

시맨틱 웹 데이터의 경로 기반 질의 처리 기법

김연희*, 김지현**

The Scheme for Path-based Query Processing on the Semantic Data

Youn-Hee Kim *, Jee-Hyun Kim **

요 약

시맨틱 웹에서는 메타데이터와 온톨로지를 이용해 정보 리소스의 개념을 정의하고 의미적 연관성을 표현함으로써 지능적인 정보 검색과 자동화된 웹 서비스의 제공이 가능하다. 이러한 시맨틱 웹의 핵심적인 기능을 구현하기 위해서는 온톨로지와 메타데이터와 같은 시맨틱 웹 데이터를 효율적으로 관리하는 것이 무엇보다 중요하다. 따라서 본 논문에서는 시맨틱 웹 데이터의 의미와 구조적인 특성을 고려하여 보다 정확한 질의 결과의 검색과 효율적인 질의 처리를 지원할 수 있는 인덱스 구조를 제안한다. 특히 시맨틱 웹 데이터의 의미와 구조적인 특성을 그대로 표현하기 위해 그래프 데이터 모델을 이용하고 다양한 질의 형태를 그래프 모델 기반의 경로식으로 처리한다. 본 논문에서 제안한 인덱스는 1차적으로 추출 가능한 구조적 경로 정보는 물론 온톨로지를 이용한 추론을 통해 2차적으로 추출 가능한 구조적 경로 정보에 대한 질의를 처리 대상으로 하여 기존 연구들과 차별화하며 시맨틱 웹의 개념을 온전히 반영하는 것을 목표로 한다. 또한 실험적 평가를 통해 본 논문에서 제안한 인덱스 구조가 정확성과 효율성 측면에서 우수하며 시맨틱 웹의 다양한 애플리케이션 개발에 적용 가능성을 보인다.

Abstract

In the Semantic Web, it is possible to provide intelligent information retrieval and automated web services by defining a concept of information resource and representing a semantic relation between resources with metadata and ontology. It is very important to manage semantic data such as ontology and metadata efficiently for implementing essential functions of the Semantic Web. Thus we propose an index structure to support more accurate search results and efficient query processing by considering semantic and structural features of the semantic data. Especially we use a graph data model to express semantic and structural features of the semantic data and process various type of queries by using graph model based path expressions. In this paper the proposed index aims to distinguish our approach from earlier studies and involve the concept of the

• 제1저자 : 김연희

• 투고일 : 2009. 07. 30, 심사일 : 2009. 08. 27, 게재확정일 : 2009. 10. 06.

* 부천대학 e-비즈니스과 강의전담교수 ** 서일대학 소프트웨어과 부교수

※ 본 논문은 2008년도 서일대학 학술연구비에 의해 연구되었음.

Semantic Web in its entirety by querying on primarily extracted structural path information and secondary extracted one through semantic inferences with ontology. In the experiments, we show that our approach is more accurate and efficient than the previous approaches and can be applicable to various applications in the Semantic Web.

▶ Keyword : 시맨틱 웹 데이터(Semantic Web Data), 인덱스(Index), 경로식(Path Expression)

1. 서론

최근 많은 웹 기반 응용 서비스가 개발되면서 정보 표현 및 접근의 단순성을 넘어서는 다양한 사용자의 요구가 발생하고 있다. 특히, 웹 상의 데이터가 계속해서 기하급수적으로 증가하고 있고, 유사한 데이터의 다양한 표현이 가능해지면서 보다 정확하고 풍부한 데이터를 검색하기 위해 단순한 키워드 검색을 넘어서 데이터의 의미 있는 해석에 기반을 두는 확장된 검색 방법이 요구된다. 이러한 여러 가지 요인으로 인하여 메타 데이터를 이용하여 정보에 추가적인 의미를 부여하고, 이를 통해 데이터의 의미를 인간뿐만 아니라 컴퓨터가 이해하고 처리할 수 있도록 하는 시맨틱 웹의 개념이 요구되고 있다[1, 2].

시맨틱 웹에서는 문서, 정보, 개념 등 웹 상에 존재하는 모든 구분되는 리소스를 URI(Uniform Resource Identifier)로 지칭한다. 그리고 메타데이터와 온톨로지를 이용해서 정보 리소스의 개념을 정의하고 의미적 연관성을 표현함으로써 보다 능동적인 정보 검색은 물론 자동화된 다양한 웹 서비스를 제공할 수 있다. 온톨로지는 메타데이터를 기술할 때 발생할 수 있는 중첩성과 모호성을 없애기 위해 용어의 개념과 용어 간의 관계를 정의한다. 메타데이터는 온톨로지에서 정의된 용어를 이용하여 정보 리소스의 의미와 정보 리소스간의 의미적 연관성을 기술하여 데이터 내에 숨겨져 있는 새로운 정보를 유도해내는 역할을 담당한다. 따라서 온톨로지와 메타데이터를 함께 이용하면 사용자의 요구에 대해 보다 정확하고 풍부한 정보 검색이 가능하기 때문에 시맨틱 웹에서의 온톨로지와 메타데이터의 역할에 많은 기대가 모아지고 있다.

메타데이터와 온톨로지를 형식적으로 기술하기 위해 RDF(Resource Description Framework), RDF 스키마, OWL(Web Ontology Language) 등의 다양한 언어들 소개되어 왔다. W3C에서 제안한 RDF와 RDF 스키마는 각각 메타데이터와 온톨로지를 기술하기 위한 기본적인 시맨틱 웹 언어로서 주어(subject)-서술어(predicate)-목적어(object)로 구성되는 트리플 구조의 문장 형태로 기술된다[3, 4]. 현재는 RDF와 RDF 스키마에 비해 풍부한 표현력과 모델링 요

소를 가지고 있는 OWL이 W3C의 권고안으로 채택되어 관심을 받고 있지만 보편화되기까지 많은 시간이 소요될 것으로 예상되므로 RDF와 RDF 스키마로 기술된 메타데이터와 온톨로지를 효율적으로 관리하기 위한 연구가 요구된다. 특히, 시맨틱 웹 상의 다양한 응용 시스템에 실제로 적용하기 위해서는 RDF와 RDF 스키마로 기술된 메타데이터와 온톨로지를 저장하고 추론에 기반한 효율적인 질의 처리를 지원하는 인덱싱 기법이 필요하다. RDF와 RDF 스키마에 관한 기존 연구들은 트리플 문장 구조를 활용하여 저장과 검색을 단순화시키려는 노력이 대부분이었다. 그러나 RDF와 RDF 스키마가 그래프 형태의 데이터 모델로 표현이 가능하고 사용자가 시맨틱 웹 환경에 적합한 다양한 질의 유형의 처리를 요구함에 따라 경로식 형태의 다양한 질의 형태는 물론 추론에 기반한 질의 처리를 통해 그래프 모델 내에 숨겨져 있는 정보에 대한 검색이 가능해야 한다.

따라서 본 논문에서는 RDF와 RDF 스키마를 모두 고려한 질의 유형을 분류하고 효율적인 질의 처리를 지원할 수 있는 인덱스 구조를 제안한다. 특히 본 논문에서 제안한 인덱스는 그래프 데이터 모델 상에서 표현 가능한 복잡한 경로 질의 처리를 지원할 수 있고 RDF 스키마에 직접 정의되어 있는 1차적인 구조 정보뿐만 아니라 클래스나 프로퍼티의 계층 관계에 의한 상속을 통해 2차적으로 추론이 가능한 구조 정보에 대한 질의 처리를 지원할 수 있다.

시맨틱 웹 상의 온톨로지와 메타데이터를 이용하여 추론에 기반한 효율적인 질의 처리를 지원하는 인덱싱 기법 개발을 목표로 하는 본 논문은 다음과 같이 구성된다. 2장에서는 시맨틱 웹 데이터의 저장과 인덱싱 기법에 관련한 기존 연구 결과물을 소개하고 문제점을 분석한다. 3장에서는 RDF와 RDF 스키마의 데이터 모델과 본 논문에서 고려하는 질의 유형을 소개한다. 4장에서는 직 · 간접적으로 정의된 RDF 스키마의 구조적 정보를 기반으로 효율적인 질의 처리를 지원할 수 있는 인덱스 구조를 제안한다. 5장에서는 본 논문에서 제안한 인덱스 구조에 대해 실험적인 방법으로 성능 평가를 수행한 결과를 설명하고 6장에서 결론을 맺는다.

II. 관련 연구

2.1 RDF와 RDF 스키마 저장 기법

RDF와 RDF 스키마를 위한 저장 및 질의 처리에 관심을 가지는 많은 연구 결과물이 소개되어 왔다. 이러한 연구 결과의 대부분이 RDF와 RDF 스키마의 트리플 구조를 기반으로 관계형 데이터베이스를 저장 및 검색에 이용한다(5). 관계형 데이터베이스를 이용한 시맨틱 웹 데이터 저장 시스템은 크게 수직-분할 방식, 바이너리 방식, 하이브리드 방식으로 구분할 수 있다(6).

수직-분할 방식은 주어-목적어-서술어의 트리플 구조를 이용한 단일 테이블에 RDF와 RDF 스키마 데이터를 모두 함께 저장하는 방식이다. 이러한 방식은 단일 테이블을 활용하기 때문에 관리가 쉽고 갱신 내용을 반영하기 용이한 장점이 있으나 모든 질의에 대해 하나의 테이블을 접근해야 하기 때문에 질의 처리 시 여러 번의 자기 조인이 필요한 단점이 있다. 또한, RDF와 RDF 스키마의 근본적인 차이점을 고려하지 않는 문제점도 존재한다. 수직-분할 방식의 저장 구조를 이용한 기존 저장 시스템에 Jena(7), RStar(8) 등이 있다.

바이너리 방식은 RDF 스키마에 정의된 각각의 클래스와 프로퍼티 별로 독립적인 테이블을 생성하여 RDF 데이터의 정보를 저장하는 방법이다. 이와 같은 방식은 프로퍼티 중심의 질의나 클래스와 리소스 관계에 대한 질의를 처리하는 경우 수직-분할 방식에 비해 효율적이나 많은 수의 테이블을 관리해야 하고 새로운 프로퍼티나 클래스가 추가될 때마다 새로운 테이블이 생성되어야 하는 문제점이 존재한다. 바이너리 방식을 채택한 기존 저장 시스템으로는 Sesame(9), DLDB(10) 등이 있다.

하이브리드 방식은 수직-분할 방식과 바이너리 방식을 모두 적용한 것으로 프로퍼티의 타입별로 수직-분할 방식의 테이블을 생성하고 바이너리 방식과 동일하게 모든 클래스에 대해 독립적인 테이블을 유지하는 방법이다. RDFSuite(11)과 같은 시스템이 하이브리드 방식을 적용한 예이다. 이 방법은 수직-분할 방식과 바이너리 방식의 장점을 모두 수용하고 있지만 여전히 몇 가지 문제점이 존재한다. 특히, RDF 스키마에 정의된 내용을 바탕으로 저장 시스템의 스키마를 설계하지만 서브 클래스, 서브 프로퍼티의 관계나 도메인, 레인지 클래스에 대한 정보 등은 고려하지 않기 때문에 RDF 스키마 중심의 질의를 처리하는데 비효율적이다. 그리고 복잡한 경로식으로 표현되는 질의를 처리하기 위해서는 많은 수의 조인 연산이 필요하다. 따라서 복잡한 경로식으로 표현된 질의의 효율적인 처리를 위한 저장 구조가 필요하다.

2.2 RDF와 RDF 스키마 인덱싱 기법

RDF와 RDF 스키마의 인덱싱 기법에 대한 기존 연구 결과는 크게 구조 기반 인덱스 구조와 키워드 기반 인덱스 구조로 분류할 수 있다. 구조 기반 인덱스에는 트리플 형태에 기반을 둔 단순 경로 인덱스와 반복되는 트리플 형태에 기반을 둔 복합 경로 인덱스로 세분화할 수 있다. 단순 경로 인덱스는 가능한 모든 질의 유형을 지원하기 위해 주어 인덱스, 서술어 인덱스, 목적어 인덱스가 별도로 유지되며 YARS(12) 시스템에서 사용된다. 복합 경로 인덱스로는 Suffix Array(13)가 대표적이며 RDF와 RDF 스키마의 그래프 데이터 모델에서 추출할 수 있는 경로에서 도달 가능한 모든 개체에 대한 정보를 유지하는 방식이다. 키워드 인덱스는 프로퍼티의 값으로 지정된 리터럴 데이터에 대한 텍스트 질의 처리를 지원하기 위한 것으로 YARS(12)와 참고문헌 [14]의 시스템에서 사용된다.

RDF와 RDF 스키마는 기존 웹 데이터와는 다른 구조적 특징을 가지고 있기 때문에 단순 경로 인덱스와 복합 경로 인덱스의 역할이 중요하다. 그러나 기존 연구 결과들은 대부분 RDF 스키마에 정의된 정보를 활용하지 않고 RDF 스키마와 RDF 데이터를 동일하게 처리하기 때문에 RDF 스키마에 정의된 의미적 관계나 계층 정보 등을 이용한 보다 정확하고 풍부한 정보의 검색을 지원하지 못한다. 그리고 저장 구조에 비해 활발한 연구가 이루어지지 않았고 저장 구조의 보조적인 수단으로서의 역할만이 강조되어 왔다. 따라서 RDF 스키마의 특징을 고려한 구조적 측면에서의 질의 처리를 지원할 수 있는 독립적인 인덱스 구조가 요구된다.

III. 데이터 모델링과 질의 형태 분류

3.1 RDF와 RDF 스키마의 데이터 모델

RDF와 RDF 스키마는 노드와 간선에 모두 레이블이 표현된 그래프 형태의 데이터 모델로 표현된다. RDF를 위한 그래프 형태의 데이터 모델에서 노드는 리소스나 리터럴 값을 의미하고 간선은 리소스간의 관계 또는 리소스와 리터럴 값의 관계를 표현하는 프로퍼티를 의미한다. RDF 스키마를 위한 그래프 형태의 데이터 모델에서 노드는 RDF 문서에서 기술하고 있는 리소스의 개념을 정의한 클래스를 의미하고 간선은 클래스간의 의미적 관계를 표현하는 프로퍼티를 의미한다. 그리고 RDF 스키마의 데이터 모델에는 클래스의 계층 관계와 프로퍼티의 계층 관계를 나타내는 부가적인 간선이 존재한다. 이러한 계층 관계를 의미하는 간선들은 RDF 스키마의 표준

안에 따라 "subClassOf" 또는 "subPropertyOf"의 레이블로 표현된다. 또한 RDF와 RDF 스키마의 데이터 모델에는 RDF 스키마에 정의된 클래스와 그 클래스로부터 인스턴스화된 RDF 내 리소스간의 관계를 표현하는 간선이 존재하며 이 간선은 "typeOf" 레이블로 표현된다.

RDF와 RDF 스키마의 그래프 모델은 다음과 같이 정의될 수 있다.

- 그래프 모델 $G = (V, E)$
 집합 $V = VR \cup VL \cup VC$
 집합 $E = ER \cup EL \cup EC$

(VR: 리소스 집합, VL: 리터럴 데이터 집합, VC: 클래스 집합)

(ER: 리소스간 관계, EL: 리소스-리터럴간 관계, EC: 클래스간 관계)

RDF와 RDF 스키마의 그래프 모델 G는 집합 V와 집합 E로 구성된다. 집합 V는 RDF 문서 내에 기술되어 있는 리소스 집합 VR과 리터럴 데이터의 집합 VL, RDF 스키마 문서 내에 정의된 클래스들의 집합 VC로 이루어져 있다. 집합 E는 RDF와 RDF 스키마 문서 내에 존재하는 프로퍼티들의 집합으로서 리소스와 리소스간의 의미적 관계, 리소스와 리터럴 데이터간의 의미적 관계, 클래스와 클래스간의 의미적 관계들로 구성된다. 집합 E에 속하는 간선들은 모두 진출 간선과 진입 간선의 역할을 담당할 수 있어 방향성 있는 그래프의 표현이 가능하다.

그림 1은 책에 관련된 정보를 기술한 RDF와 RDF 스키마 문서를 그래프 형태로 표현한 예이다. 타원으로 표현된 노드는 RDF 스키마에 정의된 클래스와 RDF에 기술된 리소스를 의미한다. 사각형으로 표현된 노드는 리터럴 값이다. 일반적인 프로퍼티를 나타내는 간선과 클래스 및 프로퍼티의 계층 관계를 나타내는 간선은 구분하여 표시하였다.

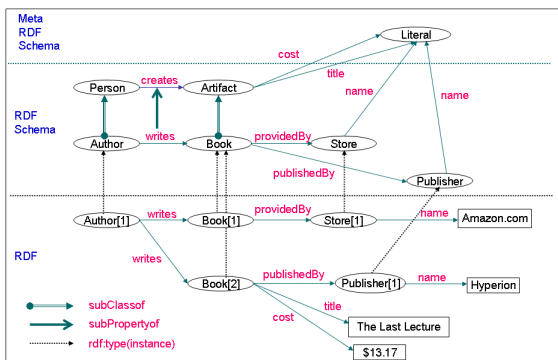


그림 1. 시맨틱 웹 데이터의 그래프 데이터 모델
 Fig. 1. Graph Data Model of Semantic Web Data

RDF와 RDF 스키마 내에 존재하는 다양한 개체들을 다루기 위해서는 각각의 개체들을 식별할 수 있는 레이블링 기법이 필요하다. 각 개체에 레이블링된 정보는 저장 및 인덱스 구조를 구축할 때 키의 역할을 담당할 수 있다. 본 논문에서는 의미 있는 정보를 키 값으로 이용함으로써 처리의 효율성을 높이고자 한다. RDF와 RDF 스키마 내에서 식별이 필요한 주요 개체는 클래스, 프로퍼티, 리소스이다. 일반적으로 RDF 스키마에 정의된 클래스나 프로퍼티의 개수에 비해 RDF에 기술된 리소스의 개수가 많으며, 특히 동일한 리소스에 대한 많은 정보가 표현되어 있다. 이때, 각 리소스는 URI로 길게 표현되기 때문에 저장 시 많은 공간의 낭비를 초래하게 된다. 따라서 본 논문에서는 긴 URI의 표현보다는 간단한 레이블링 정보를 사용함으로써 저장 공간의 효율성을 높이고자 한다.

본 논문에서는 RDF 스키마에 정의된 각 클래스와 프로퍼티에 대해 정의된 순서에 따라 일련번호를 붙여 각 개체를 식별하고 RDF에 기술된 리소스는 객체 지향 데이터베이스의 레이블링 기법을 적용하여 식별한다. RDF와 RDF 스키마를 함께 사용하는 경우 RDF 내에 정의된 리소스들은 보통 클래스 타입이 정의되어 있다. 이러한 리소스의 클래스 타입 정보는 질의 처리 시 유용하게 활용될 수 있다. 특히, 일반 사용자의 경우 RDF 스키마에 정의된 클래스로부터 인스턴스화된 리소스 정보를 검색하는 경우가 많기 때문에 리소스 정교 클래스 타입에 대한 연관 관계는 중요하다. 따라서 본 논문에서는 리소스의 클래스 타입과 RDF 내에 리소스 기술 순서에 따라 그림 1에서 정같이 "클래스타입[기술순서]"로 리소스의 레이블을 표현한다. 예를 들면, Book 클래스로부터 인스턴스화된 리소스가 RDF 내에 처음으로 기술된 것이라면 "Book(1)"의 레이블이 부여된다. 이러한 레이블을 통해 각 리소스가 어떤 클래스로부터 인스턴스화된 것인지에 대한 정보를 쉽게 판단할 수 있다.

3.2 RDF와 RDF 스키마의 질의 유형

RDF와 RDF 스키마에 대한 질의 형태는 인덱싱 기법을 설계하는데 있어 중요한 고려 사항이 된다. RDF와 RDF 스키마에 대한 질의 형태는 질의의 중심이 무엇이나에 따라 RDF 스키마에 관한 질의, RDF 데이터에 관한 질의, RDF와 RDF 스키마 혼합 질의로 분류할 수 있다. 그 중 혼합 질의 형태는 "typeOf" 프로퍼티에 의해 표현된 클래스와 리소스간의 관계에 대한 질의로, 온톨로지를 고려하여 RDF로 기술된 리소스를 검색할 수 있기 때문에 시맨틱 웹의 개념을 가장 잘 표현하고 있다고 할 수 있다. 따라서 향후 시맨틱 웹 환경에서 가장 선호되는 질의 유형이라 예상되므로 본 논문에서

는 RDF와 RDF 스키마의 혼합 질의 유형만을 고려한다. 그리고 사용자 선택에 따른 질의 처리의 적용 범위에 따라 1차 질의와 2차 질의를 분류한다. 1차 질의는 RDF와 RDF 스키마 데이터 모델에서 직접적으로 나타나는 정보에 대한 질의를 의미하고 2차 질의는 데이터 모델 상에서 수평적 관계와 수직적 관계를 함께 고려하여 간접적으로 나타나는 정보에 대한 질의를 의미한다. 질의 처리의 영역은 사용자가 선택할 수 있어야 하며, 사용자가 명시적으로 선택하지 않은 경우는 기본적으로 2차 질의를 처리 대상으로 해야 보다 논리적이고 타당한 질의 결과를 반환할 수 있다.

RDF와 RDF 스키마의 데이터 모델은 그래프 형태로 표현될 수 있기 때문에 혼합 질의 유형은 그래프에서 추출 가능한 경로식으로 표현이 가능하다. 본 논문에서는 혼합 질의 유형을 단순 경로 질의와 복합 경로 질의로 세분화한다. 단순 경로 질의는 트리플 구조에 기반을 두고 클래스 타입이 이미 정해진 리소스에 대한 질의를 의미한다. 단순 경로 질의를 통해 사용자는 RDF 스키마에 정의된 클래스와 프로퍼티 정보를 질의 조건으로 하여 관련된 리소스를 검색할 수 있다. 그리고 2차 질의를 통하여 데이터 모델에 직접적으로 표현되지는 않지만 추론을 통해 얻어진 결과까지도 반환할 수 있다. 다만, 단순 경로 질의는 RDF 스키마 상에서 트리플 구조의 단순 경로를 의미하는 것으로 실제 질의 처리 시에는 RDF와의 관계까지 고려해야 하기 때문에 보다 복잡한 경로로 해석된다. 따라서 RDF와 RDF 스키마를 모두 고려한 혼합 질의를 위해서는 이러한 복잡한 경로로 해석되는 질의를 어떻게 처리할 것인지가 무엇보다 중요하다. 복합 경로 질의는 RDF 스키마 상의 단순 경로 질의가 반복적으로 나타나는 질의 형태를 의미한다.

IV. 단순 경로 기반 구조 인덱스

3장에서 제시한 혼합 질의 유형을 처리할 경우 RDF와 RDF 스키마의 데이터 모델을 깊이 우선 탐색과 같은 일반적인 그래프 순회 방법으로 처리하는 것은 비효율적이다. 그리고 클래스와 프로퍼티의 계층 정보까지 고려하는 2차 질의를 처리하는 경우에 추론에 의해 유추할 수 있는 경로식까지 처리해야 하는 어려움이 존재한다. 이러한 이유로 기존 저장 구조들의 경우 복합 경로식으로 표현되는 질의는 물론 2차 질의를 처리하기 위해서는 많은 비교와 조인 연산을 필요로 하는 복잡한 처리 과정이 요구되고 경우에 따라서는 모든 결과가 반환되지 않는 문제점이 있다. 예를 들어, 그림 1에서 "Book.cost.Literal" 질의 경로 상에 존재하는 리소스와 터널 데이터의 쌍을 검색하고자 할 때 1차 질의에서는 어떠한 결과도 반환되지 않는다. 그림 1의 그래프 모

델에서 해당 경로식이 직접적으로 나타나지 않기 때문이다. Lite 정확한 결과의 반환을 위해서는 "Book질의클래스와 "Artifact질의 클래스 간에 서브의클래스 관계에서 음을 판별별별이 정보를 이용해서 "Artifact질의클래스다. † cost 질의를 처리를 "Book질의클래스에서 상속받아 "Book.cost.Literal 질터의는 새로운의 경로 정보에서 생성된다는 것을 이해해야만 정확한 검색의 결과를 반환할 수 있다. 이러한 literal 질 예이 앞te 정의한 2차 질의 "Book질의클래스와 "그래프 모델에 직접적으로 나타나지 않지만 분명히 존재하면 te 계층 정보를 이용해 유추해낼 수 있는 2차 질의에 대한 ra를 지원해야 정확별별별유효한 결과를 얻을 수 있다. 그러나 2차 질의를 처리하기 위해 필요한 정보들을 질의를 처리할 때마다 추론해내야 한다면 빠른 시뮬벌어 추결과를 얻어내기 어렵다. 따라서 RDF와 RDF 스키마에 대한 보다 정확별별별 빠른 itera를 지원하기 위해서 RDF 스키마에 t질의 쿼리 가능한 경로서 상속했만 아니기 Book질의추출할 수 있는 모든 경로서 상속을 질의 처리 시 쉽게 이용할 수 있도록 관리할 필요성이 있다. 따라서 본 논문에서는 저장 구조를 이용한 2차 질의 leral 질비효율성을 극복별별별 RDF 스키마에 t질의 • Book질의 표현되는 모든 경로서 상속보까지 고려하는 ra를 지원하는 인덱스 기법과 이를 이용한 질의 처리 전략을 제안한다.

4.1 인덱스 구축 과정

본 논문에서는 RDF와 RDF 스키마를 대상으로 하는 구조 기반 질의 처리를 위한 인덱스 구조를 제안한다. 구조 기반 질의는 RDF와 RDF 스키마의 그래프 모델에서 트리플 경로 식으로 표현이 가능하기 때문에 본 논문에서 제안한 인덱스는 RDF 스키마에서 직 · 간접적으로 추출 가능한 모든 기본 구조 정보, 즉 경로 정보를 관리하는 것을 목표로 한다.

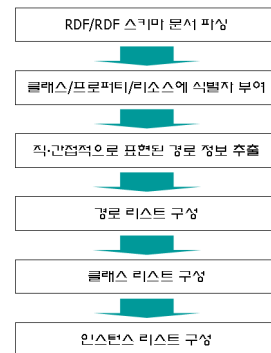


그림 2. 인덱스 구축 과정
Fig. 2. Index Construction Process

본 논문에서 제안하는 경로 기반 인덱스는 경로 리스트(PList), 인스턴스 리스트(IList), 클래스 리스트(CList)로 구성된다. 경로 리스트는 RDF와 RDF 스키마에서 직·간접적으로 추출한 경로 정보를 저장한다. 인스턴스 리스트는 특정 경로 상에 존재하는 모든 리소스 쌍의 정보를 저장한다. 클래스 리스트는 각 클래스와 관련된 경로 리스트의 노드로의 접근을 제공한다. 그림 2는 본 논문에서 제안한 인덱스의 생성 과정을 보여준다.

경로 기반 인덱스를 구축하기 위해서는 먼저, RDF와 RDF 스키마 문서를 트리플 형태의 문장으로 파싱한 후 파싱된 정보를 분석하여 클래스, 프로퍼티, 인스턴스에 식별자를 부여한다. 다음으로 RDF 스키마 그래프 모델로부터 직·간접적으로 표현된 트리플 구조 기반의 경로 정보를 추출하여 경로 테이블에 임시로 저장한 후 경로 리스트를 구성한다. 이때, 동일한 클래스를 경로 상의 시작점으로 하는 모든 경로 정보들은 서로 연결한다. 마찬가지로 동일한 클래스를 경로 상의 끝나는 점으로 하는 모든 경로 정보들도 서로 연결하여 관련된 경로 정보들에 대한 빠른 검색을 유도한다. 경로 리스트의 구성이 완료되면 각 클래스 별로 해당 클래스를 경로의 시작점으로 하는 경로 리스트 내 첫 번째 노드의 포인터와 해당 클래스를 끝점으로 하는 경로 리스트 내 첫 번째 노드의 포인터로 클래스 리스트를 구성한다. 마지막으로 RDF 문서에서 파싱된 정보로부터 각 경로 상에 존재하는 리소스 쌍을 추출하여 인스턴스 리스트를 구성한다. 이와 같은 과정을 통해 구성되는 경로 기반 인덱스는 해당 경로에 존재하는 모든 리소스 쌍을 쉽게 검색할 수 있고 간접적으로 표현되는 경로에 대한 질의 처리를 지원하는 장점이 있다.

4.2 경로 테이블

RDF 스키마의 그래프 모델로부터 직·간접적으로 추출 가능한 모든 경로는 인덱스를 구축하기 전에 그림 3과 같은 경로 테이블에 임시적으로 저장된다.

1	Person.creates.Artifact
2	Person.creates.Book
3	Author.creates.Artifact
4	Author.creates.Book
5	Author.writes.Book
6	Artifact.cost.Literal
7	Book.cost.Literal
8	Artifact.title.Literal
9	Book.title.Literal
10

그림 3. 경로 테이블 예
Fig. 3. Example of Path Table

그림 3의 경로 테이블은 그림 1의 데이터 모델 예제에서 직·간접적으로 추출한 경로 정보의 일부분을 보여주며 사각형은 "Person.creates.Artifact"의 경로식으로 표현되는 질의를 처리할 때 클래스와 프로퍼티의 계층 관계에 따라 함께 처리해야 하는 관련 경로 정보를 표시한다. 경로 테이블에 하는 관련 경로할 때는 같은 클래스로부터 시작되는 경로 정보나 같은 클래스로 끝이 나는 경로 정보는 클러스터링하여 저장함으로써 처리의 효율성을 높인다.

4.3 3단계 인덱스 구조

그림 4는 본 논문에서 제안한 경로 기반 인덱스의 구조를 보여준다. 그림 4의 경로 기반 인덱스는 3-레이어 구조로 구성되어 있으며 질의 처리 시 단계별로 각 레이어에 접근하게 된다. 최상단에는 각 클래스를 포함하는 경로 리스트 내 노드로의 접근을 위한 클래스 리스트, 중간 레이어에는 RDF와 RDF 스키마에서 직·간접적으로 표현되는 모든 경로 정보를 저장하고 경로 정보 간의 연관 관계를 표현하는 경로 리스트, 최하위 레이어에는 각 경로 상에 존재하는 리소스 쌍의 정보를 저장하는 인스턴스 리스트가 존재한다.

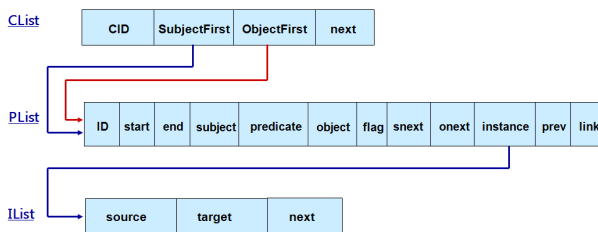


그림 4. 경로 기반 인덱스
Fig. 4. Path-based Index

경로 리스트는 RDF 스키마에서 직접적으로 기술된 경로 정보는 물론 클래스 및 프로퍼티 계층 관계에 따른 상속을 통해 간접적으로 추출 가능한 경로 정보를 모두 저장한다. RDF 스키마 모델에 정의된 경로 정보는 RDF에서 인스턴스화되기 때문에 RDF 스키마에서 추출한 경로 정보와 RDF에서 추출한 경로 정보는 동일하다. 따라서 본 논문에서는 RDF 스키마만을 대상으로 하여 경로 정보를 추출한다. 경로 리스트의 노드는 경로 정보의 저장을 위한 7개의 필드와 노드 간 연결을 위한 5개의 포인터 필드로 구성된다. 각 필드의 역할은 표 1에서 설명한다.

경로 리스트에서 관리하는 정보를 이용해 특정 클래스로부터 시작하는 경로 기반의 질의나 특정 클래스로 끝나는 경로 기반의 질의를 빠르게 처리할 수 있다. 그리고 클래스 및 프로퍼티의 계층 관계에 따른 상속의 결과로 질의 처리 시 함께

고려해야 하는 경로 정보 영역에 대한 접근을 가능하게 한다. 만약, 현재 경로 노드의 ID가 start 필드의 값과 동일한 경우는 end 필드가 지시하는 경로 노드까지를 질의 처리의 범위로 설정한다. 그리고 현재 경로 노드의 ID가 start 필드의 값보다 큰 경우는 현재 경로 노드에 대한 질의를 처리할 때 start 필드가 유지하고 있는 ID를 가진 경로 노드로부터 end 필드가 유지하고 있는 ID를 가지고 있는 경로 노드까지를 질의 처리의 범위로 설정하면 된다.

표 1. 경로 리스트의 노드 구조
Table 1. Node Structure of PList

필드	역할
ID	경로 리스트 내 각 경로 정보에 대한 식별자
start	질의 처리 시 함께 처리해야 하는 경로 영역 중 첫 번째 경로 노드의 ID
end	질의 처리 시 함께 고려해야 하는 경로 영역 중 마지막 경로 노드의 ID
subject	해당 경로를 시작하는 클래스(소스 클래스)의 ID
predicate	해당 경로에서 관계를 표현하는 프로퍼티
object	해당 경로로부터 도달하게 되는 클래스(타겟 클래스)의 ID
flag	해당 경로의 타겟 클래스가 리터럴인지 표시
snext	같은 소스 클래스로 시작하는 경로 노드들을 연결하는 포인터
onext	같은 타겟 클래스로 도달하는 경로 노드들을 연결하는 포인터
instance	해당 경로 상에 존재하는 모든 리소스 쌍을 유지하는 인스턴스 리스트로 연결하는 포인터
prev	경로 리스트 내 이전 경로 노드와 연결하는 포인터
link	경로 리스트 내 다음 경로 노드와 연결하는 포인터

인스턴스 리스트는 특정 경로 상에 존재하는 소스 리소스와 타겟 리소스로 구성된 모든 리소스 쌍의 정보를 관리한다. 인스턴스 리스트의 노드는 세 개의 필드로 구성되어 있으며 표 2에서 각 필드의 역할을 설명한다.

표 2. 인스턴스 리스트의 노드 구조
Table 2. Node Structure of IList

필드	역할
source	해당 경로 상에서 소스 클래스의 인스턴스인 리소스의 아이디(RID)
target	해당 경로 상에서 타겟 클래스의 인스턴스인 리소스의 아이디(RID)
next	같은 경로 상에 존재하는 다음 리소스 쌍을 연결하는 포인터

같은 소스 클래스나 같은 타겟 클래스를 포함하는 경로 리스트 내의 노드를 연결하기 위해서는 클래스 리스트가 필요하

다. 클래스 리스트의 노드는 4개의 필드로 구성되어 있으며 표 3에서 각 필드의 역할을 설명한다.

표 3. 클래스 리스트의 노드 구조
Table 2. Node Structure of CList

필드	역할
CID	RDF 스키마에 정의된 클래스의 식별자
SubjectFirst	해당 클래스를 소스 클래스로 하는 경로 리스트 내 첫 번째 노드에 대한 포인터
ObjectFirst	해당 클래스를 타겟 클래스로 하는 경로 리스트 내 첫 번째 노드에 대한 포인터
next	클래스 리스트 내 다음 노드를 연결하는 포인터

본 논문에서 제안한 인덱스는 RDF와 RDF 스키마를 모두 고려하기 때문에 클래스 타입이 이미 정해진 리소스에 관련된 질의 처리가 가능하면서도 클래스와 프로퍼티의 계층 관계를 통해 추론될 수 있는 새로운 정보에 대한 질의 처리가 가능하기 때문에 시맨틱 웹의 개념에 적합하다.

4.4 질의 처리 전략

본 논문에서 제안한 인덱스를 이용한 질의 처리는 다음과 같이 3단계를 거쳐 진행된다.

- STEP 1 : 사용자의 질의를 분석한 후 경로식으로 변환하여 경로를 시작하는 소스 클래스와 경로가 마무리되는 타겟 클래스를 결정한다.
- STEP 2 : 결정된 소스 클래스나 타겟 클래스에 관련된 노드를 클래스 리스트에서 검색한다.
- STEP 3 : 클래스 리스트에서 검색된 클래스와 연결되어 있는 경로 리스트의 모든 노드를 순회하면서 각 노드에 연결되어 있는 인스턴스 리스트의 노드들을 차례대로 방문하여 경로 상에 존재하는 모든 리소스의 쌍을 검색한다.

그림 5는 "Person.create.Artifact"의 경로 상에 존재하는 모든 리소스 쌍을 검색하는 질의의 처리 과정을 간단하게 보여준다.

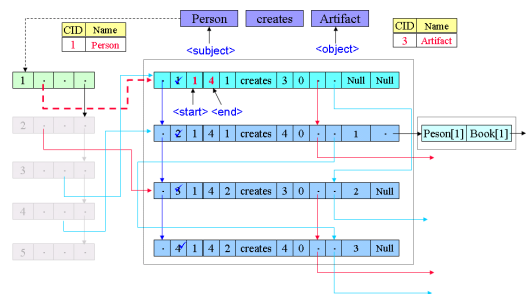


그림 5. 질의 처리 예
Fig. 5. Example of Query Processing

질의를 처리하기 위해 먼저 인덱스 구조의 첫 번째 레이어에 존재하는 클래스 리스트에서 "Person" 클래스에 대한 정보를 제공하는 노드를 검색한다. 검색된 노드에서 SubjectFirst 필드를 이용해 "Person" 클래스를 소스 클래스로 포함하는 경로 리스트 내에 첫 번째 경로 노드를 검색한다. 검색된 경로 노드의 ID 필드와 start 필드를 검사하여 ID 필드의 값과 start 필드의 값이 같은 경우에는 현재 경로 노드에서부터 end 필드가 가리키는 경로 노드까지의 영역에 존재하는 모든 경로 노드를 순차적으로 방문하면서 instance 필드가 가리키고 있는 해당 경로 상의 리소스 쌍 정보를 모두 검색하면 질의 처리의 결과가 얻어진다. 만약 경로 노드의 ID 필드와 start 필드를 검사하여 ID 필드의 값과 start 필드의 값이 다른 경우는 start 필드가 가리키는 경로 노드로 이동하여 질의 처리를 시작해야 한다.

"?.creates.Artifact"와 같이 경로 상의 소스 클래스가 질의 조건으로 주어지지 않은 경우에는 "Artifact" 클래스를 타겟 클래스로 포함하는 경로 노드들을 먼저 검색하여 질의 결과를 구성하면 된다.

V. 성능 평가

본 논문에서 제안한 인덱스는 RDF 스키마에 정의된 클래스 및 프로퍼티의 계층 관계에 의한 상속까지 고려한 2차 질의를 보다 효율적으로 처리하는데 목표를 두고 있다. 따라서 본 논문에서 제안한 인덱스는 빠른 시간 내에 1차 질의만으로는 검색이 불가능한 RDF와 RDF 스키마 정보를 모두 검색할 수 있는 장점이 존재한다. 이러한 인덱스의 성능 개선 효과를 평가하기 위해서 기존에 RDF와 RDF 스키마를 저장하는 대표적인 방식인 수직-분할 방식과 하이브리드 방식의 저장 시스템을 비교 대상으로 선택하여 질의 처리 시간과 반환되는 검색 결과의 개수를 비교하였다.

본 논문에서 제안한 인덱스와 비교 대상 저장 시스템은 펜티엄4 3.0GHz의 CPU와 1GB 메모리를 가지고 Linux 9.1 과 오라클 9i가 설치되어 있는 컴퓨터에서 C 언어로 구현하였다. 실험 데이터는 Wordnet에서 제공하는 RDF와 RDF 스키마 문서를 사용하였다. Wordnet은 온라인 어휘 참조 시스템으로 영어 단어의 품사를 명사, 동사, 형용사, 부사로 나누고 그들이 원래 가지고 있는 의미에 따라 동의어, 하위어 등의 의미적 관계를 정의한 것이다[15]. Wordnet의 RDF 스키마는 6개의 클래스와 5개의 프로퍼티를 정의하고 있다. Wordnet을 실험 데이터로 선택한 이유는 RDF와 RDF 스키마 문서가 모두 존재하고 충분한 크기의 데이터를 포함하고 있으며 많은 관련 연구에서 실험 데이터로 활용되어 이미 검

증된 데이터이기 때문이다. 데이터 크기의 변화에 따른 성능을 평가하기 위하여 Wordnet에서 제공하는 RDF 문서를 크기가 다른 3개의 파일로 나누어 실험을 수행하였다. 각 실험 데이터 파일에는 서로 다른 개수의 트리플 문장이 존재한다. 표 4는 실험에 사용된 데이터에 대한 추가적인 정보를 보여주고 있다.

표 4. 실험 데이터
Table 4. Experimental Data

분류	Data1	Data2	Data3
RDF 스키마 문서의 크기	4KB	4KB	4KB
RDF 문서의 크기	587KB	2,125KB	21,199KB
트리플 형태 문장의 수	4,198	15,059	149,081

본 논문에서는 시맨틱 웹의 개념을 가장 잘 반영하고 있는 RDF와 RDF 스키마의 혼합 질의 처리를 지원하는 것을 목표로 하고 있지만 지금까지 기존 연구들에서 많은 관심을 가져온 RDF 스키마 중심 질의 처리와 관련하여 본 논문에서 제안한 인덱스의 효율성을 확인하기 위해 RDF 스키마 중심 질의를 포함하여 표 5에서 제시한 5가지 질의를 평가 대상으로 한다.

표 5. 실험 질의
Table 5. Experimental Queries

분류	질의
RDF 스키마 중심 질의	Q1 "LexicalConcept" 클래스의 서브 클래스 검색 (?X subClassOf LexicalConcept)
	Q2 "Adjective" 클래스의 서브 클래스 검색 (?X subClassOf Adjective)
	Q3 "wordForm" 프로퍼티의 도메인 클래스 검색 (wordForm domain ?X)
혼합 질의	Q4 "hyponymOf" 프로퍼티로 의미적 관계를 맺는 LexicalConcept 클래스의 인스턴스 쌍 검색 (LexicalConcept hyponymOf LexicalConcept) and (?X typeOf LexicalConcept) and (?Y typeOf LexicalConcept)
	Q5 "wordForm" 프로퍼티로 의미적 관계를 맺는 "LexicalConcept" 클래스의 인스턴스와 리터럴 값의 쌍 검색 (LexicalConcept wordForm Literal) and (?X typeOf LexicalConcept) and (?Y typeOf Literal)

참고 문헌 [6]에서 RDF와 RDF 스키마를 저장하는 수직-분할 방식, 바이너리 방식, 하이브리드 방식에 대해 실험을

통해 하이브리드 방식이 가장 우수한 성능을 가지고 있음을 보였다. 따라서 본 논문에서는 표 5에서 제시한 5가지 질의 유형에 대해 일반적인 저장 방식인 수직-분할 방식을 채택하고 있는 Jena와 하이브리드 방식의 저장 구조, 그리고 제안 인덱스의 처리 성능을 비교하여 개선 효과를 보이고자 한다.

표 6은 데이터 파일 1을 대상으로 하여 표 5에서 제시한 질의 유형 중 RDF 스키마 중심의 질의 유형에 속하는 Q1, Q2, Q3을 처리한 결과로 반환된 리소스의 개수와 처리 시간에 대해 Jena, 하이브리드 시스템, 제안 인덱스를 비교한 결과이다. 제안 인덱스는 클래스, 프로퍼티, 리소스 개체를 구별하기 위해 레이블링 기법을 사용하고 2차 질의 처리를 지원하기 때문에 다른 비교 대상 시스템들에 비해 같은 처리 시간 내에 질의 조건에 부합하는 모든 리소스를 검색할 수 있는 장점이 있다. 표 6에서 확인할 수 있는 것처럼 기존 저장 시스템들은 1차 질의 처리를 대상으로 하기 때문에 클래스 및 프로퍼티의 계층 관계를 통해 추론될 수 있는 정보들은 질의 조건을 만족함에도 불구하고 검색 결과로 반환되지 않지만 제안 인덱스를 이용하면 이러한 문제를 해결할 수 있다.

표 6. Q1, Q2, Q3에 대한 질의 처리 결과
Table 6. Query Processing Results of Q1, Q2, Q3

분류	Jena		Hybrid		제안 인덱스	
	검색 결과	검색 시간	검색 결과	검색 시간	검색 결과	검색 시간
Q1	1	0.0000001	1	0.0000001	5	0.0000001
Q2	1	0.0000001	1	0.0000001	1	0.0000001
Q3	1	0.0000001	1	0.0000001	6	0.0000001

표 7은 세 가지 크기의 데이터 파일에 대해 RDF와 RDF 스키마를 혼합한 질의 Q4와 Q5의 처리로 반환되는 검색 결과의 개수와 초 단위의 처리 시간을 비교한 결과이다. Q4와 Q5의 경우 제안 인덱스를 이용하면 다른 비교 대상 시스템들에 비해 대부분 빠른 질의 처리 시간이 소요되는 것을 확인할 수 있다. 그리고 1차 질의만을 처리의 대상으로 하는 다른 시스템들에 비해 2차 질의까지 고려하는 제안 인덱스는 추론에 의해 질의 조건을 만족시키는 모든 결과 리소스를 반환한다. 특히 Q4와 Q5의 질의가 요구하는 리소스는 그래프 모델에서 직접적으로 표현되는 정보가 아니기 때문에 비교의 대상이 되는 저장 시스템들에서는 반환되는 결과가 없는 것을 확인할 수 있다. 따라서 반환되는 검색 결과의 크기를 고려할 때 다른 시스템에 비해 제안 인덱스의 성능이 정확성 측면과 처리의 효율성 측면에서 모두 우수하다고 판단할 수 있다.

표 7. Q4, Q5에 대한 질의 처리 결과
Table 7. Query Processing Results of Q4, Q5

분류		Jena		Hybrid		제안 인덱스	
		검색 결과	검색 시간	검색 결과	검색 시간	검색 결과	검색 시간
Q4	Data1	0	0.05	0	0.02	656	0.00
	Data2	0	0.05	0	0.03	2,485	0.01
	Data3	0	0.50	0	0.55	24,061	0.03
Q5	Data1	0	0.00	0	0.05	681	0.00
	Data2	0	0.00	0	0.16	3,876	0.01
	Data3	0	0.00	0	1.32	37,652	0.07

그림 6과 그림 7은 Q4, Q5 질의에 대한 표 7의 결과를 그래프로 나타낸 것으로 제안 인덱스를 사용하는 경우 실험 데이터의 크기가 커져도 유사한 검색 성능을 보인다는 것을 확인할 수 있다. 특히, 실험 데이터의 크기가 커지면 반환되는 검색 결과의 수가 비교 대상 시스템에 비해 크게 증가함에도 불구하고 처리 시간은 크게 증가하지 않아 제안 인덱스의 확장성이 높다고 판단할 수 있다.

RDF 스키마 중심 질의 뿐만 아니라 RDF와 RDF 스키마의 혼합 질의 유형까지 기존의 저장 시스템들에 비해 제안 인덱스가 빠른 처리 시간을 보이고 있지만 인덱스의 구축을 위한 과정이 복잡한 문제점을 가지고 있다. 하지만 데이터 크기에 따른 시스템 별 데이터 적재 시간을 측정해본 결과 데이터 파일의 크기가 커지더라도 제안 인덱스를 구축하는데 소요되는 시간은 감수할 수 있을 정도이며 시스템 구축이 자주 반복되지 않기 때문에 실제 시맨틱 웹 애플리케이션 개발 시 제안 인덱스의 활용이 가능하다고 판단된다.

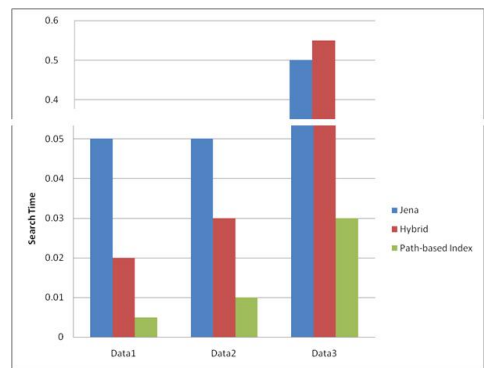


그림 6. Q4의 질의 처리 시간 비교
Fig. 6. Comparison of Search Time for Q4

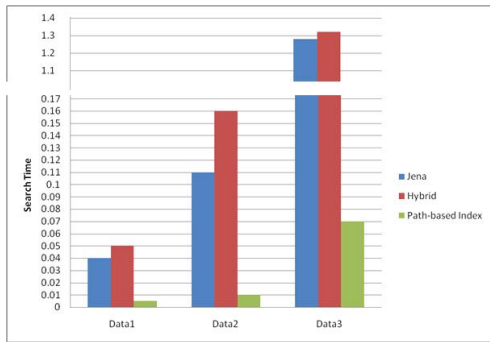


그림 7. Q5의 질의 처리 시간 비교
Fig. 7. Comparison of Search Time for Q5

VI. 결론

차세대 웹의 대안으로 인식되고 있는 시맨틱 웹에서는 정보 리소스의 의미와 개념적 관계를 정의하는 메타데이터와 온톨로지의 역할이 무엇보다 중요시되고 있다. 특히, 시맨틱 웹 언어 중 일반적으로 많이 활용되는 RDF와 RDF 스키마로 표현되는 메타데이터와 온톨로지를 어떻게 저장하고 어떻게 질의할 것인가에 대한 연구는 핵심적인 이슈라 할 수 있다. 그러나 기존 연구 결과물들은 RDF와 RDF 스키마의 근본적인 차이점은 간과하고 트리플 구조에 기반하여 RDF와 RDF 스키마 데이터를 단순화하여 저장하고 검색하는 것을 목표로 하는 경우가 많았다. 특히, 온톨로지를 통해 추론 가능한 정보들을 저장 후에도 활용할 수 있어야 하는데 이러한 정보를 질의 처리에 효율적으로 이용할 수 있도록 하는 인덱싱 기법에 대한 연구가 부족한 경향이 있었다. 따라서 본 논문에서는 RDF와 RDF 스키마로 기술되는 메타데이터와 온톨로지 정보를 저장하고 추론에 기반한 효율적인 질의 처리를 지원하는 인덱싱 기법을 제안하였다. 본 논문에서 제안한 인덱스는 문서적 관점에서 RDF 스키마로부터 1차적으로 추출 가능한 구조적 경로 정보는 물론 의미적 관점에서 2차적으로 추출 가능한 구조적 경로 정보를 질의 처리의 대상으로 하기 때문에 시맨틱 웹의 개념을 온전히 반영하고 있다고 할 수 있다. 그리고 실험적 평가를 통해 제안 인덱스가 정확성은 물론 시맨틱 웹 상의 다양한 응용 시스템에 실제로 적용할 수 있는 효율성과 확장성을 가지고 있음을 증명하였다.

참고문헌

- [1] T. Berners-Lee, J. Hendler, and O. Lassila, "The Semantic Web," *Scientific American*, Vol. 284, No. 5, pp.34-43, May, 2001.
- [2] J. Hendler, T. Berners-Lee and E. Miler, "Integrating Applications on the Semantic Web," *Journal of the Institute of Electrical Engineers of Japan*, Vol. 122, No. 10, pp. 676-680, October 2002.
- [3] RDF, <http://www.w3.org/TR/rdf-primer/>
- [4] RDF Schema, <http://www.w3.org/TR/rdf-schema/>
- [5] 이순미, "시맨틱 웹 문서를 위한 관계형 저장 스키마 설계 및 질의 처리 기법," 한국컴퓨터정보학회논문지, 제 14권, 제 1호, 35-45쪽, 2009년 1월.
- [6] Y. Theoharis, V. Christophides, and G. Karvounarakis, "Benchmarking Database Representations of RDF/S Stores," In Proc. of the ISWC, pp. 685-701, November 2005.
- [7] B. McBride, "Jena : A Semantic Web Toolkit," *IEEE Internet Computing*, Vol 6, No. 6, pp.55-59, November 2002.
- [8] L. Ma, Z. Su, Y. Pan, L. Zhang, and T. Liu, "Rstar: An RDF Storage and Query System for Enterprise Resource Management," In Proc. of the CIKM, pp. 484-491, 2004.
- [9] J. Broekstra, A. Kampman, and F. van Harmelen, "Sesame: A Generic Architecture for Storing and Querying RDF and RDF Schema," In Proc. of the ISWC, LNCS 2342, pp. 54-68, June 2002.
- [10] Z. Pan, and J. Heflin, "DLDB: Extending Relational Databases to Support Semantic Web Queries," In Proc. of the PSSS, Vol. 89, pp. 109-113, October 2003.
- [11] S. Alexaki, V. Christophides, G. Karvounarakis, D. Plexousakis and K. Tolle, "The *RDFSuite: Managing Voluminous RDF Description Basés*" Technical Report, Institute of Computer Science FORTH, 2000.
- [12] A. Harth and S. Decker, "Yet Another RDF Store: Perfect Index Structures for for Storing

Semantic Web Data With Contexts,” DERI Technical Report, 2004.

[13] A. Matono, T. Amagasa, M. Yoshikawa, and S. Uemura, “An Indexing Scheme for RDF and RDF Schema based on Suffix Arrays,” In Proc. of the SWDB, pp.151-168, September 2003.

[14] 김연희, 신혜연, 임해철, 정균락, “시맨틱 웹 데이터의 키워드 질의 처리를 위한 인덱싱 및 저장 기법,” 한국컴퓨터정보학회논문지, 제 12권, 제 5호, 93-102쪽, 2007년 11월.

[15] Wordnet, <http://www.w3.org/TR/wordnet-rdf/>

저 자 소 개



김 연 희
 2000: 홍익대학교 공학사
 2002: 홍익대학교 공학석사
 2006: 홍익대학교 공학박사
 현재: 부천대학 e-비즈니스과
 강의전담교수
 관심분야: 시맨틱 웹, XML, 분산 데이터베이스



김 지 현
 1978: 이화여자대학교 수학과 학사
 1994: 단국대학교 전자 정보 전공
 경영학석사
 2004: 단국대학교 전산통계학과
 이학박사
 1997: 정보관리 기술사
 현재: 서일대학 소프트웨어과 부교수
 관심분야: 웹 공학, 데이터베이스, 품질 관리