韓國 컴퓨터情報學會 論文誌
第12卷 第5號, 2007. 11.

2007-12-5-1-11

시맨틱 웹 데이터의 키워드 질의 처리를 위한 인덱싱 및 저장 기법

김 연 희*, 신 혜 연*, 임 해 철**, 정 균 락**

Indexing and Storage Schemes for Keyword-based Query Processing over Semantic Web Data

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

요 약

시맨틱 웹에서는 메타데이터와 온톨로지를 이용하여 질의를 처리하기 때문에 보다 정확한 검색 결과를 얻을 수 있을 뿐만 아니라 추론을 통하여 얻어진 새로운 지식도 검색 결과에 포함시킬 수 있다. 메타데이터와 온톨로지를 기술하기 위한 시맨틱 웹 언어 중 RDF와 RDF 스키마가 보편적으로 많이 활용되고 있다. 따라서 RDF와 RDF 스키마로 기술된 시맨틱 웹 언어에 대한 효과적인 검색 기법이 요구된다. 본 논문에서는 키워드 질의 처리 결과의 기본 단위를 전체 웹 문서나 부분이 아닌 정보 리소스로 정의하였다. 그리고 메타데이터와 온톨로지 정보를 모두 고려한 시맨틱 웹 환경의 키워드 질의를 3가지 유형으로 분류하고 다양한 키워드 관련 질의에 대한 처리를 효과적으로 지원하기 위하여 키워드 인덱스와 저장 구조를 제안하였다. 본 논문에서 제안한 키워드 인덱스는 질의 조건으로 주어진 키워드를 직접 포함하고 있는 리소스는 물론 의미적 관계에 의해 간접적으로 포함하고 있는 리소스에 관련된 정보를 쉽게 제공할 수 있다. 그리고 본 논문에서는 클래스와 속성의 일반적인 정보와 계층 정보를 단순한 레이블링 기법을 이용하여 표현한 후 제안된 저장 구조를 이용해 정보를 유지하여 시맨틱 웹 환경에 적합한 키워드 질의 처리를 지원하고자 한다.

Abstract

Metadata and ontology can be used to retrieve related information through the inference more accurately and simply on the Semantic Web. RDF and RDF Schema are general languages for representing metadata and ontology. An enormous number of keywords on the Semantic Web are very important to make practical applications of the Semantic Web because most users prefer to search with keywords. In this paper, we consider a resource as a unit of query results. And we classify queries with keyword conditions into three patterns and propose indexing techniques for keyword-search considering both metadata and ontology. Our index maintains resources that contain keywords indirectly using conceptual relationships between resources as well as resources that contain keywords directly. So, if user wants to search resources that contain a certain keyword, all resources are retrieved using our keyword index. We propose a structure of table for storing RDF Schema information that is labeled using some simple methods.

▸ Keyword : 키워드 질의(Keyword-based Query), 인덱싱 기법(Indexing Scheme), 저장 기법(Storage Scheme), 시맨틱 웹(Semantic Web)

• 제1저자 : 김연희 • 교신저자 : 정균락
• 접수일 : 2007. 8.20, 심사일 : 2007. 9.28, 심사완료일 : 2007. 10.8.
* 홍익대학교 컴퓨터공학과 대학원 **홍익대학교 컴퓨터공학과 교수
※ 본 연구는 한국과학재단 특정기초연구(R01-2004-000-10586-0(2006))지원으로 수행되었음.

I. 서 론

데이터를 기술하고 데이터간의 단순한 연결을 표현하는데 초점을 맞추고 있는 현재 웹 환경은 추론을 기반으로 하는 지능화된 검색이 어렵고 컴퓨터가 자동으로 데이터를 이해하고 처리하지 못하는 등 여러 한계점을 가지고 있다. 이러한 문제를 해결하기 위해 현재 웹의 확장된 개념인 시맨틱 웹이 차세대 웹으로 인식되고 있다. 시맨틱 웹에서는 텍스트뿐만 아니라 다양한 데이터를 포함하고 있는 정보 리소스 자체의 개념과 다른 정보 리소스와의 의미적 관계를 표현함으로써 보다 지능화된 정보 검색은 물론 자동화된 다양한 웹 서비스를 제공할 수 있다[1].

시맨틱 웹에서 온톨로지는 정보 리소스에 의미를 부여하기 위해 사용되는 용어의 개념을 정의하고 용어들 간의 관계를 정의한다. 메타데이터는 온톨로지에서 정의된 용어를 이용하여 정보 리소스의 의미와 정보 리소스 간의 의미적 연관성을 표현한다. 시맨틱 웹을 보편화시키고 정보 리소스에 대한 보다 지능화된 정보 검색을 제공하기 위해서는 메타데이터와 온톨로지를 이용한 효율적인 정보 검색에 대한 연구가 필요하다. 따라서 본 논문에서는 시맨틱 웹의 일반 사용자를 위한 메타데이터와 온톨로지 검색 방법에 초점을 맞추고 있다.

시맨틱 웹 상에서 중요한 역할을 담당하는 메타데이터와 온톨로지를 기술하기 위한 언어들이 소개되어 왔다. 그중에서 RDF(Resource Description Framework)와 RDF 스키마는 W3C에서 제안한 가장 기본적인 시맨틱 웹 언어로 일반적으로 많이 사용되고 있기 때문에 RDF와 RDF 스키마로 표현된 메타데이터와 온톨로지 처리에 관한 연구가 필요하다[2, 3].

웹 환경에서 이루어지는 가장 일반적인 검색 방법이 키워드 검색이다. 사용자가 제시한 키워드와 관련한 문서의 부분이나 또는 전체 문서를 검색해서 반환하는 것이 키워드 검색의 처리 과정이다. 시맨틱 웹 상에서 메타데이터와 온톨로지가 RDF와 RDF 스키마를 이용해 문서로 표현되기 때문에 RDF와 RDF 스키마 문서 또한 키워드 검색의 대상이 될 수 있으며 일반적인 사용자를 고려한 시맨틱 웹의 활성화를 위해서는 RDF와 RDF 스키마 문서에 대한 키워드 검색 기법에 대한 연구가 반드시 요구된다. 특히, RDF와 RDF 스키마에 대한 키워드 검색은 HTML이나 XML과는 조금 다른 특성을 지니기 때문에 HTML이나 XML을 대상으로 하는 키워드 검색 기법과는 차별화된 검색 방법이 필요하다.

따라서 본 논문에서는 이러한 연구의 필요성에 따라 다음과 같은 내용을 제안한다.

첫째, 시맨틱 웹 환경에서 RDF와 RDF 스키마로 기술된 시맨틱 웹 문서에 대한 키워드 검색의 유형을 단순 검색과 복합 검색으로 분류하고 각 질의 유형의 특징을 정의한다. 둘째, 분류된 질의 유형에 대해 RDF와 RDF 스키마 특성을 고려한 효과적인 키워드 질의 처리를 지원하기 위한 인덱스 구조와 저장 스키마를 제안한다. 그리고 제안한 저장 스키마와 인덱스 구조를 이용한 질의 처리 전략을 제안한다. 마지막으로 키워드 검색 결과를 위한 랭킹 평가 기법을 제안한다.

본 논문은 다음과 같이 구성된다. 2장에서는 기존 시맨틱 웹 데이터 관리 기법에 대해 소개한다. 3장에서는 시맨틱 웹 데이터에 대한 키워드 질의 유형을 분류하고, 4장에서는 본 논문에서 제안한 시맨틱 웹 데이터를 위한 키워드 인덱스 구조와 질의 처리 전략을 소개한다. 5장에서는 본 논문에서 적용한 랭킹 평가 기법을 소개하고, 6장에서는 제안한 키워드 인덱싱 기법을 적용한 실험 결과를 제시하고 7장에서 결론을 맺는다.

II. 관련 연구

2.1 RDF와 RDF 스키마 데이터

RDF는 웹 상의 정보 리소스에 대한 메타데이터의 표현 및 교환의 수단으로 W3C에서 제안된 가장 기본적인 시맨틱 웹 언어로서 데이터의 의미와 추론을 위해 필요한 정보를 기술하는데 그 목적을 두고 있다[2]. RDF는 각 정보 리소스의 특성이나 다른 정보 리소스와의 의미적 관계를 주어(subject), 서술어(predicate), 목적어(object)로 구성된 문장(statement)의 형태로 표현하며 XML의 문법 형태를 그대로 이용한다. 정보 리소스의 URI(Uniform Resource Identifier)가 주어의 역할을 담당하고 서술어는 주어인 정보 리소스에 대한 속성으로 프로퍼티라 한다. 프로퍼티 또한 URI를 갖는 일종의 정보 리소스이다. RDF 문장 내에 목적어는 프로퍼티의 실제 값으로 다른 리소스의 URI 또는 리터럴 데이터로 표현된다.

RDF 스키마는 RDF 문장을 기술하는데 필요한 어휘를 정의하기 위한 온톨로지 언어로 같은 개념을 표현하는데 서로 다른 어휘로 메타데이터를 기술하거나 또는 서로 다른 개념을 표현하는데 같은 어휘를 사용하는 등의 모호성 문제를 해결할 수 있다[3]. RDF 스키마는 메타데이터를 기술

할 때 적용될 수 있는 개념을 클래스로 정의하고 각 클래스들 간의 의미적인 관계를 프로퍼티로 정의한다. 그리고 클래스들 간의 계층 관계 또는 프로퍼티들 간의 계층 관계를 함께 정의함으로써 보다 다양한 의미적 관계를 표현할 수 있다. RDF 스키마 역시 XML 문법에 기반한 RDF 문장의 형태로 기술된다.

RDF와 RDF 스키마의 문장들은 노드와 간선에 모두 레이블이 표현된 방향성 그래프 형태의 데이터 모델로 표현될 수 있다. RDF와 RDF 스키마 문장에서 주어와 목적어는 그래프 데이터 모델에서 노드로 표현되고 서술어는 주어에 해당하는 노드에서 목적어에 해당하는 노드로 연결된 간선으로 표현된다.

시맨틱 웹의 보편화를 위해서는 메타데이터와 온톨로지의 구축과 관리가 무엇보다 중요하기 때문에 메타데이터와 온톨로지를 기술하는 언어인 RDF와 RDF 스키마 데이터를 위한 저장 및 질의 처리에 관한 연구가 무엇보다 중요하다.

2.2 시맨틱 웹 데이터 관리 기법

시맨틱 웹 데이터 관리를 위한 기존의 연구들은 대부분 RDF와 RDF 스키마 데이터에 대한 저장 및 검색 기법에 초점을 맞추어 진행되어 왔다[4,5,6,7]. 특히, RDF와 RDF 스키마 문장들을 주어/서술어/목적어의 트리플 구조로 인식하고 상용 데이터베이스 시스템의 테이블로 저장하는 다양한 방법과 저장 형태에 따른 질의 처리 기법에 대한 연구 결과가 많았다. 그러나 기존 시맨틱 웹 데이터 관리 기법에 대한 연구들은 RDF와 RDF 스키마를 독립적으로 처리하지 않고 트리플 구조의 문장으로 동일하게 처리함으로써 메타데이터와 온톨로지의 근본적인 차별성을 질의 처리 시 활용하지 못하는 한계를 가지고 있다. Sesame 시스템과 같이 RDF와 RDF 스키마를 독립적으로 처리하더라도 RDF 스키마에 기술된 클래스 간 계층 구조나 프로퍼티 간 계층 구조에 대한 충분한 고려를 하지 않는 문제가 여전히 남아 있다.[5]. 따라서 메타데이터와 온톨로지 정보의 근본적인 차이점을 인식하고 질의 처리 시 활용하는 연구가 필요하다.

기존 웹 환경에서 HTML 문서나 XML 문서에 대한 키워드 검색 기법이 중요시 되었던 것과 같은 맥락으로 시맨틱 웹 환경의 일반 사용자들을 위한 키워드 검색 기법에 대한 연구 또한 필요하다. 그러나 지금까지 시맨틱 웹 데이터의 질의 처리에 대한 연구들은 RDF 문장들에 대해 트리플 구조에 기반한 질의 형태가 가장 일반적이라 규정하고 문장을 구성하는 주어, 서술어, 목적어 요소 중 일부가 질의 조건으로 주어졌을 때 나머지 다른 구성 요소나 해당 문장 전체를 결과로 빠르게 반환하기 위한 인덱스 구조나 질의 처리 기법들에 관심을 가져왔다[8,9,10]. 물론 연구 결과의 일부분으로 키워드 검색을 지원하기 위한 인덱스 구조에 대해 제안하였으나 이들이 제안한 키워드 인덱스는 역리스트 구조나 해쉬 테이블에 기반하여 단순히 해당 키워드를 직접 포함하는 RDF 문장 내 구성 요소나 문장 전체를 결과로 반환하는 역할을 담당한다. 그러나 시맨틱 웹 환경에 적합한 키워드 검색을 지원하기 위해서는 RDF 문장의 트리플 구조뿐만 아니라 의미적 관계까지 고려해야 하고 더 나아가 RDF 스키마가 제공하는 온톨로지 정보를 활용하여 단순 키워드 검색이 아닌 추론에 기반한 의미적 키워드 검색을 제공할 필요가 있다. 따라서 본 논문에서는 시맨틱 웹 데이터의 키워드 검색과 관련한 기존 연구 결과의 한계를 인식하고 RDF와 RDF 스키마로 표현된 메타데이터와 온톨로지 데이터에 적합한 키워드 검색을 지원하기 위한 저장 및 인덱싱 기법을 제안한다.

III. 시맨틱 웹 데이터의 키워드 검색 유형

본 장에서는 기존 웹 환경에서의 키워드 검색과 시맨틱 웹 환경에서의 키워드 검색이 가지는 근본적인 차이점을 설명하고 시맨틱 웹 데이터를 위한 키워드 검색 유형을 분류한다.

기존 웹 환경에서 HTML 문서에 대한 키워드 검색은 사용자가 제시한 질의 키워드를 포함하고 있는 전체 문서를 질의 결과로 반환하고 XML 문서에 대한 키워드 검색은 질의 키워드를 포함하는 태그를 질의 결과로 반환한다. RDF와 RDF 스키마는 기본적으로 XML의 문법 형태를 그대로 따르고 있지만 메타데이터와 온톨로지를 주어/서술어/목적어의 정형화된 구조로 서술한다는 특징을 지니기 때문에 HTML이나 XML을 대상으로 하는 키워드 검색과는 차별화된 접근 방법이 필요하다.

RDF와 RDF 스키마 데이터에서는 키워드의 의미를 다양하게 정의내릴 수 있다. 즉, 리터럴 데이터뿐만 아니라 클래스나 프로퍼티의 이름 또한 넓은 의미의 키워드라 할 수 있으나 본 논문에서는 클래스나 프로퍼티가 가지고 있는 본래의 역할을 키워드 검색 시 이용하기 때문에 키워드는 리터럴 데이터로부터 추출하는 것으로 한정한다.

본 논문에서는 RDF와 RDF 스키마 문서가 웹 상의 리소스에 대한 정보를 기술한 것이므로 특정 태그를 포함한 부분이나, 전체 문서보다는 해당 정보를 포함하는 리소스를 질의 결과의 반환 단위로 고려한다. 그리고 사용자의 보다 다양하고 정확한 질의 처리를 지원하기 위해 검색 조건에 따라 RDF와 RDF 스키마 문서에 대한 키워드 검색을 다음과 같이 크게 단순 질의와 복합 질의로 분류한다.

- 단순 질의
  (1) 검색 조건으로 키워드만이 주어진 질의
  - 사용자가 제시한 키워드를 직·간접적으로 포함하는 모든 리소스를 검색한다.
- 복합 질의
  (1) 검색 조건으로 키워드와 관련 프로퍼티가 주어진 질의
  - 사용자가 제시한 프로퍼티에 대한 값의 일부로 키워드를 직·간접적으로 포함하는 모든 리소스를 검색한다.
  (2) 검색 조건으로 키워드와 관련 클래스 타입이 주어진 질의
  - 사용자가 제시한 키워드를 직·간접적으로 포함하는 모든 리소스 중에서 조건으로 주어진 클래스 타입에 해당하는 리소스만을 검색한다.

앞서 분류한 대표적인 키워드 질의 유형을 처리하기 위해 본 논문에서는 RDF와 RDF 스키마의 그래프 데이터 모델을 이용한다. 〈그림 1〉은 본 논문에서 사용할 예제 RDF와 RDF 스키마 데이터를 그래프 모델로 표현한 것으로 특정 출판사에서 보유하고 있는 책과 저자에 대한 메타데이터와 온톨로지 정보를 나타내고 있다. 〈그림 1〉의 RDF 데이터 부분에서 원형 노드는 리소스를 의미하고 직사각형 노드는 리터럴 데이터를 의미한다. 그리고 프로퍼티의 이름으로 레이블된 간선이 존재한다. RDF 스키마 데이터 부분에서 원형 노드는 클래스를 의미하고 RDF와 마찬가지로 프로퍼티의 이름으로 레이블된 간선이 존재한다. RDF 데이터와는 다르게 RDF 스키마 부분에는 클래스의 계층 구조 또는 프로퍼티의 계층 구조를 나타내는 부가적인 간선이 존재한다.

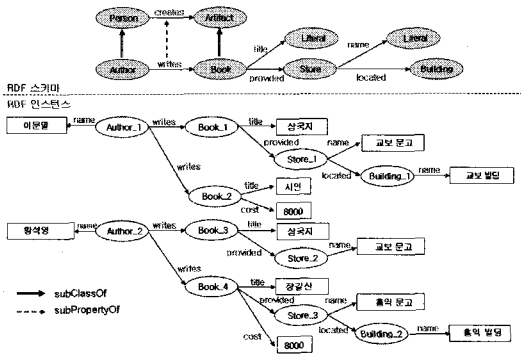


그림 1. 시맨틱 웹 데이터의 그래프 표현 예
Fig 1. Example of Graph Model for RDF and RDF Schema

IV. 시맨틱 웹 기반 키워드 질의 처리 기법

본 논문에서는 RDF와 RDF 스키마로 표현된 메타데이터와 온톨로지 정보를 모두 고려하여 시맨틱 웹 환경에 적합한 키워드 검색을 지원하기 위해 인덱스 구조와 저장 스키마를 제안한다.

4.1 키워드 인덱스

본 논문에서 제안한 인덱스 구조는 주어진 키워드를 직·간접적으로 포함하는 리소스와 그와 관련된 정보를 빠르게 검색할 수 있도록 하는 것을 목표로 한다. 텍스트 문서에 대한 키워드 검색을 지원하기 위해 일반적으로 역 인덱스 구조를 이용한다. 보통 역 인덱스는 텍스트 파일에 나타나 있는 키워드들을 인덱스의 키 값으로 하여 키워드를 포함하고 있는 텍스트 파일을 쉽게 식별할 수 있도록 인덱스 파일 부분과 포스팅 파일 부분으로 구성되어 있다. 본 논문에서 제안한 키워드 인덱스도 기존 역 인덱스 구조를 응용하여 키워드 리스트 부분과 포스팅 리스트 부분으로 나뉜다.

키워드 리스트는 RDF 문서로부터 추출한 모든 리터럴 키워드와 그 키워드를 직접 포함하고 있는 문서 내 리소스의 수를 표현하고 있어 리소스와의 관계에 기반한 키워드 빈도수와 중요도를 파악할 수 있다. 키워드 리스트를 구성하는 기본 단위는 키워드 노드로 〈그림 2〉에 본 논문에서 제안한 키워드 노드의 구조가 나타나있다.

그림 2. 키워드 노드 구조
Fig 2. Structure of Keyword Node

〈그림 2〉의 노드 구조에서 Keyword 필드는 RDF 문서에서 추출한 리터럴 키워드정보를 저장하고 Frequency 필드는 해당 키워드의 빈도수를 유지한다. 일반 텍스트 문서에 대한 키워드 질의 처리 기법에서 키워드의 빈도수는 단순히 문서 내에서 키워드의 발생 횟수를 의미한다. 그러나 RDF 문서는 리소스에 대한 의미적 정보를 표현하는 것을 목적으로 하는 특성 때문에 키워드 빈도수의 의미도 바뀔 필요가 있다. 따라서 본 논문에서는 키워드를 직접 포함하고 있는 리소스의 개수로 키워드 빈도수를 정의함으로써 리소스 중심의 키워드 질의 처리가 가능하도록 한다. 키워드 노드의 R_Pointer 필드는 해당 키워드를 포함하는 리소스

에 대한 정보를 표현하는 포스팅 리스트 노드에 대한 포인터를 유지한다. 포스팅 리스트는 키워드를 직·간접적으로 포함하는 리소스와 관련된 속성 정보를 유지함으로써 질의 조건으로 키워드가 주어졌을 때 키워드와 관련된 리소스 및 속성 정보를 쉽게 검색할 수 있도록 한다. 포스팅 리스트는 키워드를 직접적으로 포함하는 리소스들을 수직적으로 연결한 직접 리소스 리스트와 키워드를 직접 포함하는 특정 리소스를 통해 키워드를 간접 포함하는 리소스들을 수평적으로 연결한 간접 리소스 리스트로 세분화된다.

직접 리소스 리스트는 <그림 3>과 같은 노드로 구성된다.

그림 3. 직접 리소스 노드
Fig 3. Structure of Direct Resource Node

<그림 3>에서 RID 필드는 키워드를 포함하는 리소스의 아이디를 표현하고 PID 필드는 RID에 표현된 리소스와 관련된 속성으로 해당 키워드를 값으로 가지는 속성의 아이디를 표현한다. D_Pointer 필드는 같은 키워드를 직접 포함하고 있는 다음 리소스 노드에 대한 포인터를 유지하고 I_Pointer 필드는 RID에 표현된 해당 리소스를 통해 키워드를 간접적으로 포함하는 첫 번째 리소스에 대한 포인터를 유지한다.

간접 리스트 노드는 <그림 4>와 같이 직접 리스트 노드와 유사하게 RID 필드는 키워드를 간접적으로 포함하는 리소스의 아이디를 표현하고 PID 필드는 RID에 표현된 리소스와 관련된 속성으로 해당 키워드를 값으로 가지는 속성의 아이디를 표현한다. I_Pointer 필드는 RID에 표현된 해당 리소스를 통해 키워드를 간접적으로 포함하는 다음 리소스에 대한 포인터를 유지한다.

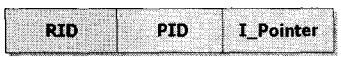


그림 4. 간접 리소스 노드
Fig 4. Structure of Indirect esource Node

키워드 리스트와 직·간접 리소스 리스트로 구성된 키워드 인덱스의 전체적인 구조는 <그림 5>와 같다.

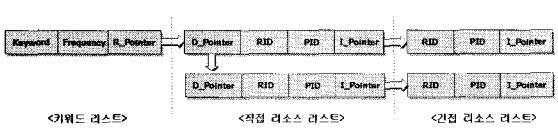


그림 5. 키워드 인덱스 전체 구조
Fig 5. Structure of Entire Keyword Index

키워드 인덱스를 구성하는데 있어 성능과 관련된 중요한 고려 사항은 키워드와의 관련성이 높은 리소스들을 인덱스에 유지시켜야한다는 점이다. 키워드와의 관련성을 고려하면 키워드를 직접 포함하는 리소스들은 모두 키워드 인덱스에 포함시켜야 한다. 하지만 간접 포함 리소스의 경우 키워드로부터의 거리가 멀수록 키워드와의 관련성이 적다고 할 수 있으므로 키워드로부터의 유효 거리를 제약 사항으로 정해놓는 것이 중요하다. 키워드와의 관련성의 경우 검색에 대한 사용자의 기호가 반영되는 사항이므로 유효 거리에 대한 조건은 인덱스 구축 시 사용자나 인덱스 개발자가 결정한다.

키워드 인덱스에서 리소스를 구별하기 위한 RID는 각 리소스에 정의된 클래스 타입을 이용해서 부여한다. 즉, 클래스 이름과 문서 내 출현 순서에 따라 RID가 지정된다. 예를 들어 어떤 리소스가 "Author" 클래스 타입으로 지정되어 있고 "Author" 클래스로 지정된 리소스들 중에서 문서 내 출현 순서가 첫 번째인 경우 그 리소스의 RID는 "Author_1"로 지정한다. 이렇게 RID를 지정하게 되면 각 리소스가 어떤 클래스 타입으로 지정되었는지를 쉽게 판단할 수 있기 때문에 클래스 정보를 이용한 복합 질의를 처리하는데 유리하다. 본 논문에서는 모든 리소스에 클래스 타입이 지정되어 있다고 가정한다.

속성을 구별하기 위한 PID는 기존 Dewey 방식을 위한 레이블링 기법을 활용하여 부여한다. Dewey 방식을 이용한 속성 아이디의 부여 방법은 4.2절에서 자세하게 설명한다.

<그림 6>은 <그림 1> 예제의 정보를 제안 키워드 인덱스로 구성한 결과의 일부분을 보여준다. 앞서 설명한대로 키워드 리스트, 직접 포함 리소스 리스트, 간접 포함 리소스 리스트 세부분으로 구성되어 있는 것을 확인할 수 있다.

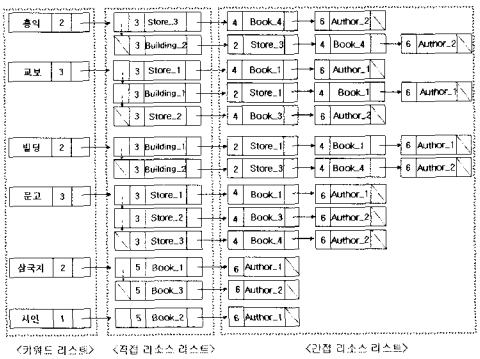


그림 6. 제안 키워드 인덱스의 구성 예
Fig 6. Example of Proposed Keyword Index

4.2 저장 구조

앞서 제안한 키워드 인덱스는 RDF 문서에 표현되어 있는 키워드와 그와 관련된 리소스 및 속성 정보로만 구성되기 때문에 RDF 스키마에 표현된 온토롤지 정보를 이용한 복합 질의를 처리하기 위해서는 RDF 스키마 정보를 관리할 수 있는 방법이 필요하다. 본 논문에서는 RDF와 RDF 스키마를 모두 고려한 키워드 질의 처리를 위해 RDF 스키마에 정의된 온톨로지 정보의 저장 구조를 제안한다. 특히 제안된 저장 구조는 RDF 스키마 문서에 기술되어 있는 클래스와 속성의 정의 정보와 클래스들간의 계층 정보와 속성들간의 계층 정보를 표현하고 클래스와 속성의 계층 정보를 이용한 이행적 추론을 지원하는데 목적을 두고 있다. 본 논문에서는 클래스와 속성들을 구별하기 위해 Dewey 방식을 이용한 간단한 레이블링 방법을 이용하는데 이러한 레이블링 방법을 통해 클래스들 간의 계층 관계나 속성들 간의 계층 관계를 쉽게 판별할 수 있다.

Dewey 방식은 트리 형태로 구성된 데이터 구조에서 우선 같은 레벨에 존재하는 노드들 간에는 왼쪽에서 오른쪽으로 순차적으로 번호를 부여한다. 그리고 자식 노드는 부모 노드의 레이블에 부여받은 번호를 덧붙여서 최종적인 레이블을 완성한다. 〈그림 7〉은 〈그림 1〉의 예제에서 RDF 스키마의 클래스들에 Dewey 방식을 이용해 레이블링 한 결과를 보여주고 있다. 〈그림 1〉에서 "Literal" 클래스는 텍스트 데이터를 위한 일반적인 클래스이므로 생략하였다. 속성도 클래스와 같은 방법으로 레이블이 결정되므로 〈그림 7〉에서는 속성의 레이블 정보는 생략하였다.

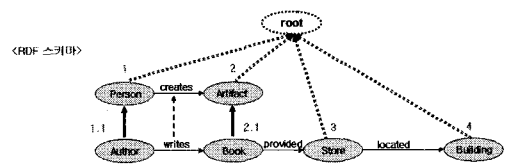


그림 7. 클래스 식별을 위한 레이블링 결과
Fig 7. Example of Labeling for Classes

본 논문에서는 RDF 스키마에 정의된 클래스와 속성 정보를 저장하기 위해 클래스 테이블과 프로퍼티 테이블로 구성된 저장 구조를 제안한다. 〈그림 8〉은 〈그림 1〉의 RDF 스키마 정보를 제안한 두 개의 테이블에 저장한 결과를 보여준다.

〈그림 8〉의 클래스 테이블에서 CID 필드는 Dewey 방식에 의해 결정된 클래스의 레이블을 저장하고 C_Name 필드는 RDF 스키마에 정의된 클래스의 이름을 저장한다. 〈그림 8〉의 속성 테이블에서 PID 필드는 Dewey 방식에 의해 결정된 속성의 레이블을 저장하고 P_Name 필드는 RDF 스키마에 정의된 속성의 이름을 저장한다. 〈그림 8〉의 저장 구조를 통해 RDF 스키마에 정의되어 있는 클래스와 속성의 기본 정보와 계층 정보를 검색할 수 있다. 물론 RDF 스키마만을 질의의 대상으로 한다면 도메인, 레인지 클래스와 같은 추가적인 정보를 저장할 필요가 있지만 본 논문에서는 RDF 문서에 대한 키워드 질의 처리를 지원하기 위한 목적으로 RDF 스키마 저장 구조를 제안한 것이므로 키워드 질의 처리에 영향을 미치는 두 개의 테이블 만을 유지한다.

〈클래스 테이블〉

CID	C_Name
1	Person
1.1	Author
2	Artifact
2.1	Book
3	Store
4	Building

〈속성 테이블〉

PID	P_Name
1	creates
1.1	writes
2	title
3	provided
4	name
5	located

그림 8. RDF 스키마를 위한 저장 예
Fig 8. Example of Storage for RDF Schema

4.3 키워드 질의 처리 과정

본 절에서는 제안 키워드 인덱스와 저장 구조를 이용하여 키워드와 클래스 타입이 질의 조건으로 주어지는 복합 질의 유형의 단계별 처리 과정을 설명한다. 예를 들어 "삼국지라는 키워드와 관련된 리소스들 가운데 Artifact 클래스 타입으로 정의된 리소스만을 검색하라." 하는 질의가 요청되었을 때 다음과 같이 4 단계를 거쳐 최종적인 결과 리소스를 반환하게 된다.

(1 단계)	클래스 테이블에서 "Artifact" 클래스의 아이디를 검색한다. 검색 결과로 "Artifact" 클래스의 아이디는 "2"이다.

(2 단계)	1단계에서 검색된 "Artifact" 클래스 아이디를 이용해서 "Artifact" 클래스의 모든 서브 클래스를 검색한다. 즉, "Artifact" 클래스 아이디를 자신의 아이디 앞부분에 포함하고 있는 클래스들이 검색 대상이 된다. 서브 클래스의 검색은 문자열 처리 함수를 이용하면 쉽게 처리된다. 검색 결과로 아이디 "2.1"을 가지는 "Book" 클래스가 서브 클래스로 반환된다.

[3 단계]	키워드 인덱스를 이용하여 "삼국지" 키워드를 직·간접적으로 포함하는 모든 리소스를 반환한다. 검색 결과로 "Author_1", "Author_2", "Book_1", "Book_3"과 같이 4개의 리소스가 반환된다.

[4 단계]	3단계에서 검색된 리소스 가운데 "Artifact"나 "Book"으로 시작되는 아이디를 가지는 리소스만을 최종 결과로 반환한다. 즉 "Book_1"과 "Book_3" 리소스가 최종 결과가 된다.

본 논문에서 제시한 바와 같이 클래스 간의 계층 구조를 이용하여 키워드 복합 질의 유형을 처리하게 되면 질의 조건으로 주어진 "Artifact" 클래스 타입의 리소스뿐만 아니라 서브 클래스인 "Book" 클래스 타입의 리소스를 결과로 반환할 수 있기 때문에 이행적 추론을 이용한 보다 정확한 검색이 가능하다.

클래스 타입과 키워드가 질의 조건으로 주어진 복합 질의 처리 과정과 유사하게 키워드만이 주어지는 단순 질의와 속성과 키워드가 질의 조건으로 주어지는 경우도 쉽게 처리할 수 있다. 키워드만 질의 조건으로 주어지는 단순 질의의 경우는 제안 키워드 인덱스 구조만을 이용해 결과 리소스를 반환할 수 있다. 속성과 키워드가 주어지는 경우도 제안 키워드 인덱스 구조만을 이용하면 결과 리소스를 반환할 수 있는데 이것은 리소스에 대한 정보를 표현하는 리소스 노드의 경우 관련 속성에 대한 정보를 함께 유지하고 있기 때문이다.

V. 시맨틱 웹 데이터를 위한 랭킹 기법

본 논문에서 제안한 인덱스와 저장 구조를 이용하여 시맨틱 웹 데이터에 대한 키워드 질의를 처리하면 사용자가 입력한 키워드를 직·간접적으로 포함하고 있는 많은 리소스들이 결과로 반환되게 된다. 이때, 질의 조건으로 주어진 키워드와 가장 관련성이 높은 리소스 순으로 결과를 반환한다면 사용자의 검색 만족도를 높일 수 있다. 현재의 웹 환경에서는 질의 조건으로 주어진 키워드를 포함하고 있는 모든 문서를 반환하되 사람들이 이전에 가장 많이 클릭한 순으로 결과를 정렬하거나 다른 웹 문서로부터 가장 많이 링크된 문서 순으로 결과를 정렬하는 랭킹 기법을 주로 사용하고 있다. 그러나 시맨틱 웹 환경은 지금 현재의 웹 환경과 다른 만큼 시맨틱 웹 환경을 고려한 랭킹 기법이 필요하다. 따라서 본 논문에서는 키워드와 관련된 각 리소스 마다 중요도과 밀접 정도에 따라 가중치를 부여하고 이에 따라 결과 반환 순서를 결정하는 시맨틱 웹 환경의 랭킹 기법을 이용한다.

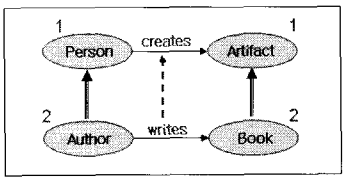
본 논문에서 이용하는 랭킹 기법은 키워드를 직접 포함하고 있는 리소스들을 대상으로 하며 각 리소스의 클래스 타입 중요도와 와 리소스에 연결된 속성의 수에 따라 결정되며 수직적 랭킹 기법이라 한다. 〈그림 9〉는 본 논문에서 이용하는 수직적 랭킹 기법(Vertical Weight)을 수식화한 것이다.

$$\text{Vertical Weight} = \text{Class Weight} + \text{Direct Resource Weight} / \text{Predicate}$$

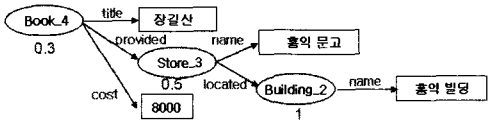
그림 9. 수직 랭킹 결정 수식
Fig 9. Expression for Vertical Weight

Class Weight는 각 리소스에 연결된 클래스의 가중치를 의미한다. 특히 클래스들의 계층 관계에서 하위 클래스가 상위 클래스보다 더욱 세부적인 정보를 제공한다고 할 수 있으므로 하위 클래스일수록 가중치를 높게 부여한다. 이러한 클래스의 가중치는 클래스와 연결되어 있는 리소스에 영향을 준다. Direct Resource Weight는 키워드를 직접 포함하고 있는 모든 리소스마다 주어지는 값으로 기본적으로 1이 부여된다. Direct Resource Weight값을 리소스에 연결되어 있는 속성들의 개수인 Predicate으로 나누는 이유는 리소스 내에서 키워드의 밀집도가 높을수록 높은 랭킹을 부여하기 위한 것이다.

〈그림 10〉에서 (a)는 클래스 계층 구조에 기반하여 리소스에 연결된 클래스마다 Class Weight를 부여한 예를 보여준다. 〈그림 10〉에서 (b)는 Direct Resource Weight/Predicate의 가중치를 각 리소스에 부여한 예를 보여준다. 이러한 가중치 부여 방법은 질의 조건으로 새로운 키워드가 주어질 때 마다 동적으로 계산되며 계산된 가중치 값에 따라 결과 리소스들의 랭킹을 결정하고 높은 순서대로 결과로 반환한다.



(a) Class Weight



(b) Direct Resource Weight / Predicate

그림 10. 가중치 값 부여 예

Fig 10. Example of Computed Weight

VI. 실험 결과

본 장에서는 시맨틱 웹 데이터의 키워드 질의 처리를 위해 본 논문에서 제안한 키워드 인덱스와 저장 구조로 구성된 프로토타입 시스템을 구현하여 질의 유형별로 처리 결과를 설명한다.

본 논문에서 제안한 키워드 인덱스와 저장 구조로 구성된 프로토타입 시스템은 펜티엄4 2.8GHz의 CPU와 1GB의 메모리를 가지고 Windows XP를 운영 체제로 이용하는 컴퓨터 에 구현하였다.

실험 데이터는 다양한 키워드를 포함하고 있는 유전자 정보[11]가 기술된 RDF와 RDF 스키마 문서를 이용하였으며 〈표 1〉과 같이 트리플 문장의 개수에 따라 세 개의 데이터 파일로 나누어 실험한 후 결과를 비교하였다.

표 1. 실험 데이터 파일

Table 1. Data Files for Experiment

	트리플 문장 수	파일 크기
실험 데이터 파일(1)	2,055개	235KB
실험 데이터 파일(2)	5,101개	592KB
실험 데이터 파일(3)	10,026개	1,215KB

본 논문에서 제시한 시맨틱 웹 데이터의 3가지 질의 유형의 대표적인 질의들을 이용해 실험을 진행하였으며 〈표 2〉에 사용된 질의에 대한 설명이 나타나있다.

표 2. 실험에 사용된 대표 질의

Table 2. Queries for Experiment

유형	질의 조건	질의 내용
질의1	키워드	"RNA"와 관련된 모든 리소스를 검색하라.
질의2	키워드 + 프로퍼티	"Pfam"을 키워드로 지니고 있는 리소스와 같은 (is_a) 리소스를 검색하라.
질의3	키워드 + 클래스 타입	"ProDom"을 키워드로 지니고 있는 리소스 중 클래스 타입이 용어 (term)인 리소스를 검색하라.

〈표 2〉에서 제시된 3가지 질의 유형에 대해 크기가 다른 실험 데이터 파일을 변화시키면서 질의 처리에 소요된 시간을 각각 측정하였다.

표 3. 첫 번째 질의에 대한 실험 결과

Table 3. Results for Query 1

	처리 시간	검색 리소스 수
실험 데이터 파일(1)	209μsec	1개
실험 데이터 파일(2)	927μsec	11개
실험 데이터 파일(3)	1.49msec	165개

〈표 3〉은 키워드만이 질의 조건으로 주어진 첫 번째 질의에 대해 크기가 다른 실험 데이터 별 처리 시간과 검색 결과로 반환된 직접 리소스와 간접 리소스의 총 개수를 보여준다.

표 4. 두 번째 질의에 대한 실험 결과

Table 4. Results for Query 2

	처리 시간	검색 리소스 수
실험 데이터 파일(1)	1.40msec	12개
실험 데이터 파일(2)	1.56msec	14개
실험 데이터 파일(3)	2.77msec	26개

〈표 4〉는 키워드와 관련 프로퍼티가 질의 조건으로 주어진 두 번째 질의에 대해 크기가 다른 실험 데이터 별 처리 시간과 검색 결과로 반환된 직접 리소스와 간접 리소스의 총 개수를 보여준다.

표 5. 세 번째 질의에 대한 실험 결과

Table 5. Results for Query 3

	처리 시간	검색 리소스 수
실험 데이터 파일(1)	876μsec	4개
실험 데이터 파일(2)	3.43msec	19개
실험 데이터 파일(3)	5.27msec	29개

〈표 5〉는 키워드와 리소스에 대한 클래스 타입이 질의 조건으로 주어진 세 번째 질의에 대해 크기가 다른 실험 데이터 별 처리 시간과 검색 결과로 반환된 직접 리소스와 간접 리소스의 총 개수를 보여준다.

본 논문에서 제안한 키워드 인덱스 및 저장 구조에 대한 실험 결과를 살펴보면 키워드만이 질의 조건으로 주어졌을

때보다 프로퍼티나 클래스 타입의 추가 조건이 주어졌을 때 질의 처리 시간이 일반적으로 조금 더 오래 걸리는 편이지만 그 차이가 크지 않은 것을 확인할 수 있다. 이러한 실험 결과에서 확인할 수 있듯이 본 논문에서 제안한 키워드 인덱스와 저장 구조가 시맨틱 웹 데이터의 특징인 프로퍼티나 클래스 타입에 대한 정보를 키워드 질의 처리에 활용할 수 있도록 지원하기 때문에 본 논문의 결과물은 시맨틱 웹 환경에서의 키워드 질의 처리에 적합한 형태라 할 수 있다. 그리고 실험 데이터 파일의 크기가 커지더라도 동일한 질의 유형에 대한 질의 처리 시간의 증가폭이 완만하여 더욱 많은 수의 트리플 문장을 포함하고 있는 시맨틱 웹 데이터에도 본 논문의 결과물을 그대로 적용시킬 수 있을 것으로 예상된다.

VII. 결론

메타데이터를 기술하는 RDF와 온톨로지를 기술하는 RDF Schema 데이터는 차세대 웹으로 각광받고 있는 시맨틱 웹을 지원하는 중요한 역할을 담당하고 있다. 메타데이터와 온톨로지는 RDF와 RDF Schema 언어에 의해 텍스트 문서로 작성된다고 할 수 있기 때문에 RDF와 RDF Schema로 작성된 시맨틱 웹 문서에 대한 키워드 검색 기법에 대한 연구가 필요하다.

시맨틱 웹 환경에서 메타데이터와 온톨로지는 정보 리소스의 개념과 다른 정보 리소스와의 의미적 관계를 표현하기 때문에 본 논문에서는 키워드 질의 처리 결과의 기본 단위를 전체 웹 문서나 부분이 아닌 정보 리소스로 정의하였다. 그리고 메타데이터와 온톨로지 정보를 모두 고려한 시맨틱 웹 환경의 키워드 질의를 3가지 유형으로 분류하고 다양한 키워드 관련 질의에 대한 처리를 효과적으로 지원하기 위하여 키워드 인덱스와 저장 구조를 제안하였다. 본 논문에서 제안한 키워드 인덱스는 질의 조건으로 주어진 키워드를 직접 포함하고 있는 리소스는 물론 의미적 관계에 의해 간접적으로 포함하고 있는 리소스에 관련된 정보를 쉽게 제공할 수 있도록 하는 것을 목표로 한다. 그리고 시맨틱 웹 환경에서 메타데이터에 대한 키워드 질의 처리 시 보다 정확하고 의미있는 결과를 얻기 위해서는 온톨로지 정보를 이용해야 하기 때문에 본 논문에서는 클래스와 속성의 정의 정보, 클래스들 간의 계층 정보와 속성들 간의 계층 정보를 단순한 레이블링 기법을 이용하여 표현한 후 제안한 저장 구조를 이용하여 정보를 유지한다. 따라서 본 논문에서 제안한 인덱스와 저장 구조는 시맨틱 웹 환경에 적합한 키워드 질의 처리를 지원할 수 있다.

참고문헌

[1] Tim Berners-Lee, James Hendler, and Ora Lassila, The Semantic Web, Scientific American, May 2001.

[2] World Wide Web Consortium, Resource Description Framework(RDF) Model and Syntax Specification, 2004

[3] World Wide Web Consortium, Resource Description Framework(RDF) Schema Specification 1.0, 2004

[4] S. Alexaki et al., "The ICS-FORTH RDFSuite: Managing Voluminous RDF Description Bases", In Proc. of SemWeb, 2001.

[5] J. Broekstra et al., "Sesame: A Generic Architecture for Storing and Querying RDF and RDF Schema", In Proc. of International Semantic Web Conference, 2002.

[6] S. Melnik, "Storing RDF in a Relational Database"

[7] J. Broekstra et al., "Sesame: A Generic Architecture for Storing and Querying RDF and RDF Schema", In Proc. of the 1st Int'l Semantic Web Conference, pp. 54-68, 2002.

[8] A. Harth and S. Decker. In Proc. of the 3rd Latin American Web Congress , 2005.

[9] A. Reggiori, D. van Gulik, and Z. Bjelogrlic, "Indexing and retrieving Semantic Web resources: the RDFStore model", In Proc. of SWAD-Europe Workshop on Semantic Web Storage and Retrieval, 2003.

[10] 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 first International Workshop on Semantic Web and Databases (SWDB), 2003.

[11] Gene Ontology, www.geneontology.org

저 자 소 개

김 연 희

2000년 2월 : 홍익대학교 컴퓨터공학과 졸업(학사)
2002년 2월 : 홍익대학교 컴퓨터공학과 대학원 졸업(석사)
2006년 8월 : 홍익대학교 컴퓨터공학과 대학원 졸업(박사)
관심분야 : 시맨틱 웹, XML, 분산 데이터베이스, 모바일 데이터베이스

신 혜 연

2005년 2월 : 대진대학교 컴퓨터공학과 졸업(학사)
2007년 2월 : 홍익대학교 컴퓨터공학과 대학원 졸업(석사)
관심분야 : 시맨틱 웹

임 해 철

1976년 2월 : 서울대학교 계산통계학과 졸업(이학사)
1978년 2월 : 한국과학기술원 전자계산학과 졸업(이학석사)
1988년 8월 : 서울대학교 컴퓨터공학과 졸업(공학박사)
1981년 ~ 현재 : 홍익대학교 컴퓨터공학과 교수
관심분야 : 시맨틱 웹, XML, 멀티미디어 데이터베이스

정 균 락

1978년 2월 : 서울대학교 계산통계학과 졸업(학사)
1980년 2월 : 한국과학기술원 전자계산학과 졸업(석사)
1991년 3월 : 미네소타대학교 Computer Science 졸업(박사)
1991년 ~ 현재 : 홍익대학교 컴퓨터공학과 교수
관심분야 : 알고리즘 설계 및 분석, VLSI 알고리즘