게 파일의 내용을 분리하는가에 따라 패턴은 달라지게 된다. 그러므로 결과파일을 분리하는 기준이 있어야 하는데, 본 논문에서는 사용자에게 보여지는 형태를 기준으로 한다. 즉 위의 아마존의 예를 표현한 것과 같이 blank tag를 이용하여 이 tag가 발생하기 이전까지를 하나의 의미단위로 판단하게 된다. 위의 예를 살펴보면 blank tag를 기준으로 3개의 의미단위로 나뉘어 있다.

3.3 패턴생성을 위한 의미파악

전처리 된 파일은 초기 HTML 파일을 재구성한 것이므로 그 자체만으로 패턴을 찾아내는 것은 불가능하다. 패턴을 찾기 위해서는 전처리 파일의 각 line이 무엇을 의미하는지 파악해야 한다. 여기서 line은 한 blank tag를 기준으로 다음 blank tag가 나올 때까지, 즉 화면상의 한 line을 의미한다. 패턴생성의 효율을 높이기 위해 각 line의 의미를 파악한 후 이것을 숫자를 이용해서 새로운 파일에 저장하게 된다. 패턴을 생성할 때는 새로 저장된 파일을 이용한다.

그림 3은 전처리 된 파일이 어떻게 숫자로 변환되는지를 나타낸 것이다. 각 숫자들은 실제 내용이 무엇을 의미하는지를 나타낸다. 여기서는 '3'은 도서명을 나타내고, '2'는 blank tag, '1' 은 가격을 나타내고, '0'은 의미를 알 수 없는 텍스트를 나타낸다.

```
<a href = CGI address> Abstract Data Type in Java </a>          ---- 3
~ <nobr><font color=#990033>Usually ships in 24 hours</font></nobr>   ---- 2
<dd>                                                            ---- 0
Michael S. Jenkins / Paperback / Published1997                  ---- 2
<br>                                                            ---- 1
Our Price: $44.95                                               ---- 2
<br>
```

그림 3 숫자파일 생성 예

3.4 패턴생성

위에서 구성된 숫자파일을 분석해 보면 같은 형태의 숫자배열로 계속 반복되는 것을 확인할 수 있다. 앞에서도 말했듯이, 쇼핑몰에 존재하는 데이터베이스는 일정 형태로 사용자에게 검색결과를 보여주게 된다. 그래서 상품검색 결과파일을 분석해 보면, 상품정보를 보여주는 부분에서 같은 숫자의 반복을 확인할 수 있다. 이것이 패턴이 되는 것이다.

아마존의 예를 보면, 위의 정보가 반복되므로 '32021' 이라는 패턴을 생성하게 된다.

패턴을 찾을 때 기준이 되는 것은 상품의 가격이다. 상품정보에는 가격이 들어가 있다. 그래서 가장 처음 가격이 발생하는 순간부터 상품정보가 나타난다고 가정한다. 즉, 가장 처음 나타나는 가격과 상품명 이전의 정보는 헤더정보로 간주하게 된다. 그리고 파일이 끝나기 전 마지막 가격정보 이후는 테일로 간주하게 된다. 이것은 에이전트가 별도의 처리 없이 본 논문에서 제안한 패턴생성 알고리즘을 수행함으로서 헤더와 테일 정보를 구별할 수 있다는 것을 말한다. 그러므로 헤더나 테일을 제거할 때 정확히 제거하지 못하여 발생했던 문제점들을 해결할 수 있다.

다음은 패턴을 찾아내는 부분을 pseudo코드로 나타낸 것이다.

```
While(True){
        CostLocation := FindCost(SaveNumberFile,Ith);
        GoodsNameLocation := FindGoodsName(SaveNumberFile);
        SavePattern := SaveNumber(GoodsNameLocation, CostLocation);

        while(Not EOF(SaveNumberFile)){
                CostLocation := FindCost(saveNumber);
                GoodsNameLocation := FindGoodsName(SaveNumberFile);
                IsPattern := SaveNumber(CostLocation, GoodsNameLocation);

                if(IsPattern = = SavePattern)
                        PatternCount++;
        }
        if(PatternCount >= Limit)
                break;
        else
                Ith++;
}
```

그림 4 패턴생성하기

루프 시작은 가격정보를 찾는 것이다. 가격정보를 찾게 되면 그 위치를 기준으로 가격에 해당하는 상품명의 위치를 확인하고 상품명과 가격 사이의 숫자들을 초기 패턴이라고 정의한다. 이것이 SavePattern에 저장되게 된다. 그리고 이후부터 계속 가격과 상품명 사이의 숫자들을 임시 버퍼에 저장한 후 초기 패턴과 비교하여 일치하면 패턴과 일치하는 횟수를 세는 PatternCount를 증가시킨다. 이런 동작을 파일을 끝까지 반복하여 PatternCount가 일정기준 이상 발생하면 이것은 해당 쇼핑몰의 패턴으로 인정하게 된다. 만약 인정되지 않으면 초기 가격정보 다음 가격정보를 기준으로 한 새로운 패턴을 정의한 후 이와 같은 동작을 반복하게 된다.

대부분의 경우 처음에 패턴을 찾고, 많아야 2번 정도면 쇼핑몰의 패턴을 찾게 된다.

4. 쇼핑몰에서 상품정보의 추출

4.1 패턴을 이용한 파일수정

실제 쇼핑 에이전트에서는 각 쇼핑몰에 사용자 질의를 전달하고 상품검색 결과를 가져오게 되면 결과파일은 전처리와 패턴생성을 위한 파일 생성을 통하여 숫자로 표현된 결과파일을 생성하게 된다. 이 파일을 통해서 패턴이 생성되고 파일내의 대부분의 상품정보는 패턴과 일치하는 숫자배열을 갖고 있지만, 어떤 경우는 상품에 대한 정보이면서도 패턴과는 일치하지 않는 경우가 있다. 이유는 전처리가 끝난 파일을 다시 의미파악 단계에서 숫자를 변경시키게 되는데, 여기서 의미파악이 잘못되면, 상품명이면서도 상품명으로 인식을 못하는 경우가 발생한다. 그러나 올바르게 인식하지 못하는 경우라도 그 구성은 패턴과 일치하는 면이 있다. 그러므로 패턴과 비교하여 어느 정도 일치하면 잘못된 부분을 패턴에 해당하는 숫자로 고치는 것이다.

만약 아마존에서 '32021'이라는 패턴을 갖고 있는데, 도서명을 잘못 인식하여 '02021'로 파일이 구성되었다면 '32021'이라는 패턴을 통하여 잘못된 부분도 수정이 가능하다는 것이다.

이것은 상품명이나 기타 상품정보에 대한 동의어 처리 없이도 올바른 정보를 추출할 수 있다는 것을 의미한다.

4.2 상품정보 추출하기

추출은 수정단계를 거친 최종 파일과 실제 내용을 담고있는 전처리 파일을 이용하여 이루어진다.

패턴을 이용하여 작성한 규칙에는 몇 번째 가격정보부터 실제 검색된 상품정보인가를 나타내는 부분과 쇼핑몰의 패턴이 같이 포함되어 있다. 이것은 실제 상품정보에서 뿐만 아니라 헤더부분에서도 가격정보가 발생할 수 있기 때문이다.

이 규칙을 이용하여 가장 처음 발생하는 상품의 상품명부터 찾아가면서 실제 내용을 추출해 내면 된다. 실제 내용은 전처리 파일에 저장되어 있으므로, 상품정보에 해당하는 숫자가 나타나면 전처리 파일의 해당내용을 추출하면 된다.

결과를 추출할 때는 상품명과 가격을 쌍으로 추출하게 된다. 만약 상품명과 가격이 쌍을 이루지 않으면 그것을 올바른 상품정보가 아닌데, 이유는 태일 부분에서도 상품명이라고 판단한 line이 존재할 수 있기 때문이다.

5. 결론 및 향후 연구 계획

기존의 쇼핑 에이전트는 새로운 도메인을 추가하기 위해서는 정보추출을 위한 규칙을 수동으로 작성해야 한다. 이런 문제점은 쇼핑 에이전트의 성능을 저하시키고, 사용자의 검색을 위한 추가 노력을 요구할 수도 있다.

본 논문에서 제안된 시스템은 규칙을 자동 생성할 수 있는 패턴생성 알고리즘을 이용하여 기존 쇼핑 에이전트에서 발생했던 수동 규칙생성의 문제점을 해결했다. 또 이 시스템에서 생성된 패턴을 이용하면 기존 쇼핑 에이전트에서 상품정보를 찾기 위해 필요한 동의어 처리를 위한 도메인 지식 없이도 정보를 추출할 수 있으므로 검색 성능 역시 향상된다. 그리고 상품정보 이외의 정보인 헤더나 테일을 별도의 처리 없이 정확히 구별함으로서 시스템의 성능을 향상시킨다.

향후 연구 계획으로는 추출된 정보를 이용한 비교 쇼핑 에이전트의 구현과 패턴생성 알고리즘을 전자 상거래 이외의 정보추출 분야에 적용시키는 작업을 하고자 한다.

[참고문헌]

[1] Naveen Ashish, Craig Knoblock, "Wrapper Generation for Semi-structured Internet Sources", ACM SIGMOD Workshop on Management of Semi-structured Data, Tucson, AZ, 1997.

[2] J. Hammer, H. Garcia-Molina, J. Cho, R. Aranha, A. Crespo, " Extracting Semistructured Information from the Web", ACM SIGMOD Workshop on Management of Semi-structured Data, Tucson, AZ, 1997.

[3] Robert B. Doorenbos, Oren Etzioni, Daniel S. Weld, "A Scalable Comparison-Shopping Agent for the World-Wide Web", Proceedings of the First International Conference on Autonomous Agents, 1997.

[4] Laura Bright, Jean-Robert Gruser, Louiqa Raschid, Maria Esther Vidal, "A Wrapper Generation Toolkit to Specify and Construct Wrappers for Web Accessible Data Source(WebSources)", International Journal of Computer Systems Science and Engineering, 1999.

[5] Paolo Atzeni, Giansalvatore Mecca, Paolo Merialdo, "Semistructured and Structured Data in the Web: Going Back and Forth", Workshop on Management of Semistructured Data, 1997.