Journal of The Korea Society of Computer and Information
Vol. 17, No. 10, October 2012
www.ksci.re.kr
http://dx.doi.org/10.9708/jksci/2012.17.10.107

시맨틱 웹 데이터에서 접미사 배열 기반의 경로 질의 처리 기법

김 성 완*

Suffix Array Based Path Query Processing Scheme for Semantic Web Data

Sung Wan Kim*

요 약

서로 연결된 데이터들의 의미를 컴퓨터가 이해하여 자동으로 처리할 수 있는 시맨틱 기술의 보급이 확산되고 있다. 시맨틱 웹에서 데이터에 대한 처리는 데이터 자체에 대한 접근뿐만 아니라 데이터 상호간의 연관성 즉, 데이터 상호간의 의미에 대한 이해와 접근을 중요시 하고 있다. 시맨틱 웹의 데이터와 그 연관성을 표현하기 위해 W3C에서는 RDF를 표준 형식으로 제정하였으며 RDF로 표현된 데이터에 대한 질의 처리를 지원하기 위해 여러 RDF 질의어가 제안되었으나 시맨틱 연관성을 고려한 질의어 정의와 이에 관련한 질의 처리 기법은 계속적인 연구가 필요한 분야이다. 본 논문에서는 RDF 질의 처리를 위해 소개된 접미사 배열 기반의 인덱싱 기법을 기반으로 시맨틱 연관성의 대표적 유형인 ρ-path 질의를 처리하기 위한 방법을 제안한다. 제안된 질의 처리 방법의 성능 평가를 위해 다른 두 가지 형태의 처리 방법을 구현하여 실험적으로 비교하였다. 평균 질의 처리 시간 측정을 통해 제안 기법이 다른 두 가지 처리 방법에 비해 각각 약 1.8~2.5배와 3.8~11배의 우수한 처리 성능을 보인다.

▸ Keywords : 경로 질의 처리, 접미사 배열, RDF 데이터, 시맨틱 웹

Abstract

The applying of semantic technologies that aim to let computers understand and automatically process the meaning of the interlinked data on the Web is spreading. In Semantic Web, understanding and accessing the associations between data that is, the meaning between data as well as accessing to the data itself is important. W3C recommended RDF (Resource Description Framework) as a standard format to represent both Semantic Web data and their associations and also proposed several RDF query languages in order to support query processing for RDF data.

• 제1저자 및 교신저자 : 김성완
• 투고일 : 2012. 07. 18, 심사일 : 2012. 08. 22, 게재확정일 : 2012. 09. 07.
* 삼육대학교 컴퓨터학부(Division of Computer, Sahmyook University)

However further researches on the query language definition considering the semantic associations and query processing techniques are still required. In this paper, using the suffix array-based indexing scheme previously introduced for RDF query processing, we propose a query processing approach to handle ρ-path query which is the representative type of semantic associations. To evaluate the query processing performance of the proposed approach, we implemented two different types of query processing approaches and measured the average query processing times. The experiments show that the proposed approach achieved 1.8 to 2.5 and 3.8 to 11 times better performance respectively than others two.

▸ Keywords : Path Query Processing, Suffix Array, RDF Data, Semantic Web

I. 서 론

네이버 등 민간 포털 사이트를 중심으로 도입됐던 시맨틱 기반의 검색 서비스가 올 7월부터 최근 행정안전부의 관광 및 재해 분야 관련 공공 부분 사업을 대상으로 추진[1]되는 등 서로 연결된 데이터들의 의미를 컴퓨터가 이해하여 자동으로 처리할 수 있는 시맨틱 기술의 보급이 확산되고 있다.

시맨틱 웹에서 데이터에 대한 처리는 단순히 데이터 자체에 대한 접근뿐만 아니라 데이터 상호간의 관련성 혹은 연관성 즉, 데이터 상호간의 의미에 대한 이해와 접근을 더욱 중요시 하며, 이러한 맥락에서 시맨틱 웹을 데이터의 웹(Web of Data)로 표현하고 있다. 또한, 이렇게 웹 상의 상호 관련된 데이터 집합의 모임을 링크드 데이터(linked data)라 부르기도 한다[2]. 이처럼 링크드 데이터의 생성과 관리를 위해서는 데이터와 그 연관성을 표현하기 위한 표준적인 형식이 필요하다. W3C에서는 RDF(Resource Description Framework)를 이를 위한 표준 형식[3]으로 제정하였으며 시맨틱 웹을 실현하기 위한 기반 기술로 널리 사용되고 있다.

시맨틱 연관성은 엔터티 사이의 복잡한 관련성을 의미하며 이를 통해 사전에 예상하지 못한 유용한 정보를 얻어 낼 수 있다. 엔터티 간의 연관성을 발견(discovery)하는 기술은 예를 들어 '탑승객 A가 요주의 목록에 포함된 기관과 연관되어 있는가?' 와 같이 항공 보안, 테러 방지 및 국가 보안 등의 업무와 시맨틱 기반의 검색 서비스 응용에서 매우 중요한 질의 요소로 활용된다[4].

RDF 데이터에 대한 질의 처리를 위해 W3C의 SPAQRL 등과 같은 RDF 질의어가 제정되었으나 질의어 차원에서 엔터티 사이의 복잡한 관련성의 발견을 위한 충분한 기능을 제공하지 못하고 있다. 또한 RDF 데이터에 대한 관리, 검색과 전형적인 질의 처리 기법에 대한 연구는 활발히 진행되어 왔으나[5] 시맨틱 연관성을 고려한 질의어 차원의 연구와 표준적인 기능의 제정 그리고 이에 대한 질의 처리 기법의 연구는 계속적으로 요구되고 있다.

한편, [6]에서는 경로에 기반한 RDF 질의 처리를 위해 접미사 배열을 응용한 기본적인 인덱싱 기법을 최초로 제안하였으며 [7]에서는 이에 대한 성능 개선을 위한 인덱싱 및 질의 처리 기법이 소개되었다. 본 논문에서는 RDF 질의 처리를 위해 제시된 접미사 배열을 이용한 인덱싱 기법[6][7]을 기반으로 시맨틱 연관성 검색의 대표적 유형인 ρ-path를 처리하기 위한 방법을 제안하고 이에 대한 성능 평가를 진행한다.

본 논문은 구성은 다음과 같다. 2장에서는 시맨틱 연관성에 대한 정의 및 개념과 RDF 질의 처리를 위한 접미사 배열을 이용한 인덱싱 및 질의 처리 기법에 대한 관련 연구를 서술한다. 3장에서는 ρ-path 질의 처리 기법을 제안하고 4장에서는 제안 기법에 대한 성능평가에 대해 서술 한다. 5장에서는 결론을 내린다.

II. 기초 연구

1. RDF와 시맨틱 연관성 검색

RDF에서는 링크드 데이터들의 연관성을 기술하기 위해 〈서브젝트, 프로퍼티, 오브젝트〉 형태로 구성된 트리플을 최소 표현 단위로 사용한다. 여기서, 서브젝트와 오브젝트는 웹 상에서 식별 가능한 데이터를 의미하며 RDF에서는 이를 리소스(resource)라 한다. 프로퍼티(property)는 두 리소스 간의 연관성을 나타낸다. 또한, RDF로 기술된 데이터는 〈그림 1〉과 같이 리소스를 노드로 하고 프로퍼티를 간선으로 하는 방향성 그래프 형태로 표현할 수 있다.

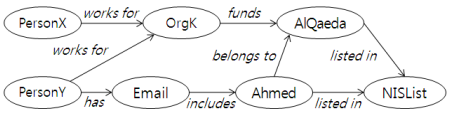

그림 1. 예제 RDF 그래프
Fig. 1. Example RDF Graph

[4][8][9]에서는 리소스 사이의 발생할 수 있는 관련성을 시맨틱 연관성으로 정의하였으며, 시맨틱 연관성의 발견은 두 리소스를 연결하는 길이를 알 수 없거나 특정한 의미를 지니는 경로를 검색하는 것이라고 하였다. [4]에서는 RDF 그래프 상에 경로 'e_1, p_1, e_2, p_2, … p_{n-1}, e_n'이 존재한다면 두 개체 e_1과 e_n 사이에는 시맨틱 연관성이 있다고 정의하고 (여기서 e_i와 p_i는 개체와 프로퍼티를 각각 의미함), 개체와 프로퍼티가 교대로 구성된 이러한 시퀀스를 시맨틱 경로(semantic path)로 정의하였다. 이러한 시맨틱 경로는 리소스 사이에 존재하는 명시적 또는 암시적인 의미적 관련성을 나타낸다. 이는 두 리소스 사이를 연결하는 경로가 존재할 경우 서로 의미적으로 관련성이 있음을 가정하는 것이다. 본 논문에서는 이처럼 임의의 두 리소스 사이를 연결하는 방향성이 있는 시맨틱 경로를 'ρ-path 시맨틱 연관성'이라고 정의한다.

한편, RDF로 표현된 데이터에 대한 질의 처리를 지원하기 위해 RQL, SPARQL 등 여러 RDF 질의어가 제안되었으나 리소스 사이의 복잡한 관계성들의 발견을 위한 충분한 지원을 하지 못하고 있다[4][10]. SPARQLeR[10]은 프로퍼티에 대한 정규식 표현을 이용하여 시맨틱 연관성 검색을 지원하도록 SPARQL을 확장한 RDF 언어이다.

예를 들어 〈그림 1〉에서 요주의 인물인 'PersonY'가 국가정보원(NIS)에서 관리하는 위험 조직 목록 'NISList'에 포함되는지 검색하는 경우를 고려해 보자. 즉, 리소스 〈PersonY〉와 〈NISList〉 사이에 존재하는 ρ-path 시맨틱 연관성을 검색하기 위한 질의를 SPARQLeR 형식으로 표현하면 다음과 같다.

SELECT %path
WHERE {〈PersonY〉 %path 〈NIS List〉}

위 질의에서 %path는 경로 변수를 의미하며 〈PersonY〉와 〈NISList〉 사이에 존재하는 임의의 경로에 매칭된다. 위 질의 처리를 위해 〈그림 1〉로부터 〈PersonY〉와 〈NISList〉 사이에 존재하는 임의의 경로를 추출하면 다음과 같은 3개의 시맨틱 경로를 결과로 반환하게 된다.

PersonY→works for→OrgK→funds→AlQaeda→listed in→NISList

PersonY→has→Email→includes→Ahmed→belongs to→AlQaeda→listed in→NISList

PersonY→has→Email→includes→Ahmed→listed_in→NISList

2. 접미사 배열 기반의 RDF 데이타 처리

RDF 데이터에 대한 효율적인 경로 질의 처리를 위해 여러 가지 인덱싱 및 질의 처리 기법이 연구되었다. [6]에서는 경로식으로 표현될 수 있는 RDF 데이터에 대한 효과적인 질의 처리를 위해 접미사 배열을 응용한 인덱싱 기법을 처음으로 제안하였다. 이 기법에서는 RDF 데이터를 사이클이 없는 방향성 그래프(DAG : Directed Acyclic Graph)로 간주하고 경로 패턴들을 추출하였다. 추출된 경로 패턴들은 변형된 접미사 배열을 이용하여 인덱스를 구축하는데 활용된다. 그러나 이 연구에서는 단방향 단순 질의만을 지원하도록 개발되었으며 시맨틱 연관성을 고려한 질의 처리에 대해서는 언급하지 않았다.

[7]에서는 접미사 배열 기반의 인덱스를 활용해 질의 처리 시 수행되는 이진 탐색의 수행 범위 축소 방안 등을 제안하여 경로 질의 처리에 대한 성능 개선을 시도하였으나 시맨틱 연관성 질의는 역시 고려하지 않았다.

[11]에서는 RDF 데이터 및 RDFS 기반의 온톨로지 정보를 저장하고 추론에 기반한 질의 처리를 지원하는 인덱싱 기법을 제안하였다. [12]에서는 RDF 그래프 상의 두 노드간의 최단 경로를 구하기 위해 Dijkstra의 알고리즘을 기반으로

idx / pid	1	2	3	4	5	6	7	8	9
1	PersonX	works for	OrgK	funds	AlQaeda	listed in	NISList		
2	PersonY	works for	OrgK	funds	AlQaeda	listed in	NISList		
3	PersonY	has	Email	includes	Ahmed	belongs to	Al-Qaeda	listed in	NISList
4	PersonY	has	Email	includes	Ahmed	listed in	NISList		

그림 2. 경로 정보 테이블 (pTab)
Fig. 2. Path Information Table (pTab)

한 내용을 소개하였다. [13]에서는 [7]에서 제안된 개선된 인덱싱 기법을 사용하여 시맨틱 연관성 유형 중 하나인 ρ-intersection 연산에 대한 처리 방안을 논의하였다. ρ-intersection 유형이란 두 개의 시맨틱 경로가 특정 리소스 상에 교차 되는 경우를 의미한다.

본 논문의 선행 연구인 [14]에서는 [7]에서 제안된 개선된 접미사 배열 기반의 인덱싱 기법을 기반으로 ρ-path 시맨틱 연관성 유형에 대한 처리 방안을 논의하고 이에 대한 처리 알고리즘을 제시하였으나 이에 대한 구현 및 성능 평가는 언급하지 않고 있다. [15]에서도 [7]에서 다루지 못한 ρ-path 시맨틱 연관성 유형에 대한 처리 방안을 언급하고 개략적 수준에서 그 처리 방법을 논의하였으나 구체적인 알고리즘의 제시와 구현 및 평가는 수행하지 않았다.

[7]의 연구에서 제안한 접미사 배열 기반의 인덱스 구성 단계를 간략하게 살펴보면 다음과 같다. 첫째, RDF 데이터로부터 모든 경로들을 추출한 후 경로 정보 테이블(pTab)을 생성한다. 이를 위해 추출된 각 경로에 대해서 접미사들을 생성하며 각 접미사에는 '접미사 레이블'을 할당한다. 접미사 레이블은 경로 식별자(pid)와 인덱스 포인트(idx) 쌍으로 구성된다. <그림 2>는 <그림 1>의 예제 RDF 그래프로 부터 구해진 경로 정보 테이블(pTab)을 나타낸 것이다.

둘째, 추출된 접미사들을 사전 순으로 정렬하고 또한 각 접미사들에 대한 LCP(Longest Common Prefix)값을 계산한다. 접미사에 대한 LCP 값은 바로 직전 접미사 패턴과 비교할 경우 최장 공통 접두사의 길이 값을 의미한다. 예를 들어 접미사 'abrada'가 바로 직전의 접미사로 'abrasic'를 갖는 경우, 최장 공통 접두사는 'abra'가 되므로 접미사 'abrada'에 대한 LCP 값은 4가 된다. LCP 값들은 접미사 배열 기반의 인덱스를 이용하여 질의 처리 시 반복적인 패턴 매칭 작업의 부담을 축소하기 위해 사용된다.

셋째, 정렬된 접미사들에 할당된 접미사 레이블 값들과 계산된 LCP 값들을 이용하여 인덱스를 생성한다. <그림 3>은 이와 같은 과정을 거쳐 생성된 인덱스의 전체적인 모습을 개념적으로 나타낸 것이다.

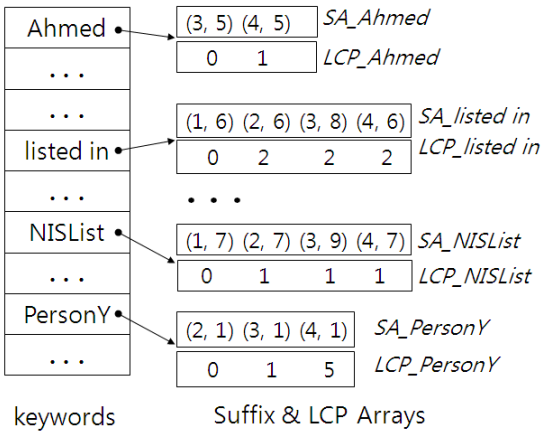

그림 3. 인덱스 구조
Fig. 3. Index Architecture Overview

<그림 3>의 오른편에 있는 각 접미사 배열 SA_i는 'i'로 시작하는 접미사 패턴들에 대한 접미사 레이블 값들만을 포함한다. 예를 들어 SA_NISList은 'NISList'로 시작하는 접미사 패턴들에 할당된 접미사 레이블 [(1, 7)(2, 7)(3, 9)(4, 7)]만을 포함한다. LCP_k 배열은 'k'를 첫 번째 요소로 갖는 접미사 패턴들의 LCP 값을 유지한다. <그림 3> 왼편의 키워드 그룹은 추출된 접미사 패턴들의 첫 번째 요소들만으로 구성된다. 각각의 키워드는 해당 접미사 배열 SA_i와 LCP 배열 LCP_i와 연결되어 있어 특정 키워드로 시작되는 접미사 패턴들만을 빠르게 접근할 수 있다.

본 논문에서는 본 연구의 선행 연구로 [14]에서 제안된 접미사 배열 기반의 ρ-path 시맨틱 연관성 검색을 위한 경로 질의 처리 방법을 확장 제안하고 이에 대한 실험적 성능평가를 다루고자 한다.

III. 본 론

본 논문에서 RDF 데이터는 DAG 형태로 가정하며 경로(path)는 다음과 같이 리소스 서술 내용에 대한 RDF 그래프 상의 경로 구성 요소들 즉, 노드 레이블과 간선 레이블이 교대로 구성된 것으로 정의한다.

path ::= (rscLbl '.' propLbl '.')*(rscLbl | literalVal
| (rscLbl '.' propLbl))
rscLbl ::= URI Reference ([3] 참조)
propLbl ::= propName ([3] 참조)
literalVal ::= Constant Values ([3] 참조)
propName ::= URI Reference ([3] 참조)

1. 단순 경로 질의 처리

ρ-path 시맨틱 연관성 검색은 [7]에서 언급된 단순 경로 질의 처리를 활용하여 처리된다. 여기서 단순 경로는 고정된 길이를 갖으며 그 구성 요소들이 중간에 빠짐없이 구성된 것 경로를 의미한다. 단순 경로 질의란 질의로 주어진 방향성 경로로 부터 도달 가능한 (reachable) 리소스를 검색하는 질의이다. 가장 단순 한 단순 경로 패턴은 위에서 언급한 경로의 정의에 따라 리소스 하나만으로도 구성될 수 있다. 예를 들어 〈그림 1〉의 RDF 그래프 상에서 단순 경로 'PersonX.works for.OrgK.funds'으로부터 도달 가능한 리소스를 검색하면 AlQaeda가 된다.

〈그림 4〉는 이러한 단순 경로 질의 처리를 위한 알고리즘 중 주어진 경로 패턴과 일치하는 접미사들에 할당된 모든 접미사 레이블들을 추출하는 함수를 나타낸 것이다. 이 함수의 첫 번째 단계는 접미사 배열 SA_i에 대한 이진 탐색과 경로 정보 테이블(pTab)을 이용하여 사용자 질의에 명세된 경로 패턴과 최초로 일치되는 배열 요소를 찾는다. 두 번째 단계는 해당 접미사 배열 SA_i상에서 찾아진 배열 요소를 기준으로 좌측 및 우측에 인접한 나머지 접미사 패턴들에 대한 접미사 레이블을 찾는다. 이 때, 직접적인 문자열 패턴 매칭을 이용한 접미사 패턴 검색 따른 부담을 축소하기 위해 LCP 배열을 이용한 산술 연산을 통해 작업이 수행된다. 예를 들어 단순 경로 질의 패턴 'PersonY.has.Email'을 〈그림 4〉의 알고리즘에 따라 처리할 경우 접미사 레이블 집합 {(3, 1), (4, 1)}이 반환된다.

만일 단순 경로 패턴이 리소스 하나만으로 경우는 해당 리소스와 연관된 접미사 배열 SA_i에 대한 이진 탐색 없이 SA_i의 내용을 순차적으로 접근하여 접미사 레이블 집합을 추출할 수 있다.

2. ρ-path 시맨틱 연관성 검색

ρ-path 시맨틱 연관성 검색을 위한 질의 형식을 SPARQLeR 표현을 사용하여 나타내면 다음과 같다.

```
Function GetSuffixLabel(usrQueryPattern)
  // 입력 : 경로 질의 패턴 usrQueryPattern (d : usrQueryPattern의 경로 길이)
  // 출력 : 입력 질의에 일치되는 접미사 레이블 (pid, idx)의 집합 resultSet

1:  usrQueryPattern에 최초로 일치하는 배열 SA_i 상의 위치 값 p를 검색
2:  SA_i[p]의 값인 접미사 레이블을 결과 집합 resultSet에 추가

  // 위치 값 p를 기준으로 배열 SA_i 상의 좌측 및 우측의 일치되는 접미사 레이블 추가 검색
3:  tp ← p
4:  While (d <= LCP_i[tp]) // 배열 SA_i 상의 좌측 검색
5:    tp ← tp - 1  // 위치 값 조정
6:    SA_i[tp]의 값인 접미사 레이블 (pid, idx)를 resultSet에 추가
    End While

7:  tp ← p
8:  While (d <= LCP_i[tp+1])  // 배열 SA_i 상의 우측 검색
9:    tp ← tp + 1  // 위치 값 조정
10:   SA_i[tp]의 값인 접미사 레이블 (pid, idx)를 resultSet에 추가
    End While
11:  접미사 레이블 (pid, idx)의 집합 resultSet를 반환
  End Function
```

그림 4. 접미사 레이블 집합 반환을 위한 알고리즘
Fig. 4. Algorithm for Returning a Set of Suffix Labels

SELECT %path
WHERE {〈단순 경로〉 %path 〈단순 경로〉}

여기서 WHERE 절의 〈단순 경로〉는 본 장의 서두에서 정의한 경로(path)를 의미한다. %로 시작되는 변수는 경로 변수이며 임의의 경로 패턴과 매칭 된다. 예를 들어 '〈단순 경로1〉 %path 〈단순 경로2〉'는 〈단순 경로1〉의 결과와 〈단순 경로2〉의 결과 사이에 존재하는 모든 경로들을 반환한다. 이 형식을 이용하여 ρ-path 시맨틱 연관성 검색 질의 예제를 표현하면 다음과 같다.

질의 1) 두 리소스 〈PersonY〉와 〈NISList〉 사이에 존재하는 ρ-path 시맨틱 연관성 반환

SELECT %path
WHERE {〈PersonY〉 %path 〈NISList〉}

질의 2) 단순 경로 패턴 〈PersonX.works for.OrgK〉와 리소스 〈NISList〉 사이에 존재하는 ρ-path 시맨틱 연관성 반환

SELECT %path
WHERE {〈PersonX.works for.OrgK〉 %path 〈NISList〉}

ρ-path 시맨틱 연관성 검색을 위한 질의는 크게 2단계로 처리되며 〈그림 5〉는 이에 대한 알고리즘을 나타낸 것이다. 첫 번째 단계는 주어진 사용자 경로 질의의 WHERE 절에 포함된 검색 조건을 두 개의 단순 경로 질의로 분할한다. 여기서 분할된 각 서브 질의는 〈그림 4〉에서 제시된 알고리즘을 이용하여 각각 처리한다. 예제 질의 2의 경우 서브 질의 'PersonX.works for.OrgK'와 'NISList'로 분할된 후 GetSuffixLabel 함수의 입력 인자 값으로 각각 전달되어 처리되며 접미사 레이블 집합 {(2, 1)(3, 1)(4, 1)}와 {(1, 7)(2, 7)(3, 9)(4, 7)}가 해당 서브 질의의 처리 결과로 각각 반환 된다.

두 번째 단계에서는 첫 번째 단계에서 구해진 각 접미사 레이블 집합에 대해 pid 값을 기준으로 정렬-합병 알고리즘을 적용하여 최종 결과를 구하게 된다. 만일 WHERE 절에 포함된 〈단순 경로〉가 리소스 하나만으로 주어진 경우는 〈그림 3〉의 해당 리소스와 연관된 접미사 배열 SA_i의 접미사 레이블들은 이미 pid 값을 기준으로 오름차순 정렬되어 있으므로 해당 리소스에 대해 구해진 접미사 레이블 집합에 대해서는 추가적인 정렬 연산을 필요로 하지 않는다.

첫 번째 단계에서 구해진 접미사 레이블 집합을 각각 P와 Q라 할 경우 P에 속한 임의의 접미사 레이블 (p1, i1)과 Q에 속한 임의의 접미사 레이블 (q1, i2)에 대한 합병 조건은 서로 동일한 pid를 가지며 또한 첫 번째 접미사 레이블의 idx 값이 두 번째 접미사 레이블의 idx 값보다 작은 경우이다

```
Function pPathQueryProcessing(UserQueryPattern)
    // 입력 : 사용자 경로 질의 패턴 UserQueryPattern
    // 출력 : 경로 패턴들의 집합 finalSet

1:  사용자 경로 질의 UserQueryPattern을 SubQuery1와 SubQuery2로 분할
2:  P ← GetSuffixLabel(SubQuery1)  // GetSuffixLabel 함수 호출 및 접미사 레이블 집합 P 구함
3:  Q ← GetSuffixLabel(SubQuery2)  // GetSuffixLabel 함수 호출 및 접미사 레이블 집합 Q 구함

4:   집합 P와 집합 Q의 접미사 레이블들을 pid 값 기준으로 각각 오름차순 정렬 수행

    // pid 값 기준으로 P와 Q에 대한 정렬-합병 알고리즘 적용
5:  Foreach  동일한 pid를 가지는 p ∈ P와 q ∈ Q  Do
6:      If (p의 idx 〈 q의 idx ) Then
7:          pTAB[pid][p의 idx]과 pTAB[pid][q의 idx]사이의 내용을 finalSet에 추가
        End if
    End For
End Functon
```

그림 5. ρ-path 시맨틱 연관성 검색을 위한 알고리즘
Fig. 5. Algorithm for ρ-path Semantic Association Discovery

(즉, p1 ≡ q1 ∧ i1 〈 i2). 합병 조건을 만족하는 경우 해당 접미사 레이블 쌍 사이에 있는 경로 패턴을 경로 정보 테이블(pTab)로부터 추출하여 최종 결과로 포함한다.

예제 질의 1에 대한 처리에서 P = {(2, 1)(3, 1)(4, 1)}, Q = {(1, 7)(2, 7)(3, 9)(4, 7)}이 추출 되며, 집합 P의 (2, 1)과 집합 Q의 (2, 7) 쌍이 합병 조건을 만족하므로 pTab[2][1]와 pTab[2][7] 사이에 존재하는 경로 패턴 'works for→OrgK→funds→AlQaeda→listed in'이 최종 결과에 포함된다. 집합 P의 (3, 1)와 집합 Q의 (3, 9) 쌍 그리고 집합 P의 (4, 1)와 집합 Q의 (4, 7) 쌍 역시 합병 조건을 만족하므로 경로 패턴 'has→Email→includes→Ahmed →belongs to→AlQaeda→listed in'과 'has→Email→ includes→Ahmed→listed in'을 추가로 구하게 되어 총 3개의 경로 패턴을 최종 결과로 얻을 수 있다. 집합 Q의 (1, 7)는 합병 조건을 만족하는 집합 P의 쌍이 없으므로 결과 처리에서 제외된다.

IV. 구현 및 성능평가

본 논문에서 제안한 ρ-path 시맨틱 연관성 질의 처리 방법에 대한 평가를 위해 3가지 방법을 구현하여 실험 및 성능 평가를 수행하였다. 첫째, [6]에서 제안한 인덱싱 기법과 본 논문의 3장에서 제안한 ρ-path 시맨틱 연관성 검색 알고리즘을 적용한 방법을 구현(비교 1 방법)하였다. 둘째, [7]에서 소개된 개선된 인덱싱 기법을 사용하되 경로 정보 테이블(pTab)에 대한 순차적인 경로 탐색을 적용한 질의 처리 방법을 구현(비교 2 방법) 하였으며, 질의 처리의 대략적인 단계는 〈그림 6〉과 같다. 마지막으로 3장에서 제안한 바와 같이 개선된 인덱싱 기법과 ρ-path 시맨틱 연관성 검색 알고리즘을 적용한 방법을 구현(제안 방법)하였다.

시스템 구현 및 실험을 위한 환경은 다음과 같다. 인덱싱 및 질의 처리 시스템의 구현은 Visual C++ 6.0과 MySQL 5.0을 사용하였다. 실험용 컴퓨터는 Intel Core2 Duo 2.20GHz CPU, 1GB 메모리와 Window XP Professional이 설치되어 있는 기계를 사용하였다.

실험용 데이터는 FOAF project[16]에서 제공하는 FOAF 온톨로지 기반의 RDF 데이터를 DAG 형태로 변형하여 사용하였으며 크기가 다른 두 가지 데이터를 사용하였다. 다음 〈표 1〉은 두 개의 실험용 데이터로 부터 추출된 경로와 접미사의 개수를 나타낸 것이다. 참고로, 실험용 데이터에서 추출된 최장 경로의 길이는 18이다.

```
1. 사용자 질의를 SubQuery1과 SubQuery2로 분할
   // 설명을 단순화 하기 위해 SubQuery2는 리소스
      하나로 구성된 경로 패턴으로 가정
2. P ← GetSuffixLabel(SubQuery1)
3. Foreach (pid, idx) in P
     i ← idx+1
     tPath ← null
     While (pTab[pid][i] ≠ null) Do
       If (pTab[pid][i] ≡ SubQuery2) Then
          tPath를 finalSet에 추가
          Exit While
       Else
          tPath ← tPath와 pTab[pid][i]를 연결
       End If
       i ← i + 1
     End While
   End For
```

그림 6. 순차적 경로 접근
Fig. 6. Sequential Path Access

표 1. 데이터 셋
Table 1. Data Set

	데이터 1	데이터 2
데이터 크기(KB)	2,000	10,000
경로 수	1,314	6,570
접미사 수	14,874	74,370

실험용 질의는 〈표 2〉와 같은 5가지 유형의 질의를 사용하였다. 여기서, 범위란 RDF 그래프 상에서 질의가 처리되는 위치를 의미한다. 예를 들어 '전체'란 RDF 그래프 상의 진입 차수가 0인 노드(예를 들어 〈그림 1〉에서 'PersonX')로 부터 진출 차수가 0인 노드(예를 들어 〈그림 1〉에서 'NISList')까지 이르는 경로 전체를 질의 처리 범위로 포함하는 것을 의미한다. '전반부'란 상위 노드들로 구성된 영역에 대한 경로를 질의 처리 범위로 갖는 경우를 의미한다. ρ-path 길이는 질의 처리 후 반환되는 ρ-path 시맨틱 연관성 경로의 길이를 의미한다. 복잡성은 질의 결과로 반환 되는 ρ-path 시맨틱 연관성 경로를 구성하는 각 리소스와 다른 리소스간의 연관성 정도를 의미하는 것으로 복잡성이 높을수록 즉, 진입 및 진출

차수가 높은 리소스를 질의 처리 범위에 포함하게 되어 실행 시간이 더 소요되게 된다.

표 2. 테스트용 질의
Table 2. Test Queries

유형	질의 특징			
	범위	p-path 길이	복잡성	경로 개수
Q1	전반부	3	낮음	1
Q2	전체	8~18	보통	3
Q3	중간	4~8	높음	2
Q4	전반부	6	높음	1
Q5	후반부	2~8	보통	2

실험 방법은 각 실험 질의 유형들에 대한 질의 처리 시간을 10회씩 측정하여 평균 실행 시간을 계산하였다. 〈표 3〉은 질의 에 포함된 경로 패턴과 최초로 일치되는 접미사 경로 패턴을 찾기 위해 접미사 배열 기반의 인덱스 상에서 수행되는 이진 탐색 횟수를 나타낸 것이다. S1과 S2는 주어진 사용자 질의를 분할하여 만들어진 두 개의 서브 질의를 각각 나타낸다.

〈그림 3〉에서 본 것과 같이 제안 방법에서는 여러 개의 접미사 배열 인덱스를 유지하므로 데이터 셋의 크기가 변해도 이진 탐색 수행 횟수 변화가 없다. 그러나, 비교 1 방법에서는 단일 인덱스를 사용하므로 데이터 셋의 크기에 커짐에 따라 이진 탐색 수행 횟수가 많아지게 됨을 알 수 있다. 또한, [7]에서 소개한 바와 같이 제안 방법의 이진 탐색 횟수가 비교 1 방법에 비해 축소되었음을 알 수 있으며 이는 질의 처리 수행 시간 단축에 영향을 주게 된다. 비교 2 방법의 경우 두 번째 서브 질의 S2는 경로 정보 테이블(pTab)에 대한 순차 경로 탐색을 통해 처리되므로 접미사 배열 인덱스에 대한 이진 탐색을 수행하지 않는다.

표 3. 접미사 배열 인덱스에 대한 이진 탐색 횟수
Table 3. The Number of Binary Search over SA index

유형	데이터 셋 1						데이터 셋 2					
	비교1		비교2		제안		비교1		비교2		제안	
	S1	S2	S1	S2	S1	S2	S1	S2	S1	S2	S1	S2
Q1	11	9	4	-	4	1	13	15	4	-	4	1
Q2	7	8	1	-	1	1	8	12	1	-	1	1
Q3	6	6	1	-	1	1	8	10	1	-	1	1
Q4	4	6	1	-	1	1	6	8	1	-	1	1
Q5	6	9	1	-	1	1	9	1	1	-	1	1

〈그림 7〉은 실험 데이터 1에 대한 평균 실행 시간을 보여주고 있다. 5가지 실험 질의 유형 모두에 대해 제안 기법이 비교 1 방법에 대해 1.8~2.5배 정도의 우수한 질의 처리 성능을 보였으며, 비교 2 방법에 대해서는 3.8~11.2 배 정도의 우수한 성능을 보였다. 이러한 이유는 〈표 3〉에서 본 것과 같이 제안 방법이 이진탐색에 소요되는 횟수가 비교 1 방법에 비해 축소되었기 때문이다. 제안 방법과 비교 1 방법의 경우 질의 복잡성에 비례하여 질의 처리 시간이 소요되나 비교 2 방법의 경우 경로에 대한 순차적인 방법으로 처리되기 때문에 질의 복잡성 외에도 최종 결과에 포함되는 경로의 길이에 비례하여 질의 처리 시간이 소요되었다.

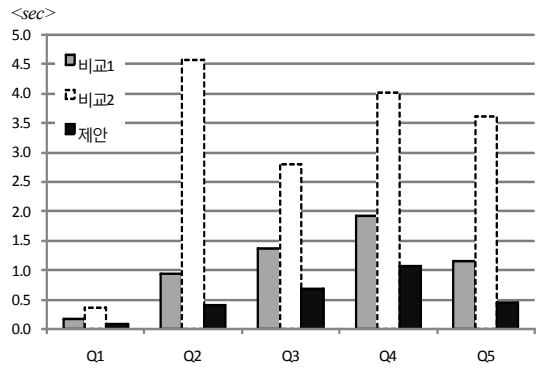

그림 7. 평균 실행 시간 (데이타 1 사용)
Fig. 7. Average Execution Time (using Data Set 1)

〈그림 8〉은 실험 데이터 2에 대한 평균 실행 시간을 보여주고 있다. 〈그림 7〉에서와 마찬 가지로 5가지 실험 질의 유형 모두에 대해 제안 기법이 비교 1 방법에 대해 1.8~2.6배

정도의 우수한 질의 처리 성능을 보였으며, 비교 2 방법에 대해서는 3.8~11.3배 정도의 우수한 성능을 보였다.

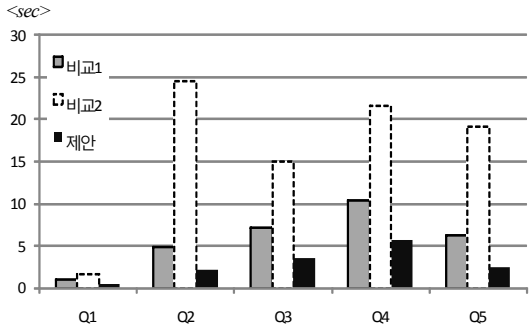

그림 8. 평균 실행 시간 (데이타 2 사용)
Fig. 8. Average Execution Time (using Data Set 2)

V. 결 론

시맨틱 웹 검색 기술의 적용이 확대 됨에 따라 리소스간의 시맨틱 연관성 검색의 효율적인 지원이 요구된다. 최근 제안된 RDF 질의어에서는 질의어 차원에서 시맨틱 연관성 검색을 위한 구문을 포함하고 있다.

본 논문에서는 시맨틱 연관성 검색의 유형 중 가장 대표적인 ρ-path 검색을 위한 질의 처리 기법을 소개하였다. 특히, 접미사 배열을 활용한 RDF 데이터 인덱싱 기법을 활용하여 전형적인 RDF 데이터 질의 유형뿐만 아니라 ρ-path 시맨틱 연관성 검색 질의도 지원할 수 있는 방안을 제안하였다. 또한, 실험적 성능 평가를 통해 제안 기법의 우수성을 보였다. 제안 처리 방안을 적용할 경우 약 1.8~2.5배의 성능 향상을 보였다. 또한, 순차적인 방법을 적용한 방법과 비교하여 3.8~11배의 성능 향상을 보였다.

한편, 본 논문에서는 방향성이 있는 DAG 형태의 RDF 데이터를 가정하였다. 방향성 경로에서는 두 개체 사이의 시맨틱 연관성을 명시적으로 나타내는 것에 반해 두 개체사이의 비방향성 경로 역시 중요한 시맨틱 연관성을 암시적으로 나타낸다[10]. 향후 이러한 비방향성 시맨틱 연관성 탐색을 위한 방안에 대한 추가 연구를 진행하고자 한다.

참고문헌

[1] K. Kim, IT Daily Newspaper, 7 May 2012 *http://www.itdaily.kr/news/articleView.html?idxno = 30352*

[2] W3C, Semantic Web, 2012 *http://www.w3.org/standards/semanticweb/*

[3] W3C, RDF Primer (W3C Recommendation), Feb. 2004, *http://www.w3.org/TR/rdf-primer*

[4] A. Sheth et al., "Semantic Association Identification and Knowledge Discovery for National Security Applications," Journal of Database Management, Vol 16, pp.33-53, 2005.

[5] A. Hertel, J. Broekstra, and H. Stuckenschmidt, "RDF Storage and Retrieval Systems," Handbook on Ontologies, Springer, pp.489-508. 2009

[6] A. Matono, T. Amagasa, M. Yishikawa, and S. Uemura, "An Indexing Scheme for RDF and RDF Schema based on Suffix Arrays," First International Workshop on Semantic Web and Databases (SWDB), pp.151-168, Sept. 2003.

[7] S. Kim, "Improved Processing of Path Query on RDF Data Using Suffix Array," Journal of Convergence Information Technology, Volume 4, Number 3, pp. 45-52 Sept. 2009.

[8] Kemafor Anyanwu and Amit Sheth, "ρ-Queries: Enabling Querying for Semantic Associations on the Semantic Web," Proc. of International Conference on World Wide Web, pp.690-699, 2003

[9] B. Aleman-Meza, C. Halaschek-Wiener, I. Arpinar, C. Ramakrishnan, and A. Sheth, "Ranking Complex Relationships on the Semantic Web," IEEE Internet Computing, Vol. 9, no. 3, pp. 37-44, 2005.

[10] Krys Kochut and Maciej Janik, "SPARQLeR: Extended Sparql for Semantic Association Discovery," LNCS, Vol. 4519, Proc. of the 4th European Conference on The Semantic Web, pp. 145-159, 2007.

[11] Y. Kim, and J. Kim, "The Scheme for

Path-based Query Processing on the Semantic Data," Journal of the Korea Society of Computer and Information, Vol.14, No.10, pp.31-41, October 2009.

〔12〕 A. Gubichev and T. Neumann, "Path Query Processing in Very Large RDF Graphs," Proc. of the 14th Int'l Workshop on the Web and Database(WebDB), June 2011

〔13〕 S. Kim and Y. Kim, "Processing of ρ-intersect Operation on RDF Data Using Suffix Array," Journal of the Korea Society of Computer and Information, Vol.16, No.7, pp.95-103, July 2011.

〔14〕 S. Kim, "Query Processing for Discovering ρ-path Semantic Association Using Suffix Array", Proc. of the 36th KIISE Fall Semiannual Conference, pp. 69-73, 2009

〔15〕 Hikmat Ullah Khan and Tahir Afzal Malik, "Finding Resources from Middle of RDF Graph and at Sub-Query Level in Suffix Array Based RDF Indexing Using RDQL Queries," International Journal of Computer Theory and Engineering, Vol. 4, No. 3, pp. 369-372, June 2012.

〔16〕 The Friend of a Friend (FOAF) project, *http://www.foaf-project.org*

저 자 소 개

김 성 완

1998 : 홍익대학교 전자계산학과 이학석사

2003 : 홍익대학교 전자계산학과 이학박사

1999 ~ 2005 : 삼육의명대학 컴퓨터정보과 조교수

2006 ~ 현 재 : 삼육대학교 컴퓨터학부 부교수

관심분야 : XML 및 웹 데이터베이스, 정보처리, 시맨틱 웹

Email : swkim@syu.ac.kr