관계형 데이터베이스 기반의 후방향 추론을 이용하는 확장 가능한 RDF 데이타 변경 탐지 기법

(A Scalable Change Detection Technique for RDF Data using a Backward-chaining Inference based on Relational Databases)

임 동 혁 † 이 상 원 †† 김 형 주 †††
(Dong-Hyuk Im) (Sang-Won Lee) (Hyoung-Joo Kim)

요 약 최근의 RDF 변경 탐지 기법들은 구조적인 변경 이외에, RDF 모델의 클로저를 적용하여 변경 부분을 탐지하는 의미적 변경도 다룬다. 하지만, 기존의 의미적 변경을 고려하는 탐지 기법들은 메모리 저장 공간에 전체 트리플 집합을 적재하여 변경 내용을 탐지하거나, RDF 모델의 클로저를 미리 계산하는 전방향 추론을 사용하기 때문에 대용량 RDF 데이터의 변경 탐지에 비효율적이다. 따라서, 본 논문에서는 관계형 데이터베이스 기반의 후방향 추론 기법을 사용하는 변경 탐지 기법을 제안한다. 제안된 기법은 관계형 데이터베이스에서 변경 탐지에 사용 가능한 트리플들에 대해서만 추론을 수행한다. 생물 정보 도메인에서 사용되는 실제 RDF 데이터들에 대한 비교 실험을 통하여 제안된 기법이 더 효율적임을 보인다.

키워드 : RDF, 변경 탐지, 추론, 관계형 데이터베이스

Abstract Recent studies on change detection for RDF data are focused on not only the structural difference but also the semantic-aware difference by computing the closure of RDF models. However, since these techniques which take into account the semantics of RDF model require both RDF models to be memory resident, or they use a forward-chaining strategy which computes the entire closure in advance, it is not efficient to apply them directly to detect changes in large RDF data. In this paper, we propose a scalable change detection technique for RDF data, which uses a backward-chaining inference based on relational database. Proposed method uses a new approach for RDF reasoning that computes only the relevant part of the closure for change detection in a relational database. We show that our method clearly outperforms the previous works through experiment using the real RDF from the bioinformatics domain.

Key words : RDF, change detection, inference, relational database

· 본 연구는 지식경제부 및 정보통신산업진흥원의 대학 IT연구센터 지원사업 (NIPA-2010-C1090-1031-0002), (NIPA-2010-C1090-1021-0008) 및 BK-21 정보 기술 사업단, 서울시의 지원으로 수행한 '서울시 산학연협력사업(PA090903)'의 연구결과로 수행되었음

† 학생회원 : 서울대학교 컴퓨터공학부
dhim@idb.snu.ac.kr
†† 정 회 원 : 성균관대학교 정보통신공학부 교수
wonlee@ece.skku.ac.kr
††† 종신회원 : 서울대학교 컴퓨터공학부 교수
hjk@snu.ac.kr

논문접수 : 2010년 2월 24일
심사완료 : 2010년 7월 22일

Copyright©2010 한국정보과학회 : 개인 목적이나 교육 목적인 경우, 이 저작물의 전체 또는 일부에 대한 복사본 혹은 디지털 사본의 제작을 허가합니다. 이 때, 사본은 상업적 수단으로 사용할 수 없으며 첫 페이지에 본 문구와 출처를 반드시 명시해야 합니다. 이 외의 목적으로 복제, 배포, 출판, 전송 등 모든 유형의 사용행위를 하는 경우에 대하여는 사전에 허가를 얻고 비용을 지불해야 합니다.

정보과학회논문지: 데이타베이스 제37권 제4호(2010.8)

1. 서 론

RDF(Resource Description Framework)는 웹 상에서 정보 자원을 기술하기 위한 메타데이터 표준 언어로 W3C에 의해 제정되었다[1]. RDF는 시맨틱 웹이 차세대 웹으로 진행됨에 따라 이질적인 다수의 정보 자원을 통합하고, 통합된 질의어 처리를 위한 온톨로지 언어로 사용이 높아지고 있다. 따라서, 사용자는 정보 자원의 구조적 특성에 관계없이 RDF 질의어(i.e., SPARQL)를 이용하여 정보 자원을 통합 검색할 수 있다. 그런데, RDF와 같은 온톨로지는 다음과 같은 요구에 의해 진화하는 특징을 가진다[2]. 첫째, 실 세계 지식 도메인 자체의 정보가 갱신되기 때문에 도메인을 모델링하는 온톨로지 역시 변화되어야 한다. 둘째, 온톨로지는 사용하는

개인의 필요에 의해 변화한다. 온톨로지의 개발 과정 자체가 다수의 사용자에 의한 협력적이고 병렬적인 속성을 가졌기 때문에, 개인의 요구에 의해 온톨로지의 변경이 일어난다. 따라서, 온톨로지의 변경 내용을 정확하고 효율적으로 탐지하고, 탐지된 변경 결과에 대한 분석(i.e., 특정 기간 동안에 빈번하게 변경된 개념 혹은 속성들을 찾는 일)은 지식 관리 시스템에서 핵심이 된다.

하지만, 텍스트, XML문서 등에 사용되는 기존의 변경 탐지 기법들은 그래프의 구조적 차이와 RDF에서 사용하는 의미적 관계들을 표현할 수 없기 때문에 RDF에 그대로 적용할 수 없다. 따라서, 팀 버너스 리는 RDF 데이터의 변경 결과에 대한 교환 및 배포, 갱신을 위해 RDF 특성에 맞는 변경 탐지 기법의 필요성과 중요성을 소개하였다[3]. 또한, 변경탐지 결과는 버전 관리 시스템이나 동기화 서비스에 사용되기 때문에, 저장 및 네트워크상의 교환을 위해 정확하면서도 크기가 작아야 한다. 이와 같은 요구사항으로 인해, 최근의 RDF 변경탐지에 대한 연구들은 변경 탐지 결과의 크기를 최소화하기 위해 단순한 구조적 차이가 아닌 추론 기반의 의미적 관계의 변경 탐지에 초점을 두고 있다[4,5].

그림 1은 RDF 모델의 갱신 전후의 한 예를 보여준다. 그림 1에서 각RDF 모델은 트리플(주어, 술어, 목적어)을 원소로 갖는 집합으로 표현되며, 두 모델의 변경 부분은 두 집합의 차이로 계산된다. 이와 같이, 명시적으로 표현된 집합들의 차이를 계산하는 방법은 RDF 변경 탐지에서 많이 사용되고 있다. 그런데, RDF 모델에서는 추론을 지원하고, 이 추론을 통해서 RDF 모델에 명시적으로 표현되지 않는 새로운 트리플들을 생성한다.

따라서, SemVersion[4], Delta Function[5]과 같은 추론을 이용하는 변경 탐지 기법들은 두 트리플 집합간의 차집합을 계산하는 것보다 더 작은 크기의 변경 부분을 계산한다. 예들 들어, 그림 1에서 RDF 모델 K의 (TA subClassOf Univ_Person) 트리플은 K, K'의 명시적인 변경 탐지에서 삭제된 트리플에 속한다. 하지만, K'을 추론하면 (TA subClassOf Univ_Person)이 생성되기 때문에 이 트리플은 삭제되는 트리플 집합에 포함되지 않는다. 따라서, 추론을 이용하는 변경 탐지는 명시적인 트리플들만의 변경 탐지보다 변경 크기를 줄여

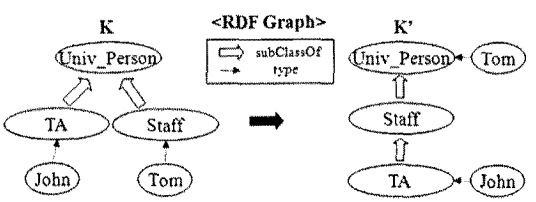


그림 1 RDF 모델의 갱신 전후의 예

주며, RDF 모델의 의미적 변경 내용을 나타낸다.

하지만, 기존의 변경 탐지 기법인 [5]는 메인 메모리 저장 공간에 전체 트리플 집합을 적재하여 두 집합간의 차집합 계산을 하며, [6]은 본 논문과 마찬가지로 후방향 추론을 사용하였으나 마찬가지로 메인 메모리 상에서 변경 탐지를 수행하기 때문에 일단 전체 트리플 집합을 메모리에 적재해야 하는 단점을 가지게 된다. RDF 데이터는 여러 웹 데이터 소스들을 트리플의 형태로 표현하여 연결시키려는 Linked Data[7]의 확대에 의해 점차 그 크기가 증가하고 있다. 이처럼 대용량의 데이터들에 있어서 전체 트리플 집합을 메인 메모리에 적재하여 변경 부분을 탐지하는 것은 어렵다.

따라서, [8]에서 XML 데이터를 관계형 데이터베이스에 저장한 후 SQL을 이용하여 효율적인 변경 탐지를 하였듯이, RDF 데이터에 있어서도 관계형 데이터베이스를 이용한 변경 탐지 기법이 필요하다. 하지만, SemVersion[4]과 같은 관계형 데이터베이스 기반의 RDF변경 탐지 기법은 조인 연산이 빈번히 사용되는 RDFS 함의 규칙을 미리 계산한 후, 추론된 트리플을 데이터베이스에 실체화하여 저장하는 전방향 추론 기법을 사용하기 때문에 대용량 RDF 데이터에 적합하지 않다.

본 논문에서는 대용량 데이터를 처리할 수 있는 변경 탐지의 확장성과 효율성을 고려한다. 대용량 데이터의 지원을 위해, 변경 탐지 연산에 적합한 관계형 데이터 베이스 스키마 설계와 SQL에 기반한 변경 탐지를 제안한다. 또한, [4]와 같이 관계형 데이터베이스에 저장된 모든 트리플 집합을 먼저 추론하고 실체화 저장하여 변경 탐지를 수행하는 것이 아니라, 집합 전체가 아닌 작은 부분의 트리플들만 후방향 추론해서 변경 탐지에 사용한다.

본 논문의 2장에서는 RDF 변경 탐지에 대한 관련 연구를 소개한다. 3장과 4장에서는 제안하는 변경 탐지 기법을 구체적으로 소개하며, 5장에서는 실험을 통해 제안된 기법의 우수성을 보이고, 6장에서 결론을 맺는다.

2. 배경 지식 및 관련 연구

RDF 모델은 DAG로 표현된다. RDF 그래프의 각 노드는 정보 자원을 표현하고, 간선은 노드들 사이의 관계를 나타낸다. 또한, RDF 그래프는 두 자원간의 이진 관계를 표현하는 트리플을 원소로 갖는 집합으로 표현된다.

정의 1. 트리플 집합 K에 속한 트리플은 주어(Subject), 술어(Predicate), 목적어(Object)로 이루어지며 t(S, P, O) ∈ K 이다. (S는 주어, P는 술어, O는 목적어)

RDF 추론은 W3C의 권고안인 RDF Semantics의 함의 규칙[9]을 이용한다. 이때 본 논문에서는 모든 함의 규칙을 적용하는 것이 아니라 [4]에서와 같이 기본 타입이나 도메인/레인지를 추가하는 함의 규칙을 제외한 클

래스 계층 관계에 대한 추론, 술어 계층 관계에 대한 추론, 클래스 계층 관계를 이용한 추론, 술어 계층 관계를 이용한 추론 등으로 추론의 범위를 한정하여 사용한다.

다음 정의는 논문에서 사용하는 RDFS 함의 규칙 중 계층 관계와 관련된 규칙을 나타낸다.

정의 2. 본 논문에서 변경 탐지에 사용되는 RDFS 함의 규칙은 다음과 같다.

1) (U subPropertyOf V) ∧ (V subPropertyOf X) ⇒ (U subPropertyOf X)
2) (U A Y) ∧ (A subPropertyOf B) ⇒ (U B Y)
3) (V type U) ∧ (U subClassOf X) ⇒ (V subClassOf X)
4) (U subClassOf V) ∧ (V subClassOf X) ⇒ (U subClassOf X)

RDF 모델의 대표적인 변경 탐지 기법은 다음과 같다. △E(Explicit)는 가장 기본적인 방법으로Jena[10]에서 사용하고 있으며, △C(Closure)는 SemVersion[4]에서 제안을 하였으며 최초로 변경 탐지에 RDFS 함의 규칙을 적용하였다. △D(Dense)는 Delta Function[5]에서 사용하는 방법으로 RDFS 함의 규칙 적용 대상을 변경하여 △E와 △C보다 최소의 변경 탐지 결과를 제공해 준다. RDF 모델에서의 변경 탐지 연산은 트리플의 삭제와 트리플의 삽입으로만 나타낸다. 이때, RDF 모델 K에 정의 2의 규칙을 적용한 집합을 C(K)로 표시하며 Ins는 삽입 연산을 Del은 삭제 연산을 표시한다.

$$\triangle E = \{Ins(t(S,P,O)) \mid t(S,P,O) \in K' - K\} \cup \{Del(t(S,P,O)) \mid t(S,P,O) \in K - K'\}$$

$$\triangle C = \{Ins(t(S,P,O)) \mid t(S,P,O) \in C(K') - C(K)\} \cup \{Del(t(S,P,O)) \mid t(S,P,O) \in C(K) - C(K')\}$$

$$\triangle D = \{Ins(t(S,P,O)) \mid t(S,P,O) \in K' - C(K)\} \cup \{Del(t(S,P,O)) \mid t(S,P,O) \in K - C(K')\}$$

아래의 3가지 변경탐지 결과는 그림 1에서 살펴본 변경 예제를 바탕으로 한 것이다. 본 논문에서는 최소 크기의 변경 탐지 기법인 △D를 다룬다.

△E={Ins(TA subClassOf Staff), Ins(Tom type Univ_Person), Del(TA subClassOf Univ_Person), Del(Tom type Staff)}
△C={Ins(TA subClassOf Staff), Ins(John type Staff), Del(Tom type Staff)}
△D={Ins(TA subClassOf Staff), Del(Tom type Staff)}

3. 후방향 추론 기반의 변경탐지 기법

앞에서 소개한 변경 탐지 기법들에 사용되는 두 가지 핵심 연산은 트리플의 차집합 계산과 추론 연산이다. 차집합 연산이 단순한 비교 연산으로 수행되는 반면, 추론 연산은 기존의 트리플 집합에서 RDFS 함의 규칙에 의해 추가로 생성되는 트리플들 찾아야 하기 때문에 의미적 변경 탐지의 핵심이 된다. RDF 추론 전략은 추론 방식에 따라 전방향 추론과 후방향 추론으로 구분된다. 전방향 추론은 현재 정의된 사실들에 기반하여 적용 가능한 규칙을 연속적으로 적용하여 최종 목표에 도달하는 방식이다. 반면, 후방향 추론은 찾고자 하는 목표에서 시작하여 이를 지지할 수 있는 사실들을 찾는 방식이다. 상당수의 RDF 저장 시스템은 사용자들에게 빠른 질의를 지원하기 위해 전방향 추론을 사용한다. 특히, 대표적인 RDF 관리 시스템인 Sesame[11]에서는 더 이상 추론되는 트리플이 없을 때 추론이 종료되는 고갈적 전방향 추론을 사용하고 있다. 또한, 변경 탐지 시스템에 있어서, SemVersion역시 미리 추론을 한 후 실체화하여 데이터베이스에 저장한 후 변경 내용을 계산하는 전방향 추론 방식을 사용한다. 데이터베이스 기반의 전방향 추론을 사용하는 최소 크기의 변경 탐지(△D)를 고려해보자. 전방향 추론 기반의 변경 탐지는 두 모델 K, K'에 대해 추론을 수행하여 C(K), C(K')을 먼저 계산하고 추론이 완료된 이후에 (K-C(K'))와 (K'-C(K))를 계산한다. 하지만, RDF 모델을 먼저 추론한 후 비교를 하면 변경 부분을 줄일 수 있지만 전체 모델을 추론하는데 많은 시간이 필요하다. 특히, 관계형 데이터베이스 기반의 RDF저장 시스템에서는 트리플 테이블에 저장한 후 조인 연산을 통해 추론을 수행하기 때문에 비효율성은 더욱 심각하다. 따라서, 본 논문에서는 먼저 트리플의 차집합 연산으로 ((K-K')∪(K'-K), △E)을 계산한 후, 그 결과 트리플들에 대해서만 후방향 추론을 적용해서 ((K-C(K'))∪(K'-C(K)), △D)를 계산하는 방식을 제안한다. 다음 정리는 제안하는 후방향 추론의 정당성을 설명한다.

정리 1. 트리플 집합 K, K에 Closure Rule을 적용한 집합 C(K)이 주어졌을 때, K⊆C(K)를 만족한다.

정리 2. △D 는 △E 의 부분 집합이다. (△D⊆△E)

증명 : △D∩△E = △D 임을 증명한다.

$(K-C(K')) \cap (K-K') = (K \cap C(K')^c) \cap (K \cap K'^c)$

$= K \cap C(K')^c \cap K'^c = K \cap (C(K') \cup K')^c = K \cap C(K')^c$

$= (K-C(K'))$ (정리 1 : $C(K') \cup K' = C(K')$)

따라서, $(K-C(K')) \subseteq (K-K')$이 성립한다. (1)

또한, $(K' - C(K)) \cap (K' - K)$ 역시 같은 방법으로 $(K'-C(K)) \subseteq (K'-K)$ 이 성립한다. (2)

(1), (2)의 조건에 의해 (K-C(K'))∪(K'-C(K))⊆(K-K')∪ (K'-K) 을 만족하므로 △D⊆△E 임을 증명한다.

정리 2는 △D가 △E의 부분 집합임을 증명한다. 즉, △E의 결과가 △D의 모든 결과를 포함하는 완전성을 보장한다. 따라서, 어떤 트리플이 △E에 포함되지 않는다면 △D에도 포함되지 않는다.

정리 3. △E에서 C(K'), C(K)에 포함되는 트리플들을 제거한 결과는 △D와 동일하다.

증명 : (K-K')-C(K')∪(K'-K)-C(K) = △D임을 증명한다.

먼저 (K-K') -C(K') = (K-K') ∩ C(K')^c = (K∩K''^{c}) ∩ C(K')^c

= K ∩ (K''∩ C(K')^c) = K ∩ C(K')^c= K-C(K')

따라서, (K-K')-C(K') = K-C(K')을 만족한다. (1)

마찬가지로, (K'-K) -C(K) = K'-C(K)가 된다. (2)

(1), (2)의 조건에 의해 △D와 동일함을 알 수 있다.

다시 말해, 정리 3은 △E에 속한 트리플들에 후방향 추론을 적용한 후, C(K'), C(K)에 포함되는 트리플들을 찾아내어 △E에서 제거하면 △D 와 동일한 결과를 가진다. 이는 K나 K'이 아닌 더 작은 크기의 (K-K')과 (K'-K)에 후방향 추론을 적용해서 C(K)와 C(K')를 찾아 최소 변경 탐지 △D를 찾을 수 있는 것을 의미한다. [6]에서도 구조적 변경을 사용한 후 [11]의 추론 엔진을 이용하여 후방향 추론을 사용하였으나 본 논문에서는 △E에서 △D를 유도하는 차이점을 보이며 또한 4장에서 설명할 관계형 데이터베이스의 SQL을 이용하여 후방향 추론을 하는 차이점을 가진다.

4. 관계형 데이터베이스 기반의 변경 탐지

관계형 데이터베이스 기반의 변경 탐지는 대용량 데이터에 대한 확장성 및 안정성 등의 장점을 제공하며, 이미 저장되어 있는 다른 데이터와 변경 탐지를 할 때마다 데이터를 파싱할 필요가 없는 장점을 가진다. 관계형 데이터베이스 기반의 변경 탐지를 위해서는 먼저 RDF데이터를 파싱하여 테이블에 저장해야 한다. 본 논문에서는 RDF를 저장하는 가장 일반적인 형태인 트리플 형식의 저장소를 사용한다. [11]에서 RDF를 subClass, subProperty, Class, Type 등의 술어 테이블과 전체 트리플을 위한 Triple 테이블로 나누어 저장하였듯이, 본 논문 역시 Triples 테이블과 2장의 정의 2 규칙에서 사용하는 술어를 포함하는 트리플을 별도의 테이블로 구성한다. 하지만, Sesame에서는 클래스, 인스턴스 등에 대해서도 별도의 테이블을 사용하였지만 본 논문에서는 이와 같은 정보는 Triples테이블에 포함하며 오직 추론을 통해 변경 탐지 결과를 줄여줄 수 있는 술어를 가진 트리플들만을 별도의 테이블로 생성하여 저장하였다. 또한, 변경 탐지 결과 역시 트리플로 표현되기 때문에 삭제 테이블(DEL)과 삽입 테이블(INS)로 구성한다. 본 논문의 RDB 스키마 구조는 다음과 같다.

Triples (Sub, Pred, Obj), SubClass (Sub, Pred, Obj), SubProperty (Sub, Pred, Obj), Type (Sub, Pred, Obj), INS (Sub, Pred, Obj), DEL (Sub, Pred, Obj)

이와 같은 구조는 변경 단위가 트리플이기 때문에 효율적으로 SQL 처리가 가능하며, 술어 별로 테이블이 구성되기 때문에 추론 시 해당 테이블만 접근하여 추론할 수 있는 장점을 가진다. 예를 들어, 상하위 관계(subClassOf)에 대한 차집합 연산 및 추론의 연산은 SubClass 테이블에만 접근해서 계산할 수 있다.

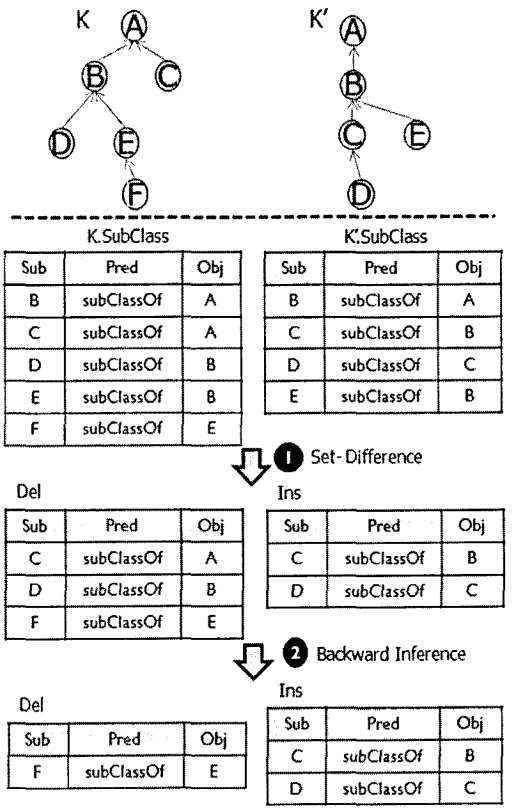


그림 2 변경 탐지의 예

그림 2는 두 RDF 데이터 K, K'의 상하위 관계에 대한 후방향 기반의 변경 탐지의 기법을 보여준다. 우선, 다음의 차집합 SQL을 이용하여 두 모델의 SubClass 테이블에서 명시적으로 삭제, 삽입되는 트리플 집합 DEL, INS를 각각 계산한다.

```
INSERT INTO DEL ((SELECT * FROM
K.SUBCLASS) MINUS (SELECT * FROM
K'.SUBCLASS))
INSERT INTO INS ((SELECT * FROM
K'.SUBCLASS) MINUS (SELECT * FROM
K.SUBCLASS))
```

다음으로, 이전 단계의 차집합 연산에 대응되어 걸러진 트리플들에 대해서 후방향 추론을 적용한다. 후방향 추론의 계산은 Sesame처럼 테이블의 조인 연산을 사용하는 것이 아니라, 재귀적 질의를 사용하며 표준 SQL인 WITH를 이용한다. 예들 들어, 다음 SQL은 그림에서의 DEL 테이블의 (C subClassOf A)가 K'SubClass 테이블에서 후방향으로 추론이 가능한지 계산한다.

```
WITH TEMP (SUB, OBJ) AS
((SELECT SUB, OBJ FROM K'SUBCLASS
WHERE SUB='C') UNION ALL (SELECT
t1.SUB, t1.OBJ FROM K'.SUBCLASS t1, TEMP
t2 WHERE t1.SUB=t2.OBJ))
SELECT * FROM TEMP WHERE OBJ ='A'
```

SELECT 연산 결과가 존재하면, 후방향 추론의 결과는 참이 된다. 그림 2에서, (C subClassOf A)는 (C subClassOf B), (B subClassOf A)이 존재하므로 추론의 값이 참이 된다. 마지막으로, 후방향 추론의 결과에 따라, Ins 혹은 Del 테이블의 결과에서 해당되는 트리플을 삭제한다. 예를 들어, 트리플 (C subClassOf A)는 후방향 추론의 값이 참이므로 Del 테이블에서 삭제한다. 이와 같은 과정에 의해 최종 변경 탐지 결과가 생성된다. 결국, 최종 변경 탐지 결과는 전방향 추론 기반의 변경 탐지 결과와 같다.

5. 성능 분석

본 실험은 리눅스 환경의 1G 메인 메모리를 가진 펜티엄 4-3.2Hz에서 수행되었으며 자바로 구현되었다. 관계형 데이터베이스로는 오라클 11G를 사용하였으며, 버퍼 캐쉬 크기는 512M로 사용하였다. 실험은 인위적으로 생성한 데이터 집합들과 생물 정보 도메인에서 많이 사용되는 Gene Ontology(G1~G9, 9개월간의 월별 TermDB 버전)와 Uniprot(U1~U6, 버전 14.8~15.3) 집합의 클래스 상하위 관계를 대상으로 하였다. 인위적인 데이터는 Jena[10]를 이용하여 트리플의 수를 증가시키며 생성하였다. 이 때 실험에서 비교 대상이 되는 데이터는 5% 변화율로 임의적으로 선택되도록 변경하였다. 모든 실험은 메모리 기반의 변경 탐지를 제외한 관계형 데이터베이스 기반의 전방향 추론 방식과 후방향 추론 방식의 최소 크기의 변경 탐지 기법(△D)의 결과 및 탐지 실행 시간을 비교하였다.

표 1 실제 데이터에서의 변경 탐지 결과

		(G1,G2)	(U1,U2)
전방향 기반 변경 탐지 (Forward)	삽입	647	5401
	삭제	115	770
후방향 기반 변경 탐지 (Backward)	삽입	647	5401
	삭제	115	770

표 1은 실험 데이터 중 (G1,G2), (U1,U2)에서 △D의 전방향 추론 기반의 변경 탐지 결과와 후방향 기반의 변경 탐지 결과(클래스 상하위 관계에 대한 삽입 연산과 삭제 연산의 개수)를 보여준다. 결과에서 알 수 있듯이 두 기법을 사용한 변경 탐지의 결과가 일치함을 알 수 있다.

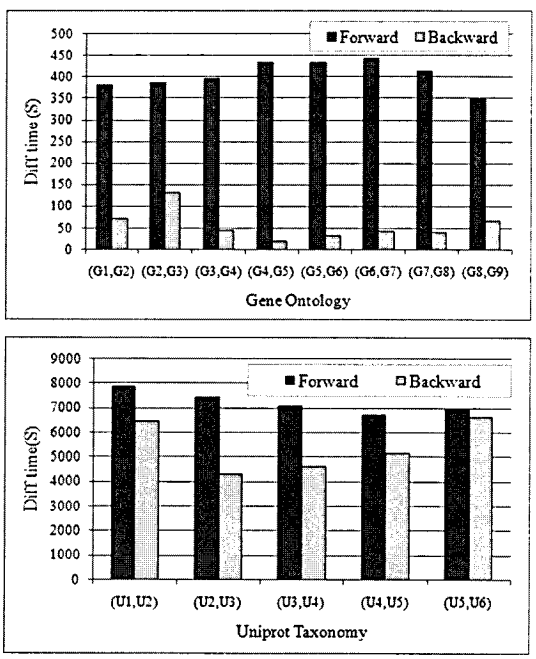


그림 3 △D 변경 탐지 성능 비교

그림 3은 실제 데이터들에 있어서의 최소 변경 탐지 △D (전방향, 후방향)의 실행 시간을 각각 보여준다. 그림에서 알 수 있듯이, 모든 경우에 있어 후방향 추론이 훨씬 빠른 실행 시간을 보인다. 전방향 추론 기법이 RDF 모델의 전체를 추론하는 반면, 후방향 추론 기법은 △E에 포함되는 트리플들만을 추론하기 때문이다.

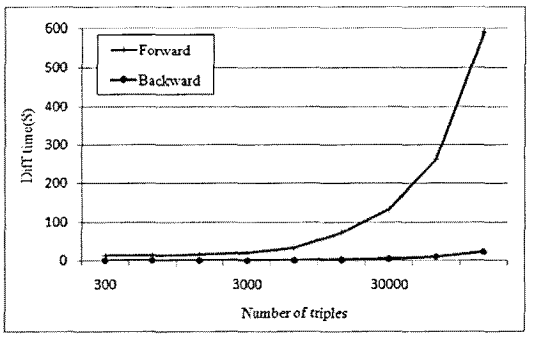


그림 4 변경 탐지의 확장 가능성 비교

또한, 변경 탐지의 확장 가능성을 평가하기 위해, 인위적인 데이터의 크기를 증가시켜 가면서 변경 탐지의 성능을 비교하였다. 그림 4는 인위적인 데이터에 있어서의 2가지 기법들의 실행 시간을 보여준다. 제안된 기법이 데이터가 증가함에 따라 훨씬 빠른 시간에 변경 내용을 탐지하는 것을 알 수 있다.

6. 결론 및 향후 연구 계획

최근의 RDF 모델의 변경탐지 기법에 대한 연구들은 RDFS 함의 규칙을 적용하여 더 작은 크기의 변경 부분을 찾으려는 연구들이 주를 이룬다. 하지만 기존의 기법들은 메모리 기반이거나, 미리 추론을 사용하여 대용량 데이터에 대하여 적합하지 않다. 본 논문에서는 이러한 문제점을 해결하기 위해 후방향 추론을 지원하는 관계형 데이터베이스 기반의 변경 탐지 기법을 제안하였다. 제안하는 변경 탐지 기법은 우선, 관계형 데이터베이스에 저장을 하고 두 모델에 대해 △E를 먼저 계산한 후, 이때 얻어진 변경 탐지 결과에 후방향 추론을 적용하는 방법을 사용하여 최소 크기의 변경 탐지 결과 △D를 얻는다. 이러한 방법은 기존의 데이터베이스 기반의 변경 탐지보다 우수한 성능을 보이며 대용량의 RDF 데이터의 변경 탐지에 적합하다.

참 고 문 헌

[1] G. Klyne et al., "Resource Description Framework (RDF): Concepts and Abstract Syntax," *W3C Recommendation*, 2004.

[2] G. Flouris et al., "Ontology Change: Classification and survey," *The Knowledge Engineering Review*, vol.23, no.2, 2008.

[3] T. Berners-Lee and D. Connolly, "Delta: An Ontology for the Distribution of Differences Between RDF Graphs," http://www.w3c.org/DesignIssues/Diff, 2004.

[4] M. Volkel et al., "SemVersion: An RDF-based Ontology Versioning System," *Proc. of ICWI*, 2006.

[5] D. Zeginis et al., "On the Foundation of Computing Deltas between RDF Models," *Proc. of ISWC*, 2007.

[6] D. H. Im and H. J. Kim, "Efficient Change Detection between RDF Models Using Backward Chaining Strategy," Journal of KIISE : Computing Practice and Letters, vol.15, no.2, 2009 (in Korean).

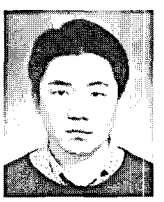
[7] C. Bizer, T. Heath and T. Berners-Lee, "Linked Data - The Story So Far," *International Journal on Semantic Web and Information Systems*, vol.5, no.3, 2009.

[8] E. Leonardi et al., "XANDY: Detecting Changes on Large Unordered XML Documents using Relational Databases," *Proc. of DASFAA*, 2005.

[9] P. Hayes et al., "RDF Semantics," Technical Report, W3C Recommendataion, 2004.

[10] J. J. Carroll et al., "Jena: implementation the semantic web recommendation," *Proc. of WWW*, 2003.

[11] J. Broekstra et al., "Sesame: A Generic Architecture for Storing and Querying RDF and RDF Schema," *Proc. of ISWC*, 2002.



임 동 혁

2003년 고려대학교 컴퓨터교육과(학사) 2005년 서울대학교 컴퓨터공학부(석사) 2005년~현재 서울대학교 컴퓨터공학부 박사과정 재학 중. 관심분야는 데이터베이스, 시맨틱 웹, 온톨로지

이 상 원

1991년 서울대학교 컴퓨터학과(학사). 1994년 서울대학교 컴퓨터학과(석사). 1999년 서울대학교 컴퓨터학과(박사). 1999년~2001년 한국 오라클. 2001년~2002년 이화여대 BK21 계약교수. 2002년~현재 성균관대학교 정보통신공학부 부교수. 관심분야는 flash memory DBMS 온톨로지

김 형 주

1982년 서울대학교 전산학과(학사). 1985년 Univ. of Texas at Austin(석사) 1988년 Univ. of Texas at Austin(박사). 1988년~1990년 Georgia Institute of Technology(부교수). 1991년~현재 서울대학교 컴퓨터공학부 교수. 관심분야는 데이타베이스, XML, 시맨틱 웹, 온톨로지