시맨틱 웹 문서에 대한 키워드 검색 및 랭킹 기법

김연희*, 오성균**

Keyword Search and Ranking Methods on Semantic Web Documents

Youn-Hee Kim*, Sung-Kyun Oh**

요 약

본 논문에서는 시맨틱 웹에서 온톨로지와 메타데이터를 기술하는 OWL 문서를 대상으로 하는 키워드 검색 기법과 랭킹 기법을 제안한다. 제안한 키워드 검색 기법은 OWL 문서에 대한 키워드 검색 결과의 단위를 정보 리소스로 정의하고 질의 키워드의 범위를 클래스와 프로퍼티의 이름은 물론 리터럴 데이터까지 확장하였다. 그리고 클래스나 프로퍼티의 계층 관계, 동등 관계 등 OWL 문서에 정의되어 있는 기본적인 추론 요소들을 고려하여 직접 기술되어 있지 않지만 새롭게 유도되는 정보도 키워드 검색에 반영하였다. 또한 키워드를 통해 간접적으로 의미적 관계를 맺고 있는 정보 리소스에 대한 검색이 가능하기 때문에 질의 키워드와 관련이 있는 많은 수의 정보 리소스들을 검색할 수 있다. 제안한 랭킹 기법은 OWL 문서의 특성을 고려하여 다양한 요소를 순위 결정에 참여시킴으로써 사용자의 검색 만족도를 높일 수 있다. 본 논문에서 제안한 키워드 검색 기법과 랭킹 기법은 방송 프로그램과 같은 디지털 콘텐츠의 검색 등 다양한 분야에서 활용될 수 있다.

Key Words : keyword search, ranking, semantic web, ontology, OWL.

ABSTRACT

In this paper, we propose keyword search and ranking methods for OWL documents that describe metadata and ontology on the Semantic Web. The proposed keyword search method defines a unit of keyword search result as an information resource and expands a scope of query keyword to names of class and property or literal data. And we reflected derived information by inference in the keyword search by considering the elements of OWL documents such as hierarchical relationship of classes or properties and equal relationship of classes. In addition, our method can search a large number of information resources that are relevant to query keywords because of information resources indirectly associated with query keywords through semantic relationship. Our ranking method can improve user's search satisfaction because of involving a variety of factors in the ranking by considering the characteristics of OWL. The proposed methods can be used to retrieve digital contents, such as broadcast programs.

I. 서 론

사용자가 제시한 질의 키워드를 포함하고 있는 데이터를 검색하여 결과로 반환하는 키워드 검색은 인터넷 상에 존재하는 웹 페이지뿐만 아니라 관계형 데이터베이스나 XML 문서 등 다양한 검색 대상을 위한 질의 처리 기법으로 널리 사용되고 있다[1, 2]. 키워드 검색은 데이터의 내부적 구조나 특별한 질의 언어를 몰라도 쉽게 원하는 데이터를 검색할 수 있다는 장점 때문에 일반 사용자들이 선호하는 검색 방식이다[1, 2]. 따라서 사용자의 검색 만족도와 검색 정확도를 향상시키면서 다양한 분야에서 키워드 검색을 활용하기 위한 연구가 계속 진행되고 있다.

키워드 검색은 검색 대상에 따라 반환되는 결과의 형태에 차이가 있다. 인터넷 상에서는 웹 페이지를 검색 대상으로 하여 사용자가 제시한 질의 키워드를 모두 포함하고 있는 웹 페이지를 검색 결과로 반환한다. 그리고 관계형 데이터베이스에서는 질의 키워드를 모두 포함하고 있는 튜플을 검색 결과로 반환하고 XML 문서는 질의 키워드를 모두 포함하고 있는 태그를 검색 결과로 반환한다. 반환되는 결과의 형태뿐만 아니라 검색 대상에 따라 키워드 검색의 처리 방식은 달

※ 본 논문은 2011년도 서일대학 학술연구비에 의해 연구되었음.
*부천대학교 e-비즈니스과 (yhkim@bc.ac.kr),
**서일대학교 컴퓨터소프트웨어과 (skoh@seoil.ac.kr)
접수일자 : 2012년 11월 21일, 수정완료일자 : 2012년 11월 27일, 최종게재확정일자 : 2012년 12월 3일

라지기 때문에 검색 대상의 특성을 고려하여 그에 적합한 키워드 검색 방식을 적용하는 것이 중요하다.

한편, 현재 웹의 확장된 개념으로 차세대 웹이라 인정받고 있는 시맨틱 웹(Semantic Web)은 정보 리소스(resource)의 개념과 다른 정보 리소스와의 의미적 관계를 사람은 물론 컴퓨터도 이해할 수 있는 정형화된 형태로 표현함으로써 보다 지능화된 정보 검색을 제공한다. 정보 리소스는 문서 전체, 문서의 일부분, 사람 등 정보를 가지고 있는 모든 대상을 의미하며 URI(Uniform Resource Identifier)로 식별한다. 시맨틱 웹에서 정보 리소스의 개념과 다른 정보 리소스와의 의미적 관계를 기술한 것을 메타데이터(metadata)라고 한다. 그리고 메타데이터를 정형화된 형태로 기술하기 위해 사용되는 개념과 개념들간의 관계를 정의한 것을 온톨로지(ontology)라고 한다. 시맨틱 웹에서는 온톨로지에 대한 의미적 해석과 추론을 통해 메타데이터에 직접 기술되어 있지 않은 새로운 지식을 생성할 수도 있기 때문에 일반적인 웹보다 더 정확하고 풍부한 정보 검색 결과를 반환하여 사용자의 만족도가 높다. 이러한 시맨틱 웹의 장점을 가능하게 하는 핵심 요소인 메타데이터와 온톨로지를 기술하기 위해 이전에는 RDF(Resource Description Framework)와 RDF 스키마가 많이 사용되었으나 최근에는 풍부한 표현력을 제공하는 OWL(Web Ontology Language)이 표준 언어로 많이 사용되고 있으며 현재 OWL 2.0까지 발표되었다[3].

시맨틱 웹 사용자들을 위해서 메타데이터와 온톨로지가 기술되어 있는 OWL 문서에 적합한 효과적인 키워드 검색 기법이 필요하다. OWL 문서는 기존 웹 페이지나 관계형 데이터베이스, XML 문서와는 다르게 그래프 형태의 구조를 가지고 있고 의미적 요소들을 포함하고 있기 때문에 그러한 특성을 반영한 키워드 검색 기법이 요구된다.

본 논문에서는 시맨틱 웹 환경에서 OWL 문서에 대한 효과적인 키워드 검색을 지원하기 위해 다음과 같은 내용을 제안한다.

첫째, OWL 문서의 특성을 고려하여 질의 키워드의 범위와 반환되는 검색 결과의 단위를 정의하고 키워드 검색의 유형을 분류한다. 둘째, OWL 문서에 적합한 키워드 검색을 지원하는 인덱스 구조와 질의 처리 전략을 제안한다. 마지막으로 키워드 검색을 요청한 사용자의 만족도를 높이기 위해 키워드 검색 결과를 랭킹하는 기법을 제안한다.

본 논문은 다음과 같이 구성된다. 2장에서는 시맨틱 웹 문서에 대한 기존 키워드 검색 기법에 대해 소개하고 문제점을 제시한다. 3장에서는 OWL 문서에 대한 질의 키워드의 범위와 반환되는 검색 결과의 단위를 정의하고 키워드 검색 유형을 분류한다. 그리고 분류된 키워드 검색 유형을 모두 지원하기 위한 인덱스 구조와 질의 처리 전략을 설명한다. 그리고 4장에서는 OWL 문서의 키워드 검색 시 반환되는 결과를 평가할 수 있는 랭킹 기법을 설명하고 5장에서 결론을 맺는다.

Ⅱ. 관련 연구

OWL 문서는 온톨로지와 메타데이터를 함께 기술하는데, 온톨로지 부분은 메타데이터를 기술하기 위해 사용되는 주요 개념들을 클래스로 정의하고 클래스의 특성이나 다른 클래스와의 의미적 관계를 프로퍼티로 정의한다. 그리고 클래스들 간의 계층 관계는 물론 같은 개념을 표현하고 있는 이음동의어 관계와 같이 클래스들 간의 다양한 관계를 정의할 수 있다. 프로퍼티에 대해서도 마찬가지이다. 메타데이터 부분에서는 정보 리소스를 온톨로지에 정의된 특정 클래스의 타입으로 선언하고 클래스의 정의 내용에 따라 정보 리소스의 특성이나 다른 클래스 타입의 정보 리소스와의 의미적 관계를 기술한다. OWL 문서는 온톨로지와 메타데이터의 내용을 주어(subject), 서술어(predicate), 목적어(object)로 구성된 트리플 구조의 문장(statement) 형태로 기술한다. 트리플 구조에서 주어는 온톨로지에서는 클래스가 되고 메타데이터에서는 정보 리소스가 된다. 서술어는 주어인 클래스나 정보 리소스에 대한 속성으로 프로퍼티가 된다. 목적어는 주어의 역할을 담당하는 클래스나 정보 리소스와 의미적 관계를 맺고 있는 또 다른 클래스나 정보 리소스, 또는 리터럴 데이터일 수도 있다. OWL 문서에 기술된 트리플 구조의 문장들은 노드와 간선에 모두 레이블이 표시된 방향성 그래프의 형태로 표현할 수 있다. 즉, 주어와 목적어는 노드로 표현되고 서술어는 주어와 목적어 노드를 연결하는 간선이 된다.

OWL 문서는 기본적으로 XML 문법 형태를 그대로 이용하여 기술되지만 트리플 구조의 정형화된 형태를 가지고 있고 트리 형태인 XML 문서와는 달리 그래프 형태이기 때문에 참고 문헌 [4]와 참고 문헌 [5]에서 제시한 XML에 대한 키워드 검색 기법과 랭킹 기법을 그대로 적용할 수 없다.

OWL은 같은 의미를 가지는 동등 클래스(equivalentClass), 프로퍼티의 역관계(inverse), 대칭관계(symmetric), 이행관계(transitive) 등 RDF와 RDF 스키마에 비해 다양한 관계의 정의가 가능하다. 따라서 RDF와 RDF 스키마를 이용해 기술된 문서에 대한 키워드 검색 기법의 기존 연구 내용을 OWL 특성에 맞게 보완하고 확장할 필요가 있다.

RDF와 RDF 스키마에 대한 키워드 검색 기법을 제안한 참고 문헌 [1]은 RDF와 RDF 스키마로 작성된 시맨틱 웹 문서를 그래프 형태로 표현하되 관련 있는 노드와 간선을 검색 결과로 함께 반환한다. 그리고 키워드 검색 성능을 향상시키기 위해 그래프의 크기를 줄이는 그래프 축약 기법을 제안하고 축약된 그래프 구조에 적용 가능한 랭킹 기법을 제안한다. 참고 문헌 [6]은 메타데이터와 온톨로지를 인스턴스 그래프와 스키마 그래프로 각각 표현하고 다양한 의미적 관계를 시맨틱 경로로 정의하여 키워드 검색 시 고려한다. 참고 문헌 [7]은 RDF와 RDF 스키마를 그래프 형태로 표현하고 키워드를 직접적으로 포함하고 있는 노드뿐만 아니라 간접적으로 포함하고 있는 노드까지 검색 결과에 포함시키기 위한

인덱스 구조를 제안한다.

RDF와 RDF 스키마를 위한 키워드 검색 기법을 제안한 기존 연구들은 클래스들 간의 계층 관계와 같이 온톨로지에 정의된 기본적인 추론 요소들을 고려하지 않고 질의 키워드를 클래스나 프로퍼티의 이름으로 제한하거나 리터럴 데이터로만 제한하는 경우가 많다. 그리고 검색된 결과와 연관된 다른 정보를 함께 제공하지 않는 경우가 많다. 또한 키워드 검색이 그래프 탐색에 기반을 두고 수행되기 때문에 사용자의 검색 요청 시 복잡한 그래프 탐색이 요구된다. 특히, 질의 키워드가 여러 개 제시된 경우 그래프 탐색 과정은 더욱 복잡해진다.

OWL 문서에 대한 키워드 검색 기법을 제안한 참고 문헌 [2]에서는 트리플 구조 단위로 키워드 검색을 지원한다. 그리고 노드 간의 연관성을 쉽게 판단하기 위해 Multi-Numbering Scheme을 사용하고 그래프 상의 노드들을 병합하는 알고리즘을 제안한다. 하지만 참고 문헌 [2]에서도 클래스의 계층 관계나 동등 관계 등 온톨로지에 정의된 추론적 요소들을 고려하지 않고 키워드 검색 처리 시 병합 과정이 실시간으로 이루어지기 때문에 검색 시간에 영향을 끼칠 수 있다. 또한 누락되는 검색 결과가 발생하는 경우도 있다.

Ⅲ. OWL 문서를 위한 키워드 검색 기법

본 장에서는 온톨로지와 메타데이터를 함께 기술하는 OWL 문서에서 질의 키워드의 범위와 키워드 검색 결과의 형태를 새롭게 정의한다. 그리고 OWL 문서에 기술된 온톨로지에 대한 기본적인 추론적 요소들을 고려하여 직접적으로 기술되지 않은 새로운 정보까지 키워드 검색 과정에 포함시키고 키워드 검색 처리 시 그래프 탐색이 실시간으로 수행될 필요가 없도록 검색에 필요한 정보를 미리 저장하는 인덱스와 저장 구조를 제안한다.

1. 질의 키워드의 범위와 키워드 검색 결과의 형태 정의

특정 도메인의 지식 구조를 표현한 온톨로지 중심의 키워드 질의에서는 클래스의 이름이나 프로퍼티의 이름이 질의 키워드로 제시되는 경우가 많다. 그리고 정보 리소스의 특성을 기술한 메타데이터 중심의 키워드 질의에서는 리터럴 데이터의 내용이 질의 키워드로 제시되는 경우가 많다. 따라서 본 논문에서는 사용자가 제시하는 질의 키워드의 범위를 클래스의 이름와 프로퍼티의 이름은 물론 프로퍼티의 값으로 기술된 리터럴 데이터까지 확대한다.

OWL 문서에 대한 키워드 검색 결과는 기본적으로 질의 키워드를 포함하고 있는 정보 리소스를 질의 결과의 반환 단위로 한다. 그런데 검색 기능을 제공하는 다양한 포털 사이트에서 제공하고 있는 연관 검색처럼 일반 사용자들은 자신이 제시한 키워드를 포함하고 있는 검색 결과와 연관된 다른 정보들까지 함께 결과로 반환받기를 원하는 경향이 있다. 그래서 본 논문에서는 질의 키워드를 포함하고 있는 정보 리소스를 검색 결과로 반환하면서 그 정보 리소스의 다른 특성 값들도 함께 반환하여 사용자의 검색 만족도를 향상시키고자 한다.

본 논문에서는 사용자가 제시한 질의 키워드의 개수에 따라 한 개의 질의 키워드가 제시되는 단순 키워드 검색과 여러 개의 질의 키워드가 제시되는 복합 키워드 검색으로 분류한다. 특히, 여러 개의 질의 키워드가 제시되는 복합 키워드 검색의 경우에는 질의 키워드들을 포함하는 모든 정보 리소스들의 집합을 검색 결과로 반환한다.

2. 키워드 검색을 위한 인덱스와 저장 구조

본 논문에서 제안한 OWL 문서를 위한 키워드 인덱스 구조를 설명하기 위해 그래프 형태로 표현한 그림 1의 온톨로지 예제와 그림 2의 메타데이터 예제를 사용한다.

그림 1은 방송 프로그램과 관련하여 OWL 문서에 기술된 온톨로지를 그래프 모델로 표현한 것이다. 타원은 클래스를 의미하고 화살표는 프로퍼티를 의미한다. 그리고 클래스가 프로퍼티의 값으로 리터럴 데이터를 가지는 제약 사항은 직사각형으로 표현한다. OWL로 표현할 수 있는 추론적 요소 중 기본 요소인 하위 클래스(subClass)와 이음동의어 클래스 관계를 표현하는 동등 클래스(equivalentClass) 관계는 각각 이중선 화살표와 일반 이중선으로 표현한다. 그림 1에서 각 클래스에 표시되어 있는 숫자는 클래스를 식별하기 위한 아이디로 Dewey 방식을 이용하여 지정한다. 예를 들어 "Creator"는 개념적으로 상위 클래스가 존재하지 않고 OWL 문서 내에서 처음으로 정의된 클래스이기 때문에 Dewey 방식에 따라 "1"의 아이디가 부여되고 "Creator" 클래스의 하위 클래스 중 첫 번째 클래스인 "Director" 클래스는 "1.1"의 아이디가 부여된다. 그리고 "Director" 클래스와 같은 의미를 가지는 동등 클래스인 "Producer" 클래스는 "Director" 클래스와 같은 아이디인 "1.1"이 부여된다. 프로퍼티도 같은 방식으로 아이디가 부여되는데 그림 1에서는 생략한다.

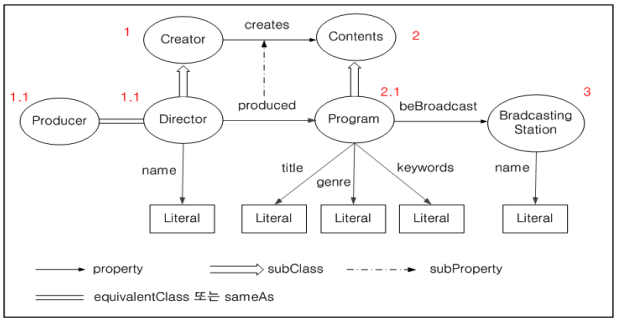

그림 1. 방송 프로그램에 대한 온톨로지 예

그림 2는 방송 프로그램과 관련하여 그림 1에 제시된 온톨로지에 기반을 두고 작성된 메타데이터를 그래프 모델로 표현한 것이다. 타원은 정보 리소스를 의미하고 화살표는 프로퍼티를 의미한다. 리터럴 데이터로 표현된 정보 리소스의 프로퍼티 값은 직사각형으로 표현한다. 본 논문에서는 모든 정보 리소스가 개념을 명확하게 제시하기 위해 온톨로지에 정의된 특정 클래스 타입으로 반드시 선언된다고 가정한다. 정보 리소스는 URI로 식별되지만 본 논문에서는 참고 문헌 [7]과 같이 정보 리소스를 식별하는 것은 물론 정보 리소스의 클래스 타입을 명시적으로 나타내기 위해 클래스의 이름과 OWL 문서 내 작성된 순서에 따라 각 정보 리소스마다 고유의 아이디를 부여한다. 그림 2에서 아이디가 "Program_1"인 정보 리소스는 그림 1에서 정의한 "Program" 클래스 타입이고 OWL 문서에서 첫 번째로 출현한 것임을 의미한다.

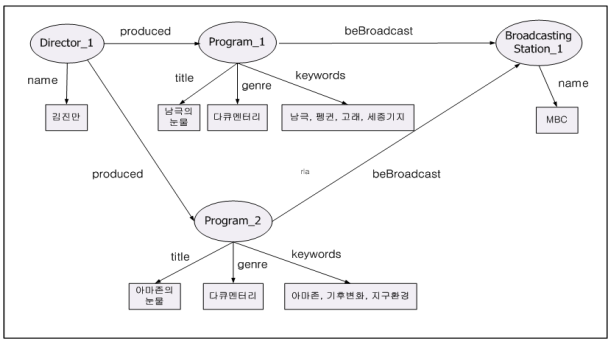

그림 2. 방송 프로그램에 대한 메타데이터 예

본 논문에서는 OWL 문서의 온톨로지에서 정의한 하위 클래스 관계와 동등 클래스 관계를 이용한 추론을 통해 새롭게 유도되는 정보도 키워드 검색의 대상으로 포함시키고 사전에 그래프 탐색을 통해 키워드 검색 처리에 필요한 정보들을 인덱스와 저장 구조에 미리 저장해둠으로써 키워드 검색에 소요되는 시간을 단축시키고자 한다. 그리고 질의 키워드를 포함하고 있는 정보 리소스가 가지고 있는 다른 유용한 정보들도 함께 검색 결과로 반환함으로써 일반 사용자의 검색 만족도를 향상시키는데 목표를 두고 있다.

본 논문에서는 클래스의 이름을 대상으로 하는 키워드 검색을 위해 클래스에 대한 정보를 그림 3과 같이 관계형 데이터베이스의 테이블 구조를 이용해서 저장한다. 그림 3에서 CID 필드는 앞서 설명한 Dewey 방식에 의해 부여된 클래스의 아이디를 저장하고 name 필드는 클래스의 이름을 저장한다. Dewey 방식을 이용해 클래스의 아이디를 부여했기 때문에 이후 키워드 검색을 위한 질의 처리 시 상위 클래스와 하위 클래스의 계층 관계를 쉽게 판단할 수 있다. 그리고 그림 3에서 제시한 "Director"와 "Producer" 클래스가 맺는 동등 관계의 경우는 같은 클래스 아이디를 부여하여 표현함으로써 마치 같은 클래스인 것처럼 동등 관계에 대한 처리를 단순화시킨다. 그림 3의 클래스 테이블은 그림 1의 온톨로지에 정의된 클래스 정보를 저장한 예를 보여준다. 클래스의 이름을 대상으로 하는 키워드에 대한 정보를 테이블을 이용한 간단한 저장 구조에 저장할 수 있는 이유는 각 정보 리소스마다 어떤 클래스 타입인지를 쉽게 판단할 수 있도록 고유의 아이디를 부여해서 사용자가 제시한 키워드를 이름으로 가지는 클래스 타입에 속하는 정보 리소스를 쉽게 판단할 수 있기 때문이다.

CID	name
1	Creator
1.1	Director
1.1	Producer
2	Contents
2.1	Program
3	BroadcastingStation

그림 3. 클래스 테이블 구성 예

본 논문에서는 프로퍼티의 이름을 대상으로 하는 키워드 검색을 위해 프로퍼티에 대한 정보를 클래스와 마찬가지로 그림 4와 같이 관계형 데이터베이스의 테이블 구조를 이용해서 저장한다. 그림 4의 프로퍼티 테이블은 그림 1의 온톨로지에 정의된 프로퍼티 정보를 저장한 예를 보여준다.

PID	name
1	creates
1.1	produced
2	name
3	title
4	genre
5	keywords
6	beBroadcast
7	name

그림 4. 프로퍼티 테이블 구성 예

본 논문에서는 리터럴 데이터의 내용을 대상으로 하는 키워드 검색을 위해 별도의 키워드 인덱스를 제안한다. 본 논문에서 제안한 키워드 인덱스는 키워드 검색에 일반적으로 많이 활용되는 역 인덱스(inverted index) 구조를 이용한다. 역 인덱스는 문서에 존재하는 키워드들을 인덱스 키 값으로 하면서 키워드를 포함하고 있는 문서들을 쉽게 추출할 수 있도록 키워드 리스트 영역과 포스팅 리스트 영역으로 나뉜다. 본 논문에서 제안한 키워드 인덱스도 키워드 리스트 영역과 포스팅 리스트 영역으로 나뉘지만 포스팅 리스트 영역에 문서에 대한 정보가 아닌 OWL 문서에 기술되어 있는 정보 리소스에 대한 정보가 저장된다는 차이가 있다.

그림 5는 본 논문에서 제안한 리터럴 키워드 인덱스의 구조를 보여준다. 키워드 리스트 영역은 OWL 문서에서 리터

럴 데이터의 내용을 대상으로 추출된 각각의 키워드와 정보 리소스들 중에서 그 키워드가 가장 많이 출현한 최대 빈도수를 저장하는 키워드 노드로 구성된다. 키워드의 최대 빈도수는 이후 정보 리소스의 랭킹을 평가하는 중요 요소로 활용된다. 리소스-프로퍼티 리스트 영역은 프로퍼티를 통해 각 키워드와 직접 연관되어 있는 리소스에 대한 정보를 저장하는 리소스-프로퍼티 노드로 구성된다. 리소스-프로퍼티 노드에는 키워드와 연관된 정보 리소스의 아이디(RID)와 관련된 프로퍼티의 아이디(PID)가 저장된다. 리터럴 키워드 인덱스를 통해 각 키워드와 직접적으로 연관되어 있는 정보 리소스에 대한 내용을 쉽게 확인할 수 있다.

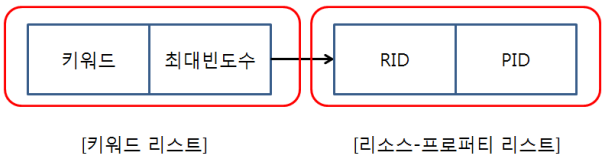

그림 5. 리터럴 키워드 인덱스의 구조

그림 6은 그림 2의 메타데이터에 기술된 내용을 가지고 그림 5의 리터럴 키워드 인덱스를 구성한 예를 보여준다. 그림 6에서는 이해를 돕기 위해 프로퍼티 아이디와 함께 프로퍼티의 이름도 함께 제시하였다. 즉, "title(3)"은 프로퍼티 아이디가 3인 "title" 프로퍼티를 의미하는 것이다.

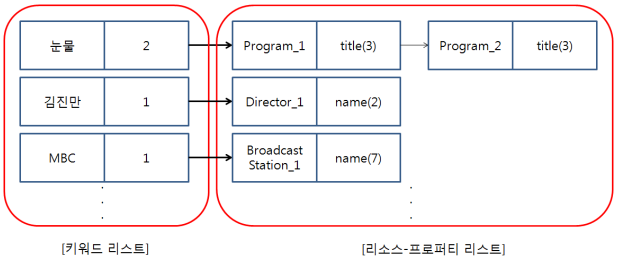

그림 6. 리터럴 키워드 인덱스 구성 예

본 논문에서는 키워드 검색 결과로 질의 키워드를 포함하고 있는 정보 리소스만을 반환하는 것이 아니라 정보 리소스가 가지고 있는 모든 속성 값, 즉 프로퍼티 값들을 반환하여 연관 검색이 가능하도록 한다. 이를 위해 정보 리소스에 대해 프로퍼티 값으로 기술된 모든 값을 클러스터링하기 위한 리소스 인덱스를 제안한다. 본 논문에서 제안한 리소스 인덱스도 역 인덱스 구조를 이용하며 리소스 리스트와 프로퍼티-값 리스트 영역으로 구성된다. 그림 7은 본 논문에서 제안한 리소스 인덱스의 구조를 보여준다. 그림 7에서 리소스 리스트 영역은 OWL 문서에서 프로퍼티의 값이 하나라도 존재하는 정보 리소스를 모두 저장한다. 그리고 프로퍼티-값 리스트 영역은 각 정보 리소스가 가지고 있는 모든 프로퍼티와 그 값을 저장한다.

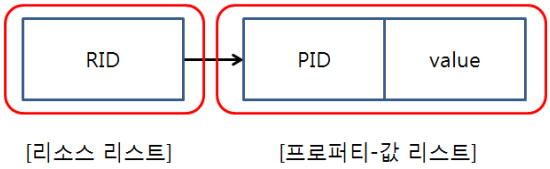

그림 7. 리소스 인덱스 구조

그림 8은 그림 2의 메타데이터에 기술된 내용을 가지고 그림 7의 리소스 인덱스를 구성한 예를 보여준다. 이해를 돕기 위해 프로퍼티는 아이디와 함께 이름도 함께 제시하였다.

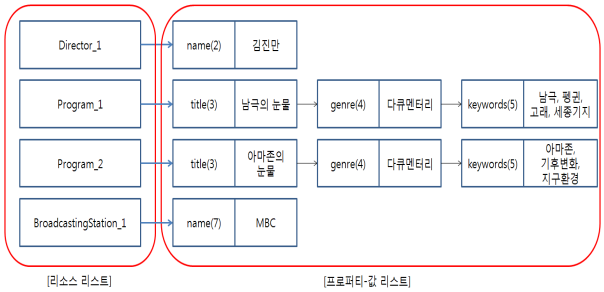

그림 8. 리소스 인덱스 구성 예

리터럴 키워드 인덱스와 리소스 인덱스를 이용하면 질의 키워드가 한 개인 단순 검색에 대해서는 쉽게 처리할 수 있지만 여러 개의 질의 키워드를 모두 포함하는 정보 리소스들의 집합을 검색해야 하는 복합 검색에 대해서는 올바른 결과를 반환할 수 없다. 따라서 복합 검색을 위한 별도의 저장 구조가 요구된다.

그림 9는 그림 1의 온톨로지를 표현한 그래프에 대한 탐색을 통해 클래스가 다른 클래스와 프로퍼티를 통해 의미적 관계를 맺고 있는 모든 경로 정보를 추출하여 저장하는 경로 테이블의 구성 예를 보여준다. 경로는 클래스들 간의 의미적 관계를 표현하는 것으로 프로퍼티를 통해 의미적 관계를 맺고 있는 클래스간의 기본 경로는 “클래스이름.프로퍼티이름.클래스이름” 으로 표현한다. 본 논문에서는 경로의 시작 클래스와 끝 클래스가 일치하지 않는다고 가정하고 경로의 길이는 해당 경로 표현에 존재하는 프로퍼티의 개수로 정의한다. 여러 개의 기본 경로로 구성된 복잡한 경로를 통해 의미적 관계를 맺는 클래스가 존재할 수도 있다. 예를 들어, 그림 9에서 “Director.produced.Program.beBroadcast.BroadcastingStation” 경로는 "Director" 클래스와 "BroadcastingStation"클래스가 직접적인 관계를 맺고 있지는 않지만 다른 클래스와의 의미적 관계를 통해 간접적으로 관계를 맺고 있음을 의미하고 경로의 길이는 2가 된다. 그림 9의 경로 테이블에서 path 필드는 온톨로지에서 추출할 수 있는 클래스간의 경로를 저장하고 length 필드는 그 경로의 길이를 저장한다.

path	length
Director.produced.Program.beBroadcast.BroadcastingStation	2
Director.produced.Program	1
Program.bebroadcast.BroadcastingStation	1

그림 9. 경로 테이블 구성 예

키워드 검색의 반환 결과는 정보 리소스이기 때문에 경로 테이블에 표현된 경로 상에 존재하는 모든 리소스들의 간접적 연결 관계도 별도의 저장 구조로 저장할 필요가 있다. 하지만 OWL 문서의 메타데이터 부분은 복잡한 그래프 형태로 표현되는 것이 일반적이기 때문에 그래프 탐색을 통해 리소스간의 복잡한 경로 관계를 모두 추출하여 저장 구조에 미리 저장해두면 키워드 검색 처리를 위한 전처리 과정에 많은 시간이 소요된다. 따라서 본 논문에서는 OWL 문서의 기본 구조인 트리플 구조에 기반하여 OWL 문서에 기술된 모든 정보 리소스들 간의 관계를 그림 10과 같이 테이블을 이용해 저장한다. 그림 10의 트리플 테이블에서 RID1과 RID2 필드는 관계를 맺고 있는 정보 리소스들을 각각 저장하고 PID 필드는 정보 리소스들이 맺고 있는 의미적 관계를 표현한 프로퍼티의 아이디를 저장한다.

RID1	PID	RID2
Director_1	produced(1.1)	Program_1
Director_1	produced(1.1)	Program_2
Program_1	beBroadcast(6)	BroadcastingStation_1
Program_2	beBroadcast(6)	BroadcastingStation_1

그림 10. 트리플 테이블 구성 예

여러 개의 질의 키워드를 모두 포함하는 정보 리소스들의 집합을 검색하는 복합 검색의 경우에는 먼저 클래스 테이블, 프로퍼티 테이블, 리터럴 키워드 인덱스를 통해서 각 질의 키워드를 직접적으로 포함하고 있는 모든 정보 리소스를 검색한다. 그리고 나서 검색된 정보 리소스들이 직·간접적으로 의미적 관계를 맺고 있는지를 판단하기 위해 검색된 정보 리소스의 타입으로 선언된 클래스들 간에 의미적 경로가 존재하는지를 그림 9에서 제시한 경로 테이블을 통해 검색한다. 마지막으로 경로 테이블에서 검색된 경로에 의해 의미적 관계를 맺고 있는 모든 정보 리소스들을 찾아 검색 결과로 반환하기 위해 트리플 테이블을 검색하면 된다. 이때, 경로의 길이에 따라 트리플 테이블에 대한 검색은 조인 연산을 필요로 한다. 경로 길이에 따라 필요한 조인 연산을 표현한 SQL 문을 표 1과 같이 미리 정의해두면 검색 시간을 단축하는데 도움이 된다. 예를 들어, 표 1에서 경로 길이가 1인 경우는 조인 연산이 필요 없지만 경로 길이가 2인 경우는 한 번의 자기 조인(self-join)이 필요하다.

표 1. 경로 길이에 따라 필요한 트리플 테이블의 SQL문

길이	SQL문
1	SELECT RID1, PID, RID2 FROM Triple WHERE RID1 = start and RID2 = end;
2	SELECT t1.RID1, t1.PID, t1.RID2, t2.PID, t2.RID2 FROM Triple t1, Triple t2 WHERE t1.RID1 = start and t2.RID2 = end and t1.RID2 = t2.RID1;

3. 키워드 검색 처리 전략

본 논문에서 제안한 인덱스 구조와 저장 구조를 이용하여 키워드 검색을 처리하는 과정은 표 2에서 제시한 단계와 같다. 단순 검색의 경우에는 1단계와 2단계 수행 후 바로 5단계의 처리 과정을 수행하면 된다. 복합 검색의 경우에는 모든 단계를 순차적으로 수행하면 된다. 본 논문에서는 클래스 이름과 리터럴 데이터에 대한 키워드에 초점을 맞추어 키워드 검색 전략을 제시하였다. 프로퍼티 이름에 대한 키워드 검색 전략은 클래스 이름에 대한 키워드 검색 전략과 같다.

표 2. 키워드 검색 처리 과정

단계	처리 내용
1	• 질의 키워드를 클래스 테이블에서 검색 - 클래스 아이디를 이용해 검색된 클래스의 하위 클래스도 모두 검색
2	• 질의 키워드를 리터럴 키워드 인덱스에서 검색 - 질의 키워드와 직접 연관된 리소스 모두 검색 - 리소스 아이디를 이용해 각 리소스의 타입 클래스를 확인
3	• 2단계에서 검색된 클래스들간에 경로가 존재하는지 경로 테이블에서 확인 - 클래스들간의 의미있는 관계가 존재하는지를 확인
4	• 3단계에서 검색된 경로 길이에 따라 미리 정의된 SQL문을 이용해 트리플 테이블에서 해당 경로에 존재하는 모든 정보 리소스를 검색
5	• 리소스 인덱스를 이용해 4단계에서 검색된 모든 정보 리소스의 연관 정보를 검색하여 결과로 반환

그림 1과 그림 2에 제시된 OWL 문서의 온톨로지와 메타데이터 예제에 대해 표 2의 질의 처리 전략을 적용한 두 가지 키워드 검색 예를 살펴보자.

먼저 "Contents"가 질의 키워드로 주어진 단순 검색의 경우를 살펴보자. 1단계에서 클래스 테이블을 검색하여 "Contents" 키워드를 클래스 이름으로 가지고 있는 클래스의 아이디를 검색하면 "2"가 된다. 그리고 클래스 테이블에서 아이디가 "2.1"인 하위 클래스 "Program"을 쉽게 검색할 수 있다. 2단계에서 리터럴 키워드 인덱스에는 "Contents"라는 키워드를 가지고 있는 정보 리소스가 존재하지 않음을 확인할 수 있다. 마지막으로 5단계에서 리소스 인덱스를 이용해 "Contents" 클래스는 물론 하위 클래스인 "Program" 클래스 타입으로 선언된 정보 리소스에 대해 그 정보 리소스가 가지고 있는 모든 프로퍼티의 값을 함께 검색하여 결과로 반환하

면 된다. 따라서 "Program_1"과 "Program_2" 정보 리소스가 가지고 있는 모든 프로퍼티의 값이 검색 결과로 반환된다.

"김진만"과 "MBC"가 질의 키워드로 주어진 복합 검색의 경우를 살펴보자. 1단계에서 클래스 테이블을 검색하면 일치하는 클래스가 없다. 2단계에서는 "김진만" 키워드를 포함하고 있는 "Director_1" 정보 리소스와 "MBC" 키워드를 포함하고 있는 "BroadcastingStation_1" 정보 리소스가 검색된다. 이제 3단계에서 "Director_1" 정보 리소스의 타입 클래스인 "Director" 클래스와 "BroadcastingStation_1" 정보 리소스의 타입 클래스인 "BroadcastingStation" 클래스로 구성된 "Director.produced.Program.beBroadcast.BroadcastingStation" 경로를 경로 테이블에서 검색한다. 이 경로는 길이가 2이므로 한 번의 자기 조인을 포함하는 SQL문을 이용해 4단계에서 의미적 관계를 맺고 있는 모든 정보 리소스의 집합을 트리플 테이블에서 검색한다. 4단계의 검색 결과로 "Director_1", "Program_1", "Program_2", "BroadcastingStation_1" 정보 리소스가 검색된다. 이제 마지막 5단계에서 "Director_1", "Program_1", "Program_2", "BroadcastingStation_1" 정보 리소스가 가지고 있는 모든 프로퍼티 값을 검색하여 함께 결과로 반환하면 된다.

Ⅳ. OWL 문서를 위한 랭킹 기법

본 논문에서 제안한 인덱스와 저장 구조를 가지고 키워드 질의 처리 전략을 적용하면 질의 키워드와 관련이 있는 많은 수의 정보 리소스들이 결과로 반환된다. 따라서 반환된 결과 중에 질의 키워드와 가장 관련성이 높고 사용자가 원하는 정보를 가지고 있는 정보 리소스를 우선적으로 반환하여 사용자의 검색 만족도를 향상시킬 필요가 있다. 기존의 웹 문서를 대상으로 하는 키워드 검색에서는 키워드가 문서에 발생하는 빈도수나 다른 문서로부터 링크된 개수, 다른 문서를 링크하고 있는 개수 등을 이용하여 검색 결과를 정렬하지만 OWL 문서를 대상으로 하는 키워드 검색에서는 기존 방식과 다른 랭킹 기법이 요구된다.

따라서 본 논문에서는 OWL 문서의 특성을 고려하여 클래스들간의 계층 구조, 도메인 전문가들의 의견, 키워드의 빈도수, 연관 정보를 포함하고 있는 정도 등 다각적인 요소들을 반영해서 결과로 반환된 정보 리소스들의 랭킹을 평가하기 위해 정보 리소스와 질의 키워드와의 관련 정도를 계산할 수 있는 여러 개의 평가 함수를 제안한다.

사용자들이 선호하는 클래스의 타입으로 선언된 정보 리소스는 그렇지 않은 정보 리소스에 비해 더 높게 평가되어야 한다. 본 논문에서 제안한 클래스의 선호 점수를 계산하기 위한 함수는 수식 (1)과 같다.

$$Score(c) = \frac{Level(c)}{MaxLevel(H)} + DomainWeight(c) + \frac{RF(c)}{MaxRF(C)} \quad (1)$$

클래스 c의 선호 점수는 세 가지 항목의 점수 합으로 계산된다. 일반적으로 사용자들은 좀 더 명확한 개념을 선호하고 클래스간의 계층 구조에서 하위 클래스로 갈수록 좀 더 자세한 개념을 표현하고 있다고 할 수 있다. 따라서 수식 (1)에서 클래스 c가 속해 있는 전체 클래스 계층 구조 H에서 클래스 c의 레벨을 의미하는 Level(c)를 계층 구조의 최대 레벨을 의미하는 MaxLevel(H)로 나누어 계층 구조에서 하위 클래스로 갈수록 높은 점수를 받도록 하는 것은 물론 0과 1사이의 값으로 정규화시킨다. 수식 (1)에서 DomainWeight(c)는 검색 대상인 OWL 문서의 온톨로지에서 클래스 c의 중요도를 도메인 전문가들이 0과 1사이의 값으로 평가한 점수를 의미한다. 그리고 수식 (1)에서 RF(c)는 클래스 c의 타입으로 선언된 리소스의 개수(Resource Frequency)를 의미한다. 많은 리소스들의 타입으로 선언된 클래스는 사용자가 선호하는 클래스일 가능성이 높으므로 수식 (1)에서 클래스 c 타입의 리소스 개수인 RF(c)를 전체 클래스 집합 C에서 타입이 선언된 리소스의 최대 개수인 MaxRF(C)로 나눈다.

정보 리소스가 여러 개의 프로퍼티로 특성을 기술하거나 다른 정보 리소스와의 관계를 맺고 있다면 그렇지 않은 정보 리소스에 비해 사용자들이 원하는 정보를 많이 가지고 있을 확률이 높으므로 높은 점수가 부여되어야 한다. 따라서 정보 리소스의 자체의 구조적 점수를 계산하는 Score(r)은 다음 수식 (2)와 같이 정의한다.

$$Score(r) = \alpha * DP(r) + (1-\alpha) * OP(r) \quad (2)$$

수식 (2)에서 정보 리소스 r의 구조적 점수는 정보 리소스 r에 연결된 프로퍼티의 개수로 결정된다. 프로퍼티는 두 가지 종류로 구분되는데 리터럴 데이터 값과 연결되는 DataProperty와 다른 정보 리소스와의 관계를 표현하는 ObjectProperty가 있다. DP(r)은 정보 리소스 r에 연결된 DataProperty의 개수를 의미하고 OP(r)은 정보 리소스 r에 연결된 ObjectProperty의 개수를 의미한다. OWL 문서가 사용되는 도메인의 특성에 따라 DataProperty의 중요성이 강조될 수도 있고 ObjectProperty의 중요성이 강조될 수도 있기 때문에 이를 조정할 수 있는 조정 계수 α를 이용한다. α는 0과 1사이의 값으로 결정한다.

Score(c)와 Score(r) 함수를 이용해 타입 클래스의 선호 점수와 정보 리소스의 구조적 점수가 각각 계산되면 최종적으로 질의 키워드와 정보 리소스와의 관련성까지 고려하여 정보 리소스의 순위를 결정할 수 있는 값을 수식 (3)에서 제시한 함수를 이용해 계산한다.

$$Rank(r, Q) = \sum_{k \in Q} \frac{KF(r,k)}{MaxKF(R,k)} + Score(c) * Score(r) \quad (3)$$

정보 리소스 r과 질의 키워드 집합 Q의 관련 정도를 고려해 정보 리소스의 최종 순위 결정 값을 수식 (3)에서 제시한

Rank(r, Q) 함수를 이용해 계산한다. 정보 리소스의 순위 결정 값은 수식 (1)과 수식 (2)에서 제시한 클래스 점수와 정보 리소스의 구조적 점수의 영향을 받는다. 수식 (3)에서 KF(r, k)는 질의 키워드 집합 Q에 속하는 각 질의 키워드 k가 리소스 r에 나타나는 빈도수를 의미한다. 키워드가 정보 리소스 r에 빈번하게 나타날수록 키워드와 정보 리소스간의 관련성이 높다고 할 수 있으므로 수식 (3)에서는 각 키워드 k가 정보 리소스 r에 발생하는 빈도수를 모든 정보 리소스 집합 R을 대상으로 각 키워드 k가 발생하는 최대 빈도수로 나누어 정규화한다. 그리고 질의 키워드 집합 Q에 속하는 모든 질의 키워드에 대해 합계를 구하여 순위 결정 값에 반영한다.

여러 개의 정보 리소스로 구성된 정보 리소스 집합이 검색 결과로 반환되는 경우에는 모든 정보 리소스의 Rank(r, Q) 함수의 계산 결과 값을 모두 더하여 그 합계가 가장 높은 정보 리소스의 집합을 결과 리스트에서 우선적으로 제시하면 된다.

Ⅴ. 결론

시맨틱 웹의 장점을 보편화시키기 위해서는 메타데이터를 기술하는 OWL 문서에 대한 키워드 검색 기법과 랭킹 기법에 대한 연구가 필요하다. 본 논문에서는 OWL 문서의 정보 저장 단위인 정보 리소스를 키워드 검색 결과의 기본 단위로 정의하고 클래스 이름과 프로퍼티의 이름, 그리고 데이터의 내용까지 질의 키워드의 범위로 확장하여 정의하였다. 그리고 한 개의 질의 키워드가 주어지는 경우는 물론 여러 개의 질의 키워드가 주어지는 복합 검색의 경우에도 효과적으로 키워드 검색을 수행할 수 있도록 필요한 인덱스 구조와 저장 구조를 제안하였다.

본 논문에서 제안한 키워드 인덱스와 리소스 인덱스를 통해 질의 키워드를 직접 포함하고 있는 정보 리소스를 빠르게 검색할 수 있는 것은 물론 정보 리소스가 가지고 있는 다른 정보도 함께 검색 결과로 반환하는 연관 검색이 가능하다. 그리고 경로 테이블과 트리플 테이블을 이용해 질의 키워드가 여러 개 주어진 복합 검색에서 정보 리소스들이 직접적으로 관계를 맺고 있지 않더라도 다른 정보 리소스를 통해 간접적 관계를 맺고 있는지를 쉽게 판단하여 정보 리소스간의 의미적 관계에 기반을 둔 키워드 검색이 가능하게 된다. 또한 클래스와 프로퍼티에 대해 계층 관계와 동등 관계를 쉽게 판단할 수 있도록 아이디를 부여하여 계층 관계는 물론 동등 관계를 이용한 키워드 검색을 지원한다. OWL 문서에 기술된 다양한 요소들을 이용하여 키워드 검색을 하게 되는 경우 많은 정보 리소스가 결과로 반환된다. 본 논문에서는 정보 리소스의 클래스 타입과 정보 리소스 자체의 점수를 고려하고 질의 키워드와 정보 리소스의 관련성을 수치화하여 평가함으로써 검색된 정보 리소스들을 중요도에 따라 정렬하여 제공함으로써 사용자의 검색 만족도를 높이고자 한다.

향후에 본 논문에서 제안한 키워드 검색 기법과 랭킹 기법을 적용한 프로토타입 시스템을 개발하여 정확성과 재현율, 그리고 질의 처리 시간에 대해 제안한 기법의 우수성을 평가하고자 한다.

참 고 문 헌

[1] 김진하, 송인철, 김명호, “RDF 데이타에 대한 효율적인 키워드 검색 기법”, 정보과학회논문지, 제35권 제6호, pp. 495-504, 2008.

[2] 김학수, 손진현, “RDF/S 및 OWL 문서에 대한 키워드 검색 알고리즘”, 한국정보처리학회 춘계학술발표대회 논문집, 제16권 제1호, pp. 321-324, 2009.

[3] OWL 2 Web Ontology Language Primer, “http://www.w3.org/TR/2009/REC-owl2-primer-20091027/”.

[4] Lin Guo, Feng Shao, Chavdar Botev, and Jayavel Shanmugasundaram, "XRANK: Ranked Keyword Search over XML Documents", Proceedings of ACM SIGMOD Conference, pp. 16-27, 2003.

[5] Ziyang Liu, Jeffrey Walker, and Yi Chen, "XSeek: A Semantic XML Search Engine Using Keywords," Proceedings of International Conference on Very Large Data Bases, pp. 1330-1333, 2007.

[6] Jihyun Lee, Jun-Ki Min, and Chin-Wan Chung, "An Effective Semantic Search Technique using Ontology", Proceeding of WWW, pp. 1057-1058, 2009.

[7] 김연희, 신혜연, 임해철, 정균락, “시맨틱 웹 데이터의 키워드 질의 처리를 위한 인덱싱 및 저장 기법”, 한국컴퓨터정보학회 논문지, 제12권 제5호, pp. 93-102, 2007.

저자

김 연 희(Youn-Hee Kim)

- 2000년 : 홍익대학교 컴퓨터공학과 (공학사)
- 2002년 : 홍익대학교 컴퓨터공학과 (공학석사)
- 2006년 : 홍익대학교 컴퓨터공학과 (공학박사)
- 2007년 ~ 현재 : 부천대학교 e-비즈니스과 강의전담교수

<관심분야> : 시맨틱 웹, 데이터베이스, XML, 이러닝 시스템

오 성 균(Sung-Kyun Oh)

- 1981년 : 홍익대학교 전자계산학과(이학사)
- 1984년 : 연세대학교 전자계산학과(공학석사)
- 1999년 : 홍익대학교 전자계산학과(이학박사)
- 1987년 ~ 현재 : 서일대학교 컴퓨터소프트웨어과 교수

<관심분야> : 능동데이터베이스, XML모델링, 소프트웨어공학