제한된 언어집합과 온톨로지를 활용한 반자동적인 규칙생성 방법 연구[1]
(Methodology for semi-autonomous rule extraction based on Restricted Language Set and ontology)

손미애, 최윤규
성균관대학교 산업공학과
(myesohn@skku.edu, saintcyg@gmail.com)

초록

지능정보시스템 구축에 있어서 자동화가 어려운 단계중의 하나인 규칙 습득을 위해 활용되는 방법중의 하나가 제한된 언어집합 기법을 이용하는 것이다. 그러나 제한된 언어집합 기법을 이용해 규칙을 생성하기 위해서는 규칙을 구성하는 변수와 그 값들에 대한 정보가 사전에 정의되어 있어야 하는데, 유동성이 큰 웹 환경에서 예상 가능한 모든 변수와 그 값을 사전에 정의하는 것이 매우 어렵다. 이에 본 연구에서는 이러한 한계를 극복하기 위해 제한된 언어집합 기법과 온톨로지를 이용한 규칙 생성 방법론을 제시하였다. 이를 위해 지식의 습득 대상이 되는 특정 문장은 문법구조 분석기를 이용해 파싱을 수행하며, 파싱된 단어들을 이용해 규칙의 구성 요소인 변수와 그 값을 식별한다. 그러나 규칙을 내포한 자연어 문장의 불완전성으로 인해 변수가 명확하지 않거나 완전히 빠져 있는 경우가 흔히 발생하며, 이로 인해 온전한 형식의 규칙 생성이 어렵게 된다. 이 문제는 도메인 온톨로지의 생성을 통해 해결하였다. 이 온톨로지는 특정 도메인을 구성하고 있는 개념들간의 관계를 포함하고 있다는 점에서는 기존의 온톨로지와 유사하지만, 규칙을 완성하는 과정에서 사용된 개념들의 사용빈도를 기반으로 온톨로지의 구조를 변경하고, 결과적으로 더 정확한 규칙의 생성을 지원한다는 점에서 기존의 온톨로지와 차별화된다. 이상의 과정을 통해 식별된 규칙의 구성요소들은 제한된 언어집합 기법을 이용해 구체화된다. 본 연구에서 제안하는 방법론을 설명하기 위해 임의의 인터넷 쇼핑몰에서 수행되는 배송관련 웹 페이지를 선정하였다. 본 방법론은 XRML에서의 지식 습득 과정의 효율성 제고에 기여할 수 있을 것으로 기대된다.

Keywords
제한된 언어집합, knowledge acquisition, ontology, XRML

1. 서론

컴퓨터(machine)가 웹 문서에 내포되어 있는 정보나 지식 등을 이해한 후, 사용자의 요구에 맞게 처리할 수 있도록 하는 일련의 연구가 시맨틱 웹 분야를 중심으로 적극적으로 추진되고 있다[1]. 이를 위해 웹 콘텐츠를 컴퓨터가 처리할 수 있는 XML로 표현한다거나, 데이터 모델과 온톨로지를 이용해 웹 문서들 간의 관계와 그 안에 포함된 콘텐츠들이 의미하는 바와 관계를 엄격하고 명료하게 나타낸다고 하더라도, 웹 문서가 함축하고 있는 규칙을 이해하고 처리하는 주체는 여전히 사람이다. 웹 문서에 함축하고 있는 규칙을 생성하기 위한 연구가 XRML을 중심으로 이루어지고 있으나[7, 8 and 10], 현재까지 수행된 XRML 연구에서는 에디터의 지원을 받는다 하더라도 규칙 생성의 매 단계에 지식관리자가 개입하는 것이 필수적이었다. 이는 현재의 자연어 처리 기술 수준에서 보면 아주 당연한 사실이다 [6].

자연어 처리의 한계를 극복하기 위해 수행되고 있는 연구들 중[1, 3, 4, 9 and 11], 제한된 언어 집합 기법을 사용함으로써 웹 문서에 포함되어 있는 규칙 습득의 가능성을 확인하는 것이 본 논문의 목표이다. 이를 위해 본 논문에서는 제한된 언어 집합 기반의 규칙 습득 도구인 RESO-RULE(REstricted Language Set and Ontology for RULe Extraction)을 개발하는 중이다. RESO-RULE을 개발하는 과정에서 규칙 습득 도구가 가지고 있어야 할 몇 가지 속성을 식별되었다. 본 논문에서 식별한 속성은 다음과 같다.

[1] 본 연구는 한국과학재단 특정기초연구 (R01-2006-000-10303-0)지원으로 수행되었음

- 습득의 완전성: 지식 식별의 대상이 되는 웹 문서 중, 규칙을 내포하고 있는 자연어 형태의 문장은 빠짐없이 식별되어야 한다.

- 표현의 다양성: 구조적인 규칙으로 변환하기 위해 식별된 비구조적인 자연어 문장(unstructured natural language statement)은, 지식관리자의 이해를 돕기 위해 구조적인 자연어 문장(structured natural language statement)으로 변환되어야 한다. 비구조적인 자연어 문장은 웹 문서에 포함된 자연어 문장을 의미하며, 구조적인 자연어 문장은 If-Then과 같은 규칙의 형식을 포함하고 있는 자연어 문장을 의미한다. 구조적인 자연어 문장은 비구조적인 자연어 문장에 비해 규칙으로의 변환을 용이하게 한다.

- 변환의 용이성: 지식관리자가 도구에 대한 특별한 학습을 하지 않아도 지식을 변환할 수 있어야 한다. 전문가가 아닌 누구라도 제안된 절차에 따르기만 하면 규칙을 생성할 수 있어야 한다.

- 변화에 대한 적응성: 비구조적인 자연어 문서나 이미 생성된 규칙이 환경의 변화나 지식관리자의 학습에 의해 변경 요인이 발생한 경우, 변화를 손쉽게 반영할 수 있어야 한다.

위와 같은 속성을 가진 도구를 사용하게 되면, 기존 연구들에서 제안하고 있는 반자동화된 XRML 에디터가 가진 한계를 극복할 수 있다. 즉, XRML에디터를 이용해 웹 문서에 포함되어 있는 규칙을 식별하기 위해서는 반드시 새로운 규칙표현 언어에 대한 추가적인 학습이 필요하며, 복잡한 규칙 생성 절차로 인해 지식관리자가 어려움을 겪을 수도 있고, 그로 인해 지식관리자나 시스템이 정확한 규칙을 추출하지 못할 수도 있다. 이러한 문제를 해결하기 위해 RESO-RULE에서는 다음과 같은 방법을 사용하였다.

- 규칙을 내재하고 비구조적인 자연어 문장 자체를 식별하는 과정 즉 자연어를 이해하는 절차는 지식관리자가 전담하고, 규칙을 내포하고 있다고 인식된 비구조적인 자연어 문장으로부터의 규칙 식별성은 제한된 언어 집합 기법을 사용해 개발한 RESO-RULE 에디터와 지식관리자의 상호작용을 통해 수행한다.

- RESO-RULE 에디터는 사람과 컴퓨터가 동시에 이해할 수 있는 If-Then 형식의 문장을 먼저 생성함으로써, 지식관리자로부 터 규칙의 완전성에 대한 검증을 받은 이후에 구조적인 규칙을 생성한다.

- 지식관리자가 규칙 생성에 필요한 도구나 방법론에 대한 추가적인 학습 없이도 규칙을 생성할 수 있도록 하였다.

그러나 자연어로 표현된 문장의 불완전성으로 인해 변수가 명확하지 않거나 완전히 빠져 있는 경우가 흔히 발생하며, 이로 인해 If-Then 형식의 규칙 생성이 어려울 수도 있다. 이를 해결하기 위해 언어 집합에 포함되어 있는 어휘들에 대한 온톨로지를 구축하였다. 이 온톨로지는 특정 도메인을 구성하고 있는 개념들간의 관계를 포함하고 있다는 점에서는 기존의 온톨로지와 유사하지만, 규칙을 완성하는 과정에서 사용된 개념들의 사용빈도를 기반으로 온톨로지의 구조를 변경하고, 결과적으로 더 정확한 규칙의 생성을 지원한다는 점에서 기존의 온톨로지와 차별화된다. 이상의 과정을 통해 식별된 규칙의 구성요소들은 제한된 언어 집합 기법을 이용해 구체화된다. 본 연구에서 제안하는 방법론을 설명하기 위해 임의의 인터넷 쇼핑몰에서 수행되는 배송관련 웹 페이지를 선정하였다. 본 방법론은 XRML에서의 지식 습득 과정의 효율성 제고에 기여할 수 있을 것으로 기대된다.

RESO-RULE을 이용한 규칙 생성 절차를 보여주기 위해 본 논문은 다음과 같이 구성하였다. 2장에서는 제한된 언어집합을 이용한 자연어 처리 방법을 살펴보고, 3장에서는 RESO-RULE의 아키텍처를 도시할 것이다. RESO-RULE을 이용한 규칙 생성 방법은 4장에서 설명하였다. 5장에서는 현재 개발중인 RESO-RULE의 프로토타입의 일부를 보여주고, 6장에서는 RESO-RULE의 규칙 습득 능력을 분석할 것이다. 7 장에서는 RESO-RULE이 기여한 바와 기존 연구와의 차별성을 보이고, 마지막으로 8장에서는 결론과 향후 연구 과제에 대하여 정리하였다.

2. 제한된 언어 집합을 이용한 자연어 처리

지능정보시스템의 구축의 병목지점으로 알려져 있는 규칙의 획득[5]을 지원하기 위해 수동적인 방법과 반자동화된 방법 등이 제안된 바 있다[1]. Lee and Sohn[7]이 제안한 extensible Rule Markup Language(XRML)에서의 규칙 획득하는 방법 역시 반자동화된 방법에 속한다. 그러나 규칙의 습득 방법이 수동적인 반자동적이든 반드시 수반되어야 하는 작업 과정이 획득한 규칙을 추론엔진이 활용할 수 있는 형식으로 변환하는 것이다. 이러한 변환을 수행하는 주체인 지식관리자가 이러한 변환을 수행하기 위해서는 추론엔진이 사용하는 언어의 문법 구조에 대한 추가적인 학습이 필요하게 된다. 도메인 전문가가 아닌 지식관리자가 도메인 지식을

정확하게 습득하고 표현해야 한다는 부담에 더해 언어 학습의 부담까지 갖게 되는 것이다.
이에 본 논문에서는 지식관리자가 추론엔진이 사용하는 언어의 문법 구조에 대한 학습 없이도 추론엔진이 요구하는 규칙을 생성해 주는 ' 제한된 언어 집합' 기법을 활용하고자 한다[6]. 제한된 언어집합은 특정 추론엔진이 사용하는 언어의 문법 구조를 사전에 정의한 후, 사전에 정의된 구조와 절차에 따라 규칙을 생성해 주는 기법이다. 이 기법의 특징은 비구조적인 자연어 문장을 제한된 언어집합이 지원하는 문법과 유사한 형식을 갖는 구조적인 자연어 문장으로 수정한 후, 그 결과를 이용해 컴퓨터가 이해할 수 있는 전형적인 규칙(canonical rule)으로 변환하는 것이다. 이 기법을 사용할 경우, 지식관리자가 해야 할 일은 획득한 지식을 추론엔진이 요구하는 규칙으로 직접 변환하는 것이 아니라, 비구조적인 자연어 문장을 생략된 변수나 값이 없는 구조적인 자연어 문장으로 변환하기만 하면 되는 것이다. 예를 들어, 그림 1의 비구조적인 자연어 문장에 ' 제한된 언어 집합' 기법을 적용하면 구조적인 자연어 문장으로 변환된다.

Unstructured NL statement	We can ship to addresses in Korea.		
If-Then을 표현하는 제한된 언어 집합	<start>	::=	the <rule_name> is that <rule_main>.
	<rule_name>	::=	"enter the rule name"
	<rule_main>	::=	if <rule_part> then <rule_part>
	<rule_part>	::=	<vav>(, <opertor> <vav>)*
	<vav>	::=	<variable> is <value>
	<operator>	::=	and | or
	<variable>	::=	"select one of the variables suggested by the system, and if needs be, add variables."
	<value>	::=	"select one of the values suggested by the system, and if needs be, add values."
Structured NL statement	The shipping decision is that if country is Korea, then shipping is permitted.		

그림 1 - 제한된 언어집합을 이용해 자연어 문장을 수정하는 절차

이러한 제한된 언어집합을 기반으로 하는 규칙생성의 전 과정은 RESO-RULE editor의 도움을 받는다. 즉, 에디터가 유도하는 대로 필요한 변수나 값을 입력하거나 선택하기만 하면 자동적으로 규칙이 생성된다. 에디터의 도움을 받음으로써, 지식관리자가 에디터의 능력을 지나치게 낮게 보거나 높이 평가해서 규칙 생성에 실패하는 것을 방지하게 된다[3, 4, and 11]. 그러나 제한된 언어집합 기법만을 이용해 구조적인 자연어 문장을 생성할 수는 없다. 자연어 문장을 분석해 제한된 언어집합으로 전달해 주는 방법과 절차에 대해서는 4장에서 상세하게 논의할 것이다.

3. RESO-RULE의 아키텍처

본 장에서는 RESO-RULE의 계층적 아키텍처와 각 계층의 특징에 대해 간략히 설명하고자 한다. 그림 2는 RESO-RULE의 계층적 아키텍처를 보여주고 있다.

GUI 계층

GUI 계층은 지식관리자가 비구조적인 자연어 문장으로부터 규칙을 식별하여 구조적인 자연어 문장과 더 나아가 전형적인 규칙(canonical rule)의 생성을 지원해 주는 계층이다. GUI 계층의 RESO-RULE editor를 이용해, 그림 1에서 도시한 것과 같은 If-Then 형태의 구조적인 자연어 문장을 생성하게 되며, XRML Generator를 통해 구조적인 자연어 문장을 처리함으로써 RSML 형식의 규칙을 얻게 된다.

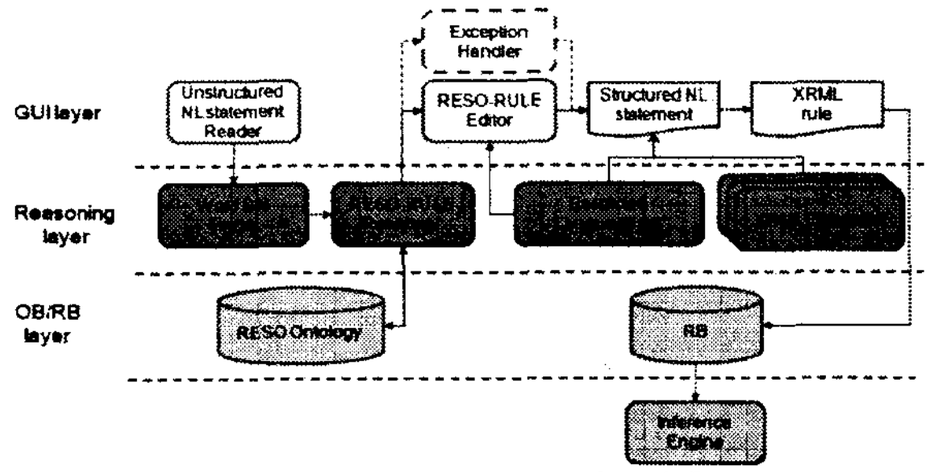


그림 2 - RESO-RULE의 계층 구조

Reasoning 계층

RESO-RULE의 핵심계층으로서 다음과 같은 두 가지 역할을 수행한다. 첫째, 규칙을 포함하고 있는 비구조적인 자연어 문장을 구조적인 자연어 문장으로 변환해 주는 역할과 둘째, 구조적인 자연어 문장을 추론엔진에 적합한 형식의 규칙으로 변환해 주는 역할을 수행한다. Reasoning 계층의 구성 요소 중 Word Set Finder는 규칙을 포함하고 있는 비구조적인 자연어 문장을 파싱하여 규칙을 생성하는데 필요한 단수명사들의 집합을 식별하는 역할을 수행한다. 이에 더해 파싱된 자연어 문장 중 일반동사가 트리의 최하위 노드에서 발견되면 이 동사의 명사형 또한 단수 명사의 집합에 추가한다. 식별된 단수명사의 집합은 그림 1의 <vav>의 variable 또는 value와 매칭될 것이다. Variable과 value이 정확하게 매칭되지 않는 경우, 즉 variable이 생략된 경우에는 RESO-RULE reasoner가 온톨로지를 참조해 variable의 대안이 될 수 있는 어휘들을 제안하게 된다. 지식관리자는 RESO-RULE editor상에 나열된 어휘의 목록 중 적절한 어휘를 선택해 규칙을 완성하고, RESO-RULE reasoner는 지식관리자에 의해 선택된 어휘를 RESO Ontology에 알려준다. 이 어휘의 선택 빈도는 온톨로지가 저장하고 있다가 RESO Ontology의 구조를 변경하는 데 활용한다.

OB/RB 계층

지식관리자가 사전에 구축한 도메인의 온톨로지이다. 이 온톨로지는 특정 도메인을 구성하고 있는 개념들간의 관계를 포함하고 있다는 점에서는 기존의 온톨로지와 유사하지만, 규칙을 완성하는 과정에서 사용된 개념들의 사용빈도를 기반으로 온톨로지의 구조를 변경하고, 결과적으로 더 정확한 규칙의 생성을 지원한다는 점에서 기존의 온톨로지와 차별화된다. 웹 문서로부터 식별된 XRML rule들을 규칙베이스에 저장되었다가 XRML 규칙을 처리할 수 있는 추론엔진에 의해 사용될 것이다.

4. RESO-RULE을 이용한 규칙습득 방법

4.1 Unstructured NL statement Reader

자연어로 쓰여진 웹 문장으로부터 규칙을 습득하기 위한 첫 번째 단계는 지식관리자가 비구조적인 자연어 문장을 선택하는 것이다. 전술한 바와 같이, RESO-RULE에서 규칙을 내포하고 있을 것 같은 비구조적인 자연어 문장을 선택하는 주체는 지식관리자이다. 다음 문장은 인터넷 서점이 amozon.com에서 웹사이트에 게시한 배송관련 문서의 일부이다. 이 문서에 포함되어 있는 규칙을 식별하기 위해 우선 첫 번째 문장을 선택하였다고 가정하자.

> We are currently able to ship books, CDs, DVDs, VHS videos, music cassettes, and vinyl records to European addresses. We can also ship some software, electronics accessories, kitchen and housewares, and tools to addresses in Denmark, Finland, France, Germany, Ireland, the Netherlands, Sweden, and the United Kingdom.

지식관리자는 위 문장에 다음과 같은 If-Then 형식의 규칙이 포함되어 있음을 알고, RESO-RULE을 이용하여 자신이 인지하고 있는 형식의 구조적인 자연어 문장을 생성하려고 한다.

> The shipping decision is that if item are book, CD, DVD, VHS video, music cassette, and vinyl record, and region is European address, then shipping is permitted.

Unstructured NL Statement Reader는 지식관리자가 규칙을 포함하고 있을 것이라고 판단한 문장을 읽은 후, Word Set Finder에게 전달하는 역할을 수행한다.

4.2 Word Set Finder

Text Reader가 전달한 자연어 문장의 구조를 분석하는 역할을 수행하는 것이 Word Set Finder이다.

4.2.1 문장구조분석기

Word Set Finder의 문장구조 분석기는 JavaNLP를 이용하여 문장의 구조를 분석하며, 4.1절에서 선택한 문서를 분석하면 그림 3과 같다. 그림 3에서 NNP는 단수 명사(singular noun), NNS는 복수 명사(plural noun), CC는 접속사(conjunction coordination) 및 TO는 전치사 to를 의미한다.

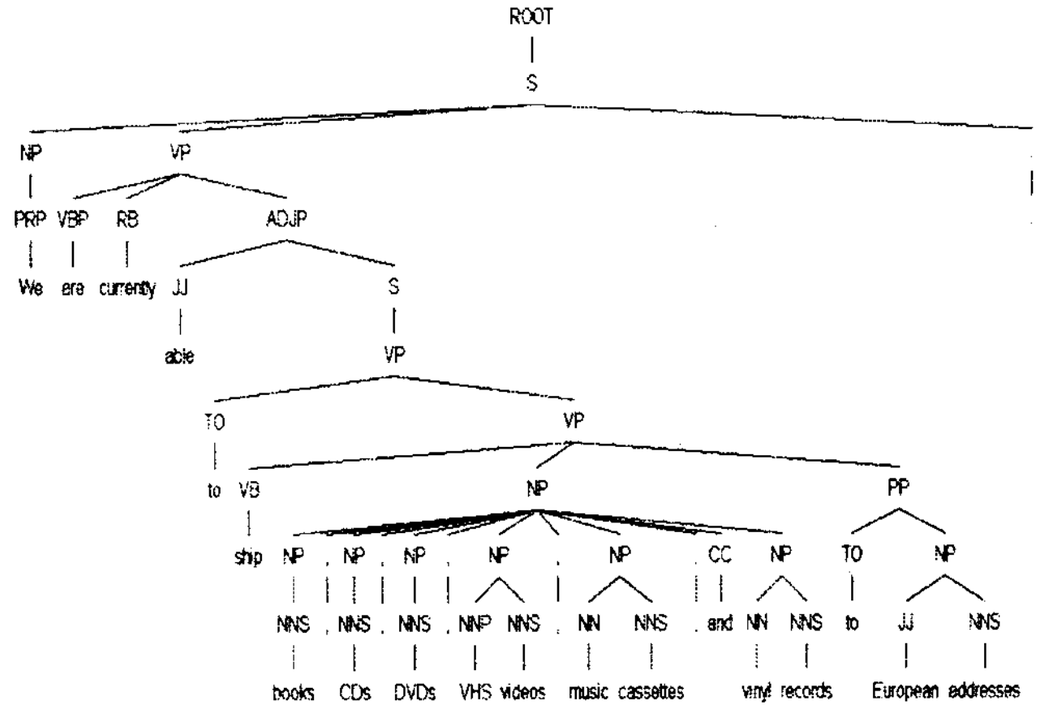


그림 3 - 문장구조분석기의 실행 결과

4.2.2 Noun set classifier

다음은 문장구조 분석기가 추출한 일반 명사들의 집합은 다음과 같은 원칙에 의거 재구성한다.

- 일반 명사와 be동사를 제외한 동사에 해당하는 {ship, books, CDs, DVDs, VHS videos, music cassettes, vinyl records, European addresses}를 규칙 생성을 위한 1차 요소로 선택한다.
- 분리된 단어 집합 중 동사 부분인 {ship}와 명사 부분인 {books, CDs, DVDs, VHS videos, music cassettes, vinyl records, European addresses}를 분리한다.
- 두개의 단어로 이루어진 명사 단어들은 '_'를 단어 사이에 삽입한다. 예를 들어 music cassettes는 music_cassettes로 vinyl records는 vinyl_records로 수정한다.
- 콤마(,)를 사이에 두고 연속적으로 나열되어 있는 명사의 집합들을 하나의 단어집합으로 식별한다. 이때 콤마(,)를 사이에 두고 연속적으로 나열되어 있는 명사들이 출현하다가 and나 or와 같은 연결사가 뒤이어 나오면 연결사 바로 뒤의 단어까지 하나의 집합으로 묶어 준다. 그 결과 {books, CDs,

DVDs, VHS videos, music_cassettes, vinyl_records}와 {European addresses}이 분리된다.

- 이 과정을 거쳐 분류한 세 개의 단어 집합, {ship}, {books, CDs, DVDs, VHS videos, music_cassettes, vinyl_records} 및 {European_ addresses}에서 복수형을 단수형으로 변환한 후 RESO_RULE Reasoner에 전달한다.

4.3 RESO-RULE Reasoner

Word Set Finder로부터 전달받은 명사들의 집합을 RESO Ontology를 이용해 분석함으로써, 생략된 변수를 식별하는 것이 RESO-RULE reasoner의 임무이다.

4.3.1 Variable Matchmaker

GINO_RUEL Reasoner의 Variable Matchmaker는 전달받은 명사들의 집합과 매칭되는 클래스를 찾기 위해 RESO Ontology에 접속한다. RESO Ontology에 접속한 후 다음과 같은 절차를 통해 변수를 찾게 된다. 우선 전달받은 단어들이 온톨로지에 존재하는 여부를 파악한다. RESO Ontology에 해당 클래스가 존재하며, 해당 클래스와 관련되어 있는 대표어인지, 대표어의 동의어인지를 검사하여 특정 대표어의 동의어일 경우 4.4절에서 제시한 방법대로 추가 작업을 한다. 대표어란 해당 도메인에서 사용되는 클래스 및 그 클래스와 동의어 관계에 있는 클래스들의 집합 가운데에서 사용 빈도가 제일 높은 클래스를 의미한다. 우선 대표어들로 이루어진 {book, CD, DVD, VHS_video, music_cassette, vinyl_record}의 경우, 이 단어 집합 중 온톨로지와 매칭되는 부분은 그림 4에서 보는 바와 같이 {book, CD, dvd, music_cassette, VHS_video}이다.

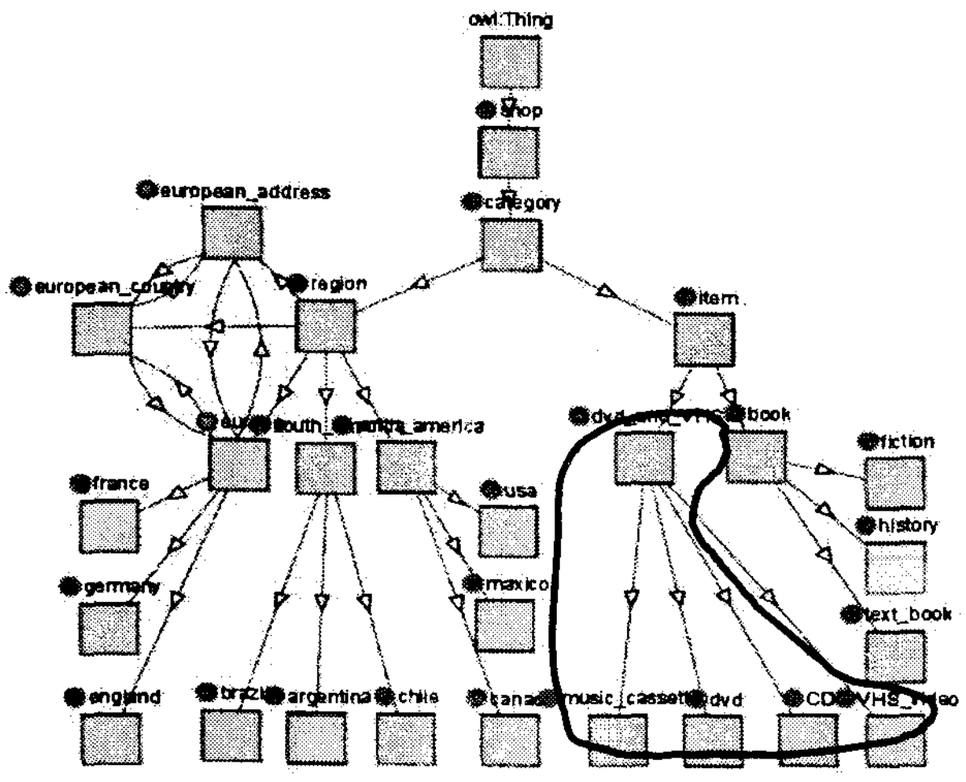


그림 4 - RESO Ontology

온톨로지와 매칭되는 클래스들을 근간으로 그림 5의 return_Variables 알고리즘을 사용하여 그림 6과 같이 변수를 찾아낸다.

```
Function return_Variables
  find WN[] in ontology tree
  for 1 to n
    CN <- WNn
    while CN != RN
      CN.count++
      change CN to PN
  for 1 to n
    CN <- WNn
    while PN != RN
      if CN.count != PN.count & PN doesn't exist in VN[]
      save PN in VN[]
    change CN to PN
  return VN[]
end
```

그림5 - Pseudo code for return_Variables algorithm

그림 5에서 CN은 현재 노드, PN은 현재 노드의 부모 노드, RN은 루트 노드, VN은 변수 노드를 나타내고 WN은 단어집합내의 단어를 포함한 노드를 의미한다.
이상의 과정을 거치면 온톨로지로부터 {item}과 {DVD_and_VHS}이라는 변수가 식별된다. 식별된 변수들은 RESO-RULE Editor에 전달되어, 지식관리자가 규칙을 생성하고자 할 경우 표 1과 같이 변수와 값, 그리고 문장형태로 제안되어 선택할 수 있도록 한다.

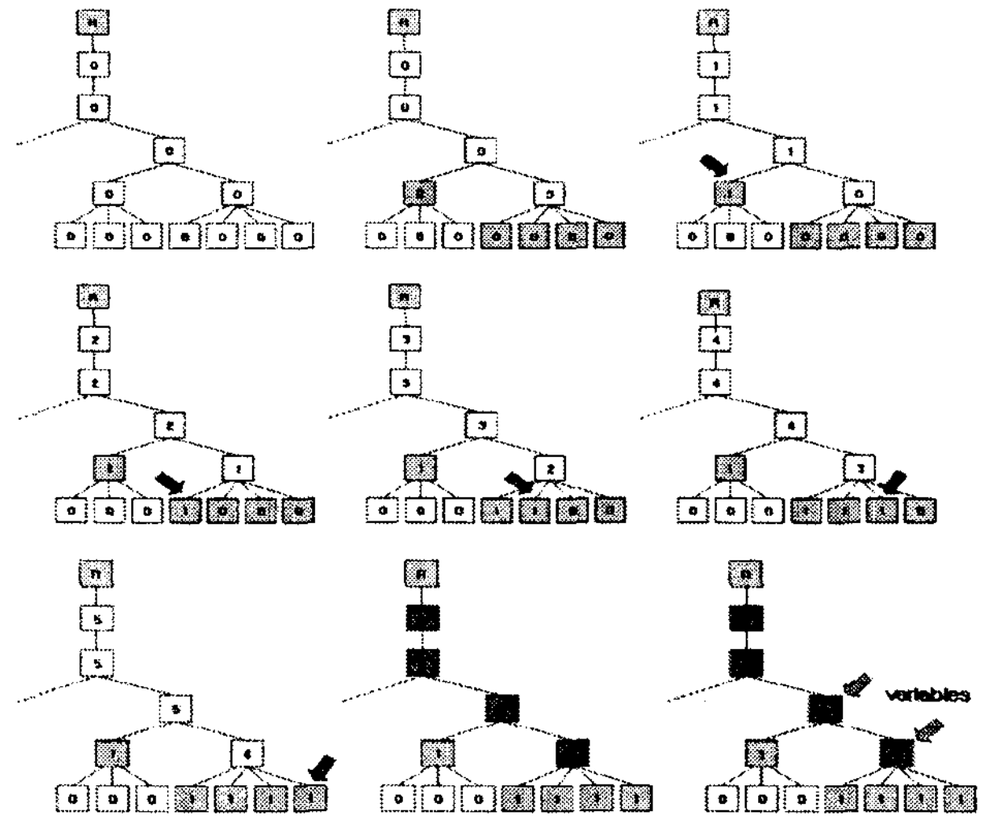


그림 6 - 단어집합에 대한 변수 추출

표 1 - 제안된 변수/값과 그에 의해 생성되는 문장

레인	변수	값	문장
1	item	book, CD, DVD, VHS video, music cassette, vinyl record	if item is book, CD, DVD, VHS video, music cassette and, vinyl record –
2	DVD_and_VHS	CD, DVD, VHS video, music cassette, vinyl record	if DVD_and_VHS is CD, DVD, VHS video, music cassette, and vinyl record and item is book –
	item	book	

위의 절차는 RESO-RULE Editor가 지원하며, 이에 대한 상세한 설명은 5장에서 할 것이다. 그러나 RESO-RULE Editor가 구조적인 자연어 문장 생성 과정을 지원한다고 하더라도, RESO 온톨로지가 비구조적인 자연어 문장이 내포하고 있는 모든 변수와 값들의 관계를 정의할 수는 없기 때문에, 자동적인 매칭이 어려운 경우가 발생한다. 이러한 경우는 Exception Handler가 처리한다.

4.3.2 Exception Handler

이때, 표 1의 변수 값으로 나타나 있는 ' vinyl_record' 의 경우, RESO Ontology에 존재하지 않는 클래스이기 때문에 어느 변수의 값인지 자동적으로 매칭해 줄 수 없다. 이렇게 예외적인 경우는 지식관리자에 판단에 의해 해결한 후, 온톨로지를 수정한다. 온톨로지의 수정 문제는 추후 논문에서 다루기로 한다. 위의 두 단계를 거쳐 선택한 규칙의 대안은 구조적인 자연어 문장 형식으로 자동 변환된다.

4.4 RESO Ontology

전술한 바와 같이 RESO Ontology는 지식관리자가 배송과 관련되어 있다고 판단되는 개념들의 상하관계와 동의어 및 유의어 관계를 사전에 식별해 구축한 것으로써, 그림 4와 같은 구조를 가지고 있다. RESO Ontology를 지식공학자가 구축했기 때문에 사용한 클래스와 클래스들 간의 관계 설정이 개인적인 관심도에 따라 편향될 수도 있다. 예를 들어, 지식공학자가 특정 클래스의 대표어로 European_address를 선택한 후 Europe과 European_country을 동의어로 지정했다고 하자. 그러나 실제로 지식공학자들이 RESO-RULE을 이용하여 규칙을 습득하는 과정에서 European_address보다 Europe가 더 많이 선택한 변수인 경우, RESO Ontology를 수정하여 Europe을 대표어로 그리고 European_address를 동의어로 지정해야 한다는 것이다. 이를 위해, 기존의 온톨로지에 특정 클래스가 출현할 때마다 그 출현 빈도를 데이터베이스에 저장한다. 이후, 데이터베이스에 저장되어 있는 대표어와 그 동의어들의 출현 빈도를 비교하여, 출현 빈도가 가장 많은 클래스를 대표어로 사용한다는 것이다. 그림 7은 클래스는 RESO 온톨로지의 대표어 변경과정을 나타내고 있다.

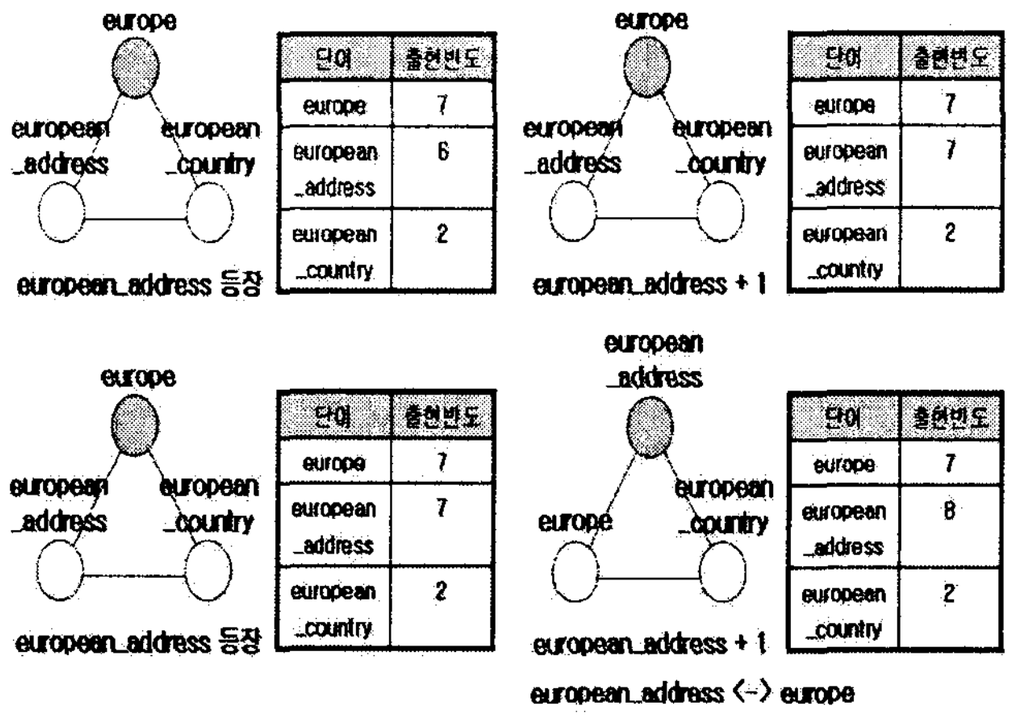


그림 7 - 출현빈도에 따른 대표어 변경 과정

이상과 같이 온톨로지의 대표어를 변경하는 전 과정은 알고리즘은 그림 8의 find_Representative Class을 사용하여 지원한다.

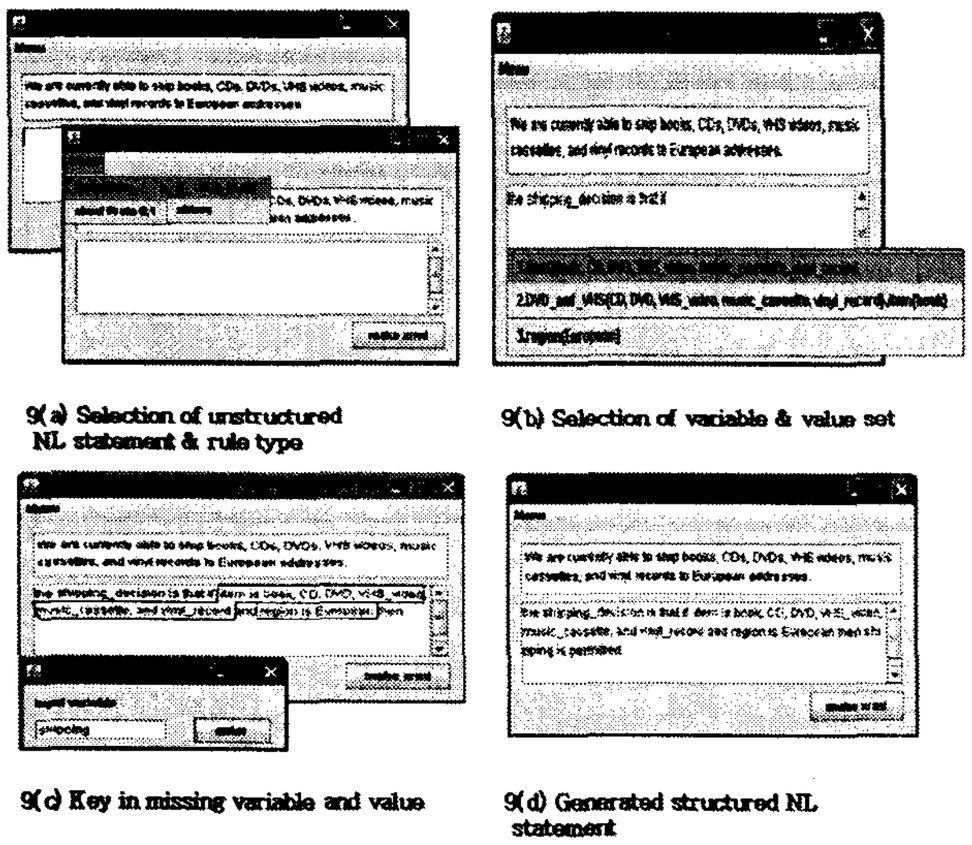
```
Function find_RepresentativeClass
  if currentClass is synonym
    find currentClass.count in classListWithCount
    find PC.count in classListWithCount
    currentClass.count ++
    if currentClass.count > RepresentativeClass.count
      currentClass <-> RepresentativeClass
    return RepresentativeClass
  else
    currentClass.count ++
  return currentClass
end
```

그림 - 8 Pseudo code for find_RepresentativeClass Algorithm

5. 프로토타입 구현: RESO-RULE

본 논문에서 제안하고 있는 프로토타입인 RESO-RULE은 JDK 1.5를 기반으로 JAVA CC, JAVA NLP, Jena 및 JDOM 등을 이용하여 개발 중에 있다. 분석의 대상으로 선택한 다수의 비구조적인 자연어 문장들은 웹사이트에 공개되어 있는 배송 관련 문장들이며, 온톨로지는 Protégé를 이용하여 구축하였다. 그림 9는 프로토타입 시스템의 예제 스크린이다. 본 시나리오에서 지식관리자는 amazon.com을 포함된 온라인 서점의 배송 정책을 규칙베이스로 생성하기 위해, 웹 서핑을 통해 비구조적인 자연어 문장들을 수집하였다. 그림 9(a)는 수집된 비구조적인 자연어 문장 중 하나를 Unstructured NL statement Reader에 탑재한 화면이다. 전술한 바와 같이 비구조적인 자연어 문장에 규칙이 포함되어 있는지 여부를 판단하는 주체는 지식관리자이다. 다음

단계는 비구조적인 자연어 문장에 포함되어 있는 규칙의 형태를 판단해야 한다. 규칙의 형태는 제한된 언어집합 기법을 통해 사전에 정의했기 때문에 RESO-RULE Editor가 지원하는 형식 중 선택을 하게 된다. 그림 9(a)에서 보는 바와 같이 현재 RESO-RULE Editor가 생성할 수 있는 것은 If-Then 형식의 규칙이다. 다음 단계에서는 선택된 규칙의 형태에 맞춰 단계적으로 구조적인 자연어 문장을 생성한다. 이때 If-Then 형식의 규칙에 필요한 변수와 그 값들은 4.3.1절에서 설명한 과정을 거쳐 식별하게 된다. 그림 9(b)는 표 1을 에디터로 구현한 것이다. 그림 9(c)의 에디터 화면에 표시된 붉은색 사각형이 그림 9(b)의 과정을 거쳐 생성된 것이다. 그림 9(c)는 비구조적인 자연어 문장에 명시적으로 나타나 있지도 않고, 온톨로지에도 포함되어 있지 않은 변수들을 처리하기 위해 지식관리자가 직접 입력하는 것을 보여주는 것이다.



9(a) Selection of unstructured NL statement & rule type

9(b) Selection of variable & value set

9(c) Key in missing variable and value

9(d) Generated structured NL statement

그림 9 - RESO-RULE 프로토타입의 예제 화면

이상의 과정을 모두 거치고 나면, If-Then 형식을 완전하게 갖춘 구조적인 자연어 문장을 생성하게 된다. 생성된 비구조적인 자연어 문장이 그림 9(d)에 나타나 있다. 이후 ' make xrml' 이라는 버튼을 누르게 되면 다음과 같은 XRML 규칙이 생성된다. XRML 형식의 규칙은 그림 10에 도시하였다.

```
<XRML>
  <RuleTitle>shipping_decision</RuleTitle>
    <IF>
      <or>
        <Item>book</Item>
        <Item>CD</Item>
        ……
        <Item>vinyl_record</Item>
      </or>
    </IF>
    <THEN>
      <Shipping>permitted</Shipping>
    </THEN>
</XRML>
```

그림 10 - 생성된 XRML 형식의 규칙

6. 평가

RESO-RULE의 규칙 식별능력을 평가하기 위해 amazon.com과 유사한 온라인 서점인 barns & novels와 powells.com의 배송 정책 문서로부터 비구조적인 자연어 문장을 수집한 후, 전 절에서 제안한 절차에 따라 규칙을 식별하면 표 2와 같다.

표 2 - 기존의 XRML 연구와 RESO-RULE의 비교

	규칙을 포함하고 있다고 식별된 자연어 문장	파싱된 단어집합	구조화된 자연어 문장
Barns&Nobles	If your order includes Used & Out of Print Books, Gift Cards, PC & Video Games, Prints and Posters, or specially designated oversized items, please see the appropriate shipping charts below	{Used & Out of Print Books, Gift Cards, PC & Video Games, Prints and Posters, specially designated oversized items } {shipping chart}	The international_shipment _decision is that if item is **Used_Out_of_Print_Book, PC_ Video_Game, Print_Poster, specially_designated_oversized_ Item** then delivery_policy is **shipping_chart**
Powells.com	Import duties are the responsibility of the recipient	{import duty} {responsibility} {recipient}	The import_duty_policy is that if duty_type is **import_duty** then taxpayer is **recipient**
	Shipping time in business days (weekdays) after your order is processed.	{shipping time} {business days} {weekdays} {order} {processing	The shipping_time_dicision is that if current_day is **business_day** and customer_action is **order** *and* order_phase *is done* then shipping_time *is started.*
	Books ordered from our Chalmers Warehouse ship separately.	{book} {order} {Chalmers Warehouse} {ship}	The international_shipment _decision is that if item is **book** and order_place is **Chalmers_Warehouse** then shipping_method is *separated_delivery*

위 표에서 ' 구조화된 자연어 문장' 열에 나타난 밑줄 친 단어는 온톨로지에 명시된 변수를 의미하고, 볼드체로 표시된 단어는 문장에 포함되어 있는 변수 값을 의미하며, 이탤릭체로 표시된 단어는 문장에 포함되어 있지 않아 실제로 식별할 수 없는 변수 값을 표시하고 있다. 결과적으로 이탤릭체로 표시된 단어를 포함하고 있는 If 파트나 Then 파트는 본 도구를 사용해서 자동적으로 생성할 수 없는 부분이다. 또한 붉은 색 사각형이 표시된 Then 파트는 온톨로지와 파싱된 단어집합을 이용해

문장은 생성했으나, 의미가 정확하게 전달되지 않고 있음을 알 수 있다. Amazon.com을 비롯해 3개의 인터넷 서점에서 국제배송과 관련되어 5개의 자연어 문장을 선택해 RESO-RULE을 운영한 결과를 표 3에 요약하였다.

표 3 - RESO-RULE을 이용한 규칙식별 결과

	총 문장 수	완전히 구조화된 문장 수	완전히 구조화되었으나 의미가 변질된 문장 수	완전히 구조화되지 않은 문장 수
규칙이 포함되어 있는 서술형 문장	3	2	1	
비구조화된 문장에 포함된 변수 값의 수	-	3	2	
구조화된 문장에 포함된 변수 값의 수	-	3	2	
규칙이 포함되어 있는 개조식 문장	2	-		2
비구조화된 문장에 포함된 변수 값의 수	-	-	-	2
구조화된 문장에 포함된 변수 값의 수	-	-	-	4

이들 문장 중 RESO-RULE을 이용해 완전한 If-Then 규칙으로 변환된 것은 서술형 문장 3개였고, 이들 중 두 개의 문장은 의미의 변질없이 구조화된 자연어 문장으로 변환되었으며, 하나는 변환과정에서 의미의 변질이 일어 났다. 나머지 두 개의 자연어 문장은 문장의 구조가 완전하지 않은 개조식 문장으로서, 축약된 표현을 특징으로 갖는 개조식 문장의 특성상 변수 값 자체가 생략된 것이 있어 완전한 형태의 생성하지 못했다. 현재까지의 분석 결과를 요약하면 규칙이 포함되어 있는 서술형 문장의 자동적인 식별은 어느 정도 가능하지만, RESO-RULE이 채택한 파서가 서술식 문장의 분석을 위해 개발된 것이기 때문에 개조식 문장에 포함되어 있는 규칙은 식별하지 못했다.
본 실험은 5개의 문장만을 대상으로 했다는 점과 온톨로지에 모든 변수들이 포함되어 있음을 가정하고 수행했다는 점 등을 한계로 가지고 있으며, 이를 보완하기 위한 추가적인 연구와 실험이 진행 중이다.

7. 기존 연구와의 차별성

RESO-RULE은 제한된 언어집합을 이용해 규칙을 내포하고 있는 웹 문장으로부터 규칙을 식별한 후 구조화하는 과정을 지원하는 도구로써, 지식관리자가 규칙 생성도구에 대한 학습이 없이도 규칙을 생성할 수 있으며, 온톨로지를 활용함으로써 좀 더 정확한 규칙의 생성이 가능하다는 장점을 가지고 있다. 본 연구의 장점을 기존의 XRML 관련 연구들과 비교하면 다음과 같다.
비정형화된 웹 문서에 내포되어 있는 지식을 식별할 수 있는 언어인 XRML이 Lee와 Sohn[7]에 의해 제안된 후, 본 연구를 포함해 몇 가지의 후속 연구가 수행된 바 있다. 첫 번째 후속연구인 XRML 2.0에서는 웹 문서의 표(table)에 포함되어 있는 규칙의 식별과 함께, 다양한 operator의 사용을 가능하게 하였고 [8], OntoRule은 온톨로지를 이용해 규칙 식별 과정을 지원하였다[10]. 이러한 연구를 통하여 XRML의 개념이 더욱 발전되었으며, 본 연구 또한 기존 연구의 한계를 극복함으로써 XRML을 한 단계 더 발전시키는 데 기여할 것이다.
기존의 OntoRule을 제외한 XRML 연구와 RESO-RULE에서는 규칙을 포함하고 있는 비구조적인 자연어 문장의 선정 주체가 지식관리자를 포함한 사람이므로, 사람이 규칙을 내포하고 있는 모든 비구조적인 자연어 문장을 찾아낼 수 있다고 가정한다면 모든 비구조적인 문장은 찾아질 수 있다. 그러나 온톨로지를 이용하는 경우는 온톨로지의 완전성에 따라 달라질 수 있다. 두 번째, 지식관리자가 이해할 수 있는 구조적인 자연어 문장으로의 변환은 기존 XRML 연구에서는 전혀 고려되지 않은 부분이다. 구조적인 자연어 문장으로의 변환이 필요한 이유는 사람과 컴퓨터간의 협력을 원활하게 하기 위해서이다. 사람과 컴퓨터가 협력적으로 문제를 해결하기 위해서는 규칙의 형식과는 무관하게 동일한 의미를 갖는 규칙을 공유해야만 한다. 기존의 XRML 연구에서는 사람은 웹 문서 그대로의 비구조적인 자연어 문장을, 그리고 컴퓨터는 XML 형식의 규칙을 공유하도록 설계한 것에 비해, RESO-RULE은 If-Then 형식의 구조적인 자연어 문장을 지식관리자에게 제공함으로써, 규칙에 재한 검증 및 개인차에 의해 의미가 달라질 수 있는 상황을 완전히 배제하였다. 또한 RESO-RULE은 지식관리자가 규칙생성에 필요한 도구를 별도로 학습하지 않아도, 본 방법에서 지식하는 절차를 따르기만 하면 규칙이 습득되도록 하였다. 그러나 기존의 연구들에서는 지식관리자가 비구조적인 자연어 문장에 rule과 관련되어 있다고 판단되는 부분을 식별한 후, 규칙 생성에 필요한 다양한 예약어, 예를 들어 If, Then, and 및 or 등을 수동으로 삽입하도록 되어 있다. 마지막으로, 비구조적인 자연어 문서나 이미 생성된 규칙이 환경의 변화나 지식관리자의 학습에 의해 변경 요인이 발생한 경우, 변화를 손쉽게 반영할 수 있어야 하는 변경에 대한 적응성은 온톨로지를 사용한 OntoRule과 RESO-RULE만이 가질 수 있는 특징이다. 이상의 분석 결과를 요약하면 표 4와 같다.

표 4 - 기존의 XRML 연구와 RESO-RULE의 비교

	습득의 완전성	표현의 다양성	변환의 용이성	변경에의 적응성
XRML 1.0	O	X	X	X
XRML 2.0	O	X	X	X
OntoRule	X	X	X	O
RESO-RULE	O	O	O	O

그러나 표 2는 본 연구에서 주장하는 비구조적인 웹 문서로부터의 지식 획득을 지원하는 도구와

방법론들이 가져야 할 특징을 중심으로 하는 제한적인 비교 결과이기 때문에, 또 다른 요소를 기준으로 분석하면 다른 결과가 도출될 수도 있다.
XRML 이외에 RESO-RULE은 제한된 언어집합 연구들과도 관련이 있다. Thompson과 Panandak[11]은 메뉴에 기반한 자연어 처리 방법인 LingoLogic을 개발하였다. LingoLogic은 처리 가능한 자연어의 범위를 임의의 도메인에 맞도록 줄였으며, 메뉴방식을 통해 자연어를 처리하였다. Ginseng은 자연어를 시맨틱 웹 쿼리 언어인 RDF Data Query Language(RDQL)로 변환하기 위한 시스템의 유도 방식의 검색엔진이며[3], GINO에서는 온톨로지의 클래스와 속성들을 추가하기 위해 시스템 유도 기반의 제한된 언어집합 기법을 활용하였다[4]. 기존 연구의 대상이 웹 문서 포함된 규칙을 식별하는 것이 아니라는 것과, 온톨로지를 이용해 규칙의 생성을 지원하지 않는다는 점에서 RESO-RULE은 기존의 연구들과 차별화된다.

8. 결론 및 향후 연구 과제

본 연구에서는 제한된 언어집합을 이용해 비구조적인 자연어 문장으로부터 규칙을 생성하는 과정을 지원해 줄 수 있는 방법론을 제안한 후, 프로토타입 시스템을 개발하였다. 본 연구가 규칙 습득 방법에 기여한 바를 요약하면 다음과 같다.

- 제한된 언어집합을 이용해 규칙 생성과정을 지원함으로써, 지식관리자가 규칙생성에 필요한 도구에 대한 학습없이도 규칙을 생성할 수 있도록 하였다.

- 컴퓨터가 처리할 수 있는 전형적인 규칙(canonical rule)으로 변환하기 전 단계에서 구조화된 자연어 문장으로 변환을 해줌으로써 규칙 생성 과정에서 발생할 수도 있는 오류를 사전에 제거할 수 있도록 하였다.

- 또한 지식 관리자의 행태를 반영하여 온톨로지를 지속적으로 개선함으로써, 사용자 친화적인 규칙을 생성할 수 있도록 하였다.

- 이상의 과정을 지원할 수 있는 프로토타입 시스템을 개발 중에 있다.

이러한 장점은 기존에 수행되었던 XRML 관련 연구를 진일보 하는데 기여할 것으로 판단되며, 이를 뒷받침하기 위해 기존의 XRML 관련 연구들과 RESO-RULE을 본 논문에서 제시한 비구조적인 웹 문서로부터의 지식 획득을 지원하는 도구들이 가져야 할 특징의 관점에서 비교하였다. 또한 RESO-RULE의 지식 식별 능력을 보이기 위해 몇 개의 인터넷 서점들이 제시하고 있는 배송 정책과 관련된 문서로부터 규칙이 습득되는 정도를 보여주었다.
그러나 현재는 서술형 문장에 포함되어 있는 If-Then 형식의 규칙만을 식별할 수 있으며, 프로토타입 시스템을 위한 제한적인 온톨로지만을 구축했다는 것 그리고 발생할 수 있는 다양한 예외사항을 처리하지 못하는 점 등이 추후 해결 과제로 남겨놓고 있다.

참고문헌

[1] Alexander Gelbukh, Natural Language Processing. Proceedings of the Fifth International Conference on Hybrid Intelligent Systems (HIS' 05), 2005.
[2] Berners-Lee, T., Hendler, J. and Lassila, O.. The Semantic Web, Scientific American, 2001.
[3] Bernstein, A., Kaufmann, E., Kaiser, C.. Querying the Semantic Web with Ginseng: A Guided Input Natural Language Search Engine. 15th Workshop on Information Technology and Systems (WITS 2005). 2005.
[4] Bernstein, A., Kaufmann, E.. GINO - A Guided Input Natural Language Ontology Editor. 5th International Semantic Web Conference, Athens, GA, USA, November 5-9, 2006.
[5] Byrd, T.A. Expert Systems Implementation: Interviews with Knowledge Engineers. Vol.33, No.10, 1995.
[6] Hulth, A., Karlgren, J., Jonsson, A., Bostrom, H., Asker, L.. Automatic Keyword Extraction using Domain Knowledge. Proceedings of the Second Computational Linguistics and Intelligent Text Processing, Mexico, 2001.
[7] Kang, J., Lee, J.K.. Rule Identification from Web Pages by the XRML approach. Decision Support Systems, Vol 41, , 2005.
[8] Lee, J.K., Sohn, M.. Extensible Rule Markup Language - toward Intelligent Web Platform, Communications of the ACM 46, 2003.
[9] Napier H. Albert, David M. Lane, Richard R. Batsell, and Norman S. Guadango. Impact of a Restricted Natural Language Interface on Ease of Learning and Productivity. Communications of ACM,. Vol. 32, 1989.
[10] Park Sangun and Jae Kyu Lee. Rule identification using ontology while acquiring rules from Web pages, International Journal of Human-Computer Studies, In Press, Corrected Proof, Available online 7 March 2007,
[11] Thompson, C.W., Pazandak, P. Tennant, H.R.. Talk to your semantic Web. IEEE Internet

Computing, Nov.-Dec. 2005.
[12] Turban Efraim, Jay E. Aronson, Ting-Peng Liang. Decision Support Systems and Intelligent Systems (7th Edition), Wiley,2005.