

온톨로지를 활용한 자동화된 지식 습득 방법론 및 효과 분석

박상언

한국과학기술원 테크노경영대학원
(mascon@kgsn.kaist.ac.kr)

이재규

한국과학기술원 테크노경영대학원
(jkleee@kgsn.kaist.ac.kr)

강주영

아주대학교 e비즈니스학부
(jykang@ajou.ac.kr)

시맨틱 웹 관련연구가 증가함에 따라 지능형 에이전트 혹은 규칙기반 시스템 등의 지능적인 웹 환경에 대한 기대 역시 커지고 있다. 그러나 규칙기반 시스템의 활용에는 아직도 규칙습득이 많은 제약이 되고 있다. 이와 같은 제약을 극복하기 위해 웹 페이지로부터 규칙을 습득하기 위한 XRML 방법론이 제안되었다. XRML 방법론은 웹 페이지로부터 규칙을 식별하고 식별된 결과로부터 자동으로 규칙을 생성하는 두 단계로 구성되어 있다. 여기서 규칙의 식별은 규칙생성의 자동화 정도에 매우 중요한 영향을 미친다. 그러나 규칙을 식별하는 작업은 대부분 지식관리자의 수작업에 의존하고 있다. 이러한 지식관리자의 부담을 줄이기 위해 본 논문에서는 온톨로지 기반의 개선된 규칙식별 방법론을 제안하고자 한다. 이를 위해 먼저 OntoRule이라는 이름의 온톨로지를 설계하였다. OntoRule은 자동화된 규칙 식별을 지원하기 위해 사용되며, 규칙의 구성요소들과 구조에 대한 정보를 포함하고 있다. 그리고 OntoRule을 이용하여 규칙을 식별하는 절차를 제안하였다. OntoRule과 규칙식별 절차를 제안하는 과정에서 온톨로지 학습효과, 하향식 접근방식과 상향식 접근방식의 차이, 온톨로지 사용범위 관리, 규칙구성요소의 식별순서, 생략된 변수의 식별과 같은 논점들이 고려되었다. 마지막으로 실험을 통해 제안된 방법론의 효과를 보였다.

논문접수일 : 2005년 11월

게재확정일 : 2005년 11월

교신저자 : 박상언

1. 서론

오늘날의 웹은 수많은 지식들이 산재해 있으며 이러한 지식들은 단순한 데이터로부터 시작해서, 다양한 실시간 정보, 여러 가지 개념, 법률, 문제를 해결하기 위한 규칙에 이르기까지 그 종류도 매우 다양하다. 따라서 이러한 웹 상의 지식을 효율적으로 습득하고 활용하기 위한 연구들이 활발하게 진행되고 있다. 그러나 대부분의 연구들이 온톨로지 수준의 사실이나 개념 [33]을 추출하는 데 초점을 두고 있으며, 규칙 기반 시스템이나 인텔리전트 에이전트에서 사용되는 추론가능한 규칙을 추출하

는 연구는 많지 않다 [14, 33]. SWRL [14]과 같은 규칙 표식 언어에 관한 연구가 추론 가능한 규칙을 완전하게 표현한다고 해도, 사용될 규칙이 충분하지 않다면 많은 응용을 갖기 어려우며 규칙 습득은 여전히 병목으로 자리잡게 될 것이다 [31]. 따라서 규칙을 활용한 지능형 웹을 구축하기 위해서는 웹으로부터의 규칙 습득에 대한 연구의 역할이 중요할 것으로 기대된다.

한편 웹으로부터 지식을 추출하는 연구는 온톨로지를 활용하여 좀 더 쉽고 효율적인 지식 습득 방법론을 제안하고자 노력하고 있다. 따라서 이러한 온톨로지의 활용이 규칙 습득에도 도움을 줄

수 있을 것으로 기대된다. 시맨틱 웹의 계층 구조를 살펴보면 [32] 규칙은 온톨로지 위에 구축되기 때문에 규칙을 습득하고자 하는 도메인(domain)에 온톨로지가 이미 존재하고 있는 경우, 이러한 온톨로지는 규칙습득에 매우 유용하게 사용될 수 있다. 이에 본 연구에서는 문제를 해결하기 위한 규칙을 웹 공간으로부터 습득하기 위한 방법론을 제시하고자 하며 그 과정에서 온톨로지를 사용하여 보다 효율적인 규칙의 습득이 가능함을 보이고자 한다.

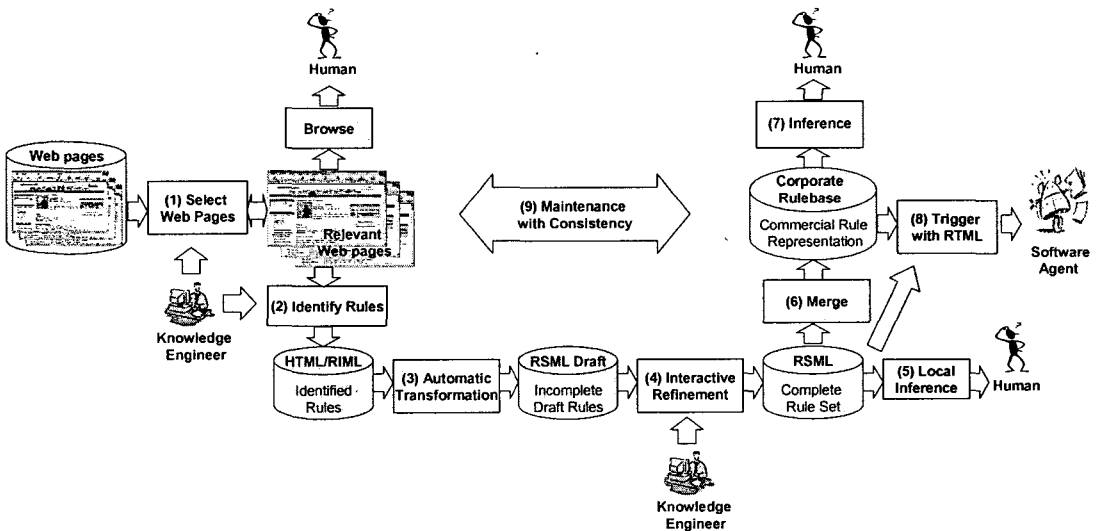
XRML(eXtensible Rule Markup Language) [21]은 웹으로부터 구조화된 규칙을 습득하기 위해 고안된 방법론이다. 그러므로, 본 연구에서는 XRML 구조에 온톨로지를 채택하여 향상된 규칙 습득 방법론을 제안하고자 한다. 이를 위해 기존의 XRML 방법론에 대해 먼저 살펴보도록 하겠다.

1.1 XRML을 활용한 규칙 습득

Kang과 Lee [16]는 자연어와 표를 중심으로 표

현되어 있는 웹 페이지로부터 규칙을 습득하고 이를 관리할 수 있는 XRML 방법론을 제안하였다. XRML은 웹 페이지로부터 규칙과 규칙구성요소를 식별하기 위한 언어인 규칙 식별 표식 언어(Rule Identification Markup Language, RIML)와 추론가능한 규칙을 구조적으로 표현하기 위한 규칙 구조 표식 언어(Rule Structure Markup Language, RSML)로 구성되어 있다.

XRML을 이용하여 지식을 습득하는 과정은 규칙에 대해 설명하고 있는 웹 페이지로부터 규칙 구성요소를 추출하여 RIML 문서를 작성하는 것으로 시작된다. 다음 과정에서 RIML의 형태로 식별된 규칙구성요소들은 RSML에서 정의하고 있는 추론 가능한 규칙 표현 양식으로 자동 변환된다. 그러나 자동 생성된 규칙은 초안으로서 완전한 규칙은 아니다. 이러한 초안을 보완하여 완성된 RSML 문서는 일반적인 규칙기반시스템에서 사용되는 규칙으로 자동변환이 가능하다. [그림 1]은 이와 같은 규칙 습득 과정을 보여주고 있다.



[그림 1] XRML 접근 방법에 의한 규칙습득의 절차

RIML을 이용한 규칙식별을 통해 얻을 수 있는 혜택은 규칙식별의 결과로부터 규칙에 가까운 규칙 초안을 자동으로 생성할 수 있다는 점이다. 자동 생성된 규칙 초안은 규칙을 수정 및 보완하여 완전한 규칙으로 변환하여야 하는 지식관리자의 부담을 크게 줄일 수 있다. 또한 RIML과 RSMIL 간의 연결성을 이용하여 웹 페이지에 변경이 발생했을 때 이를 쉽게 규칙베이스에 반영할 수 있도록 지식관리자에게 적절한 정보를 제공할 수 있다.

이러한 규칙 습득과정은 [그림 1]에서 두번째 단계에 해당되는 규칙구성요소의 식별이 어떻게 이루어지느냐에 의해 좌우된다. 지식 관리자는 가능한 많은 규칙 구성요소들을 식별해 낸다. 일단 규칙식별이 완료되면 이후는 대부분 자동화되어 있기 때문에 어렵지 않게 추론가능한 규칙을 얻을 수 있다. 실제로 Kang과 Lee [16]의 논문을 살펴 보면 Amazon.com, BarnesandNoble.com and Powells.com을 대상으로 실험한 결과 97.7%의 규칙이 규칙 식별단계에서 식별되어 자동으로 규칙으로 변환되었음을 알 수 있다. 따라서 전체과정에서 규칙 식별단계가 매우 중요하고 결정적인 역할을 하고 있음을 알 수 있다.

그러나 다른 단계들이 비교적 자동화된 것에 비해 지식 식별단계는 지식관리자의 수작업에 많이 의존하고 있다. 따라서 지식 식별단계에서 이러한 지식관리자의 수작업을 줄여줄 수 있는 방법론이 요구된다. 본 연구는 온톨로지를 사용하여 지식의 습득단계를 최대한 자동화하고 지식관리자가 보다 쉽게 규칙구성요소를 식별할 수 있도록 돕고자 한다. 그러나, Hulth[15]가 지적한 바와 같이 자연어로 표현된 문장을 구조화된 규칙으로 바로 전환하는 것은 복잡하기 때문에 규칙 습득 과정에서 지식관리자의 개입이 불가피하므로 지식관리자의 역할을 완전히 제거할 수는 없다.

1.2 연구의 목적

유사한 도메인에 속하는 웹 사이트들로부터 지식관리자가 반복해서 규칙을 습득하는 경우, 규칙을 습득하는 과정에서 이전 사이트에서 했던 경험을 이용하는 것이 가능하다. 예를 들어, Amazon.com으로부터 배송관련 규칙을 습득한 경험이 있는 지식관리자는 Powells.com에서 같은 도메인의 규칙을 습득할 때, 자신의 경험에 의해 보다 빠르고 정확하게 규칙을 습득할 수 있다. 이러한 지식 관리자의 경험이 잘 설계된 온톨로지로 표현된다면, 경험이 없는 지식관리자도 보다 이 온톨로지를 활용하여 쉽게 규칙을 습득하는 것이 가능하다. 뿐만 아니라 이러한 온톨로지를 이용한 체계적인 규칙 습득 방법이 제안되면 지식관리자가 규칙 식별 단계에서 해야 할 많은 일들을 자동화하는 것이 가능해진다. 이를 위해 온톨로지는 같은 도메인의 다른 사이트로부터 습득된 규칙 구성요소와 구조에 대한 정보를 포함하여야 할 것이다. 그리고 이러한 온톨로지는 규칙 습득이 반복됨에 따라 계속 축적되어 점차 보다 나은 효과 및 효율을 보일 수 있을 것이다.

그러므로 본 연구의 첫번째 목표는 규칙 습득 과정에서의 온톨로지의 역할을 보여주고, 온톨로지 설계 및 활용 시 발생하는 몇가지 중요한 현상들을 제시하고 이를 해결하기 위한 대안을 제시하고자 한다 (3장). 두번째 목표는 규칙 습득을 도울 수 있는 온톨로지의 요구조건들을 정리하고 이에 따라 규칙 습득을 위한 온톨로지를 설계하는 것이다 (4장). 세번째 목표는 제안된 설계에 따라 온톨로지가 구축되면, 이 온톨로지를 이용하여 XRMIL 기반의 향상된 규칙 식별과정을 제안하고자 한다 (5장). 마지막으로는 실험을 통해 간단하게 본 연구의 성과를 측정하고(6장), 결론을 맺는다.

본 연구에서는 온톨로지를 활용한 규칙 습득 과정을 예를 통해 설명하기 위해 예제 사이트로 실제 온라인 서점인 Amazon.com, Barnes&Noble.com, and Powells.com의 배송 규칙과 환불 정책이 게시되어 있는 웹페이지들을 선정하였다. 습득된 규칙은 배송과 교환 및 환불에 대한 전문가 시스템을 구축하기 위해 이용되는데, 이렇게 구축된 시스템은 물품의 가격뿐만 아니라 배송비까지도 함께 비교해주는 가격비교사이트에 통합될 수 있다.

2. 관련 연구

본 장에서는 온톨로지, 규칙 표식 언어, 온톨로지에 기반한 지식 추출에 대한 연구를 조사하고 본 연구의 접근방법과 비교하여 설명하겠다.

2.1 온톨로지 및 규칙 표식 언어

온톨로지란 보통 도메인 지식의 개념화를 위한 명세서로 정의될 수 있다 [34]. 그러므로 응용프로그램은 온톨로지를 통해 웹 상의 다양한 데이터들을 이해할 수 있게 된다 [6]. 이를 위해 많은 연구들이 RDF(S) [4, 24], DAML-OIL [7, 13], OWL [30]과 같은 온톨로지 표준 언어를 개발하는 데 주력해 왔다. 또한 온톨로지를 구축하고, 편집하고, 읽고, 검색할 수 있는 많은 도구들이 개발되고 있다 [5, 11, 18, 22]. 한편 시맨틱 웹과 관련하여 웹으로부터 자동으로 온톨로지를 추출하고자 하는 많은 연구들이 진행되고 있다 [12, 20, 23]. 이러한 연구들에서 사용되는 자동화 또는 반자동화된 온톨로지 추출 시스템은 대부분 미리 정의된 템플릿이나 문법적 패턴에 기반한 추출 방법 또는 텍스트 문서에서 개체를 식별할 수 있는 기계 학습

(machine learning) 방법론에 의존하고 있다 [2, 8, 28, 35].

이렇게 습득된 온톨로지 형태의 지식은 검색의 용도로 주로 사용되지만 점차 추론을 통해 문제를 해결하기 위한 기반으로 이용하고자 하는 연구가 많아지고 있다. 그 대표적인 연구가 Description logic에 기반한 온톨로지를 활용한 추론 [9, 10, 13, 19]에 대한 것이다. 그러나 Description logic이 갖고 있는 한계점으로 인해 점차 온톨로지보다 상위의 규칙에 대한 필요성이 인식되고 있다 [33]. 주어진 문제를 해결하기 위해 Agent를 동작시키기 위한 규칙베이스가 그 예이며, 이러한 필요성에 의해 온톨로지보다 상위의 규칙을 표현하기 위해 규칙 표식 언어 [29]와 관련된 연구가 진행 중에 있다. 규칙표식언어의 주 목적은 추론이 가능한 논리적 규칙을 웹 상에서 원활히 교환할 수 있도록 XML을 이용하여 표현하는데 있다. 이러한 노력의 결과로 SWRL [14]이 현재 표준으로 제안되고 있는 중이다.

그러나 규칙 표식 언어와 관련된 연구는 규칙 습득보다는 규칙 표현에만 그 초점을 두고 있다. 웹 상의 방대한 지식을 활용하기 위해서는 규칙 습득과 관련된 연구가 필히 요구되므로, 본 연구에서는 웹으로부터 규칙을 습득하는 방법론을 개발하고자 한다.

2.2 온톨로지를 활용한 지식 추출

웹으로부터 온톨로지를 구축하는 연구 외에도, 온톨로지를 활용하여 웹으로부터 지식을 습득하는 것과 관련된 많은 연구들이 진행되고 있다. 이들 연구의 일례로 문서에 부가적인 메타 정보를 표시하는 데 온톨로지를 이용하고 있다 [17, 26, 27].

이렇게 함으로써 소프트웨어는 스스로 문서를 분류하고 조직화하며 필요한 정보를 검색할 수 있게 된다. 그러나 이 접근 방법은 온톨로지를 활용하여 문서로부터 일반적인 지식을 추출하는 것 보다는 정보의 분류에 초점을 맞추고 있다.

다른 접근방법으로는 데이터의 구조적인 집합이 온톨로지와 같은 일정한 틀로 표현되어 있을 때, 이러한 온톨로지 틀을 이용하여 문서로부터 자동으로 일반화되고 추상화된 지식들을 추출하는 연구들이 있다 [1, 35]. 이 과정에서 문서를 파싱할 때 WordNet [25]이나 GATE [3]와 같은 외부 온톨로지가 이용되기도 한다.

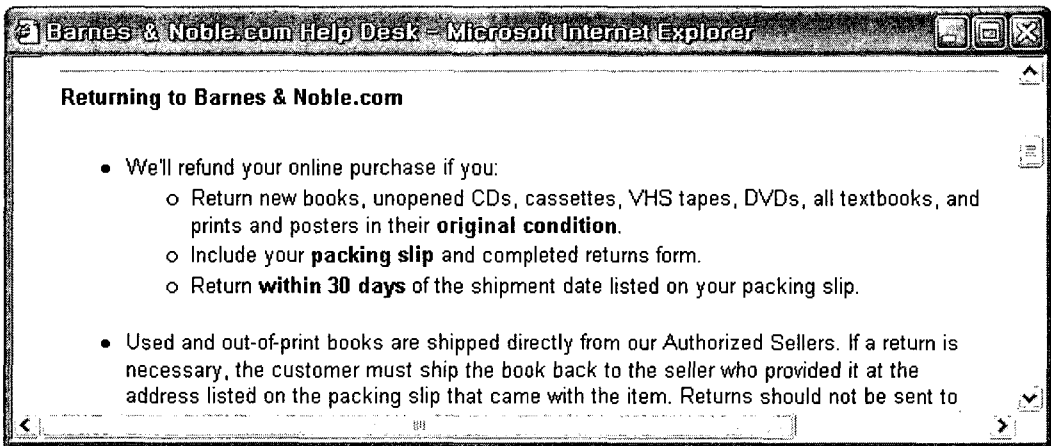
이들 연구와 본 연구의 차이점은 본 연구에서는 웹으로부터 개념위주의 지식보다는 추론 가능한 규칙을 추출하고자 한다는 점에 있다. 그 외에도, 규칙 습득에 필요한 새로운 온톨로지를 설계하고 이를 활용하는 방법론을 제안하고자 한다. 이렇게 설계된 온톨로지는 각 규칙 습득 때마다 축적되고, 공유되어 재사용될 수 있게 된다.

3. 온톨로지에 기반한 규칙 식별 접근 방법

본 장에서는 규칙 식별 시 온톨로지의 역할 및 온톨로지를 활용한 규칙 식별과 관련된 기본 접근 방법을 설명하고자 한다. 또한, 규칙 식별에서 온톨로지를 활용할 때 발생하는 다양한 이슈들을 다루도록 하겠다.

3.1 규칙 식별에서 온톨로지의 역할

1장에서 설명한 바와 같이, 규칙 식별이란 지식 관리자가 규칙을 이루고 있는 기본 구성요소들을 웹 페이지로부터 식별하는 과정이다. 이를 위해 RIML 버전 1.0 [21]에서는 *RuleGroup*, *Rule*, *variable*, *value*와 같은 기본적인 요소를 정의하였다. 예를 들어, Barnes&Noble.com으로부터 [그림 2]와 같이 책의 환불 정책에 대해 설명하고 있는 웹 페이지를 골랐다고 가정해 보자. 만일 지식관리자가 *books*와 *CDs*를 변수값(*value*)으로 인식한다면, `<value>books</value>`과 `<value>CDs`



[그림 2] Barnes&Noble.com의 환불정책을 기술하고 있는 웹 페이지

</value>와 같은 태그를 사용하여 규칙 구성요소가 표현될 수 있다. RIML 버전 2.0 [16]에서는 규칙 구성요소의 표현력을 증가시키기 위해 RuleTable, IF, THEN, AND, OR, NOT과 기본적인 연산자를 추가하였다.

Amazon.com, Barnes&Noble.com, Powells.com으로부터 규칙 습득을 수행한 실험 결과를 살펴보면, 규칙 식별단계에서 97.7%의 규칙과 88.5%의 규칙 구성요소가 식별될 수 있었다 [16]. XXML 방법론에서는 많은 규칙이 식별될수록 XXML 편집기는 많은 구조화된 규칙을 자동으로 생성해낼 수 있으므로, 이러한 실험결과는 규칙 식별이 모든 규칙 습득 단계에서 매우 중요하고 결정적인 역할을 하고 있음을 보여주고 있다. 그러나 규칙 식별은 여전히 지식관리자의 수작업에 많이 의존하고 있다.

규칙 식별과 관련된 수작업으로는 웹 페이지로부터 규칙을 구성하고 있는 변수(variable) 나 변수값(value)을 찾고 변수와 값들 간의 연결관계를 식별하여 서로 쌍을 이루게 한 후 규칙을 구성해 나가도록 하는 것들이 있다. 지식관리자는 이러한 작업 과정에서 관련 도메인과 규칙에 관한 사전 지식이나 경험이 요구된다. 이러한 사전 지식과 경험은 잘 정의된 온톨로지에 의해 대체될 수 있다. 이렇게 구축된 온톨로지를 활용하게 되면, 지식 관리자의 수작업을 줄여줄 수 있게 되므로 지식 식별 과정이 자동화될 수 있게 된다.

3.2 규칙 식별에서 온톨로지의 활용

그럼 과연 어떻게 규칙 식별에서 온톨로지가 이용될 수 있을 것인가? 이를 예를 들어 설명해보도록 하겠다. 먼저 [그림 3]과 같이 Amazon.com으로부터 생성된 ontology를 이용하여 [그림 2]와 같

은 Barnes&Noble.com의 웹 페이지로부터 지식을 습득한다고 가정해 보자. [그림 3]의 온톨로지로부터 [그림 2]의 *refund and Return within 30 days*를 변수로 식별될 수 있고, *books, CDs, VHS tapes*는 값으로 식별될 수 있다. 왜냐하면, 온톨로지에 이들 단어가 변수와 변수값으로 정의되어 있기 때문이다.

온톨로지는 규칙 습득과 관련하여 위의 경우 외에도 더 많은 정보를 제공할 수 있는데, 웹 페이지에 생략되어 있는 구성요소를 식별할 때에도 도움을 줄 수 있다. 예를 들어, [그림 3]에서 *books, CDs, VHS tapes*가 변수 *item*의 변수값으로 정의되어 있으므로, [그림 2]에서 변수 *item*이 생략되어 있음을 알 수 있게 된다. 이 밖에도 온톨로지는 규칙의 *IF* 혹은 *THEN* 부분을 식별할 때에도 도움을 줄 수 있다. 이와 관련한 상세한 규칙 식별 과정은 5장에서 설명하도록 하겠다.

```

{{ "Refund Policy Rule5"
  IS-A: Rule
  subRuleOf: "Return Policy"
  IF: (variables "Return within 30 days" Item "Returned status")
      (values BOOK "Obvious signs of use" CD DVD
        "VHS Tape" Opened "Original condition")
  THEN: (variables Refund)
        (values Partial) }}
{{ "Return within 30 days"
  IS-A: Variable
  Values: Yes No }}
{{ "Returned status"
  IS-A: Variable
  Values: Opened "Obvious sign of use" "Original condition" }}
{{ Item
  IS-A: Variable
  Values: Book CD DVD "VHS tape" }}
{{ Refund
  IS-A: Variable
  Values: Full Partial }}
    
```

[그림 3] Amazon.com으로부터 습득된 온톨로지

[그림 3]의 온톨로지는 다음과 같은 Amazon.com의 규칙을 통해 생성될 수 있는데, 다음의 규칙으로부터 온톨로지를 추출하는 것은 서로 구조화되어 있는 상태이므로 바로 이루어질 수 있게 된다.

```
IF "Return within 30 days" = No
  OR (Item = BOOK AND "Returned status" =
    "Obvious signs of use")
  OR ((Item = CD OR Item = DVD OR Item = "VHS
    Tape")
    AND "Returned status" = Opened)
  OR "Returned status" != "Original condition"
THEN Refund = Partial
```

3.3 규칙 식별에서 온톨로지 활용과 관련된 이슈

앞 절까지는 규칙식별에서 온톨로지를 활용하는 방안에 대해 기본적인개념을 설명하였다. 이 절에서는 규칙식별과정에서 온톨로지를 활용하는 것과 관련된 논점들을 정리하고자 한다.

3.3.1 전용 온톨로지와 외부 온톨로지

첫째 논점은 규칙식별에서 사용할 온톨로지의 종류이다. 우선 WordNet [25]이나 GATE [3]와 같은 외부 온톨로지를 사용하는 것을 고려할 수 있다. 이 경우, 변수나 변수값으로 사용되는 단어들을 검색하거나 동의어를 검색할 때 사용될 수 있다. 그러나, 온톨로지의 사용효과를 극대화하기 위해서는 단순한 단어나 동의어 정보외에 규칙의 구조와 규칙구성요소간의 관계와 같은 정보가 필요하다. 외부 온톨로지만으로는 검색된 단어가 변수로 사용될지 변수값으로 사용될지조차 결정하기 쉽지 않다. 따라서 본 논문에서는 규칙구성요소와 규칙의 구조를 표현하는 새로운 온톨로지를 OntoRule이라는 이름으로 설계하고자 한다.

3.3.2 반복된 규칙습득에의한 온톨로지 학습효과

동일한 분야의 여러 사이트에서 규칙습득이 반복되는 경우에 온톨로지의 내용이 계속축적될 수 있다. 이와 관련하여 규칙습득에 온톨로지를 사용하는 다양한 시나리오를 생각해볼 수 있다. 가장 간단한 시나리오는 한 사이트에서 온톨로지를 습득하고 그 사이트에서만 사용하는 경우이다. 이 경우 온톨로지는 재사용되지 않는다. 둘째는 한 사이트에서 온톨로지를 습득하고 이 온톨로지를 다른 사이트에서 반복해 사용하는 시나리오이다. 그러나 사이트마다 어휘나 규칙구조에 차이가 있기 때문에 온톨로지의 효과가 높지 않을 수 있다. 가장 이상적인 시나리오는 다른 사이트에서 온톨로지를 활용하는 동시에 온톨로지를 계속 수정하는 것이다. 이 경우, 온톨로지 학습의 효과로 규칙식별의 정확도가 규칙습득의 반복에 따라 증가할 것으로 기대된다.

3.3.3 상향접근방식과 하향접근방식의 비교

규칙식별의 하향식 접근방식은 규칙그룹, 규칙과 같은 큰 규모의 구성요소로부터 시작하여 점차 작은 규모의 구성요소로 식별해가는 방식이다. 지식관리자가 온톨로지 없이 웹 페이지의 전체 내용을 파악하면서 규칙을 습득할 때 유용할 수 있으나, 온톨로지를 이용한 자동식별에서는 처음부터 규칙을 식별하기에는 정보가 부족하기 때문에 적합하지 않다. 상향식 접근방법은 변수나 변수값들로부터 식별을 시작하여 나중에 이를 조합하여 규칙을 식별하는 방식이다. 온톨로지로부터 변수와 변수값을 먼저 식별하는 것이 비교적 쉽고, 식별된 변수와 변수값 등 축적된 정보를 기반으로 규칙을 식별할 수 있기 때문에 보다 적합한 방식이라 할

수 있다. 따라서 본 연구에서는 상향식 접근방법을 규칙습득 편집기에 적용하고 OntoXRML이라 이름을 붙였다.

3.3.4 규칙식별과정에서 온톨로지 적용범위의 관리

규칙식별의 성능은 식별과정에서 식별하지 못한 규칙구성요소의 수와 잘못 식별된 규칙구성요소의 수로 측정할 수 있다. 만일 온톨로지의 양이 충분하다면 식별하지 못한 규칙구성요소의 수는 줄일 수 있다. 반면, 잘못 식별된 규칙구성요소의 수는 늘어날 수 밖에 없다. 이를 해결하기 위해 규칙식별에 사용하는 온톨로지의 범위를 조절할 필요가 있다. 지식관리자는 현재의 웹 페이지에 적절한 온톨로지의 범위를 선택함으로써 식별하지 못한 규칙의 수를 줄이면서 동시에 잘못 식별된 규칙구성요소도 적절한 범위 안으로 제어할 수 있다.

3.3.5 변수와 변수값의 검색 순서

변수와 변수값 중에서 어느 것을 먼저 검색할 것인가 하는 것도 검토되어야 할 논점이다. 온톨로지에서 변수값은 변수에 연결되어 존재하기 때문에 만일 변수를 먼저 식별하면 다음으로 식별할 변수값의 범위를 급격히 줄이는 것이 가능하다. 또한 일반적으로 변수의 종류가 변수값의 종류에 비해 더 적기 때문에 변수부터 시작하는 것이 복잡도를 줄이는 효과를 가져온다. 셋째로 변수의 유형을 알고 있으면 이를 이용하여 이 변수에 연결된 변수값들을 보다 효율적으로 검색할 수 있다. 예를 들어 변수 *Item*은 *Book*, *CD*, 그리고 *DVD*와 같은 문자열을 변수값으로 갖는다. 따라서 웹 페이지에서 먼저 *Item*를 검색했다면 다음은 이 변수값들을 웹페이지로부터 찾아내면 된다. 변수의 유형 외

에도 규칙식별에 필요한 여러가지 정보들을 온톨로지에 추가할 수 있다.

3.3.6 생략된 변수의 식별

규칙식별에서 생략된 변수를 식별하는 것은 가장 어려운 작업 중 하나이다. [그림 2]에서 웹페이지에 *original condition*에 대응되는 생략된 변수 *Returned Status*를 찾는 것은 매우 어렵다. 그러나 만일 [그림 3]과 같은 온톨로지를 통해 *original condition*이 *Returned Status*에 연결된 변수값이라는 것을 알 수 있으면 쉽게 생략된 변수를 식별할 수 있다. 생략된 변수의 식별은 RSML 초안의 자동생성에서 중요한 역할을 한다. 규칙식별의 효과에 대한 실험에서 [16] 생략된 규칙구성요소는 전체 구성요소의 20%에 해당될 정도로 빈번히 사용된다.

4. 규칙 습득을 위한 온톨로지의 설계

본 장에서는 *OntoRule*의 설계 시 고려해야 할 요구조건을 보이고, 그 요구조건에 따라 *OntoRule*을 설계하고자 한다. 온톨로지로 설계되는 결과물에는 변수, 변수값, 규칙 구조, 규칙 식별에 사용되는 변수와 변수값에 대한 메타 정보들이 포함된다.

4.1 *OntoRule*의 요구조건

3.3절에서 언급했던 온톨로지를 이용할 때 발생할 수 있는 문제점으로부터 온톨로지 설계에 필요한 몇가지 요구조건을 도출해냈다. 요구조건은 다음과 같이 5개로 분류된다.

4.1.1 온톨로지 범위의 제어를 위한 도메인 관련 규칙 정보

지식관리자는 규칙 습득 과정을 시작할 때 목표 웹 페이지를 선택한다. 만일 ontoRule이 도메인과 응용에 대한 정보를 보유할 수 있도록 설계된다면, 지식관리자는 규칙 습득을 위한 적절한 온톨로지의 범위를 선택할 수 있다.

4.1.2 변수와 변수값에 대한 어휘와 연관관계 정보

웹 페이지로부터 적절한 변수와 변수값을 추천하기 위해 빈번하게 변수와 변수값으로 사용되었던 단어들에 대한 정보를 포함할 수 있도록 OntoRule이 설계되어야 한다. 또한 OntoRule은 XRML 에디터인 OntoXRML이 변수와 변수값의 관계를 추천할 수 있도록 변수와 변수값의 관계에 대한 정보도 포함되도록 설계되어야 한다.

4.1.3 도메인 관련 단어에 대한 동의어

동의어를 표현할 수 있는 것이 온톨로지의 또 다른 중요한 기능이다. WordNet [25]과 같은 범용 온톨로지는 주어진 단어에 대해 동의어(synonym)와 등위어(coordinate term)을 제공하지만 규칙 습득을 위해 필요한 도메인에서 사용되는 동의어는 제공하지 않는다. 따라서, OntoRule에 이러한 도메인 전용 동의어를 추가하였다. 참고로 Kang과 Lee [16]가 수행했던 실험을 보면, 전체 식별된 규칙 구성요소 중 동의어가 차지하는 비중이 15%였다.

4.1.4 규칙구조에 대한 정보

규칙, IF, THEN을 식별할 수 있으려면 온톨로지에 변수와 변수값 및 규칙간의 관계를 표현하고

있는 규칙 구조에 대한 정보가 포함되어야 한다. 규칙 구조에 대한 온톨로지는 변수, 변수값, 그리고 IF THEN을 포함하는 규칙에 대한 일반화된 정보로 구성된다.

4.1.5 변수와 변수값에 대한 부가정보

3.3.5절에서 언급한 바와 같이 변수에 대한 유형을 결정하게 되면 적절한 변수값을 식별하는 데 도움을 줄 수 있으므로 OntoRule은 규칙구성요소에 대한 다양한 숫자와 변수의 유형, 변수의 단위 및 범위에 대한 정보를 추가하였다.

4.2 OntoRule의 설계

이 절에서는 4.1절에서 설명한 요구조건에 기반하여 OntoRule이 어떻게 설계되는지에 대해 논하고자 한다. 온톨로지를 표현하기 위해 프레임(frame)을 기법을 도입했는데, 그 이유는 프레임은 이해하기 쉽고 간단하며 프레임으로 표현된 온톨로지는 쉽게 그리고 자동으로 RDF나 OWL로 전환가능하기 때문이다.

4.1에서 기술한 요구조건이 반영되도록 프레임 기반으로 설계한 전체 온톨로지 스키마(schema)는 [그림 4]와 같이 표현된다. 온톨로지 적용 범위 관리를 위해 *Domain* 클래스(class), *Application* 클래스, *RuleGroup* 클래스를 정의하였으며, 규칙의 구조를 표현하기 위해 *Rule* 클래스를 정의하였다. 또한 변수와 변수값을 표현하기 위해 *Variable* 클래스, *Value* 클래스, *Synonym* 클래스를 정의하였다. 상세한 온톨로지 설계에 대해 [그림 5]의 Amazon.com으로부터 습득한 온톨로지 예제와 함께 다음 절에서 설명하도록 하겠다.

4.2.1 온톨로지 범위의 제어를 위한 도메인 관련 규칙 정보

온톨로지 적용 범위를 관리한다는 것은 규칙 습득에서 이용될 온톨로지의 범위를 좁혀나가는 것을 의미한다. [그림 4]에서와 같이 *Domain*, *Application*, *RuleGroup*의 순서로 온톨로지 적용 범위를 줄여나가기 위해 *Domain* 클래스, *Application* 클래스, *RuleGroup* 클래스를 정의하였다. 세 클래스는 서로 연결되어 있으며 슬롯 *source*는 애플리케이션이 구축된 웹 사이트를 의미한다. *RuleGroup*, *Rule*, *Variable*, *Value*는 상위의 서로 연결되어있는 클래스로부터 *source* 슬롯 값을 상속받는다. 만일 한 클래스가 두개 이상의 웹 사이트에서 이용되었을 경우 *source* 슬롯은 여러 개의 값을 가질 수도 있다.

<pre> {{ Domain IS-A: Class Applications: }} {{ Application IS-A: Class source: RuleGroups: }} {{ RuleGroup IS-A: Class source: Rules: }} {{ Rule IS-A: Class source: IF: (variables) (values) THEN: (variables) (values) }} {{ Term IS-A: Class }} </pre>	<pre> {{ Variable IS-A: Term source: variableType: values: synonyms: unit: range: (MAX) (MIN) omittedCount: }} {{ Value IS-A: Term source: synonyms: }} {{ Synonym IS-A: Term source: }} </pre>
--	---

[그림 4] OntoRule 스키마

[그림 5]에서 *Delivery*는 가장 상위의 분류에

속하는 도메인 이름이 되고, “*Shipping Rates*”는 *Delivery* 도메인에 속하는 애플리케이션들 중 하나가 된다. 또한, “*Shipping Rates Domestic*”은 “*Shipping Rates*” 애플리케이션에 속하는 규칙 그룹 중 하나가 된다.

4.2.2 변수와 변수값의 표현

[그림 4]의 온톨로지와 같이 변수와 변수값을 표현하기 위해 *Variable* 클래스와 *Value* 클래스를 정의하였다. 그리고 변수와 변수값을 연결하기 위해 *Variable* 클래스에 *values* 관계를 정의해 두었다. [그림 5]의 예제를 보면, *Shipping Method*는 변수가 되고, *values* 슬롯에 *Standard Shipping*을 값으로 가진다. 만일 *Shipping Method*가 *Two-Day Shipping*이라는 또 다른 변수값을 가지게 되면 이를 *values* 슬롯에 추가하고 새로운 *Value* 인스턴스(instance)인 *Two-Day Shipping*을 작성한다.

4.2.3 규칙 구조의 표현

규칙과 규칙 그룹도 변수와 변수값과 마찬가지로 규칙 구성요소가 되므로 이들을 표현하기 위해 *Rule* 클래스와 *RuleGroup* 클래스를 정의하였다. 그리고, *Rule*과 *RuleGroup*을 서로 연결하기 위해 *RuleGroup* 클래스에 *rules*라는 관계를 추가하였다. [그림 5]에서 규칙 그룹 *Shipping Rates Domestic*은 *Standard Shipping Rule*를 규칙으로 가지고 있다.

*IF*와 *THEN*은 규칙과 변수와의 관계를 표현한다. 따라서 *Rule* 클래스의 슬롯으로 *IF*와 *THEN*을 정의하였고, *IF*와 *THEN* 슬롯의 패시(facet)으로 *variables*와 *values*를 정의하였다. 예를 들어 [그림 5]를 보면, *Standard Shipping Rule* 규칙은

```

{{ Delivery
  IS-A: Domain
  Applications: "Shipping Rates" }}
{{ "Shipping Rates"
  IS-A: Application
  source: Amazon.com
  RuleGroups: "Shipping Rates Domestic" }}
{{ "Shipping Rates Domestic"
  IS-A: RuleGroup
  source: Amazon.com
  Rules: "Standard Shipping Rule" }}
{{ "Standard Shipping Rule"
  IS-A: Rule
  source: Amazon.com
  IF: (variables "Shipping Region"
        "Shipping Method")
      (values Domestic "Standard Shipping")
  THEN: (variables "Delivery Time"
            "Per Shipment" "Per Item")
        (values "3 to 5 Business Days" ) }}
{{ "Shipping Region"
  IS-A: Variable
  source: Amazon.com
  values: Domestic Europe Africa "Latin America"
  variableType: OAV
  omittedCount: 73 }}
{{ "Shipping Method"
  IS-A: Variable
  source: Amazon.com
  values: "Standard Shipping" "Two-Day Shipping"
  variableType: OAV }}
{{ "Delivery Time"
  IS-A: Variable
  source: Amazon.com
  values: "3 to 5 Business Days"
  variableType: OAV
  omittedCount: 80 }}
{{ Per Shipment
  IS-A: Variable
  source: Amazon.com
  variableType: Numeric
  unit: $
  range: (max 1.99) (min 31.99) }}
{{ "Per Item"
  IS-A: Variable
  source: Amazon.com
  variableType: Numeric
  unit: $
  range: (max 0.99) (min 9.99) }}
{{ Domestic
  IS-A: value
  source: Amazon.com }}
{{ "Standard Shipping"
  IS-A: value
  source: Amazon.com }}
{{ "3 to 5 Business Days"
  IS-A: value
  source: Amazon.com }}

```

[그림 5] OntoRule 예제

*Shipping Region*과 *Shipping Method*를 *variables* 패킷값으로 가지고 있고, *Domestic*과 *"Standard Shipping"*을 *values* 패킷값으로 가지고 있다.

간단한 프레임 표현안에서 복잡AND와 OR와 같은 접속어(connective)는 복잡한 중첩 구조를 가지고 있고 온톨로지에서 이들 접속어를 모두 표현한다고 해도 일반화하는 효과가 거의 없기 때문에 OntoRule에 포함시키지 않았다.

4.2.4 변수와 변수값 특성에 대한 온톨로지

지금까지 규칙으로부터 추가적인 분석 과정이 추출해낼 수 있는 기본 정보에 대해 기술하였으나, 좀 더 향상된 식별과정을 제공하기 위하여 변수와 변수값에 몇가지 추가적인 특성을 추가하고자 한다.

변수의 특성으로는, [그림 4]에서와 같이 *Variable* 클래스의 슬롯으로 *variableType*, *unit*, *range*, *omittedCount*를 정의하였다. *variableType* 슬롯은 *OAV*, *Numeric*, *Fact* 세 개 중 하나의 값을 가진다. *OAV*(Object Attribute Value) 유형 변수는 미리 결정된 문자열을 변수값으로 가질 수 있다. *Fact* 유형의 변수는 *OAV*의 특별한 경우로 *Yes* 혹은 *No*의 변수값을 가질 수 있다. *Numeric* 유형 변수는 *integer* 또는 *float*와 같은 모든 유형의 숫자를 포함한 변수값을 가질 수 있다. 여기서 *Numeric* 유형의 변수만이 *unit*과 *range*를 속성으로 가질 수 있다. 이를 위해 *Numeric* 유형의 변수는 *unit*과 *range* 슬롯을 통해 가질 수 있는 변수값의 패턴을 표현할 수 있다. 예를 들어 [그림 5]에서 변수 *Per Shipment*는 *Numeric* 유형의 변수로서 *Per Shipment*의 변수값은 \$3.00이다. 이 때 \$3.00은 \$라는 *unit*과 3.00이라는 숫자로 구성된다. 슬

롯 *range*는 *MIN*과 *MAX* 패킷으로 구성된다. [그림 5]에서 *Per Shipment* 변수의 *range* 슬롯은 *MIN*의 패킷값으로 1.99를 갖고, *MAX*의 패킷값으로 31.99를 갖게 된다. 슬롯 *omittedCount*는 그 애플리케이션에서 생략된 변수의 개수를 의미하며 생략된 변수를 식별할 때 사용된다.

4.2.5 동의어를 위한 온톨로지

OntoRule이 동의어를 표현가능하도록 하기 위해 *Synonym* 클래스를 정의하였다. *Synonym* 인스턴스는 *synonyms* 슬롯을 통해 *Variable*과 *Value*의 인스턴스와 연결된다. 다음의 예는 변수값인 *International Surface*가 동의어로 *Standard Surface Mail*을 가지고 있는 걸 표현하고 있다.

```

{{ "International Surface"
  IS-A: Value
  synonyms: "Standard Surface Mail" }}
{{ "Standard Surface Mail"
  IS-A: Synonym
  source: Amazon.com }}
    
```

4.3 규칙으로부터의 온톨로지 생성

온톨로지 생성과 관련하여 추가, 충돌(Conflict), 일반화와 같은 다양한 현상들이 발생할 수 있다. 그러나, OntoRule의 설계와 사용에 초점을 맞추기 위해 본 논문에서는 그러한 현상들과 상세한 온톨로지 생성 과정은 다루지 않는다.

OntoRule의 대부분은 RIML과 RSML로부터 자동으로 생성될 수 있다. <표 1>은 온톨로지의 출처와 생성된 온톨로지의 구성요소를 보여주고 있다. OntoRule에서의 정보는 RIML과 RSML 파일의 내용과 유사하나, OntoRule이 좀 더 높은 추상화된 정보를 가지게 된다. 온톨로지는 <표 1>에 있는 세가지의 출처로부터 생성된 온톨로지 구성

요소를 결합하여 생성된다.

<표 1> OntoRule 생성 출처

출처	생성된 온톨로지 구성요소
RIML	Variable: omittedCount Synonym: Name
RSML	RuleGroup: Name, RuleGroup-rule relation Rule: Name, IF (variables, values), THEN (variables, values) Variable: Name, variableType, unit, range (MAX, MIN), Variable-value relation Value: Name
The Knowledge Engineer	Domain: Name, Domain-Application relation Application: Name, Application-RuleGroup relation

5. 온톨로지 기반 규칙 식별 과정

이 장에서는 OntoRule을 이용하여 자동으로 규칙구성요소를 식별하는 과정을 제안하고자 한다. 식별과정은 3장에서 기술된 논점들을 기반으로 구성되었다. 첫째, 적절한 온톨로지의 범위를 선택하고, 둘째, 상향식 접근방식을 선택하였으며, 셋째 식별은 변수에서 변수값의 순서로 진행하였다. 마지막으로 생략된 변수를 식별할 수 있도록 하였다.

식별과정을 표현하기 위하여 다음과 같이 식별과정에서 사용되는 표기법을 정리하였다.

r: 규칙 그룹의 색인.

T: 웹 페이지의 텍스트 전체. $T = \{T_1, \dots, T_r, \dots, T_n\}$. *T_r* 은 규칙 그룹, *G_r*을 내포하고 있는 텍스트이다. 예를 들어 *r*='Return Policy' 이면 *G_r* 은 반환정책에 대한 규칙그룹이다.

$O = \{O_1, \dots, O_r, \dots, O_n\}$: 규칙그룹들에 대한 온톨로지의 집합이다. O_r 은 규칙그룹에 G_r 에 대한 온톨로지이다. G_r 은 T_r 에 대응된다.

5.1 T_r 선택

지식관리자가 웹 페이지로부터 규칙이 있을 것으로 예상되는 T_r 을 선택한다.

5.2 O_r 지정

지식관리자가 규칙그룹 리스트, O 로부터 선택된 T_r 에 대응될 것으로 예상되는 O_r 을 선택한다.

5.3 O_r 을 이용하여 T_r 로부터 변수 식별

변수식별과정을 표현하기 위하여 우선 다음과 같이 변수 식별에 사용되는 표기법을 정리하였다.

i : 변수후보의 색인.

j : 선택된 변수에 대한 변수값후보집합의 색인.

$V = \{V_1, V_2, \dots, V_i, \dots, V_n\}$: T_r 에 대한 변수후보의 집합. 한 변수후보, V_i 에 대한 데이터구조는 다음과 같이 정의된다. $V_i = \{variableType, values, synonyms, unit, rangeMIN, rangeMAX\}$: V_i 의 값들은 O_r 로부터 추출된다.

$Type(V_i)$: V_i 의 $variableType$ 을 반환한다.

$Values(V_i)$: O_r 에서 V_i 의 변수값들을 반환한다.

$Unit(V_i)$: V_i 의 $unit$ 을 반환한다.

$MIN(V_i)$: V_i 의 $rangeMIN$ 을 반환한다.

$MAX(V_i)$: V_i 의 $rangeMAX$ 을 반환한다.

$U = \{U_1, U_2, \dots, U_i, \dots, U_n\}$: T_r 에서 검색된 변수값후보들의 집합. V_i 에 연결된 변수값후보집합은 다음과 같이 정의된다. $U_i = \{u_{i1}, u_{i2}, \dots, u_{ij}, \dots, u_{im}\}$. u_{ij} 는 하나의 변수값후보를 나타낸다.

• $Variable(O_r) \cap Word(T_r) \Rightarrow V$

위 식에서 $Word(T_r)$ 은 T_r 에서 파싱된 단어들이다. 이 단어들을 $Variable(O_r)$ 로 표시된 O_r 의 변수들과 비교함으로써 변수후보집합인 V 를 추출한다. 추출된 변수후보집합들에 대하여 적절한 기준 혹은 평가함수를 이용해 다시 실제로 식별하여야 할 변수들을 선택할 수 있다. 그러나, 본 연구에서는 이러한 기준을 향후 연구로 남겨두고 모든 변수후보들을 변수로 식별하였다.

5.4 T_r 로부터, 식별된 변수들에 연결되는 변수값들을 식별

• For each V_i ,

- If $Type(V_i) = 'OAV'$

then $Values(V_i) \cap Word(T_r) \Rightarrow U_i$

T_r 에서 파싱된 단어들을 $Values(V_i)$ 로 표시되는 OAV 타입인 V_i 의 변수값들과 비교함으로써 변수값후보집합을 추출한다. 추출된 변수값 후보집합은 $U_i = \{u_{i1}, u_{i2}, \dots, u_{ij}, \dots, u_{im}\}$ 로 표기된다. u_{ij} 는 OAV 타입 변수값이며 문자열이다. 텍스트로부터 추출되는 변수값들은 같은 값이 중복될 수 있다.

- Elseif $Type(V_i) = 'Fact'$

then $extendedWords({'Yes', 'No'}) \cap Word(T_r) \Rightarrow U_i$

진위형 변수의 값은 'Yes'와 'No'만 존재한다. 따라서 동의어등을 이용해 'Yes'와 'No'를 확장한 단어들과 T_r 에서 파싱된 단어들을 비교하여 진위형 변수의 변수값후보들을

추출하였다.

- Elseif $Type(V_i) = 'Numeric'$
 then $NumericValues(V_i) \cap Word(T_r)$
 $\Rightarrow U_i$

$NumericValues(V_i)$ 는 V_i 가 가질 수 있는 모든 값들을 반환한다. 이 변수값들은 다음과 같이 $Unit(V_i)$, $MIN(V_i)$, $MAX(V_i)$ 를 조합하여 구할 수 있다. For all $v \in NumericValues(V_i)$, 여기서 v 는 다음 조건을 만족하여야 한다: $v=concatenation(Unit(V_i), num)$ or $v=concatenation(num, Unit(V_i))$, where $MIN(V_i) \leq num \leq MAX(V_i)$ and num is an any number. 이 값들을 T_r 에서 파싱된 단어들을 비교함으로써 수치형 변수에 대한 변수값후보들을 추출할 수 있다.

이상에서 추출된 모든 변수값후보들을 변수값으로 식별한다.

5.5 생략된 변수의 식별

만약 온톨로지의 도움이 없다면 생략된 변수를 찾는 일은 매우 어려운 일이다. OntoXRML은 연결변수값을 이용하여 생략된 변수를 찾아낼 수 있다. 식별과정은 온톨로지로부터 과거에 생략된 적이 있는 변수를 찾는 과정과 이 변수와 연결된 변수값에 대해 웹 페이지를 검색하는 두 과정으로 이루어져 있다.

5.5.1 O_r로부터 생략된 변수 검색

- $OV = \{OV_1, OV_2, \dots, OV_i, \dots,$

$OV_n | OmittedCount(OV_i) > 0$ and
 $VariableType(OV_i) = 'OAV'$

O_r로부터 omittedCount가 0이상인 변수들을 검색한다.

5.5.2 Search for values linked to OV from Tr.

- For each OV_i ,
 $Values(OV_i) \cap Word(T_r) \Rightarrow OU_i$

검색된 OV_i 에 대하여, 온톨로지에서 OV_i 와 연결된 변수값들과 T_r 에서 파싱된 단어들을 비교함으로써 생략된 변수의 변수값후보들을 추출한다. OV 에서 변수값후보가 T_r 에 존재하는 변수들은 모두 생략된 변수후보로 식별된다.

5.6 온톨로지로부터 유사한 규칙 검색

규칙을 구성하는 기본요소인 변수와 변수값들이 식별되면 이와 같은 구성요소들을 결합하여 규칙을 만든다. 여기서 온톨로지는 주어진 변수와 변수값들로 구성할 수 있는 가장 유사한 규칙을 제공하는 역할을 한다. 이 과정을 표현하기 위하여 다음과 같은 표기법을 정리하였다.

I : 규칙그룹 G_r 내에 있는 규칙들의 색인

$G_r = \{R_{r1}, R_{r2}, \dots, R_{ri}, \dots, R_{rk}\}$: 온톨로지로부터 선택된 규칙그룹. R_{ri} 은 G_r 내의 1번째 규칙 온톨로지이다.

O_i : R_{ri} 에 대한 온톨로지

T_r : 하나의 규칙을 내포하고 있는 텍스트 범위 $T_r \in T$. T_r 은 지식관리자에 의해 지정된다.

5.6.1 T_R 선택

지식관리자가 O_I 에서 유사한 규칙을 찾아내기 위해 웹 페이지로부터 하나의 규칙을 내포하고 있을 것으로 생각되는 T_R 을 선택한다.

5.6.2 O_I 에 있는 규칙들에 대해 T_R 과의 유사도를 계산

- For each O_I in O_T ,

다음 값들을 계산한다:

$MVU(O_I|T_R)$: T_R 에서 식별된 변수 및 변수값과 매칭되는 O_I 의 변수와 변수값의 개수.

$MVU(T_R|O_I)$: O_I 의 변수 및 변수값과 매칭되는 T_R 의 식별된 변수와 변수값의 개수.

$VU(O_I)$: O_I 의 변수와 변수값의 개수

$VU(T_R)$: T_R 에서 식별된 변수와 변수값의 개수

$$Sim(O_I, T_R) = (MVU(O_I|T_R)/VU(O_I)) * MVU(T_R|O_I)/VU(T_R).$$

OntoXML은 $Sim(O_I, T_R)$ 가 가장 큰 O_I 을 T_R 에 대해 식별된 규칙으로 추천한다.

$MVU(O_I|T_R)/VU(O_I)$ 는 O_I 에 있는 변수와 변수값들 중에서 T_R 의 식별된 변수와 변수값에 매칭되는 것들의 비율을 의미한다. 마찬가지로 $MVU(T_R|O_I)/VU(T_R)$ T_R 의 식별된 변수와 변수값 중에서 O_I 에 있는 변수와 변수값들에 매칭되는 것들의 비율을 의미한다. 이 둘을 곱하여 유사도를 위한 척도를 만들었다. 만일 O_I 에 있는 변수, 변수값이 T_R 의 식별된 변수, 변수값과 완전하게 일치하면 $Sim(O_I, T_R)$ 은 1이 된다.

유사도를 이해하기 쉽도록 설명하기 위해 간단한 예제를 보이교자 한다. [그림 6]은

BarnesAndNoble.com로부터 반환정책과 관련해 식별된 내용을 보여준다.

```
We'll <variable vid="1">refund</variable> your online purchase if you:
- Return new <variable vid="2" name="Item"> <value vid="2">books</value>, unopened <value vid="2">CDs</value>, <value vid="2">cassettes</value>, <value vid="2">VHS tapes</value>, <value vid="2">DVDs</value>, all textbooks, and prints and posters in their <variable vid="3" name="returned status"><value vid="3">original condition</value>.
- Include your packing slip and completed returns form.
- Return within <value vid="4">30</value> <variable vid="4">days of the shipment</variable> date listed on your packing slip.
```

[그림 6] 변수 및 변수값 식별 후의 RIML

[그림 6]에서 식별된 변수는 4개, 변수값은 7개이다. 온톨로지로부터 유사한 규칙을 찾기 위해 [그림 7]의 두 개의 규칙에 대해 $Sim(O_I, T_R)$ 을 계산하였다.

```
{ { Rule1
IS-A: Rule
subRuleOf: "Return Policy"
IF: (variables "Days of delivery" "Accept return") (values)
THEN: (variables Refund) (values Full) } }
{ { Rule2
IS-A: Rule
subRuleOf: "Return Policy"
IF: (variables "Accept return" "Days of shipment" Item "Returned status")
(values BOOK "Obvious signs of use" CD DVD "VHS Tape" Software "Video Game" "Cassette" "Vinyl Record" Opened "Original condition" Damaged "Missing parts")
THEN:(variables Refund) (values Partial) } }
```

[그림 7] 유사도 계산을 위한 온톨로지 규칙

$Rule1$ 에서는 굵게 표시된 두 개의 변수가 [그림 6]의 식별된 변수와 일치하였다. 따라서 $MVU(O_{Rule1}|T_R)$ 는 2가 되고 $VU(O_{Rule1})$ 는 4이다. 마찬가지로 방법으로 $MVU(T_R|O_{Rule1})$ 는 2 이고

$VU(Tr)$ 는 11이다. 다음 식은 유사도의 계산과정이다.

$$Sim(ORule1, Tr) = (MVU(ORule1|Tr)/VU(OR1)) * (MVU(Tr|ORule1)/VU(Tr)) = (2/4)*(2/11) = 0.1$$

같은 방식으로 Rule2의 유사도를 계산하면 다음과 같다.

$$Sim(ORule2, Tr) = (MVU(ORule2|Tr)/VU(OR2)) * (MVU(Tr|ORule2)/VU(Tr)) = (10/19)*(10/11) = 0.48$$

계산과정을 살펴보면 $MVU(ORule1|Tr)/VU(OR1)$ 와 $MVU(ORule2|Tr)/VU(OR2)$ 는 각각 2/4와 10/19로 비슷한 반면, $MVU(Tr|ORule1)/VU(Tr)$ 와 $MVU(Tr|ORule2)/VU(Tr)$ 은 서로 상이하다. 이것이 유사도의 계산에서 $MVU(ORule|Tr)/VU(OR)$ 뿐만 아니라 $MVU(Tr|ORule)/VU(Tr)$ 를 함께 고려하는 이유이다. 이 예에서는 Rule2가 [그림 6]에 대한 규칙으로 식별되었다.

5.7 식별된 변수와 변수값에 IF와 THEN 부여

IF와 THEN의 부여는 매우 간단하다. 만일 식별된 변수가 위에서 식별된 규칙에서 IF 부분에 속하면 IF를 부여하고 THEN에 속하면 THEN을 할당한다. 변수값도 마찬가지로 부여한다. 만일 식별된 변수나 변수값이 온톨로지로부터 식별된 규칙에 존재하지 않으면 IF와 THEN을 부여할 수 없다.

5.8 수작업에 의한 규칙구성요소 식별의 완성

규칙식별을 마무리하기 위하여 지식관리자는 자동으로 식별된 내용을 확인하고 온톨로지로부터 식별되지 못한 부분들을 수동으로 식별한다.

IF나 THEN이 부여되지 못한 변수 혹은 변수값들에 수동으로 IF, THEN을 부여하고 AND, OR와 같은 연결자를 추가로 식별한다.

6. 온톨로지를 이용한 규칙식별의 성능 평가

6.1 실험의 설계

이 장에서는 5장에서 제안된 방법론, 즉 OntoRule을 이용한 자동화된 규칙 식별효과를 측정하고자 한다. 본 실험을 위해 규칙 식별에 이용할 온톨로지가 필요하므로, 먼저 Amazon.com으로부터 온톨로지를 구축하였다. 이렇게 구축된 온톨로지는 하나의 도메인(domain), 하나의 응용(application), 13개의 규칙그룹(RuleGroup), 36개의 규칙(Rule), 17개의 변수(variable)과 494개의 변수값(value)로 구성되어 있다.

평가를 위하여 널리 알려진 두개의 온라인 서점인 Barnes&Noble.com (이하 BN)과 Powells.com (이하 Powells)를 선정하였다. 각 서점들로부터 배송과 교환 및 환불 정책을 기술해둔 웹 페이지를 분석하여 OntoRule을 이용하여 규칙을 식별하였다. 실험 순서는 BN을 먼저 하고 Powells를 나중에 하였다. 이 과정을 통해 BN으로부터 규칙을 습득한 후 수정된 온톨로지를 이용해 Powells에 적용해봄으로써 온톨로지의 학습효과를 측정할 수 있었다.

본 연구에 대한 성과 평가를 위해 고려한 사항은 다음과 같다.

1. OntoRule을 이용해 얼마나 많은 변수와 변수값이 자동으로 식별되었는가?
2. OntoRule로부터 얼마나 많은 변수와 변수값이

잘못 추천되었는가?

3. 온톨로지 학습효과가 존재하였는가?

위의 관심사를 측정하기 위해 잘못 식별된 규칙 구성요소(wrong recommendation)와 식별하지 못한 규칙 구성요소(missed recommendation) 두가지의 개념을 도입하였다. 잘못된 규칙 구성요소란 5장에서 제안된 알고리즘에 의해 추천된 변수와 변수값 중 잘못 추천된 변수와 변수값을 의미한다. 식별하지 못한 규칙 구성요소란 웹페이지에 규칙 구성요소가 존재하는데, 알고리즘이 추천하지 못한 변수와 변수값을 의미한다.

실험에서 필요한 측정 변수를 정의하면 다음과 같다.

CR: 온톨로지를 통해 제대로 추천된 개수

WR: 온톨로지를 통해 잘못 추천된 개수

MR: 식별되지 못한 개수

DR: 온톨로지를 통해 추천된 총 개수.

$$DR = CR + WR. \tag{1}$$

TT: 식별되어야 하는 총 개수.

$$TT = CR + MR. \tag{2}$$

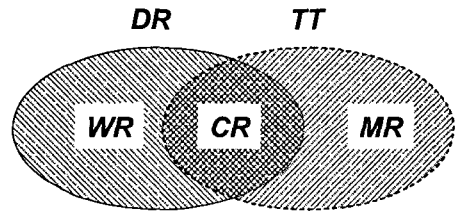
이러한 정의를 통해, WRR (Wrong Recommendation Ratio)과 MRR (Missed Recommendation Ratio)는 각각 (3)과 (4)의 식으로 정의된다.

$$WRR = WR/DR \tag{3}$$

$$MRR = MR/TT \tag{4}$$

[그림 8]은 측정 변수간의 관계를 보여주고 있다. 본 실험에서는 위의 측정 변수를 규칙 구조에 대한 걸 제외하고, 변수와 변수값에 대해서만 계산

하였다. 규칙 구조에 대한 실험은 추후 연구로 수행 중에 있다. WRR 과 MRR 은 변수와 변수값에 대해 각각 $WRR(V)$, $WRR(U)$, $MRR(V)$, $MRR(U)$ 로 표현된다.



[그림 8] 변수간의 관계도

6.2 실험 결과

전체 실험결과를 요약하면, BN은 배송비와 환불정책에 대해 11개의 규칙 그룹을 갖고 있었으며, Powells는 배송비, 무료배송, 환불정책에 대해 6개의 규칙 그룹을 갖고 있었다.

<표 2>는 변수와 변수값을 식별했을 때의 온톨로지의 기여를 보여주고 있다. <표 2>에서 변수에 대해서는 $WRR(V)$ 의 값은 27.04%였고, $MRR(V)$ 의 값은 17.04%의 결과를 보였다. 변수값에 대해서는 $WRR(U)$ 의 값은 2.55%였고, $MRR(U)$ 의 값은 5.84%로 매우 만족스러운 결과를 보여주었다. 특히, Powells의 $WRR(U)$ 의 값과 $MRR(U)$ 의 값이 각각 0.35%와 0.89%로 매우 낮았는데, 이는 Powells가 배송비를 설명할 때 좀 더 잘 구조화된 매우 큰 표를 사용하여 온톨로지를 이용하여 쉽게 식별할 수 있었기 때문이다.

<표 2>에서 전체적인 평균을 살펴보면, WRR_{Total} 의 값과 MRR_{Total} 의 값이 각각 6.11%와 7.26%가 되어 규칙 식별에 온톨로지를 활용하였을 때 만족할 만한 성과를 얻었다. 왜냐하면, 본 수치는 93.89%의 변수와 변수값이 온톨로지를 이용

<표 2> 변수와 변수값의 식별에 대한 실험 결과

		CR	WR	MR	DR	TT	WRR	MRR
Variable	BN	185	76	37	261	222	29.12%	16.67%
	Powells	39	7	9	46	48	15.22%	18.75%
	Total	224	83	46	307	270	27.04%	17.04%
Value	BN	317	41	96	358	413	11.45%	23.24%
	Powells	1440	5	13	1445	1453	0.35%	0.89%
	Total	1757	46	109	1803	1866	2.55%	5.84%
Total		1981	129	155	2110	2136	6.11%	7.26%

하여 올바르게 식별될 수 있었으며, 식별되어야 할 전체 변수와 변수값 중에서 92.74%의 변수와 변수값이 온톨로지를 이용하여 식별될 수 있음을 의미하고 있기 때문이다. 본 연구의 성과를 다른 연구와 비교할 수 없었으나, 본 실험의 결과로 나타나는 WRR과 MRR은 비교적 좋은 성과를 보이고 있다.

온톨로지 학습 효과를 측정하기 위해, Powells의 텍스트에 대해 두개의 서로 다른 규칙 식별 실험을 수행하였다. 첫번째 실험에서는 원래 Amazon.com의 온톨로지를 이용하여 규칙을 식별하였고, 두번째 실험에서는 BN의 규칙을 통해 수정된 Amazon.com의 온톨로지를 이용하여 규칙을 식별하였다. 두 실험 결과의 차이를 보여주고 있는 <표 3>을 통해 온톨로지 학습 효과가 존재하고 있음을 알 수 있다. 우리는 실험결과로 잘못 식별된 규칙 구성요소의 개수는 증가하는 반면, 식별하

지 못한 규칙 구성요소의 개수는 감소할 것으로 기대하였다. 실험 결과를 살펴보면, 기대한 대로 MR은 15에서 10으로 감소한 반면, WR은 5에서 6으로 증가하였다. 결과적으로 온톨로지 학습 효과를 통해 CR이 WR보다 더 많이 증가되었기 때문에 WRR과 MRR의 값은 둘 다 감소되었다.

6.3 실험의 한계

본 연구의 실험은 온톨로지를 활용한 규칙 식별의 효율성을 어느 정도 예를 통해 보여주었으나 몇가지 한계점을 가지고 있다. 먼저, 본 실험에서는 온톨로지를 통해 추천된 결과에 대해 평가함수를 적용하여 부적절한 후보를 제거하지 않고 모두 수용하였다. 만일 적절한 평가함수를 적용한다면 WRR의 결과값을 향상시킬 수 있을 것이다. 따라서 다음 연구로 적절한 평가함수와 평가함수를 적용한 알고리즘을 개발하고 있다. 둘째로, 본 실험은 변수와 변수값의 식별 결과에 대해서만 성과를 측정하였고 규칙, IF, THEN과 같은 다른 규칙의 구성요소에 대한 결과는 측정하지 않았다. 이에 본 실험을 확장하여 모든 규칙 구성요소를 포함시켜 평가함수를 적용하여 좀 더 정확한 결과를 측정할 수 있는 실험을 계획 중에 있다.

<표 3> Powells의 텍스트에서의 온톨로지 학습 효과의 비교 결과

	CR	WR	MR	DR	TT	WRR	MRR
Without Learning	23	5	15	28	38	17.86%	39.47%
With Learning	28	6	10	34	38	17.65%	26.32%

7. 결론

본 연구에서는 XXML 접근방법을 이용하여 규칙 식별을 하는 과정에서 지식관리자의 수작업을 줄이기 위해 온톨로지를 활용한 향상된 규칙 식별 방법론을 제안하였다. 이를 위해 첫째로 규칙식별의 자동화를 위한 규칙구성요소와 규칙구조에 대한 정보를 포함하는 온톨로지인 OntoRule을 설계하였다. OntoRule의 설계시, 온톨로지 학습효과, 하향식 접근방식과 상향식 접근방식의 차이, 온톨로지 사용범위 관리, 규칙구성요소의 식별순서, 생략된 변수의 식별과 같은 몇가지 중요한 현상을 발견하고 이를 해결하고자 하였다. 또한, 이러한 몇가지 논점을 반영하여 OntoRule을 활용한 상세한 규칙 식별 과정을 제안하였다.

제안된 XXML 접근 방법의 성능을 측정하고자, 본 논문에서는 3개의 대표적인 온라인 서점인 Amazon.com, BarnesandNoble.com, Powells.com의 실제 웹 페이지들로부터 배송 및 환불과 관련된 규칙을 습득하여 XXML의 효과를 측정하는 실험을 수행하였다. 실험 결과를 보면, 웹 페이지로부터 규칙의 습득은 97.7%로 매우 높은 정확성을 보였으며, 생성된 규칙의 완전성은 88.5%로 측정되었다. 이러한 실험결과를 통해 XXML이 특정 주제에 관한 전문가 시스템을 구축하기 위해 웹 페이지로부터 규칙을 추출할 때 매우 효율적인 도구가 될 수 있으며, 또한 추출된 규칙과 웹 페이지 간의 일관성이 효과적으로 유지될 수 있음을 알 수 있었다.

제안된 온톨로지를 이용한 규칙 식별 효과를 측정하기 위해, 본 논문에서는 XXML 에디터를 이용하여 Amazon.com으로부터 얻은 온톨로지를 이용해 BN.com과 Powells.com으로부터 자동으로 규칙을 식별하는 과정에 대해 실험을 수행하였다.

실험 결과, 전체 변수와 변수값 중 7.26%에 대해 식별을 하지 못하였고, 추천된 변수와 변수값 중 잘 못 식별한 값은 6.11%로 측정되었다. 이러한 실험결과를 통해 온톨로지가 규칙 식별시 높은 효율성을 보임을 알 수 있었다. 추후 시맨틱 웹의 발달로 인해 온톨로지가 풍부해 진다면, 규칙 습득시 활용할 수 있는 더욱 더 많은 온톨로지를 가질 수 있게 될 것이므로, 본 연구에서 제안된 방법론이 규칙 습득시 효율적인 도구가 될 수 있을 것으로 기대된다.

본 연구는 다양한 분야에서 적용될 수 있을 것으로 기대된다. 예를 들어, 쇼핑물에서 약관으로부터의 규칙 추출, 각 보험 회사들의 보험 산정 규칙, 그리고 은행의 대출 평가 규칙 등이 있다.

본 연구의 방법론을 좀 더 향상시키기 위해 추후 다음과 같은 연구가 진행되어야 한다. 첫째로, 변수, 변수값 그리고 규칙을 선택할 때 적절한 평가함수를 적용함으로써 식별의 정확성을 높일 수 있으므로 평가함수에 대한 보다 정교한 연구가 요구된다. 둘째로, 식별된 변수와 변수값을 이용해 좀 더 자동화된 규칙 식별에 대한 방법을 연구함으로써 규칙 식별 전체 과정을 완전 자동화시키는 것이 가능할 것이다. 마지막으로 본 연구의 방법론을 일반화하기 위해 좀 더 다양한 분야에서 본 연구의 효과를 입증하는 연구가 요구된다.

참고문헌

- [1] Alani, H., Kim, S., Millard, D.E., Weal, M.J., Hall, W., Lewis, P.H., Shadbolt, N.R., 2003. Automatic Ontology-Based Knowledge Extraction from Web Documents. IEEE Intelligent Systems 18(1), 14-21.

- [2] Babowal, D., Joerg, W., 1999. From Information to Knowledge: Introducing WebStruct's Knowledge Engineering Approach. In: Proceedings of the 1999 IEEE Canadian Conference on Electrical and Computer Engineering, Edmonton, Alberta, 1525 -1530.
- [3] Bontcheva, K., Tablan, V., Maynard, D., Cunningham, H., 2004. Evolving GATE to Meet New Challenges in Language Engineering. *Natural Language Engineering* 10(3), 349-473.
- [4] Brickley, D., Guha, R.V., 2004. RDF Vocabulary Description Language 1.0: RDF Schema. W3C Recommendation, <<http://www.w3c.org/TR/rdf-schema/>>.
- [5] Brockmans, S., Volz, R., Eberhart, A., Loffler, P., 2004. Visual Modeling of OWL DL Ontologies Using UML. In: Proceedings of 3rd International Semantic Web Conference, Hiroshima, Japan, 198-213.
- [6] Chan, K., Lam, W., 2005. Extracting Causation Knowledge from Natural Language Texts. *International Journal of Intelligent Systems* 20(3), 327-358.
- [7] Connolly, D., van Harmelen, F., Horrocks, I., McGuinness, D.L., Patel-Schneider, P.F., Stein, L.A., 2001. DAML+OIL Reference Description, W3C Note <<http://www.w3.org/TR/2001/NOTE-daml+oil-reference-20011218>>.
- [8] Craven, M., DiPasquo, D., Freitag, D., McCallum, A., Mitchell, T., Nigam, K., Slattery, S., 2000. Learning to Construct Knowledge Bases from the World Wide Web. *Artificial Intelligence* 118(1-2), 69-113.
- [9] Crow, L., Shadbolt, N., 2001. Extracting Focused Knowledge from the Semantic Web. *International Journal of Human-Computer Studies* 54, 155-184.
- [10] Donini, F.M., Lenzerini, M., Nardi, D., Schaerf, A., 1996. Reasoning in Description Logics, Principles of Knowledge Representative. Ed: G. Brweka. CSLI Publications, Stanford, pp. 191-236.
- [11] Dzbor, M., Domingue, J., Motta, E., 2003. Magpie - Towards a Semantic Web Browser. *Lecture Notes in Computer Science* 2870, 690-705.
- [12] Hemnani, A., Bressan, S., 2002. Extracting Information from Semi-Structured Web Documents. *Lecture Notes in Computer Science* 2426, 166-175.
- [13] Horrocks, I., 2002. DAML+OIL: A Description Logic for the Semantic Web. *IEEE Data Engineering* 25(1), 4-9.
- [14] Horrocks, I., Patel-Schneider, P.F., Boley, H., Tabet, S., Grosz, B., Dean, M., 2004. SWRL: A Semantic Web Rule Language Combining OWL and RuleML, W3C Member Submission, <<http://www.w3.org/Submission/2004/SUBM-SWRL-20040521/>>.
- [15] Hulth, A., Karlgren, J., Jonsson, A., Bostrom, H., Asker, L., 2001. Automatic Keyword Extraction using Domain Knowledge. In: Proceedings of the Second Computational Linguistics and Intelligent Text Processing, Mexico, 472-482.
- [16] Kang, J., Lee, J.K., 2005. Rule Identification from Web Pages by the XRML Approach. *Decision Support Systems*, Forthcoming.
- [17] Kiryakov, A., Popov, B., Terziev, I., Manov, D., Ognyanoff, D., 2004. Semantic Annotation, Indexing, and Retrieval. *Web Semantics* 2(1), 49-79.
- [18] Knublauch, H., Fergerson, R. W., Noy, N.

- F., Musen, M. A., 2004. The Protégé OWL plugin: An Open Development Environment for Semantic Web Applications. *Lecture Notes in Computer Science* 3298, 229-243.
- [19] Kopena, J., Regli, W.C., 2003. DAMLJessKB: A Tool for Reasoning with the Semantic Web. *IEEE Intelligent Systems* 18(3), 74-77.
- [20] Lebbink, H.J., Witteman, C.L.M., Meyer, J.-J.Ch., 2002. Ontology Based Knowledge Acquisition for Knowledge System. In: *Proceedings of the 14th Dutch-Belgian Artificial Intelligence Conference*, Leuven, Belgium, 195-202.
- [21] Lee, J.K., Sohn, M., 2003. Extensible Rule Markup Language - toward Intelligent Web Platform, *Communications of the ACM* 46, 59-64.
- [22] Liebig, T., Noppens, O., 2004. ONTOTRACK: Combining Browsing and Editing with Reasoning and Explaining for OWL Lite Ontologies. *Lecture Notes in Computer Science* 2870, 244-258.
- [23] Maedche, A., Staab, S., 2000. Mining Ontologies from Text, *Lecture Notes in Artificial Intelligence* 1937, 189-202.
- [24] Manola, F., Miller, E., 2004. Resource Description Framework (RDF) Primer. W3C Recommendation, <<http://www.w3.org/TR/REC-rdf-syntax/>>.
- [25] Miller, G.A., 1995. WordNet a Lexical Database for English, *Communications of the ACM* 38(11), 39-41.
- [26] Mukherjee, S., Yang, G., Ramakrishnan, I.V., 2003. Automatic Annotation of Content-Rich HTML Documents: Structural and Semantic Analysis. *Lecture Notes in Computer Science* 2870, 533-549.
- [27] Popov, B., Kiryakov, A., Ognyanoff, D., Manov, D., Kirilov, A., 2004. KIM - A Semantic Platform for Information Extraction and Retrieval. *Natural language Engineering* 10(3/4), 375-392.
- [28] Ruiz-Sánchez, J.M., Valencia-García, R., Fernández-Breis, J.T., Martínez-Béjar, R., Compton, P., 2003. An Approach for Incremental Knowledge Acquisition from Text. *Expert System with Applications* 25(1), 77-86.
- [29] RuleML, 2003. The Rule Markup Initiative. <<http://www.dfki.uni-kl.de/ruleml/>>.
- [30] Smith, M.K., Welty, C., McGuinness, D., 2004. OWL Web Ontology Language Guide. W3C Recommendation, <<http://www.w3c.org/TR/owl-guide/>>.
- [31] Tang, Y.Y., Yan, C.D., Suen, C.Y., 1994. Document Processing for Automatic Knowledge Acquisition. *IEEE Transactions on Knowledge and Data Engineering archive* 6(1), 3 - 21.
- [32] Tim Berners-Lee, 2002. The Semantic Web, Academic discussion, Japan Prize.
- [33] van Harmelen, F., Fensel, D., Practical knowledge representation for the Web. In: *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, Stockholm, Sweden.
- [34] van Heijst, G., Schreiber, A.T., Wielinga, B.J., 1997. Using Explicit Ontologies in KBS Development. *International Journal of Human-Computer Studies* 45, 183-292.
- [35] Vargas-Vera, M., Motta, E., Domingue, J., Shum, S.B., Lanzoni, M., 2001. Knowledge Extraction by using an Ontology-based Annotation Tool, In: *Proceedings of the Knowledge Markup and Semantic Annotation Workshop*, Canada.

Abstract

The Effect of Knowledge Acquisition through OntoRule: XRML Approach

Sangun Park* · Jae Kyu Lee* · Juyoung Kang**

We developed a methodology of rule acquisition from texts such as Web pages which utilizes ontology in identification of rule components. We expect that the proposed methodology can reduce the bottleneck of rule acquisition and contribute to the utilization of rule based systems. As parts of our research, we designed an ontology for rule acquisition named OntoRule and proposed a rule acquisition methodology through OntoXRML which is an acquisition tool using OntoRule. Also, we evaluated our approach by calculating missed recommendations and wrong recommendations of rule components in rule acquisition experiments over three online bookstores.

Key words : Knowledge Acquisition, Rule Acquisition, Rule Identification, XRML, Ontology, OntoRule, Ontology Engineering, RuleML, XML

* Management Engineering, KAIST Graduate School of Management

** Department of E-Business, School of Business, Ajou University

