# XML 기반의 Wrapper 자동 생성 에이전트

서희경[0] 양재영 정현섭 최중민
한양대학교 전자계산학과
{hkseo,jyyang,hsjung,jmchoi}@cse.hanyang.ac.kr

## Automatic Wrapper Generating Agent based on XML

Heekyoung Seo[0] Jaeyoung Yang Hyunsup Jung Joongmin Choi
Dept. of Computer Science & Engineering, Hanyang University

요 약

본 논문은 사용자를 대신해서 웹상의 여러 곳에 존재하는 정보를 추출하고 통합하여 사용자에게 제공하기 위한 에이전트 시스템을 설계하고자 한다. 정확한 정보 추출을 위해서는 추출하고자 하는 정보의 위치를 찾아내는 정보 추출 규칙이 요구된다. 이러한 규칙을 알아내기 위해서 본 논문에서 제안하는 시스템은 XML로 기술된 도메인 지식을 이용한다. 이 도메인 지식은 논리적 라인의 의미 분석에 사용되며, 논리적 라인의 의미를 기반으로 도메인 문서에서 추출해야 하는 정보의 패턴을 학습한다. 학습된 패턴에서 XML로 기술된 규칙을 생성하는데, 이 규칙은 Wrapper이 된다. 이렇게 생성된 규칙을 이용해서 정보를 추출하게 되며, 추출된 정보를 통합해서 사용자에게 제공하게 된다.

## 1. 서론

현재 인터넷의 발달로 웹상에 존재하는 정보들은 꾸준히 늘어가는 추세이다. 웹상에 정보의 양이 증가함으로써 사용자의 요구에 맞는 정보를 찾을 확률이 높아진다는 좋은 점이 있지만, 정보의 다량화로 인해 사용자가 정보를 검색할 때 더 많은 시간과 노력을 소비해야 한다는 단점이 있다. 이러한 문제점을 해결하기 위해서 사용자의 검색시간을 줄이는 시스템이 필요하다. 사용자의 검색 시간을 줄이기 위해 여러 사이트를 검색하는 대신 하나의 인터페이스를 통해서 여러 사이트의 정보를 한 번에 볼 수 있도록 하는 통합 시스템이 제안되었다. 즉 웹 상점에서 상품을 검색할 때 사용하는 비교쇼핑 시스템과 같은 시스템이 부동산 매물 검색에서도 요구된다.[9,10]

지금까지 많은 도메인에 대해 정보를 추출하기 위한 연구가 진행되어 왔다. 하지만 지금까지의 이러한 분야의 연구들은 어떤 한정된 도메인에만 적용 가능하도록 수행되어 왔다. 본 논문에서는 XML 기반의 도메인 지식을 변경해 주면 다른 도메인에도 적용 가능하도록 하는 시스템을 연구했다.

현재 이러한 연구를 위해 도메인을 부동산 매물 정보 검색으로 하고 XML 기반의 도메인 지식을 구성하였다. 또한 도메인 지식을 기반으로 패턴을 추출한 후 패턴에 맞는 XML 기반의 규칙을 생성하도록 하였다.

## 2. 관련연구

AutoSlog, LIEP, PALKA, HASTEN 시스템들은 문법이 정확하고 간단한 텍스트로 구성된 문서에서 정보를 추출하기 위한 규칙을 학습하는 시스템들이다.[1] 이러한 시스템들은 자연어 처리를 바탕으로 한 시스템들이다. ACI(Autonomous Citation Indexing)[2] 시스템은 자동으로 웹상에서 문헌의 위치를 파악하고 인용구를 추출한다. 이를 이용해 다른 형식의 같은 정보를 구별해내고, 학술 문헌에서 인용 문맥을 구분한다. 이 시스템은 PostScript, PDF 파일들의 텍스트 파일로의 변환을 통해서 문서의 정보를 추출한다. 따라서 웹상에서 HTML 태그 등을 이용하지 않기 때문에 Web 문서 처리를 기반으로 하는 시스템은 아니다.

온라인 상의 문서에서 정보를 추출하기 위해 나타난 연구 분야가 Wrapper Induction 분야이다. Wrapper는 하나의 문서에서 필요한 정보를 추출하기 위한 규칙을 말한다.[6] 이 분야에서 연구되는 시스템들은 대부분 HTML의 특성을 이용해 구분기호(delimiter) 기반의 추출 패턴을 생성한다. 이것이 Wrapper가 된다.[7] 이러한 시스템으로 처음 개발된 것이 WIEN이다.[3,4,5] WIEN을 확장한 시스템이 SoftMealy이다.[1]

웹상의 정보 추출을 위한 또 하나의 시스템은 XWRAP라는 시스템이 있다. XWRAP는 문서를 HTML 태그의 특성을 나타

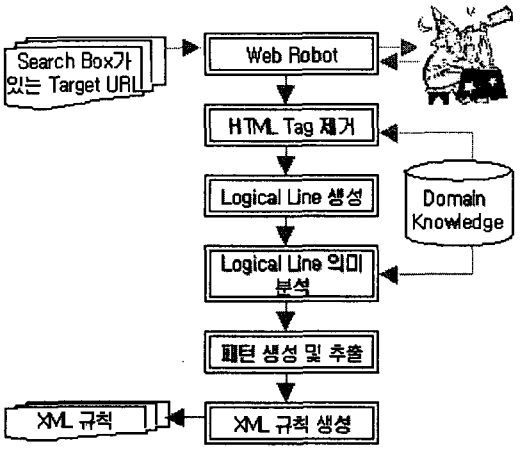


그림 1. 시스템 구조도

내는 계층구조로 파싱해서 정보의 의미를 파악하고, 해당 사이트에 맞는 정보 추출규칙을 XML 파일로 생성하는 시스템이다.[8]

이 논문에서 제안하고자 하는 시스템은 HTML 태그 중 테이블 태그(<td>, <tr>) 와 리스트형 태그(<p>, <br>, <li>) 만을 남긴 HTML 정보 소스를 이용해서 해당 사이트를 학습하여 XML 기반의 정보 추출 규칙을 생성한다. 생성된 규칙을 이용해 여러 사이트의 정보를 추출하고 통합해 사용자에게 제공한다.

## 3. 시스템 구조

본 논문의 시스템 구조는 그림 1과 같다. 아래에 그림 1의 부분별 역할을 설명하였다.

- Query 요청 : 검색 창이 있는 URL의 질의문에 질의를 추가해서 해당 사이트에 결과를 요청한다. 질의에는 도시, 주, 최소 가격, 최대 가격, 침실 수, 욕실 수 중에서 해당 사이트에서 입력으로 받아들이는 것들만 질의문에 입력해 준다.
- HTML 태그 제거 : 테이블 태그와 리스트형 태그, 이미지 태그와 하이퍼 링크를 제외한 나머지 모든 태그를 제거한다.
- Logical Line 생성 및 의미 분석 : 테이블 태그와 리스트형 태그는 사람에게 보이는 형식으로 변환 후 태그는 제거한다. 도메인 지식을 참고해서 각 라인마다 의미하는 것이 무엇인지를 분석한다. 한 라인에 두 가지 이상의 의미가 존재하면 각각 다른 라인에 존재하도록 새로운 라인을 생성해 준다.
- 패턴 생성 및 추출 : 생성된 논리 라인들에서 반복되는 패턴을 해당 사이트의 패턴으로 추출한다.
- XML 규칙 생성 : 생성된 패턴을 이용해서 XML 규칙을 생성한다.
- XML 규칙을 이용한 정보 추출 : 사용자가 질의를 주게 되면 입력 각 각의 질의문에 사용자 질의를 추가해 요청하고 결과를 가져온다. 각 사이트의 XML 기반 규칙을 이용해 문서에서 필요한 정보만을 추출한 후 사용자에게 제공한다.

## 4. 의미 분석 및 규칙 생성

```
<KNOWLEDGE>
  <OBJECTS>
    <OBJECT> PRICE </OBJECT>
    <OBJECT> BED </OBJECT>
    <OBJECT> BATH </OBJECT>
    <OBJECT> CITY </OBJECT>
    <OBJECT> MLS </OBJECT>
    <OBJECT> DETAIL </OBJECT>
  </OBJECTS>

  <PRICE>
    <ONTOLOGY>
      <TERM> PRICE </TERM>
      <TERM> $ </TERM>
    </ONTOLOGY>
    <FORMAT>
      <FORM> <REPLACE> ONTOLOGY </REPLACE> DIGITS </FORM>
      <FORM> DIGITS <REPLACE> ONTOLOGY </REPLACE> </FORM>
    </FORMAT>
  </PRICE>

  <BED>
    <ONTOLOGY>
      <TERM> BEDROOM </TERM>
      <TERM> BED </TERM>
      <TERM> BEDS </TERM>
      <TERM> BR; </TERM>
      <TERM> BEDS </TERM>
      <TERM> BEDROOMS </TERM>
    </ONTOLOGY>
    <FORMAT>
      <FORM> <REPLACE> ONTOLOGY </REPLACE> DIGITS </FORM>
      <FORM> DIGITS <REPLACE> ONTOLOGY </REPLACE> </FORM>
    </FORMAT>
  </BED>
```

그림 2. 도메인 지식

### 4.1 의미 분석

도메인 지식을 이용해서 논리 라인의 의미를 분석한다. 도메인 지식의 형태는 그림 2와 같다. 각 OBJECT에 존재하는 태그는 항상 2가지(ONTOLOGY, FOMAT) 이다. ONTOLOGY는 OBJECT를 찾기 위해 사용되는 정보 집합이다. ONTOLOGY 만으로는 찾은 정보가 해당 OBJECT인지 알 수 없다. 따라서 이때 정보가 OBJECT 인지 알아내기 위해 FORMAT의 형태를 띄고 있는지 판별을 통해 알아낸다.

매물 정보에서 추출해야 하는 정보는 <OBJECTS> 태그안에 <OBJECT>로 나타나 있다. 하나의 논리 라인에 OBJECT가 존재하는지 판별하기 위해 도메인 지식을 어떻게 사용하는지를 다음 Pseudo Code로 나타내었다.

```
For  ( each <OBJECT> )
  If a line includes a term in <ONTOLOGY>
    If a line conforms to a form in <FORMAT>
      If a line includes other OBJECT information
        Separate a line into two lines
        Recognize the separate line as OBJECT
      Else
        Recognize a logical line as OBJECT
```

위와 같은 방식으로 각 논리 라인의 분석과정을 거친 후 반복 되는 패턴을 찾게 된다.

### 4.2 패턴 추출

각 논리 라인의 의미 분석을 통해서 패턴을 찾게 되면 찾아진 여러 패턴 중 어떤 것을 규칙 생성을 위한 패턴으로 정할지를 결정해야 한다. 패턴은 반복되는 것을 선택한다. 만약 속성(attribute)의 수가 다른 경우는 가장 많은 속성을 가지는 패턴을 선택한다. 즉, 어떤 검색 결과의 경우 가격, 침실, 욕실, MLS #, 상세페이지 정보를 모두 포함하는 매물이 있는 경우가 있고, 가격, MLS # 만을 포함하는 경우가 있다면 가격, 침실, 욕실, MLS #, 상세페이지가 패턴으로 찾아진다. 이처럼 패턴 추출 시에는 가장 많은 속성을 가지는 패턴을 선택하여 가능한 많은 정보를 추출하도록 하였다.

```
<Start>
    <Form>
        Query 관련 규칙
    </Form>

    <Home>
        <Price>
            <Ontology> $ </Ontology>
            <Ident> : </Ident>
            <Format> Digits,Digits </Format>
            <Operation> Ontology*Format </Operation>
        </Price>

        <Bed>
            <Ontology> BR </Ontology>
            <Ident> : </Ident>
            <Format> Digits </Format>
            <Operation> Ident*Format*Ontology </Operation>
        </Bed>
        .....
    </Home>
</Start>
```

그림 3. XML-based Rule

### 4.3 규칙 생성

패턴이 찾아지면 이것을 이용해서 XML 기반의 규칙을 생성해 내야 한다.

● **규칙 형태 :** 그림 3 과같다. <Ontology> ... </Ontology>는 OBJECT를 추출하기 위해 맨 먼저 검색하는 정보이다. <Format>은 추출 할 정보의 타입이고, <Ident>는 구분기호(delimiter)를 나타내기 위해 사용되는 정보이며, NULL이 될 수도 있다. <Operation>은 정보 추출 시 추출해야 하는 정보의 위치를 나타낸다.

● **규칙 생성**

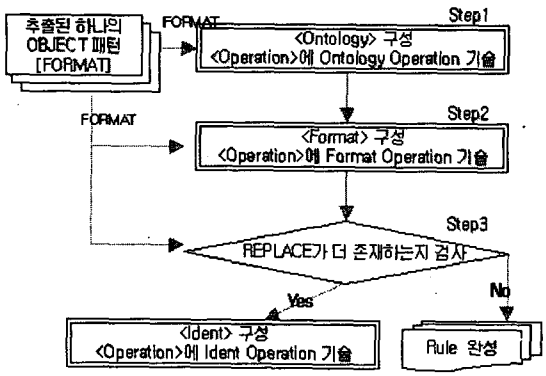


그림 4. 규칙 생성

위의 그림 4 의 방식으로 각 OBJECT 별 패턴에서 규칙을 생성하게 되면 그림 3 과 같은 형태의 규칙 파일이 생성된다.

● **생성 규칙의 이용 :** 생성된 XML 기반의 규칙 파일은 각 규칙 파일 하나하나가 각 사이트의 wrapper가 된다. 사용자의 입력을 받아 규칙 파일의 Form 부분의 질의 규칙에 따라 질의문에 추가해서 결과를 요청한다. 가져온 결과 문서들에 wrapper의 Form 부분의 규칙을 적용해서 필요한 정보들을 추출해 낸다. 추출된 정보를 통합해서 사용자에게 제공한다.

## 5. 결론

본 논문에서 제안하는 시스템을 이용해서 정보를 검색하게 되는 경우 얻을 수 있는 이점은 사용자의 시간과 노력을 아낄 수 있다는 것이다. 또한 기존의 도메인 지식만으로 학습할 수 없는 문서가 있을 때 도메인 지식만을 추가해주게 되면 학습할 수 있게 된다. 하지만, 여러 사이트를 동시에 검색해야 하기 때문에 네트웍 속도에 영향을 많이 받는 단점이 있다.

현재 부동산 매물 검색 시스템의 경우 입력정보가 비교적 많고, 다양하기 때문에 현재의 시스템 구현과 동시에 자동으로 입력 정보를 찾아내는 것은 어려운 일이다. 입력 정보 자동 검색은 향후 연구 과제로 남아있다.

## 6. 참고 문헌

[1]. Ion Muslea, "Extraction Patterns for Information Extraction Tasks: A Sur vey", The AAAI-99 Workshop on Machine Learning for Information Extraction., 1999

[2]. Steve Lawrence, C. Lee Giles, Kwt Bollaker, " Digital Libraries and Autonomous Citation Indexing" , IEEE Computer pp. 67-71, 1999

[3]. Nicholas Kushmerick, Brett Grace, " The Wrapper Induction Environment" , AAAI98, 1998

[4]. Nicholas Kushmerick, " Wrapper Induction for Information Extraction " , Doctor of Philosopy, University of Washington, 1997

[5] Nicholas Kushmerick, Daniel S. Weld, Robert Doorenbos, IJCAI-97, 1997

[6]. Jim Cowie, Wendy Lehnert, " Information Extraction" , Communication of ACM pp. 80 -91, January 1996

[7]. Alon Y. Levy, " Information Integration" , IEEE Intelligent Systems pp. 12-24, 1998

[8]. Ling Liu, Wei Han, David Buttler, Calton Pu, Wei Tang, " An XML -based Wr apper Generator for Web Information Extraction " , Proceedings of the ACM SIGMOD International Conference, June 1 -4, Philadelphia

[9]. 서희경, 양재영, 구남숙, 최중민, " 전자 상거래에서 Ontology 생성을 위한 인터페이스 에이전트" , 정보과학회 1999 추계 학술대회(II), pp. 30-32, 1999

[10]. Jaeyoung Yang, Joongmin Choi, " A More Scalable Comparison Shopping Agent " , Engineering of Intelligent System(EIS 2000), June 2000