한국컴퓨터정보학회 동계학술대회 논문집 제19권 제1호 (2011. 1)

시맨틱 연관성 검색을 위한 ρ-intersect 연산의 처리

김성완°
°삼육대학교 컴퓨터학부
e-mail: swkim@syu.ac.kr

Processing of ρ-intersect Operation for Semantic Association Discovery

Sung Wan Kim°
°Division of Computer, Sahmyook University

● 요 약 ●

시맨틱 웹상에서 메타 데이터를 표현하는 RDF 데이터에 대한 질의 처리를 위해 여러 가지 RDF 질의어가 제안되었으나 리소스 간의 복잡한 관계성들의 발견(discovery)을 위한 충분한 지원을 하지 못하고 있다. 본 논문에서는 시맨틱 연관성 검색 유형의 하나인 ρ-intersect 연산의 처리 방법을 소개한다. 이를 위해 접미사 배열을 이용한 인덱싱과 ρ-intersect 연산의 특징을 고려한 최적화 방법을 활용한다. 제안된 처리 기법을 통해 전형적인 RDF 질의 유형뿐만 아니라 시맨틱 연관성 질의 유형도 지원할 수 있도록 한다.

키워드: 시맨틱 연관성(semantic association), 접미사 배열(suffix array), 질의 처리(query processing)

I. 서 론

본 논문에서는 RDF 질의 처리를 위해 제안된 접미사 배열을 이용한 인덱싱 기법[6][8]을 기반으로 시맨틱 연관성 검색 유형의 하나인 ρ-intersect 연산의 처리 기법을 소개한다. 이를 통해 전형적인 RDF 질의 유형뿐만 아니라 시맨틱 연관성 질의 유형도 지원할 수 있도록 한다.

본 논문은 다음과 같이 구성된다. 2장은 관련 연구로서 시맨틱 연관성에 대한 개념 그리고 접미사 배열을 이용한 RDF 데이터의 인덱싱 및 질의 처리 기법에 대해 서술한다. 3장에서는 시맨틱 연관성의 한 가지 유형인 ρ-intersect 연산의 처리를 위한 인덱싱 및 질의 처리 방안을 제안하고 4장에서 결론 및 향후 연구에 대해 언급한다.

II. 관련 연구

1. 시맨틱 연관성

시맨틱 웹상에서 메타 데이터를 표현하는 W3C의 표준인 RDF(Resource Description Framework)에서는 웹상의 데이터들 간의 연관성을 표현하기 위해 <subject, property, object> 형태로 구성된 트리플을 최소 표현 단위로 사용한다[1]. 여기서 subject와 object는 데이터를 의미하는 리소스(resource)라 하며, property는 두 리소스간의 관련성을 표현한다. RDF로 기술된 데이터는 <그림 1>과 같이 subject와 object를 노드로 하고 property를 간선으로 갖는 방향성 그래프 형태로 표현할 수 있다.

시맨틱 웹의 활성화에 따라 RDF로 표현된 데이터에 대한 질의 처리를 위해 RQL, SPARQL 등의 RDF 질의어가 제안되었다. 초기에 제안된 대부분의 RDF 질의어에서는 리소스간의 복잡한 관계성들의 발견(discovery)을 위한 충분한 지원을 하지 못하고 있다[4][7]. RDF 질의의 전형적인 유형은 임의의 리소스 A로부터 특정한 관계성 R을 갖는 모든 리소스들을 질의 결과로 검색하는 것이다. 이러한 질의 유형은 반환되어야 할 리소스에 대한 조건 즉, 관계성 R에 대한 명세가 질의 조건으로 주어져야 하며, 관계성 R은 조인 조건 혹은 경로식으로 표현될 수 있다.

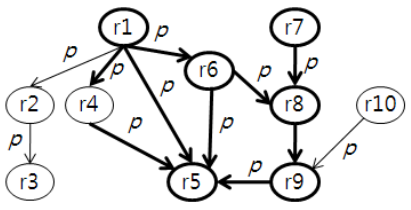

그림 1. 예제 RDF 그래프
Fig. 1. Example RDF Graph

한편, 또 다른 유형의 질의는 리소스들 간의 관계성 R을 검색하는 것이다[2][7]. 예를 들어 리소스 A와 리소스 B 사이에 어떠한 관계가 있는지 검색하는 경우, 두 리소스 사이의 연관성 R이 질의 결과로 반환되어져야 한다.

[2][3][4]에서는 이러한 리소스 사이의 발생하는 복잡한 관계성(complex relationships)을 시맨틱 연관성(semantic association)으로 정의하였다. 결국, 시맨틱 연관성의 발견(discovery)은 두 리소스 개체를 연결하는 길이를 알 수 없는 특정한 의미를 지니는 경로를 검색하는 것이라고 할 수 있다.

[4]에서는 만일 RDF 그래프 상의 임의의 한 시퀀스 'e1, p1, e2, ..., en' 가 있다면 두 개체 e1과 en 사이에는 시맨틱 연관성이 있다고 정의하고 (여기서 ei와 pi는 리소스 개체와 프로퍼티를 각각 의미함), 리소스 개체와 프로퍼티가 교대로 구성된 이러한 시퀀스를 시맨틱 경로(semantic path)로 정의하였다. 또한, 시맨틱 연관성의 유형으로 ρ-intersect 연관성 등을 정의하였다. ρ-intersect 연관성은 두 개의 시맨틱 경로가 특정 리소스 상에 교차 되는 경우를 의미한다[2][7]. 예를 들어 <그림 1>에서 리소스 r1과 r7 사이의 ρ-intersect 연관성은 {(r1→r4→r5, r7→r8→r9→r5), (r1→r5, r7→r8→r9→r5), (r1→r6→r5, r7→r8→r9→r5), (r1→r6→r8, r7→r8)} 등 4개가 존재한다.

2. 접미사 배열을 활용한 인덱싱

[6]에서는 경로식으로 표현될 수 있는 RDF 데이터에 대한 질의의 처리를 위해 인덱싱 기법을 제안하였다. 이 기법에서는 RDF/S 데이터를 DAG로 간주하고 경로 패턴들을 추출하였다. 추출된 경로 패턴들은 변형된 접미사 배열을 이용하여 인덱스를 구축하는데 활용된다. 그러나 이 연구에서는 단방향 단순 질의만을 지원하도록 개발되었으며 시맨틱 연관성 질의 처리에 대해서는 고려하지 않고 있다. [8]에서는 접미사 배열의 이진 탐색 범위 축소를 통한 질의 처리 성능을 개선하도록 하였으나 시맨틱 연관성 질의는 역시 고려하지 않았다.

III. 제안 기법

본 장에서는 [6][8]에서 제안한 것과 같이 접미사 배열을 사용한 인덱싱 기법을 기반으로 시맨틱 연관성 특히, ρ-intersect 연산에 대한 질의 처리 방법을 제안한다. 본 논문에서 RDF 데이터는 DAG 형태로 가정하며, RDF 그래프상에서 추출된 시맨틱 경로는 노드(즉, 리소스)들 만으로 구성된 것으로 정의한다. 예를 들어 <그림 1>로부터 'r1.r2.r3'과 같은 경로들이 추출될 수 있다.

1. 접미사 배열을 사용한 인덱싱

인덱스를 구성하기 위해서는 첫째, RDF 데이터로부터 모든 경로들을 추출하고 각 경로에 대해 경로 식별자(PID)를 할당한다. 경로 식별자는 추출된 경로들을 유일하게 구별하기 위한 값이다.

둘째, 추출된 각 경로부터 접미사들을 생성한다. 예를 들어 <그림 1>로부터 추출된 경로 패턴 'r1.r2.r3' (PID는 1로 가정 시)로부터 3개의 접미사 즉, 'r1.r2.r3', 'r2.r3', 'r3'가 생성된다. 이렇게 생성된 접미사들에 대해서는 경로 식별자와 인덱스 포인트(idx) 쌍으로 구성된 '접미사 레이블'을 할당한다. 인덱스 포인트는 경로 내에서 해당 접미사의 위치 값을 의미한다. 예를 들어, 경로 'r1.r2.r3'로부터 생성된 접미사 'r2.r3'는 접미사 레이블 (1, 2)가 할당 된다.

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

PID \ idx	1	2	3	4	5
1	r1	r2	r3		
2	r1	r4	r5		
3	r1	r5			
4	r1	r6	r5		
5	r1	r6	r8	r9	r5
6	r7	r8	r9	r5	
7	r10	r9	r5		
8	r10	r11			

셋째, 구해진 경로, 접미사 그리고 접미사 레이블들을 테이블 형태로 표현하는데 이를 경로 정보 테이블(pTab)이라 한다. <그림 2>는 <그림 1>로부터 구해진 경로 정보 테이블이다. 다음 단계는 추출된 접미사들을 사전 순으로 정렬하고, 정렬된 각 접미사 레이블 값을 배열 요소의 값으로 갖는 인덱스 즉, 접미사 배열을 생성한다. <그림 3>은 추출된 접미사들을 사전 순으로 정렬하는 과정을 나타낸 것이다. 이와 같은 과정을 통해 유사한 접미사 패턴들은 인덱스 상에서 물리적으로 인접하는 특징을 가지게 된다.

접미사 (정렬 전)	접미사레이블
r1.r2.r3	(1, 1)
r2.r3	(1, 2)
r3	(1, 3)
r1.r4.r5	(2, 1)
r4.r5	(2, 2)
r5	(2, 3)
. . 중략 . .	. . . .
r10.r9.r5	(7, 1)
r9.r5	(7, 2)
r5	(7, 3)
r10.r11	(8, 1)
r11	(8, 2)

정렬 〉

접미사 (정렬 후)	접미사레이블
r1.r2.r3	(1, 1)
r1.r4.r5	(2, 1)
r1.r5	(3, 1)
r1.r6.r5	(4, 1)
r1.r6.r8.r9.r5	(5, 1)
r2.r3	(1, 2)
. . 중략 . .	. . . .
r9.r5	(6, 3)
r9.r5	(7, 2)
r10.r9.r5	(7, 1)
r10.r11	(8, 1)
r11	(8, 2)

그림 3. 인덱스 구축 과정
Fig. 3. Index Construction Process

본 논문에서는 접미사 배열을 이용한 질의 처리 시 수반되는 이진 탐색 범위를 축소하여 질의 처리 성능을 높이도록 한다. 이를 위해 단일 접미사 배열을 사용하는 대신 <그림 4>와 같이 동일한 리소스로부터 시작하는 접미사 패턴들을 그룹핑 하여 각각의 독립적인 접미사 배열에 유지한다. <그림 4>에서 SA_i는 리소스 i로 시작하는 접미사 패턴들에 대한 접미사 레이블들을 유지하는 접미사 배열을 의미한다.

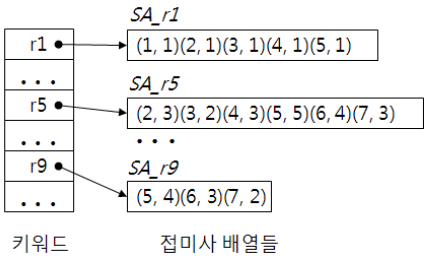

그림 4. 인덱스 구조

Fig. 4. Index Architecture Overview

2. ρ-intersect 연산의 처리

ρ-intersect 연산이 포함된 시맨틱 연관성 검색을 위한 가장 기본적인 질의는 2개의 리소스가 주어지며 두 리소스로부터 시작하는 경로 상에 교차되는 리소스를 검색하는 것이다. ρ-intersect 연관성 검색을 위한 가장 직관적인 처리 방법은 질의에 주어진 각 리소스로부터 시작되는 모든 경로들을 추출한 후 이 경로들 간에 교차되는 노드가 있는지 검색하는 것이며, 이 때 경로를 구성하는 모든 노드들을 대상으로 교차 여부를 확인하게 된다.

본 논문에서 제안하는 질의 처리 성능 향상을 위한 기본 아이디어는 진입 차수가 1인 노드는 절대로 교차(intersect)될 수 없다는 사실을 활용하여 질의 처리 대상을 축소시키는 것이다. 이러한 기본 아이디어의 적용을 위한 선행 작업으로 진입 차수가 2이상인 노드들을 별도로 식별해 둔다. <그림 2>의 경로 정보 테이블(pTab)에 진입차수가 2이상인 노드에 대해 별도로 표기 한다(그림에서는 볼드체로 표기).

<그림 5>는 본 논문에서 제안한 질의 처리 알고리즘을 나타낸 것으로 ProcessingRhoOperation 함수부터 질의 처리 단계가 시작된다. 이 함수의 처음 실행 단계는 GetSuffixLabel 함수를 사용하여 질의에 주어진 두 리소스를 시작 요소로 갖는 접미사 패턴에 대한 접미사 레이블 집합 A와 B를 각각 구하는 것이다.

GetSuffixLabel 함수의 첫 번째 단계는 접미사 배열 SA_i에 대한 이진 탐색과 경로 정보 테이블(pTab)을 이용하여 주어진 질의 패턴과 최초로 일치되는 배열 요소를 찾는다(이때 찾아진 배열 요소의 위치 값을 p라 하자). 두 번째 단계는, 찾아진 배열 요소를 기준으로 좌우에 인접한 나머지 접미사 패턴들을 찾는다. 예를 들어, 질의 패턴이 'r1.r6'일 경우 접미사 레이블 (4, 1)과 (5, 1)를 얻게 된다.

본 논문에서는 접미사 레이블집합 A와 B를 구하는 과정에서 pTab을 이용하여 진입차수가 2개 이상인 노드를 포함하지 않는 접미사 패턴에 대한 접미사 레이블은 필터링 하여 후보 집합에 포함시키지 않도록 한다. 예를 들어, r1에 대한 접미사 레이블 집합은 {(2, 1), (3, 1), (4, 1), (5, 1)}이 된다. 여기서 접미사 레이블 (1, 1)로부터 시작되는 경로는 진입차수가 2이상인 노드가 없으므로 접미사 레이블 (1, 1)은 포함되지 않는다. 만일 직관적 방법을 적용할 경우 접미사 레이블의 집합은 {(1, 1), (2, 1), (3, 1), (4, 1), (5, 1)}가 될 것이다.

두 번째 단계는 첫 번째 단계에서 얻어진 두 개의 접미사 레이블 집합 A와 B에 포함된 각각의 접미사 레이블로부터 시작하는 접미사 패턴 상에 교차되는 노드의 유무를 반복적으로 확인하는 과정으로 경로패턴을 구성하는 각 구성 요소를 비교하면서 처리한다. 본 논문에서는 교차 여부를 평가할 때 진입차수가 2이상이 아닌 노드는 평가 대상에서 제외하여 질의 처리 대상을 축소하여 성능 향상을 얻도록 한다. 각 리소스의 진입차수가 2 이상인지의 확인은 pTab 테이블에서 쉽게 확인할 수 있다.

예를 들어, 접미사 레이블 (2, 1)에 대해 pTab으로 부터 경로 'r1.r4.r5'이 추출되며, 각 경로 요소에 대해 순차적으로 접근하여 처리 시 r1과 r4는 진입차수가 2미만 이므로 질의 처리 시 평가 대상에 포함되지 않고 r5만이 평가된다. 접미사 레이블 (2, 1)과 (6, 1)로부터 추출되는 경로에 대해 위 알고리즘에 따라 진입 차수가 2이상인 노드만을 대상으로 평가를 계속 진행하면 r5를 교차노드로 구하게 된다.

```
Function GetSuffixLabel(QueryPattern)
  // 입력 : 질의 패턴 QueryPatttern
  // 출력 : 접미사 레이블의 집합 tempSet

  1단계) 접미사 배열 SA_i 상 에서 QueryPattern에 첫 번째로 일치하는 배열 요소 위치 값 p를 검색 후 SA_i[p]의 내용(즉, 접미사 레이블
       값)을 결과 집합 tempSet에 추가
  2단계) 위치 p를 기준으로 SA_i 의 좌측 및 우측 배열 요소 중에 QueryPattern에 일치되는 접미사 레이블들을 추가 검색하여 tempSet에 추가

End  Function

Function ProcessingRhoOperation(QueryPattern)
// 입력 : 사용자 질의 패턴 usrQueryPatttern
// 출력 : 교차하는 노드 집합

Call Function GetSuffixLabel(usrQueryPatternA)  // 접미사레이블 집합A를 구함
Call Function GetSuffixLabel(usrQueryPatternB)  // 접미사레이블 집합A를 구함

Foreach S in A  //  let S be a suffix label (spid, sidx)
  While ( R ≠ Null AND Indegree(R) >= 2 )  //  let R be a resource at pTab[spid][sidx]
      Foreach T in B  // let T be a suffix label (tpid, tidx)
        While ( Q ≠ Null AND Indegree(Q) >= 2 )   // let Q be a resource at pTab[tpid][tidx]
            If R == Q  Then // found
                add this resource in the final result set
                Exit While
            End If
            tidx ← tidx+1
         End While
      End For
      sidx ← sidx+1
  End While
End For
```

Ⅳ. 결 론

본 논문에서는 ρ-intersect 연산을 기반으로 하는 시맨틱 연관성 검색에 대한 처리 기법에 대해 제안하였다. 제안 기법에서는 접미사 배열을 이용한 인덱싱과 ρ-intersect 연산의 특징을 고려한 최적화 방법을 활용하여 질의 처리 성능을 향상하도록 하였다. 제안 기법은 전형적인 RDF 질의 유형 뿐만 아니라 다양한 시맨틱 연관성 검색을 위한 기반 기술로 활용될 수 있을 것이다. 향후 제안된 기법의 실험적 성능 평가를 진행할 예정이다.

참고문헌

[1] W3C, RDF Primer, http://www.w3.org/TR/rdf-primer

[2] K. Anyanwu, A. Sheth, ρ-Queries: Enabling Querying for Semantic Associations on the Semantic Web, Proc. of Int'l Conf. on WWW, 2003, pp.690~699

[3] B. Aleman-Meza et al. Ranking Complex Relationships on the Semantic Web, IEEE Internet Computing, Vol. 9, no. 3, pp. 37~44, 2005,

[4] A. Sheth et al, Semantic Association Identification and Knowledge Discovery for National Security Applications, Journal of Database Management, Vol 16, pp33~53, 2005

[5] F. Al-Khateeb et al., Complex path queries for RDF Graph. Proc. of the 4th Int'l Semantic Web Conf, Poster paper (poster id 52), 2005

[6] A. Matono, et al., "An Indexing Scheme for RDF and RDF Schema based on Suffix Arrays", First Int'l Workshop on SWDB, pp.151~168, Sept. 2003

[7] K. Kochut and M. Janik, SPARQLeR: Extended Sparql for Semantic Association Discovery, LNCS, Vol. 4519, Proc. of the 4th European Conf. on The Semantic Web, pp. 145~159, 2007

[8] S. Kim, Improved Processing of Path Query on RDF Data Using Suffix Array, Journal of Convergence Information Technology, Volume 4, Number 3, 2009.