

시맨틱 웹 데이터를 위한 키워드 인덱싱 기법

신혜연[○], 김연희*, 정균락*, 임해철*

*홍익대학교 컴퓨터공학과

hyshin[○]@mail.hongik.ac.kr, {kyh, chong, liim}@cs.hongik.ac.kr

Indexing Scheme for Keyword-based Query Processing on Semantic Web

Hye Yeon Shin[○], Youn Hee Kim*, Kyun Rak Chong*, Hae Chull Lim*

*Dept. of Computer Engineering, Hong Ik University

요 약

시맨틱 웹은 현재 웹의 확장된 개념으로 사람뿐만 아니라 컴퓨터 스스로가 데이터를 이해하고 처리할 수 있도록 정보에 의미를 부여하는 것이다. 시맨틱 웹 데이터를 기술하는 RDF를 통해 메타데이터를 표현하고 의미론적 추론이 가능하게 되었다. 따라서 기존에 일반 사용자가 쉽게 사용할 수 있는 키워드 검색 방법을 시맨틱 웹 데이터인 RDF/RDF 스키마에 적용함으로써 차세대 웹으로 인식되고 있는 시맨틱 웹을 일반 사용자도 쉽게 활용할 수 있도록 한다. 본 논문에서는 RDF 문서의 효율적인 검색을 위해 RDF 인스턴스와 RDF 스키마 정보를 저장하고, 키워드, 속성, 클래스 타입의 복합 조건 검색을 만족시키는 키워드 인덱스와 스키마 테이블 구조를 제안한다. 본 논문에서 제안한 구조는 다양한 조건들을 만족하는 리소스 정보의 빠르고 정확한 검색이 가능하도록 한다.

1. 서 론

시맨틱 웹이란 현재 웹의 확장된 개념으로 사람뿐만 아니라 컴퓨터가 이해 할 수 있는 웹을 의미한다. 시맨틱 웹에 관한 연구와 표준화 작업이 이루어지고 있는 W3C[1]에서 문서의 출력형식이 아닌 문서의 구조를 기술하는 XML(eXtensible Markup Language)[2]의 등장 이후, 시맨틱 웹을 구현하기 위해 메타데이터를 표현하는 RDF(Resource Description Framework)[3] 등의 온톨로지 언어들이 발전하였다. 웹에서의 메타데이터의 목적은 사용자가 원하지 않는 데이터를 미리 걸러주어 정보검색의 처리 과정을 줄여줌으로써 관련성이 많은 정보의 발견 가능성을 높여 정보검색의 질을 향상시키기 위한 것이다[6]. 정보 검색을 위해 현재 인터넷 검색엔진에서 널리 사용되고 있는 키워드 검색 방법은 간단한 키워드 입력을 통하여 원하는 정보를 찾을 수 있어 일반 사용자들이 선호하는 검색방법이다. 이러한 키워드 검색을 메타데이터를 기술하는 리소스, 속성, 속성값의 RDF 뿐만 아니라 RDF에 적용된 클래스와 속성, 클래스들간의 관계를 정의한 RDF 스키마까지 고려하여 적용함으로써 시맨틱 웹을 일반 사용자도 쉽게 활용할 수 있도록 한다.

예를 들면, "교보"에서 제공하는 책과 관련된 리소스를 검색 한다고 가정해보자. 이런 질의는 "교보"라는 키워드를 가지고 관련 리소스를 검색하는 것 보다 키워드 조건 검색에 RDF 스키마 정보인 속성이나 클래스 타입 조건을 추가하여 검색함으로써 좀 더 정확한 리소스 정보를 얻을 수 있게 된다.

본 논문은 RDF와 RDF 스키마 정보를 저장하여 단순한

키워드 검색이 아닌 키워드, 속성, 클래스 타입의 조건을 복합적으로 이용하여 관련된 리소스들을 빠르고 정확하게 검색하기 위한 구조를 제안한다.

본 논문에서 제안하는 검색 방법은

- 역 인덱스 구조를 이용하여 사용자가 입력한 키워드와 속성과 관련된 리소스들을 빠르게 찾는다.
- RDF 스키마에 정의된 클래스 타입 정보와 클래스 계층 정보를 이용하여 단순 키워드 검색 보다 정확한 검색이 가능하다.
- 키워드를 직접 포함하는 리소스뿐만 아니라 의미적인 연관성을 가지고 있는 간접 포함하는 리소스도 함께 검색이 가능하다.

본 논문의 구성은 다음과 같다. 2장에서는 RDF/RDF 스키마의 기본 개념, 역 인덱스의 구조, 기존 웹 환경에서의 키워드 검색의 한계를 살펴본다. 3장에서는 RDF/RDF 스키마에 관한 질의 유형과 질의처리를 위한 인덱스 구조를 제안한다. 4장에서는 결론 및 향후 연구 방향을 제시한다.

2. 관련 연구

시맨틱 웹 데이터의 의미를 기술할 수 있는 RDF의 등장 이후, RDF로 작성된 문서들이 늘어나게 되었고, 사용자들은 문서의 내용만을 검색하기보다 메타데이터의 정보에 관심을 가지기 시작하였다. 예를 들면, "삼국지"라는 책의 내용 검색과 더불어 책의 저자, 출판사와 같은 메타데이터 정보 검색을 원하게 된 것이다. 따라서 본 논문에서는 RDF 형태로 작성된 문서들에 대한 키워드 중심의 다양한 질의 처리를 수행하기 위해 역 인덱스 구조와 스키마 정보를 저장한 테이블을 이용하

본 연구는 한국과학재단 특정기초연구(과제번호 : R01-2004-000-10586-0(2006))의 지원으로 수행되었음

여 기존의 인덱스 구조를 이용한 검색의 속도와 정확성을 향상시키고자 하는데 초점을 둔다.

2.1 RDF/RDF 스키마

RDF는 웹 리소스들의 메타데이터를 기술하기 위한 모델로 (subject, predicate, object)의 트리플 구조로 작성된다.

RDF 스키마는 RDF의 리소스 개념을 나타내는 클래스와 클래스의 속성과 타입, 클래스들간의 계층구조를 정의해 놓은 것이다.

리소스의 메타데이터 기술이 가능한 RDF와 클래스, 속성, 클래스들간의 계층관계를 정의해 놓은 RDF 스키마의 특징을 함께 고려한 다양한 검색이 요구된다.

2.2 역 인덱스

기존의 인덱스 기법을 이용한 검색은 RDF/RDF 스키마의 구조적 질의에 초점을 맞춘 것으로 단일 트리플 정보 검색만 가능하다. 본 연구에서는 기존 연구와는 달리, 키워드 검색을 통하여 주어진 키워드를 포함하는 리소스와 속성의 정보를 얻을 수 있는 효율적인 인덱스 구조를 제안하고자 한다.

역 인덱스[5]는 데이터 파일에 나타나 있는 키 값을 인덱스의 키로 모두 포함 하는 방법으로 검색엔진에서 질의처리를 위해 이용되는 가장 일반적인 구조이다. 역 인덱스는 인덱스 파일, 포스팅 파일, 데이터 파일로 구성된다.

- 인덱스 파일은 키워드 리스트와 키워드를 포함한 레코드 수, 포스팅에 대한 포인터로 구성된다.
- 포스팅 파일은 키워드를 포함하는 데이터 레코드에 대한 포인터 리스트로 구성된다.
- 데이터 파일은 실제 문서 파일들로 구성 된다.

본 논문에서는 역 인덱스의 구조를 이용하여 키워드 검색을 수행한다.

웹에서의 정보 검색 방법 중 하나인 키워드 검색 방법은 사용자가 찾고자 하는 정보와 관련된 핵심적인 언어인 키워드를 직접 입력하여 검색엔진으로부터 키워드를 포함하고 있는 정보를 찾는 방식이다. 현재 웹은 수많은 정보들로 이루어져 있기 때문에 주어진 키워드를 포함하는 많은 양의 웹 문서들이 검색 결과로 나와 사용자의 판단을 흐리게 할 뿐만 아니라 사용자는 자신이 원하는 정보인지 확인하기 위해 일일이 검색 결과의 리소스 링크를 클릭해야하기 때문에 사용자가 원하는 정확한 정보를 빠른 시간 내에 찾기가 쉽지 않다. 또한 특정 키워드만의 검색에 의한 결과이므로 웹 문서의 신뢰도가 낮다는 단점을 가지고 있다.

본 논문은 역 인덱스 구조를 이용하여 웹에서의 검색 방법 중 하나인 키워드 검색을 RDF 문서에 적용한다. 이러한 검색은 메타데이터 내 사용자가 입력한 키워드를 포함하고 있는 직접 포함하는 리소스뿐만 아니라 의미적 연결성을 고려한 간접 포함 리소스도 질의 처리 대상으로 하여 키워드를 포함하는 세부적인 정보를 얻을 수 있게 된다. 더 나아가 RDF스키마에 정의된 클래스 정보를 이용하여 확장된 키워드 검색이 가능한 구조를 제안함으로써 추론과 다양한 검색으로 보다 정확한

정보를 얻을 수 있다.

3. 구조와 질의 처리과정

3.1 검색 조건과 질의 유형 분류

사용자들의 보다 다양하고 정확한 질의처리를 위해 <표 1>과 같이 검색 조건을 키워드, 속성, 키워드와 속성의 복합, 키워드와 클래스 타입의 복합 조건으로 분류하였으며, 검색 조건에 따라 질의유형도 4가지로 분류하였다.

표 1. 검색조건과 질의 유형 분류

| 검색 조건 | 질의 유형 |
|----------|-----------------------------------|
| 키워드 | 주어진 키워드를 직접, 간접적으로 포함하는 모든 리소스 검색 |
| 속성 | 주어진 속성을 포함하는 모든 리소스 검색 |
| 키워드 + 속성 | 주어진 키워드와 속성을 포함하는 리소스 검색 |
| 키워드 + 타입 | 주어진 키워드와 타입을 포함하는 리소스 |

RDF 문서 검색에 관련된 기존연구를 보면 키워드에 대한 검색 결과로 리소스 중심, 속성 중심의 단일 조건으로 분류하였다[7]. 그러나 본 논문에서는 검색의 범위를 RDF에서 타입의 계층, 클래스와 프로퍼티, 클래스들 간의 관계를 정의한 RDF 스키마까지 확대하여 다양하고 정확한 검색을 위해 검색 조건을 4가지로 분류한다.

- 첫 번째로 키워드만 주어지는 경우, 사용자가 입력한 키워드를 직접, 간접적으로 포함하는 모든 리소스를 검색한다.
- 두 번째는 속성만 주어지는 경우, RDF 문서 내에 주어진 속성과 연결된 모든 리소스를 검색한다.
- 세 번째는 키워드와 속성의 복합 조건으로 1단계 키워드 조건 검색 후 2단계 속성 조건 검색 결과를 조인하여 반환한다.
- 네 번째 조건은 키워드와 클래스 타입의 복합 조건으로 1단계 키워드 조건 검색 후 2단계 RDF 스키마를 저장한 테이블로부터 클래스 타입 조건을 만족하는 결과를 조인하여 반환한다.

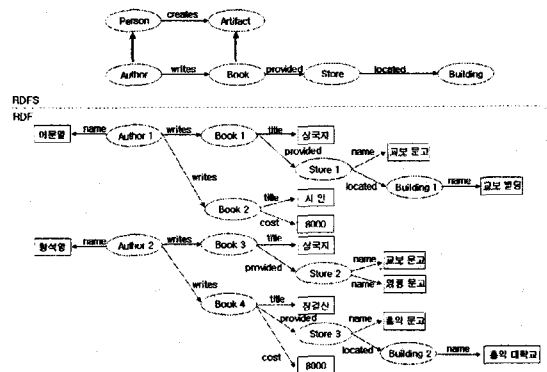


그림 1. RDF/S 그래프

3.2.1 키워드 인덱스

키워드 인덱스는 역 인덱스 구조를 이용한 것으로 RDF 문서로부터 키워드를 추출하여 구성되었다. 키워드 인덱스를 통해 다양한 질의처리를 수행할 수 있고, 검색하고자 하는 키워드와 관련된 리소스의 정보를 빠르고 쉽게 얻을 수 있게 해 준다.

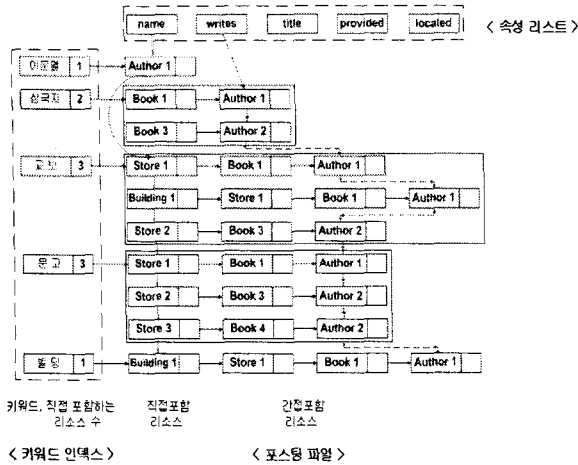


그림 2. 키워드 인덱스 구조

<그림 2>의 키워드 인덱스는 <그림 1>의 RDF/RDF 스키마 문서로부터 키워드와 속성을 추출하여 작성한 구조이다.

본 논문에서 제안한 인덱스 구조는 키워드 인덱스, 포스팅 파일, 속성 리스트 세 부분으로 구성되었다.

키워드 인덱스는 키워드와 키워드를 직접 포함하는 리소스의 수로 구성되어 결과를 반환할 때 리소스 수가 많은 키워드 순으로 검색 결과를 반환하여 키워드의 중요도와 빈도수를 파악할 수 있다.

포스팅 파일은 주어진 키워드를 직접 포함하는 리소스와 간접포함 하는 리소스를 링크드 리스트 순서대로 표현한다. 여기서 직접포함 리소스는 RDF 인스턴스에서 주어진 키워드와 직접적으로 연결된 리소스를 의미하며 간접포함 리소스는 직접 포함 리소스를 제외하고 키워드와 연결된 리소스의 수를 최대 5개로 정의한다. 이것은 검색 시 키워드와 관련된 리소스의 경로 길이를 4 이하로 간접 포함 리소스의 수를 한정하여 구현 시 오버헤드의 발생을 막는다.

향후 연구에서 랭킹기법을 도입하여 일정한 가중치 값에 따라 간접 포함 리소스의 범위를 제한하도록 한다. 직접포함 리소스와 간접포함 리소스에 대해 예를 들어 살펴보면, "빌딩"라는 키워드를 지니는 리소스를 검색하고자 할 때, <그림 2>에서 "빌딩"과 관련된 리소스들 중 "Building 1"과 같이 링크드 리스트의 첫 번째 부분을 키워드를 직접 포함하는 리소스라고 하고 그를 제외한 연결된 나머지 리소스들 "Store 1", "Book 1", "Author 1"을 간접 포함하는 리소스라고 한다. 기존의 RDF 문서에 대한 키워드 검색 연구에서는 키워드를 직

접적으로 포함하고 있는 리소스를 결과로 반환하는 질의처리에만 초점을 맞추어 간접 포함하는 리소스들은 고려하지 않았다. 그러나 본 논문에서는 주어진 키워드를 간접적으로 포함하는 리소스도 고려함으로써 의미적 연관성을 가진 리소스 검색이 가능하게 되어 주어진 키워드의 세부적인 정보들까지도 얻을 수 있다.

속성 리스트는 RDF의 모든 속성들이 나열되고, 각각의 속성들과 관계가 있는 리소스들과 연결된다.

<그림 2>의 키워드 인덱스는 <표 1>의 검색 조건 중 키워드, 속성의 단일 조건과 키워드와 속성의 복합 조건을 만족시키는 구조로 키워드 인덱스는 키워드와 키워드를 포함하는 리소스에 대한 포스팅 파일의 포인터를 가리키고 있으며 포스팅 파일은 키워드를 포함하는 직접, 간접 리소스들의 연결하고 있어 주어진 키워드, 속성 또는 키워드와 속성을 동시에 포함하는 모든 리소스를 빠른 시간 내에 검색할 수 있게 해 준다.

3.2.2 클래스, 타입 테이블

| 타입 | 클래스 아이디 | 서브클래스 | 깊이 |
|----------|---------|-------|----|
| Person | c1 | c5 | 1 |
| Artifact | c2 | c4 | 1 |
| Author | c3 | c3 | 0 |
| Book | c4 | c4 | 0 |
| Store | c5 | c5 | 0 |
| Building | c6 | c6 | 0 |

클래스 테이블

| 리소스 | 클래스 아이디 |
|------------|---------|
| Author 1 | c3 |
| Book 1 | c4 |
| Book 2 | c4 |
| Store 1 | c5 |
| Building 1 | c6 |
| Author 2 | c3 |
| Book 3 | c4 |
| Book 4 | c4 |
| Store 2 | c5 |
| Store 3 | c5 |
| Building 2 | c6 |

타입 테이블

그림 3. RDF 스키마 정보를 저장한 테이블

<그림 3>은 RDF 스키마 정보를 저장한 클래스 테이블과 타입 테이블을 보여준다.

<그림 3>의 클래스, 타입 테이블은 RDF 인스턴스에 국한되어 키워드와 속성에만 한정되어 있는 검색을 확장하여 RDF 스키마까지 포함하는 키워드와 클래스 타입의 복합 조건의 검색을 지원한다.

클래스 테이블은 클래스 타입과 클래스 아이디, 서브 클래스 아이디, 깊이로 구성된다. 클래스 테이블의 타입 필드는 클래스 이름을 나타낸다. 클래스 아이디는 RDF 스키마에서 각 클래스의 식별자이고, 서브 클래스 아이디는 주어진 클래스 타입에 직접 연결된 서브 클래스로써 클래스 계층구조를 이용한 이행적 추론이 가능하게 한다. 깊이 필드를 통해 클래스들의 계층구조를 알 수 있고, 클래스 타입에 종속된 서브 클래스 수를 확인할 수 있기 때문에 깊이 수에 따라 조인이 몇 번 일어나는지 미리 예측할 수 있다. 따라서 조인 횟수에 의한 연산의 오버헤드 발생을 막을 수 있다.

타입 테이블은 리소스와 각 리소스의 클래스 타입 정보를 저장한다. 리소스 아이디는 클래스 타입을 한 번에 알아볼 수 있도록 클래스 타입에 연결된 리소스에 타입 이름과 일련번호를 붙여서 표현한다. 리소스의 실제 URI는 리소스 테이블에 따로 저장한다.

클래스와 타입 테이블을 이용한 키워드와 타입의 복

합 조건 검색의 단계별 처리 과정은 다음 예와 같다.

질의 1) 타입이 "Artifact"인 "시인"과 관련된 리소스를 검색하라.

| 타입 | 클래스 아이디 | 서브클래스 | 깊이 |
|----------|---------|-------|----|
| Person | c1 | c3 | 1 |
| Artifact | c2 | c4 | 1 |
| Author | c3 | c3 | 0 |
| Book | c4 | c4 | 0 |
| Store | c5 | c5 | 0 |
| Building | c6 | c6 | 0 |



| 리소스 | 클래스 아이디 |
|------------|---------|
| Author 1 | c3 |
| Book 1 | c4 |
| Book 2 | c4 |
| Store 1 | c5 |
| Building 1 | c6 |
| Author 2 | c3 |
| Book 3 | c4 |
| Book 4 | c4 |
| Store 2 | c5 |
| Store 3 | c5 |
| Building 2 | c6 |

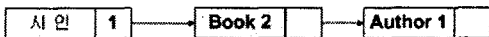


그림 4. 복합 조건 검색의 질의 처리 과정 예

- [1 단계] 클래스 테이블의 타입 필드를 순차적으로 검색하여 "Artifact" 정보를 추출한다.
- [2 단계] 타입 테이블에서 "Artifact"의 클래스와 서브 클래스에 관련된 리소스 정보 추출
- [3 단계] 키워드 인덱스를 이용하여 "시인"의 키워드를 포함하는 리소스를 검색한다.
- [4 단계] 2단계, 3단계를 통해 검색된 리소스들을 조인하여 최종 결과를 반환한다.

클래스 테이블의 서브 클래스 정보를 이용한 추론 질의 처리를 통해 "Artifact"와 연관된 리소스뿐만 아니라 서브클래스인 "Book"과 연관된 리소스까지 결과로 반환한다. 따라서 앞선 질의 예에서는 "Artifact"와 연관된 리소스는 존재하지 않으므로 "Book" 클래스에 연관된 4개의 리소스가 결과로 반환된다. 깊이 필드의 정보는 질의 처리 시 고려해야 할 서브 클래스의 범위를 한정시키는 역할을 담당한다. 앞선 질의 예에서 깊이 정보에 따르면 "Book" 클래스의 서브 클래스는 존재하지 않기 때문에 "Artifact" 클래스의 계층 구조에서는 "Book" 클래스까지만 질의 처리 시 고려한다.

3.3 질의 예와 질의 처리 과정

본 질에서는 <표 1>에서 분류한 4가지 검색 조건을 이용하여 질의 예를 제시하고 질의 처리 과정을 제시한다.

3.3.1 키워드를 포함하는 리소스 검색

질의 1) "삼국지"를 포함하는 리소스 검색

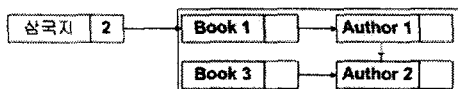


그림 5. 질의 1) 검색 결과

<그림 5>의 검색 결과는 키워드 인덱스 구조를 통해 "삼국지" 키워드를 포함하는 직접, 간접 포함 리소스를 보여준다.

질의 1)과 같은 질의 처리는 역 인덱스 구조의 키워드 리스트에서 주어진 키워드를 직접 포함하는 리소스 뿐만 아니라 의미적 연결성을 가지고 있는 간접 포함 리소스까지도 반환해주기 때문에 빠른 시간 내에 원하는 정보를 얻을 수 있다.

3.3.2 속성을 포함하는 모든 리소스 검색

질의 2) "title"속성을 포함하는 모든 리소스 검색

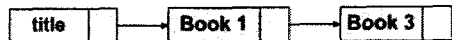


그림 6. 질의 2) 검색 결과

<그림 6>는 키워드 인덱스의 키워드 리스트를 통해 "title" 속성을 포함하고 있는 모든 리소스들을 보여준다.

질의 2)와 같은 질의는 속성 리스트를 이용하게 되면 속성 리스트에서 주어진 속성과 연결되어 있는 모든 리소스들을 반환해 주어 속성을 포함하고 있는 리소스들을 쉽게 얻을 수 있게 해 준다.

3.3.3 키워드, 속성 복합조건을 포함하는 리소스 검색

질의 3) "교보"에서 제공하는("provided")것과 관련된 리소스 검색

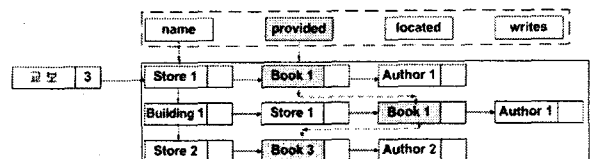


그림 7. 질의 3) 검색 결과

<그림 7>은 키워드 인덱스와 키워드 리스트를 통해 "교보" 키워드와 "provided" 속성을 모두 포함하는 리소스를 보여준다.

질의 3)과 같은 질의는 키워드 인덱스를 이용하여 키워드와 속성의 조건을 독립적으로 검색하여 주어진 키워드와 속성을 포함하는 직접, 간접 리소스들을 모두 검색하는 것이 아니라 키워드와 속성이라는 복합조건을 주어 키워드를 포함하는 리소스의 검색 결과를 반환한 후 속성과 연결된 리소스 검색 결과를 조인하여 두 조건을 모두 만족하는 직접 포함 리소스가 결과로 도출되기 때문에 보다 정확한 정보를 얻을 수 있게 된다.

3.3.4 키워드, 클래스 타입의 복합 조건을 만족하는 리소스 검색

질의 4) "Building" 타입을 가진 "교보"와 관련된 리소스 검색

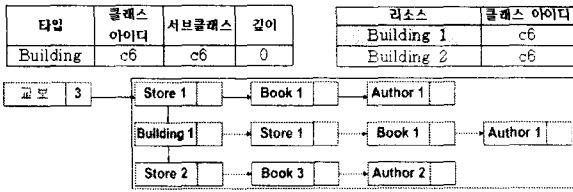


그림 8. 질의 4) 검색 결과

<그림 8>은 클래스vfg, 타입 테이블과 키워드 인덱스를 통해 "Building"타입과 "교보" 키워드를 모두 포함하는 리소스를 보여준다.

RDF 스키마까지 확대한 키워드와 타입의 복합 조건 질의 처리를 위해서 아래와 같은 단계별 과정이 수행된다.

- [1단계] 클래스 테이블에서 타입이름 "Building"을 순차적으로 검색한다.
- [2단계] "Building"의 클래스 아이디 정보를 추출 한다. 클래스 아이디 값이 c6임을 알 수 있다.
- [3단계] "Building"의 서브클래스의 정보를 추출한다. 서브클래스 값이 c6임을 알 수 있다.
- [4단계] 값이 정보를 추출한다. 값이의 값이 0임을 알 수 있다. 이는 주어진 클래스가 단일 노드임을 추론할 수 있다.
- [5단계] 타입 테이블의 클래스 아이디를 순차적으로 검색하여 c6과 관련된 리소스를 추출하여 반환한다. "Building 1"과 "Building 2"가 결과로 반환된다.
- [6단계] 키워드 인덱스에서 "교보"를 포함하는 리소스를 검색한다. <표 2>와 같이 직접 포함하는 리소스, 간접 포함하는 리소스들을 반환한다.

표 2. 교보를 포함하는 직접, 간접 리소스

| 직접포함 리소스 | 간접포함 리소스 | | |
|------------|----------|----------|----------|
| Store 1 | Book 1 | Author 1 | |
| Building 1 | Store 1 | Book 1 | Author 1 |
| Store 1 | Book 3 | Author 2 | |

- [7단계] 5,6단계의 결과 값을 조인한다. 조인 결과로 "Building 1"을 반환한다.

키워드의 단일조건 검색인 "교보"와 관련된 리소스를 검색하라는 질의처리 결과와 비교해 보면 단일 조건 검색의 경우 키워드를 포함하는 직접, 간접 리소스 정보를 모두 반환하기 때문에 빠른 검색이 가능하지만 검색된 리소스의 수가 많아지면 사용자의 판단을 흐리게 할 수도 있다. 키워드와 타입의 복합조건 검색 "Building" 타입을 가진 "교보"와 관련된 리소스 검색의 경우 키워드인덱스 구조를 통한 결과와 스키마 테이블을 이용한 결과를 조인하여 보다 정확한 리소스 정보의 빠른 검색이 가능해진다.

4. 결론 및 향후 연구

본 논문에서 제안한 키워드 인덱스와 스키마 테이블 구조를 이용하여 키워드와 속성의 단일 조건에 따른 리소스 검색과 키워드, 속성의 복합 조건, 키워드, 클래스 타입의 복합 조건 검색으로 보다 다양하고 정확한 리소스 정보의 빠른 검색이 가능하여 검색의 정확성과 속도의 향상을 기대할 수 있다.

사맨틱 웹이 차세대 웹으로 정립하기 위해서는 단순한 구조적 질의뿐만 아니라 의미론적 추론이 가능한 다양한 질의 처리기법들이 연구되어야한다.

향후 연구에서는 검색 조건에 따른 결과 리소스들이 많이 나오는 상황을 고려하여 RDF/RDF 스키마에 랭킹 기법을 추가 함으로써 자주 검색되는 키워드와 리소스 순으로 정렬하여 원하는 리소스를 빠르고 쉽게 얻을 수 있도록 하고자 한다.

참고문헌

- [1] W3C, <http://www.w3.org>
- [2] XML, <http://www.w3.org/XML/>
- [3] RDF Primer, <http://www.w3.org/TR/rdf-primer/>
- [4] RDF Schema, <http://www.w3.org/TR/rdf-schema/>
- [5] 이홍래, 이형동, 유상원, 김형주, XML 문서상에서 키워드 검색을 위한 색인 분할 기법, 데이터베이스 연구회, 2004.5
- [6] 김은주, 사맨틱 웹, TTA저널(표준기술 동향)
- [7] Youn Hee Kim, Byung Gon Kim, Jaeho Lee, and Hae Chull Lim, "The Transform of XML and Managing for Keyword Search over RDF", The International Conference on Information Networking (ICOIN '2004), Korea, Vol.2, pp.929-938, 2004.2