

시멘틱 웹 기반 DQL 검색 시스템 설계

김 제 민[†] · 박 영 택^{††}

요 약

인터넷의 등장 이후 폭발적으로 증대되는 웹 정보를 효율적으로 사용하기 위한 방안이 연구 되고 있다. 현재, 웹 서비스에서 사용하고 있는 키워드 기반 검색은 syntactic 정보만을 제공하므로 자신이 원하는 서비스를 받고자하는 사용자의 의도와는 상관없는 정보를 검색하여 제공하는 경우가 많다. 웹 서비스의 이러한 단점을 보완 하고자 시멘틱 웹(semantic web)기술이 제안되고 있는데, 시멘틱 웹을 이용하면 사용자가 원하는 의미와 일치하는 정보를 찾아줄 수 있다. 시멘틱 웹을 기반으로 만들어진 시스템은 DQL (Daml Query Language)을 이용한 시멘틱 검색(Semantic search)기법을 이용하고 있다. 시멘틱 검색은 사용자가 입력한 질의들을 해석하여 사용자의 원하는 정보와 의미가 같은 해답을 얻어내는 것이다. 이에 본 논문에서는 효율적인 정보 검색을 위해 시멘틱 웹 기반의 검색 시스템을 설계하는데 목적을 가지며, 효과적인 검색 시스템을 설계하기위해 사용자 중심의 키워드 형태 질의를 시스템 중심의 DQL로 변환하는 RDQL 변환기와 추론엔진을 불러내어 추론을 실행하는 RDQL 엔진을 제안한다.

Semantic Web based DQL Search System

Je-Min Kim[†] · Young-Tack Park^{††}

ABSTRACT

It has been proposed diverse methods to use web information efficiently as the size of information is increasing. Most of search systems use a keyword-based method that mostly relies on syntactic information. They cannot utilize semantic information of documents and thus they could generate to users. To solve shortcoming in searching documents, a technique using the Semantic Web is suggested. A semantic web can find relevant information to users by employing metadata which are represented using standard ontologies. Each document is annotated with a metadata which can be reasoned by agents. In this paper, we propose a search system using semantic web technologies. Our semantic search system analyzes semantically questions that user input, and get resolution information that user want. To improve efficiency and accuracy of semantic search systems, this paper proposes DQL(DAML Query Language) engine that employs inference engine to execute reasoning and DQL converter that changes keyword form question of the user to DQL.

키워드 : 시멘틱 검색(Semantic Search), DQL(Daml Query Language), 웹 서비스(Web Service), 시멘틱 웹(Semantic Web), 전방향 추론(Forward Chaining)

1. 서 론

현재 대부분의 웹 문서는 HTML로 작성되어 있다. HTML 문서는 사람이 웹 브라우저를 통해 문서의 의미를 이해하기는 쉽지만 소프트웨어 에이전트가 스스로 문서의 의미를 파악하기 어렵다. 이와 같은 이유 때문에 웹 문서에 대한 정보를 검색하면, 검색 결과의 정확도가 저하되며 에이전트간의 정보를 공유하기가 어려워진다. 시멘틱 검색의 궁극적인 목적은 시멘틱 웹 기술을 바탕으로 사용자가 원하는 정보와 그 의미가 같은 정보들을 검색하는 것이다. 시멘틱 검색은 사용자가 원하는 정보들의 의미를 파악하기 위한 한가지 방

법으로 RDF(Resource Description Framework) 트리플 구문을 기반으로 사용되는 RDQL을 이용한다. 이러한 RDQL기반의 시멘틱 검색 기술은 사용자가 복잡한 RDQL 질의 문법들을 미리 학습한 후, 검색 질의를 수작업으로 입력하기 때문에 일반 웹 검색에서 제공하는 키워드 검색에 비해서 사용하기에 많은 어려움이 따른다. 또 RDQL(RDF Data Query Language)를 사용하여 검색을 실행하는 RDQL 검색 엔진에 입력된 질의문과 지식 베이스에 저장된 각 메타데이터들을 단순 매치하여 질의에 맞는 결과를 검색한다. 이와 같은 경우, 질의 조건이 복잡해지거나, 메타데이터의 양이 많아지면 검색 시간이 많이 소요될 수 있고, 철자는 다르지만 의미가 같은 동의어들을 가지는 정보들을 제대로 검색하지 못하므로 검색 정확도가 저하될 수 있다.

본 논문은 키워드로 입력 받은 사용자의 검색 질의를

[†] 준 회 원 : 숭실대학교 대학원 컴퓨터학과

^{††} 정 회 원 : 숭실대학교 컴퓨터학과 교수

논문접수 : 2004년 8월 25일, 심사완료 : 2005년 2월 1일

RDQL 변환기를 이용하여 RDQL 질의문으로 자동으로 변환함으로써 사용자가 원하는 정보를 쉽게 검색할 수 있도록 도움을 주고, 검색 과정에서 온톨로지 기반의 시멘틱 추론을 실행함으로써 검색 정확도를 높일 수 있는 시스템을 설계하는 것이 주된 목적이다. 이에 본 논문에는 시멘틱 검색 시스템의 사용 효율성과 검색 정확도를 높이기 위해서 시멘틱 웹 기반의 DQL 검색 시스템을 제안한다.

본 논문에서 제안하는 시스템은

- 첫째, 복잡한 DQL형식의 질의문을 사용자가 키워드 형태로 쉽게 입력하기 위한 질의문 인터페이스
- 둘째, 사용자 중심의 키워드 형태의 질의를 시스템 중심의 RDQL로 변환하는 RDQL 컨버터
- 셋째, 입력된 질의를 처리하여 정보를 검색한 후 출력해주는 RDQL 엔진
- 넷째, RDQL 엔진이 추론 엔진을 호출하여 시멘틱 추론을 실행하는 RDQL-추론엔진 인터페이스로 구성된다.

본 논문의 구성은 다음과 같다. 2장에서는 본 연구에 기초가 되는 기본 개념과 관련 연구에 관하여 설명하고, 3장에서는 전체적인 시스템을 설명한다. 이러한 시스템을 구성하기 위해 4장에서는 RDQL 컨버터, 5장에서는 RDQL 엔진과 RDQL-추론 엔진 인터페이스, 6장에서는 시멘틱 질의를 위한 사용자 RDQL 검색 인터페이스, 7장에서는 질의 메뉴 생성기에 대해서 자세히 설명하도록 한다. 8장에서는 가상의 메타데이터를 기반으로 실시된 실험을 통해서 본 논문에서 제안하는 DQL 검색 시스템의 효율성과 검색 정확도에 대해 기술한다. 끝으로 9장에서는 결론 및 향후연구 방향에 대해 기술한다.

2. 관련 연구

시멘틱 웹은 웹에 존재하는 문서나 정보의 의미를 표현하기 위하여 DAML+OIL과 OWL을 이용하여 온톨로지를 구축하고, 이를 바탕으로 각 문서의 핵심 정보들을 메타 데이터로 생성한다. RDQL 검색 엔진은 RDF 트리플 구문으로 변환된 메타 데이터들을 기반으로 사용자가 입력하는 질의에 대한 결과와 같은 의미를 지니는 정보를 검색한다. 본 절에서는 본 논문에서 제안한 시멘틱 웹 기반 DQL 검색 시스템을 구성하기 위한 관련된 기본 개념과 관련 연구를 소개한다.

2.1 DAML+OIL / OWL

DAML+OIL은 DAML(DARPA Agent Markup Language) 프로그램의 DAML-ONT와 주로 유럽에서 개발된 OIL(Ontology Inference Layer)의 결합을 통하여 만들어진 웹 온톨로지 언어이다. DAML+OIL은 객체지향적인 방법을 사용하여 관심영역(domain)의 구조를 클래스(class)와 속성(attribute) 및 속성 값(value)을 써서 표현한다. 또 XML 스키마의 모든 데이터형을 지원하여 문자열, 십진수, 실수, 정

수 범위 등을 사용할 수 있다[1]. DAML+OIL은 RDF와 밀접하게 연관되어 있는 동시에 기본적으로 표현력을 강화된 Description logic 구조를 띤다. 그러므로 Description logic의 추론 능력과 강화된 표현력을 가지고 있기 때문에, 이러한 장점을 DAML+OIL로 작성된 메타데이터를 기반으로 하는 시멘틱 웹 서비스에 활용할 수 있다[2,3].

W3C에서는 시멘틱 웹의 구현을 위해 또 다른 웹 온톨로지 언어로서 OWL(Web Ontology Language)을 제안하고 있다. OWL 역시 웹상에서 온톨로지를 표시하고 공유하기 위한 시멘틱 마크업 언어(Semantic Markup Language)다 [4]. OWL은 논리적인 관계를 표현하는 기능이 포함되어 있기 때문에 추론을 통한 고급 정보의 표현을 위해서는 OWL의 사용이 필요하다. 또 OWL에서는 여러 가지 속성에 대해 그 특성을 기술해주는 어휘들을 제공하기 때문에, 속성을 제한하거나, 속성들 간의 논리적인 관계, 클래스들 간의 논리적인 관계를 사용하여 온톨로지를 구성하면 웹상에서의 여러 가지 정보들의 복잡한 관계들을 표현하는데 유용하다.

2.2 JTP 및 DQL

JTP(Java Theorem Prover)는 스탠포드 대학에 소속되어 있는 지식시스템 연구소의 Gleb Frank에 의해 개발된 객체 지향 추론 시스템이다. JTP는 DAML+OIL과 OWL로 작성된 온톨로지를 KIF(Knowledge Interchange Format) 형식으로 읽어 들여 추론을 할 수 있도록 개발 되었다. 또 KIF형태로 메타데이터가 저장된 지식베이스에 질의를 보내면 검색된 결과를 얻을 수 있는 인터페이스를 제공한다[5].

United States/European Union ad hoc Agent Markup Language Committee에 의해 개발된 DQL(DAML Query Language)은 시멘틱 웹 환경에서 에이전트가 서로에게 질의를 던져서 응답을 받기위해 만들어진 언어다[6]. 즉 응답을 도출해내기 위해 시멘틱 웹상의 지식 베이스를 기반으로 여러 질의들과 응답들 간의 의미적인 연관 관계(Semantic Relationship)를 정밀하게 분류한다. DQL이 데이터 베이스 시스템을 위한 표준 질의 언어(SQL)나 웹에서 정보를 검색하는 언어(XML query)와의 가장 큰 차이점은 JTP와 같은 자동 추론 엔진과 연결하여 시멘틱 웹상에 존재하는 다중 지식 베이스에서 질의에 대한 답을 도출할 수 있다는 것이다.

2.3 RDF

RDF(Resource Description Framework)는 W3C의 가장 기본적인 시멘틱 웹 언어로서 시멘틱 웹의 구현을 위한 최하위 계층의 역할을 담당하며 웹에 있는 자원에 관한 메타정보를 표현하기 위한 언어다. 또한 에이전트 같은 프로그램이 처리할 수 있는 간단한 데이터 모델과 표준화된 문법을 제공한다. RDF의 기본적인 데이터 모델은 Subject, Property, Object 타입으로 구성된다. 이렇게 RDF가 구조화된 메타 데이터를 생성해줌으로써 검색 엔진이 웹상에서 보다 정확히 정보를 검색할 수 있는 기능을 제공해준다. 또 정보의 의미(semantics)가 자원(Resource)과 그 특성 값으로 표현되므로

같은 내용에 대해서는 해석이 하나로만 정의되기 때문에 기존의 XML이 가지는 문제점을 해결하고 있다[4].

2.4 DQL Server

DQL Server는 스탠포드 대학 지식 연구소에서 연구한 서버 시스템으로써 DQL을 사용하여 질의와 응답을 제공한다 [7]. DQL Server는 각 클라이언트의 질의를 받아서 적절한 응답을 해주기 위해 XML 문법을 사용하여 서버에 질의와 응답에 대한 태그들을 추상적으로 명세한다. DQL Server는 서로 상호작용을 하는 웹 서버(front-end)와 DQL 문법을 사용하여 질의를 보내고 응답을 받는 클라이언트(Back-end)서버로 구성된다. 질의를 보내기 위한 메시지는 크게 query, queryPattern, answerKBPattern 부분으로 나뉘어 지는데, query는 DAML문서를 명료하게 표현하기 위해 각각의 URL을 정의하고, query Pattern은 각 질의를 RDF 구문으로 구체화하여 정의하며, answerKBPattern은 정의된 질의에 대한 응답을 찾아내기 위해 사용되는 DAML +OIL 문서의 URI를 명세 한다. 응답을 보내기 위한 메시지는 크게 queryPattern, answer, continuation 부분으로 나뉘어 지는데, queryPattern은 DQL 서버에 들어온 질의 메시지를 반환함으로써 DQL 서버에 보내진 메시지를 확인할 수 있고, answer는 DQL 서버가 보내준 응답 정보를 각각 binding set에 할당하여 질의에 대한 응답정보를 정의하며, continuation은 더 이상 응답할 질의가 없거나, 질의에 대한 응답 정보가 존재하지 않을 경우 (failed), 마지막 응답 정보에 none이라는 termination-token을 할당하여 응답 메시지가 끝났음을 표시한다.

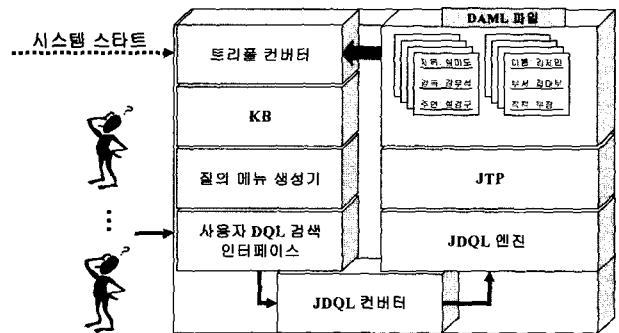
3. 시맨틱 웹 기반 DQL 검색 시스템

본 논문에서는 시맨틱 검색 시스템의 쉬운 사용과 검색 정확도를 높이기 위하여 시맨틱 웹을 기반으로 DQL 검색 시스템을 구축하기 위한 방안을 제시한다. DQL 검색 시스템의 핵심은 사용자가 입력한 키워드를 정확한 RDQL 질의문으로 자동생성하고, 추론 엔진의 추론기능을 적절히 활용하여 검색 정확도를 높이는데 있다.

본 논문에서 제안하는 DQL 검색 시스템은, 사용자 질의를 키워드 형태로 쉽게 입력하기 위한 사용자 중심의 사용자 RDQL 검색 인터페이스, 입력된 키워드 형태의 질의를 검색 엔진이 이해할 수 있는 RDQL 형태로 변환하는 RDQL 변환기, 추론을 실행하는 추론엔진을 호출하여 검색 정보들의 속성과 관계를 추론하고 사용자가 원하는 정보를 검색해주는 RDQL 엔진, 사용자가 선택할 질의어를 자동 생성하는 질의 메뉴 생성기로 구성된다.

첫째, 사용자 RDQL 검색 인터페이스는 시맨틱 검색 시스템을 이용하는 사용자의 검색어를 키워드 입력기를 통해 입력 받고, 검색된 결과를 사용자에게 보여준다. 이때 키워드 입력기는 단순한 키워드만을 입력받는 것이 아

니라 사용자에게 원하는 정보를 검색해 주기위해 입력된 키워드를 통해 사용자의 의도를 이해해야 한다. 이를 위해 질의 메뉴 생성기를 이용하여 검색 도메인의 온톨로지와 그 구조가 일치하도록 온톨로지를 분석하여 동적으로 질의 메뉴를 구성한다. 동적으로 생성된 질의 메뉴를 통하여 사용자가 입력한 키워드는 RDQL 변환기 시스템에서 처리할 수 있는 데이터로 추상화된다. 또 검색 과정을 로그 파일로 만들어서 각 사용자가 이전에 검색했던 정보를 정확히 기록하는 기능을 수행한다. 생성된 로그파일을 바탕으로 시스템 구축자는 검색 과정을 파악할 수 있기 때문에 시스템에 오류를 수정하고 성능을 파악하는데 용이하며, 검색 과정 등이 상세하게 기록되어 있기 때문에 사용자 입장에서 같은 검색을 반복하지 않도록 할 수 있다.



(그림 1) 전체적인 시스템 구조도

둘째, RDQL 변환기는 사용자 RDQL 검색 인터페이스에서 추상화하여 전송한 사용자의 검색 정보 데이터로부터 정확한 RDQL 질의문을 생성하는 기능을 한다. 본 논문에서는 효과적인 질의를 위해서 SQL과 유사한 문법을 가진 RDQL질의 언어를 사용한다. RDQL은 DAML API가 내장되어 DAML 온톨로지 작업이 가능한 Jena에서 제공된다. RDQL 변환기는 정확한 질의문 생성을 위하여 지식 베이스(Knowledge Base)에 메타데이터로 저장되어 있는 온톨로지 구조를 파악하여, 형식에 맞는 질의문을 생성한다.

셋째, RDQL 엔진은 RDQL 변환기에서 생성한 질의문을 받아들인다. RDQL 형식의 질의문은 Subject, Property, Object순으로 짜여진 트리플 구문을 바탕으로 구성되어 있다. 정보를 검색하기 위해 사용되는 메타데이터를 구성하는 트리플 구문들은 실제로 그 양이 방대하기 때문에 입력된 질의문과 모든 트리플 구문들을 매치할 수 없다. 이와 같은 문제를 해결하기 위해서 트리플 형태를 갖는 각각의 질의 문장을 규칙이 적용된 추론 엔진에 보내어 추론을 실행한 후, 추론을 통해 정제된 트리플 구문만을 바탕으로 검색을 실행한다. 또한 검색 정보들의 속성과 관계를 추론하여 검색의 정확도를 높인다. 본 논문에서는 DAML+OIL로 만들어진 메타 데이터를 KIF(Knowledge Interchange

Format) 형식으로 읽어 들여 추론을 할 수 있도록 개발된 JTP를 적극 활용하였다.

마지막으로 질의 메뉴 생성기는 검색 도메인의 온톨로지를 분석하여 자동으로 질의어들을 생성해준다. 미리 생성된 질의어들을 이용함으로써 사용자에게 효과적인 질의어 입력을 유도함으로써 부정확한 키워드 입력에 따른 검색률 저하를 예방한다. 질의어에 사용되는 부분은 트리플문 중에서 Property 부분이기 때문에, 각각의 property를 저장하고, property에 대한 값들을 따로따로 저장하여 형식에 맞는 질의어를 생성한다.

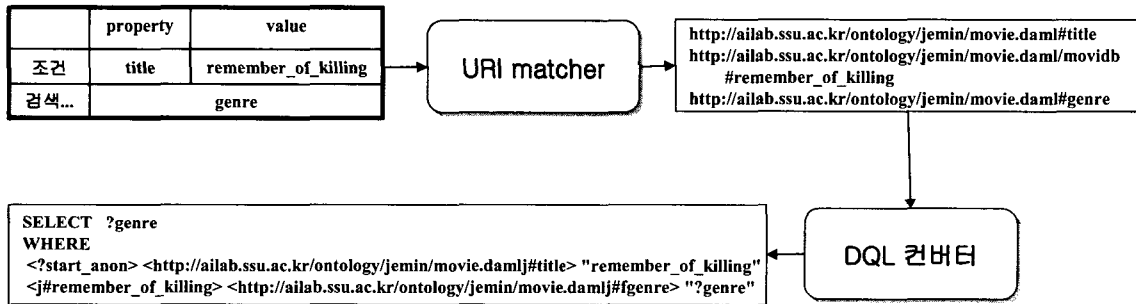
4. RDQL 변환기

본 논문에서 제안하는 사용자 RDQL 검색 인터페이스는 사용자의 질의를 키워드로 입력받도록 설계되었다. 사용자의 편의를 고려하여 설계된 입력 인터페이스에서 입력된 키워드 정보는 일반적으로 시멘틱 검색에서 사용하기에는 어려움이 따른다. 때문에 사용자가 입력한 키워드 형태의 질의어를 시멘틱 검색 시스템에서 인식하여 검색을 실행할 수 있도록 RDQL형태로 적절히 변환하는 변환기가 필요하다.

본 논문의 RDQL 엔진에서 검색을 위해 사용하는 DQL은 온톨로지 구조를 바탕으로 구축되어진 트리플 구문 형식의 메타데이터에 SQL과 유사한 문법을 사용하여 정보를 검색하는 RDQL 질의 언어를 사용하였다. Jena는 DQL을 이용하여 검색을 하기위해 DAML+OIL문서를 트리플로 변환하여 지식 베이스에 메타 데이터로 저장한다.

RDQL은 크게 WHERE 구문, SELECT 구문, AND 구문, USING 구문으로 구성된다. RDQL의 WHERE Clause는 트리플과 같이 Subject, Property, Object순의 형태로 자신의 검색 조건을 표현한다. 만약 사용자가 object에 해당하는 어떤 정보를 검색하고자 하면 object위치에 SELECT 구문에서 지정한 변수를 배치하면 사용자가 지정한 조건에 맞게 검색을 시행하는 질의문을 구성할 수 있게 된다. 예를 들어, "20세기 폭스의 영화에 출연한 모든 배우와 그들이 출연한 다른 영화의 목록?"을 DQL 질의문으로 구성하게 되면, 다음과 같다.

```
SELECT ?actor ?other_movie
WHERE (<vcard:twentythCentury>, <vcard:product>, ?movie)
(?movie, <vcard:actor>, ?actor)
(?actor, <vcard:appearance>, ?other_movie)
USING vcard FOR http://ailab.ac.kr/jemin/movie#
```



(그림 2) RDQL 형식의 질의문 변환 과정

(그림 2)는 RDQL 변환기가 입력된 질의 조건들과 URI 매칭기를 통해 적절한 RDQL 질의문을 생성하는 과정을 보여준다. 사용자 검색 인터페이스에서 입력된 키워드 형태의 데이터들은 RDQL 변환기에서 사용할 수 있는 데이터 형태로 정제된 후 전송된다. 전송된 데이터는 RDQL의 문법의 맞게 순서(SELECT-WHERE- AND)를 정해주는 키워드 추상화 과정을 거친다. 즉 사용자 검색 인터페이스의 검색 패널에 선택된 항목들은 사용자가 찾기를 원하는 정보로서 SELECT 구문에서 변수로 변환되고, 검색 조건 패널에 선택된 항목들은 사용자가 정보를 검색하기 위한 조건으로써 WHERE 구문에 트리플로 변환된다. WHERE 구문을 구성할 때에는 키워드만을 가지고 트리플 구문의 형태로 조건을 표현해야하기 때문에 URI 매칭기를 이용하여 각각의 트리플 구조를 표현된 질의문을 생성할 수 있다.

이와 같이 RDQL은 트리플과 같은 형태로 자신의 조건을 표현하는데, <Subject>, <Property> 부분은 URI를 반드시 갖기 때문에 변환된 질의문의 조건부분에 해당하는 WHERE 구문도 URI형식으로 작성되어야 한다. 키워드로 입력된 데이터를 URI로 표현하기 위해 URI 매칭기를 통해서 각각의 데이터에 맞는 URI로 매치시킨다. 즉, URI 매칭기는 각 트리플 구문의 URI를 파악하는 것과 동시에 입력된 데이터에 정확히 연결된 URI를 매치시킨다. 이러한 과정을 통해서 사용자 DQL 검색 인터페이스에서 입력된 키워드와 각 키워드에 매칭된 URI는 RDQL형식 변환기에 의해 RDQL형식의 질의문으로 변환된다.

RDQL 변환기는 RDQL 엔진의 검색된 정보들을 전송받는데 이 역시 URI 형식으로 표현된다. 때문에 사용자가 사용자 검색 인터페이스 통해서 쉽게 인식할 수 있도록

URI 제거기를 이용하여 검색된 정보에 붙은 URI 형식의 태그들을 제거하고 데이터 정제과정을 통해 정리한 후 사용자 DQL 검색 인터페이스에 전송한다.

5. RDQL 엔진

RDQL 엔진은 RDQL 컨버터에서 생성한 DQL 질의문이 찾고자 하는 정보를 검색한다. 본 논문에서 제안한 RDQL 검색 시스템은 보다 검색 정확도를 내기 위해 2단계 구조를 가지는 RDQL 엔진을 연구한다. 먼저 질의문이 들어오면 추론엔진은 내부적으로 정의된 Axiom을 사용하여 검색 대상이 되는 정보의 속성과 관계를 파악한 후, 블랙보드에 파악된 정보를 추가로 적재한다. 두 번째 단계에서는 질의문의 조건부분에 해당하는 트리플을 추론 엔진에 다시 전송하여 추론을 실행한다. 이 때 실행되는 두 번째 추론은 방대한 양의 메타 데이터로부터 실제 검색에 필요한 데이터만을 추출하는 단계로서 추출된 메타데이터로부터 정보가 검색된다.

RDQL 엔진을 구축하기 위한 가장 중요한 단계인 추론 엔진 시스템은 입력된 메타 데이터를 여러 사실들로 구성하고 추론을 실행할 수 있는 규칙들이 삽입된 지식 베이스와 지식 베이스내의 규칙을 이용하여 사실들을 추론하여 단일화하는 추론 엔진으로 구성된다. 본 논문에서는 DAML+OIL로 만들어진 메타데이터를 KIF(Knowledge Interchange Format) 형식으로 읽어서 추론을 하도록 개발된 JTP를 추론 엔진으로 활용하였다.

본 논문에서 사용하는 추론은 귀납적 추론 방식중의 하나인 전방향 추론(Forward Chain Reasoning)이다. 전방향 추론은 사용자가 추론하고자 하는 문제와 관련된 모든 사실들이 제공되면, 추론 엔진이 각각의 사실(fact)과 지식베이스(knowledge Base)에 있는 규칙(Rule)들의 전제조건을 매치하여 일치하는 규칙을 찾아 해당 규칙을 수행하고 그 규칙의 결론부분을 새로운 사실로서 추가한다. 정방향 추론의 이러한 과정은 추론엔진이 더 이상 지식베이스에 있는 규칙들의 조건 부분과 사실들 간의 매치를 통해 새로운 결론을 내릴 수 없을 때까지 진행된다. 전방향 추론시스템을 구축하기 위해서는 작업메모리(Working Memory), 규칙베이스(Rule-Base), 전방향 추론이 가능한 추론엔진이 있어야 한다. 본 논문에서의 작업메모리에는 규칙들의 조건문과 매치하기 위해 트리플 형태의 메타데이터가 저장되어 있고, 규칙 베이스에는 전방향 추론을 하기 위한 IF-THEN 형태의 규칙들이 구성되어 있다.

(그림 4)는 RDQL 엔진의 실행 과정을 설명하고 있다. 먼저 RDQL 변환기에서 생성되어 전달된 질의문들은 질의문 분석기에 의해 추론엔진의 근거 데이터가 되는 조건 질의문과 RDQL 엔진이 검색을 할 수 있게 하는 검색 질의문으로 나누는 질의 정제 과정을 거친다.

분리된 트리플 구문 중 검색 질의문은 JTP 질의문으로 변환되어 추론엔진으로 전송되고, 추론엔진은 내부적으로

정의된 Axiom과 질의문을 이용하여 검색 대상이 되는 모든 메타데이터의 속성과 관계를 1차적으로 추론한다. 예를 들어 “SM5의 제조업체는 어디인가?”

라는 질의문에 대해 온톨로지에 정의된 속성의 관계를 바탕으로 새로운 데이터를 추론하여 검색한다. 즉, 제조업체에 대한 속성 값이 존재하지 않더라도 제조업체와 같은 속성(samePropertyAs)인 브랜드에 대한 속성 값이 존재하면, 그 속성 값이 검색하고자 하는 정보가 될 수 있다. 다음은 방금과 같은 상황을 메타데이터의 속성과 관계를 1차적으로 추론한 예이다.

```

질의문:
SELECT ?pro
WHERE (<vcard:SM5>, <vcard:production>, ?pro)
USING vcard FOR http://ailab.ac.kr/jemin/car#

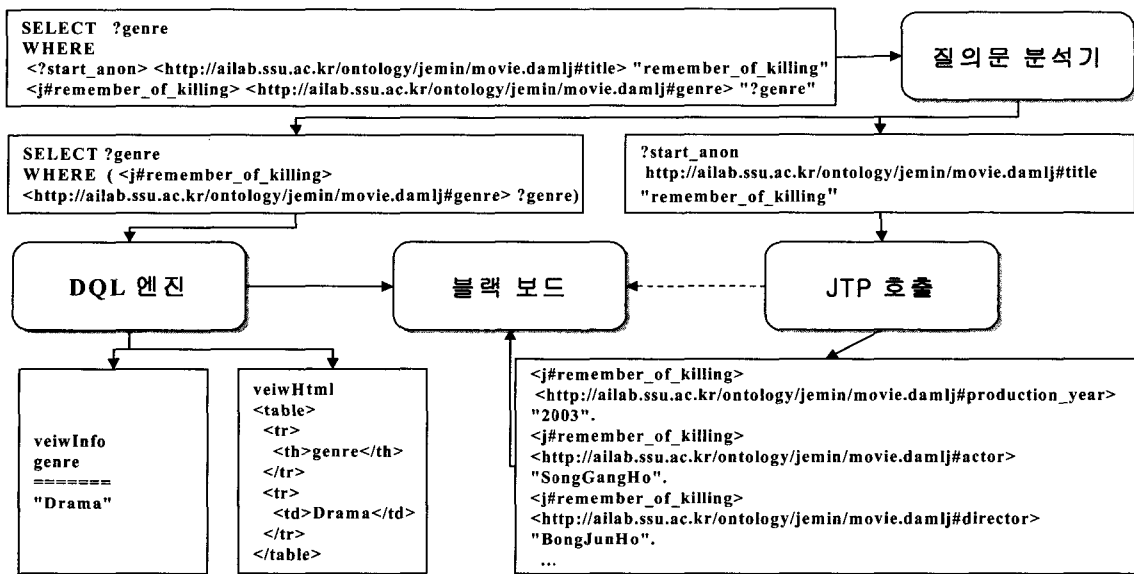
메타데이터:
<http://ailab.ac.kr/jemin/car#production>,
<http://ailab.ac.kr/jemin/car#subPropertyOf>,
<http://ailab.ac.kr/jemin/car#brand>
...
<http://ailab.ac.kr/jemin/car#brand>,
<http://ailab.ac.kr/jemin/car#subPropertyOf>,
<http://ailab.ac.kr/jemin/car#production>
...
<http://ailab.ac.kr/jemin/car#SM5>,
<http://ailab.ac.kr/jemin/car#brand>,
<http://ailab.ac.kr/jemin/car#Samsung>

Axiom:
(<=> (PropertyValue samePropertyAs ?P1 ?P2)
      (and (subPropertyOf ?P1 ?P2) (subPropertyOf ?P2 ?P1)))

결과 :
<http://ailab.ac.kr/jemin/car#SM5>,
<http://ailab.ac.kr/jemin/car#production>,
<http://ailab.ac.kr/jemin/car#Samsung>
    
```

이렇게 추론된 정보들 역시 트리플 구조를 가지며, 블랙보드에 추가적인 메타 데이터로 적재되게 된다. 분리된 트리플 구문 중 조건 질의문은 2차적인 추론 과정에서 시스템 구축자가 정의한 규칙들의 전제 조건과 비교된다. 이때 올바르게 매치되는 규칙들이 존재하는지 살펴보고 존재한다면 그 규칙을 실행한다. 규칙이 실행되면 그 결과에 해당되는 모든 사실(트리플 데이터)들을 블랙보드에 재 저장한다.

즉, 블랙보드에 있는 기존의 메타데이터는 지워지고 규칙을 실행하여 생성한 새로운 메타데이터만이 적재된다. 질의문들의 조건부분을 분석하여 전방향 추론을 실행하는 일련의 과정은 controller에서 제어하는데, 이 과정은 재귀적으로 이루어진다. 즉 조건 부분이 여러 개 존재할 경우 처음 조건부분을 이용하여 추론을 실행한 결과를 바탕으로 그 다음 전제조건을 이용해서 전방향 추론을 실행한다. 이러한 추론 과정을 통해서 블랙보드에 저장되어 있는 메타데이터들이 실제 검색 엔진에서 검색하기 위한 유용한 트리플 데이터가 되며, RDQL 검색엔진은 블랙보드에 저장된 데이터를 바탕으로 시멘틱 검색을 실행하고 그 결과를 RDQL 변환기에 전송한다.



(그림 3) JDQL 엔진의 실행과정

<표 1>, <표 2>, <표 3>, <표 4>, <표 5>는 기업의 정보 검색을 예로 들어 설명한 RDQL 엔진 실행 과정이다.

<표 1> step 1-RDQL 컨버터에서 전송된 질의문

```
SELECT ?document
WHERE ( <http://seoul/company/> <http://information/com#CN> "송실전자".
( <http://company/송실전자> <http://information/송실전자#부서> "총무부".
( <http://송실전자/총무부> <http://information/송실전자#직급> "대리".
( <http://송실전자/사원정보/대리/홍길동> <http://information/문서#DC> ?document )
( ?document <http://information/문서#date> ?date )
AND ?date >=20010804
```

<표 2> step 2-질의문 분석기에 의해 분리된 트리플 형식의 조건문

```
#1<http://seoul/company/> <http://information/com#CN> "SK".
#2< http://company/SK> < http://information/SK#부서> "총무부".
#3< http://SK/총무부> < http://information/SK#직급> "대리".
#4<http://SK/사원정보/대리/홍길동> < http://information/문서#DC > ?document.
#5?document <http://information/문서#date> ?date.
```

<표 3> step 3-전방향 추론을 통해 규칙의 결론 부분에 단일화되는 트리플 구문들을 블랙 보드에 적재

```
?document
=====
document = "pdp의 강도실형"
document = "2002년도 4분기 소비자 소비 현황"
document = "2003년도 추진 프로젝트"
document = "2003년도 휴가 계획서"
document = "2002년도 휴가 계획서"
```

<표 4> step 4-JTP 질의문과 JTP 내부의 정의된 Axiom을 사용하여 검색 대상이 되는 메타데이터의 속성과 관계 추론

```
<http://seoul/company/> <http://information/com # CN> "SK".
-----
<http://seoul/company/samePropertyAs> <http://information/com # CN>
<http://information/com # EN>
( <= <http://seoul/company/>.<http://information/com # CN>."NIKE")
.<http://company/송실전자> ?ano00 ?ano01
( <= <http://seoul/company/>.<http://information/com # CN>."SK")
.<http://company/SK> ?ano02 ?ano03
( <= <http://seoul/company/>.<http://information/com # EN>."G&Tech")
.<http://company/G&Tech> ?ano04 ?ano05
( <= <http://seoul/company/>.<http://information/com # EN>."현대건설")
.<http://company/현대건설> ?ano06 ?ano07
...
```

<표 5> step 5- 조건 질의문과 지식베이스의 규칙의 조건과 매치 시도

```
# 1<http://seoul/company/> <http://information/com # CN> "SK".
-----
( <= <http://seoul/company/>.<http://information/com # CN>."NIKE")
.<http://company/송실전자> ?ano00 ?ano01
( <= <http://seoul/company/>.<http://information/com # CN>."SK")
.<http://company/SK> ?ano02 ?ano03
( <= <http://seoul/company/>.<http://information/com # CN>."G&Tech")
.<http://company/G&Tech> ?ano04 ?ano05
( <= <http://seoul/company/>.<http://information/com # CN>."현대건설")
.<http://company/현대건설> ?ano06 ?ano07
...
```

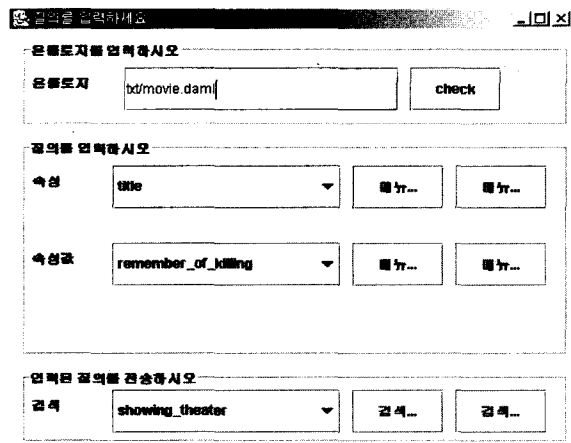
<표 6> step 6- 전방향 추론을 통해 규칙의 결론 부분에 단일화되는 트리플 구문들을 블랙 보드에 적재

```
( <= <http://seoul/company/>.<http://information/com #CN>,"SK")
.<http://company/ SK > ?ano00 ?ano01)
...
<http://company/SK><http://infomation/SK#부서>"인사부",
<http://company/SK><http://infomation/SK#부서>"총무부",
<http://company/SK><http://infomation/SK#부서>"영업부",
<http://company/SK><http://infomation/SK#부서>"개발부",
...
<http://company/SK><http://infomation/SK#owner>"최태원",
<http://company/SK><http://infomation/SK#LOC>"한국",
...
```

6. 사용자 검색 인터페이스

사용자들 입장에서는 검색에 쓰일 질의문을 보다 쉽게 사용하기를 원한다. 그래서 본 논문에서는 사용자 중심의 사용자 검색 인터페이스를 설계하여 시멘틱 검색을 이용하는 사용자가 검색하고자 하는 정보에 대한 질의문을 쉽게 입력할 수 있게 하였다.

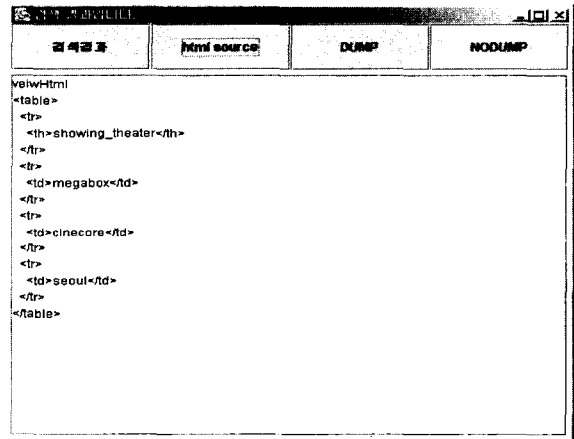
본 논문에서 제안한 사용자 검색 인터페이스는 콤보 박스와 텍스트 필드를 사용하여 키워드 형태로 질의어를 구성하게 함으로써 시멘틱 검색을 편하게 할 수 있게 구성되었고, 단순한 키워드 입력만으로 통해서 사용자가 원하는 정보의 의미를 파악할 수 있는 근거를 획득하게 된다. 또한 시스템에서는 검색된 결과를 사용자가 이해하기 쉽게 보여주는 역할을 수행하며, 각 사용자의 검색 과정을 상세히 기록한 로그파일을 제공한다. 로그 파일은 사용자가 검색한 정보와 검색 과정 등이 정확히 기록되어 있어서 어떠한 정보를 검색하고 어떠한 과정을 거쳤는지를 알 수 있도록 한 것이다. 사용자 검색 인터페이스를 통해서 얻어진 키워드들은 필요한 정보만을 골라서 저장하는 데이터 정제과정을 거친 후 정제된 데이터만을 RDQL 변환기에 전송한다.



(그림 4) 사용자 검색 인터페이스의 입력창

(그림 5)는 본 논문에서 제안하는 사용자 검색 인터페이스의 프로토타입을 나타낸 것이다. 사용자는 자신이 원하는 의미의 정보를 검색하기 위해서 키워드 입력기에 질의문을 간단

한 키워드를 입력하게 된다. 예를 들어, 영화 정보를 검색하기 위해 영화정보 관련 온톨로지를 기반으로 시멘틱 검색 서비스를 실행하는 문화 정보 검색 시스템을 사용하게 된다고 가정했을 때 사용자가 “remember_of_killing을 상영하는 극장”에 대한 정보를 원한다면 먼저 사용자는 검색조건에 해당하는 “remember_of_killing”을 지정하기 위해 키워드 입력기의 메뉴창의 title이라는 질의 아이템 과 아이템 값인 remember_of_killing을 선택한다. 그리고 사용자가 검색하기를 원하는 정보에 해당하는 상영 극장 정보를 검색 필드에서 선택하고 입력버튼을 클릭하게 된다. 이러한 방법을 통해서 입력된 키워드들은 RDQL 변환기에서 사용가능한 데이터로 정제된다.



(그림 5) 사용자 검색 인터페이스의 출력창

한편 검색 엔진에서 검색된 결과 정보들은 검색 결과 출력기에 의해 사용자가 이해할 수 있는 형태로 출력되며, 사용자가 각각의 정보를 선택하면 여러 가지 정보의 세부적인 내용을 자세히 보여주는 정보 뷰어 창을 통해 정보의 상세한 부분들을 볼 수 있다. 예를 들어 사용자가 아래 결과 출력기의 HTML버튼을 클릭하게 되면 검색 결과가 HTML 코드로 보여 지게 된다.

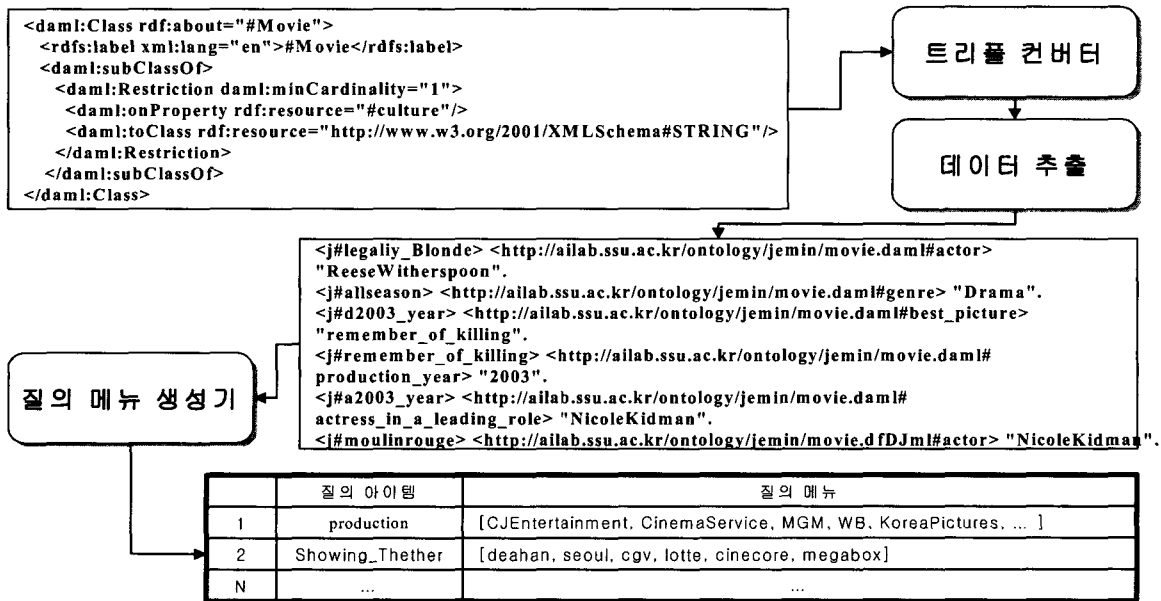
7. 질의 메뉴 생성기

사용자가 시멘틱 검색 시스템을 쉽게 사용하기 위해서는 기존 키워드 검색 방식의 인터페이스를 유지하면서 내부적으

로는 RDQL 검색 엔진과 호환되는 시스템을 설계해야만 한다. 사용자 RDQL 검색 인터페이스에서 사용자가 검색 질의를 쉽게 사용하기 위해서는 시스템 내부적으로 검색 질의어를 동적으로 제공해야한다. 즉, 기업 정보나 문화 정보 또는 뉴스 정보처럼 각 정보의 종류에 맞는 질의 메뉴들을 생성해주어야 한다.

(그림 6)은 질의 메뉴 생성기의 실행 과정을 나타낸 것이다. 시스템이 처음 구동이 되면 사용자가 원하는 검색 카테고리(영화

정보, 차 정보, 기업 정보, 학교 정보 등등)에 해당하는 온톨로지와 인스턴스 파일들을 시스템에 호출한다. 소프트웨어 에이전트나 시스템 구동기는 검색 대상이 되는 도메인에 대한 온톨로지를 트리플 컨버터를 이용하여 트리플로 변환한다. 트리플중에서 질의 아이টে을 구성하는데 중요한 역할을 하는 것은 Property와 각각의 Property의 값(object)들 이다. 또 각각의 클래스의 계층 및 상속관계를 나타내는 subClassOf와 subPropertyOf 역시 메뉴 아이টে을 구성하는데 중요한 역할을 한다.



(그림 6) 질의 메뉴 생성기의 실행과정

변환된 트리플은 Element 분류기를 통해 <subject>, <property>, <object>가 각각 분리되며 Element 분류기가 각각의 Element를 백터에 저장한다. 이렇게 저장된 백터 리스트를 바탕으로 동적으로 변하는 메뉴 아이টে을 생성한다. 예를 들어 회사란 클래스 내에 position, department라는 property가 존재하면, position과 department는 질의 메뉴 아이টে으로서 분류되며, 부장, 과장, 대리나 총무부, 경리부는 각각의 property의 값이므로 선택된 질의 아이টে의 선택 값이 된다.

8. 실험 및 결과

본 논문에서 제안하는 시멘틱 웹 기반 DQL 검색 시스템은 사용자가 간편하게 입력한 질의어를 RDQL 질의문으로 변환하고, 전방향 추론을 사용하는 추론 엔진을 이용하여 검색 데이터의 관계와 속성을 파악하며, 방대한 양의 메타데이터로부터 실제 검색 엔진이 사용하는데 필요한 데이터만을 추출하여 시멘틱 검색을 실행한다. 이와 같은 기능은 RDQL 검색 시스템이 다양하게 입력되는 사용자의 질의어들의 조합을 파악하고, 이를 기반으로 RDQL 엔진에 필요한 질의문을 생성하며 필요한 메타데이터를 추론해줌으로써

이루어진다. 사용자가 입력한 질의어들의 조합에 대한 파악과 RDQL 엔진을 위한 메타데이터 추론은 추론 엔진과 RDQL 변환기가 정확하게 구축되어야 가능하다.

그래서 본 논문에서 제안하는 DQL 검색 시스템의 성능을 평가하기 위해서 사용자가 입력한 키워드 데이터를 RDQL 질의문으로 변환해주는 질의문 변환에 대한 정확성 평가와 추론 엔진을 적용하여 데이터들의 관계와 속성들을 추론하였을 경우의 검색 정확도 대한 평가를 수행하였다.

<표 7> 실험의 활용된 자료

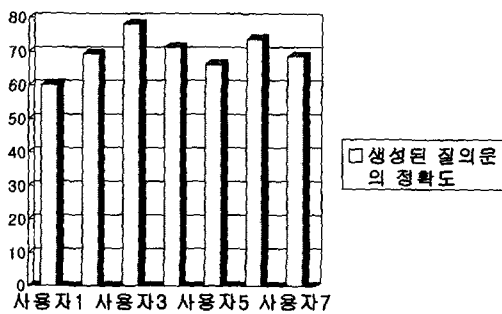
	사용 온톨로지	트리플 수	사용 질의수	비교대상
실험 1	1개	1500개	200개	없음
실험 2	3개	3000개	150개	DQL 검색 엔진

* 실험을 평가하기 위한 척도로 정확도의 평균을 이용
 * 기존 시스템은 추론 엔진이 적용되지 않은 기존의 RDQL 시스템을 이용

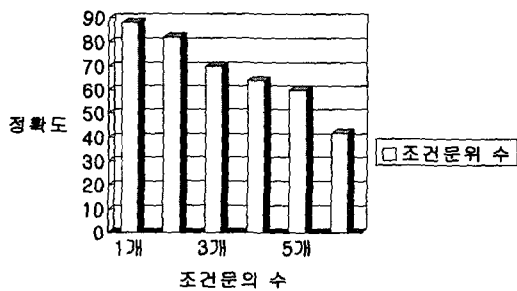
첫째, 사용자가 입력한 키워드를 RDQL 질의문으로 변환해주는 질의문 변환에 대한 정확성 평가이다. 구축된 RDQL 변환기의 정확성을 평가하기 위해서는 RDQL 변환기에 의해 생성된 질의문이 각 사용자의 의도 반

영에 대한 정확도와 각 질의문에 대해 검색하고자 하는 정보의 조건문들이 2개 이상 존재할 경우 각 조건문들을 얼마나 정확하게 조합되어 질의문을 생성해 내는지에 대한 실험이 필요하다.

(그림 7)은 본 논문에서 제안하는 RDQL 변환기가 어느 정도 정확하게 각 사용자 질의에 따른 RDQL 질의문을 생성해주는지 그래프로 표현한 것이다. 총 7명의 사용자가 각각 20개의 검색 시나리오에 따라 검색 인터페이스에 질의 키워드를 입력한 후에 생성되는 RDQL.txt를 확인하여 문법에 맞는 질의문이 생성되는지 확인한 결과이다.



(그림 7) 사용자의 다른 질의문 생성의 정확도



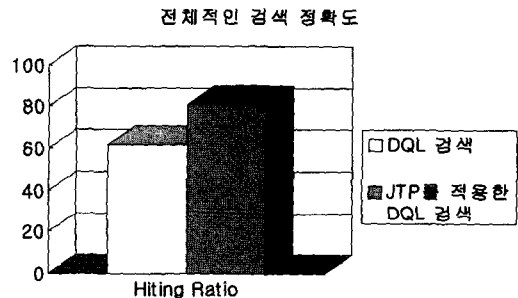
(그림 8) 조건문 수에 따른 정확도

(그림 8)는 RDQL 변환기가 질의문의 조건의 수에 따라 RDQL 질의문을 어느 정도 정확하게 생성해주는지 그래프로 표현한 것이다. 조건문 수의 다른 질의 생성의 정확도의 경우, 조건의 수가 많아질수록 정확도가 떨어지는 것으로 나타났다. 이는 조건문이 3개이하인 경우에는 사용자의 의도가 제대로 파악된 질의문이 생성되었지만 평균적으로 4개 이상인 경우에는 사용자의 의도를 정확하게 반영하지 못했다. 첫째, 실험 결과에 따르면 본 논문에서 제안하는 RDQL 변환기는 단순한 시나리오의 질의문은 정확하게 생성해 주었다. 즉, 조건문이 3개이하일 경우나 지식 베이스 안에 사용자가 찾고자 하는 정보가 존재할 경우에는 질의문으로 정확하게 변환되지만, 조건문이 많아지고, 지식 베이스 안에 사용자가 찾고자 하는 정보가 존재하지 않을 경우 생성된 질의문은 사용자의 의도를 정확하게 파악하지 못한다.

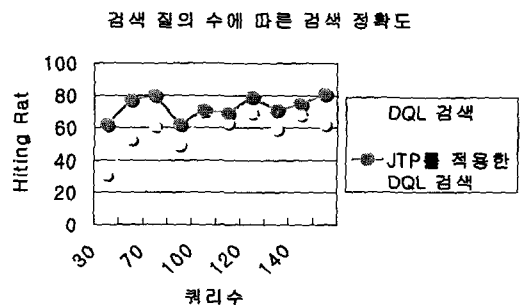
둘째, RDQL 검색 엔진에 추론 엔진을 적용하여 검색 대상

이 되는 데이터의 속성과 관계를 추론한 후, 검색 결과의 정확도에 대한 실험을 해보았다. 본 논문에서는 JTP의 API를 사용하여, 추론이 가능한 새로운 RDQL 검색 엔진을 구축하였다. 구축된 RDQL 엔진의 검색 정확도를 평가하기 위해서 추론 기능이 적용되지 않은 RDQL 엔진과 JTP 추론 기능이 적용된 RDQL 엔진에 같은 검색을 실행하여, 목적에 맞는 결과가 출력되는지 확인·비교하는 실험을 실시했다.

(그림 9)은 본 논문에서 제안하는 RDQL 엔진이 일반적인 RDQL 엔진에 비해서 어느 정도의 정확도를 갖는지에 대한 결과를 그래프로 표현한 것이다. 총 150개의 질의문을 입력한 후 검색이 제대로 되었는지 확인한 결과다. 검색의 정확도의 경우, 일반 RDQL 엔진은 62%의 정확도를 나타내었다. 반면 추론 엔진을 적용한 RDQL 엔진의 경우 81%의 정확도를 나타내었다. 기존 RDQL 엔진의 경우 트리플로 정확히 명시된 정보밖에 검색하지 못하기 때문에 정보들의 속성과 관계가 추론되어 추가적인 트리플 데이터를 생성하는 RDQL 검색 엔진에 비해 정확도가 떨어지는 것으로 분석되었다. 그러나 RDQL 엔진 역시 온톨로지에 영향을 받는다. 즉 온톨로지에 정보들의 관계와 속성에 대해 명시가 되어 있어야 추론 역시 가능하며, 때문에 검색 엔진의 성능은 온톨로지가 얼마나 잘 구축되어 있는가에 따라 좌우되는 것으로 분석되었다.



(그림 9) 검색 정확도



(그림 10) 검색 시도에 따른 정확도

(그림 10)은 RDQL 엔진이 검색 시도 횟수에 따라 일반 RDQL 엔진의 비해 어느 정도의 검색 정확도를 갖는지 측정한 것이다. 검색 시도횟수에 따른 실험에서 추론 엔진이 적용된 RDQL 엔진이 일반 RDQL 엔진에 비해 10~20%정도 정확도가 높았다.

9. 결론 및 향후 연구

일반적인 시멘틱 검색은 사용자가 복잡한 질의들을 수작업으로 처리하여 원하는 정보를 검색한다. 또 DQL (Daml Query Language)은 자체 추론 기능을 갖지 않기 때문에 검색될 정보의 속성 및 관계를 알 수 없기 때문에 검색 정확도가 떨어진다. 본 논문은 시멘틱 검색의 효율성과 검색 정확도를 높이기 위하여 다음 두 가지 시스템을 제안하였다.

- 첫째, 사용자 중심의 키워드 형태 질의를 시스템 중심의 RDQL로 변환하는 RDQL 변환기를 제안하였다.
- 둘째, 추론엔진을 불러내어 추론을 실행하는 RDQL 엔진을 제안하였다.

본 논문에서 제안하는 DQL 검색 시스템은 시멘틱 검색 방식의 사용 효율성과 검색 정확도를 높이기 위하여 설계되었다. 즉, 사용자의 질의를 단순한 키워드로 입력받아, RDQL 변환기가 RDQL 질의를 자동으로 생성하여, 사용자가 원하는 정보를 검색하는데 도움을 주고, RDQL 엔진이 추론 엔진을 불러내어 전방향 추론을 실행함으로써 검색 정확도를 높이기 위한 시스템을 설계하여 사용자가 시멘틱 웹 검색을 보다 쉽고 효율적으로 실행할 수 있다.

시멘틱 웹 환경으로부터 본 논문에서 제안한 시멘틱 웹 기반 DQL 검색 시스템을 활용함으로써 HTML로 작성되어 있는 웹 문서들을 에이전트 같은 프로그램이 자동으로 문서의 의미를 파악하여, 문서에 대한 정보를 검색 시 검색 결과의 정확도를 향상시키고 정보 공유의 어려움을 해결하는데 도움이 된다.

실험 결과에 따르면 본 논문에서 제안하는 RDQL 엔진은 일반 RDQL 엔진보다 다소 높은 정확도를 보였다. 그러나 RDQL 엔진은 미리 구축되어진 온톨로지와 인스턴스 데이터를 바탕으로 추론을 하여 검색을 하기 때문에 온톨로지에 의존적이다. 또한 RDQL 질의문이 문법에 맞게 입력이 되어야 정확한 검색 결과를 얻을 수 있기 때문에 제대로 된 RDQL 질의문이 입력되지 않는다면 정확한 검색 결과는 얻어질 수 없다. 즉, 본 논문에서 제안한 DQL검색 시스템의 문제점은 온톨로지에 대한 의존도이다. 온톨로지를 잘 구축하는 문제에 앞서 생각해야 할 점은 온톨로지의 표준화이다. 예를 들어 어떤 기업에서 DQL 검색 시스템을 사용하고자 한다면 자사만을 위한 온톨로지를 구축해 사용하는 것 보다는 모든 기업에서 사용할 수 있는 표준적인 온톨로지를 사용하는 것이 시스템의 성능을 높일 수 있을 뿐만 아니라 각 기업간의 정보 공유 하거나 에이전트를 이용한 효과적인 서비스가 가능하다. DQL 검색 시스템뿐만 아니라 다른 시멘틱 웹 관련 기술 또한 온톨로지에 대한 의존도가 크므로 향후에는 각 기관에서 표준적으로 사용할 수 있는 온톨로지를 개발할 필요가 있다.

참고 문헌

- [1] 이재호, "시멘틱 웹의 온톨로지 언어", 정보과학회지, 2003년 3월.
- [2] <http://www.xml.com/pub/a/2002/01/30/daml1.html>
- [3] <http://www.xml.com/pub/a/2002/03/13/daml.html>
- [4] <http://www.w3.org/TR/owl-ref>
- [5] Gleb Frank, "JTP user manual", KSL, Stanford University, <http://www.stanford.edu/~gkfrank/jtp/#proof>
- [6] Richard Fikes, Pat Hayes, Ian Horrocks, "DQL-A Query Language for the Semantic Web", WWW 2003, Budapest, Hungary, pp.20-24, May, 2003.
- [7] DQL Project for the Stanford Knowledge Systems Laboratory, <http://ksl.stanford.edu/projects/dql>
- [8] 최중민, "시멘틱 웹 개요와 연구동향", 정보과학회지, 2003년 3월.
- [9] RDF Primer, W3C Working Draft 23 January 2003, <http://www.w3.org/TR/rdf-primer>
- [10] Larry Kerschberg, Wooju Kim, Anthony Scimek, "Intelligent Web Search via Personalizable Meta-Search Agents," E-Center for E-Business, George Mason University, Chonbuk National University, SUNY-Brockport.
- [11] Andy Seaborne, "Jena Tutorial: A Programmer's Introduction to RDQL," April, 2002.
- [12] John Davies, Richard Weeks, Uwe Krohn, "QuizRDF: Search Technology for the Semantic Web," BTexact Technologies, Orion 5/12, Adastral Park.
- [13] 이재규, 최형립, 김현수, 서민수, 주식진, 지원철, "전문가 시스템-원리와 개발", 법영사, 1996.



김 제 민

e-mail : kimjemins@hotmail.com
 2001년 숭실대학교 컴퓨터학과(학사)
 2004년 숭실대학교 대학원 컴퓨터학과(석사)
 2004년~현재 숭실대학교 대학원 컴퓨터학과 박사과정

관심분야 : 인공지능, 시멘틱 웹, 유비쿼터스 컴퓨팅



박 영 택

e-mail : park@computing.soongsil.ac.kr
 1978년 서울대학교 전자공학과(학사)
 1980년 KAIST 전산학(석사)
 1992년 Univ. of Illinois at Urbana-Champaign(박사)

1981년~현재 숭실대학교 컴퓨터학과 교수
 관심분야 : 인공지능, 에이전트, 전문가 시스템, 유비쿼터스 컴퓨팅