SemFilter: 단순하며 효율적인 시맨틱 XML 메시지 필터링

(SemFilter: A Simple and Efficient Semantic XML Message Filtering)

김 재 훈 † 박 석 ††
(Jaehoon Kim) (Seog Park)

요 약 XML 메시지 필터링에 관한 최근의 연구들은 모든 출판되는 데이타 소스들이 필터링 시스템에 정의된 유일한 전역 스키마를 따르는 것을 가정한다. 하지만 이러한 가정을 넘어서, 데이타 제공자들이 그들 자신의 스키마를 자유롭게 사용할 수 있는 서비스를 고려할 수 있다. 즉, 데이타 소스들이 이질적인 환경이다. 하지만 XML 필터링 시스템에서 데이타 소스는 다수이며, 또한 출판되는 데이타들은 수시로 생성되고, 갱신되며, 사라진다. 즉, 매우 다이내믹한 환경이다. 본 논문에서는 그러한 다이내믹한 환경을 고려하여 고안된 단순하며 효율적인 의미적 XPath 질의 번역 구현을 소개한다. 특별히 제안되는 질의 번역 기법은 어떤 비주얼한 데이타 가이드가 제공되지 않는 환경에서 사용자가 자신의 지식과 경험에만 의존하여 작성한 질의를 번역하는 것에 초점을 맞춘다. 이러한 환경에서, 사용자는 다수의 이질적인 데이타를 질의하기 때문에, 사용자의 기억상의 스키마에 의존하여 작성된 질의는 실제 스키마와 불일치할 수 있다. 본 연구에서는 제안하는 의미적 XPath 질의 기법이 이러한 문제를 고려하도록 설계한다. 몇 가지 실험 결과는 제안된 질의 번역 기법이 수용할 만한 질의 번역 시간을 제공하며, 기존의 방법과 비교하여 실제적임을 보여 준다.

키워드 : 의미적 XML 필터링 시스템, 이질적 데이타, 데이타 통합, XPath 질의 번역

Abstract Recent studies on XML filtering assume that all data sources follow a single global schema defined in a filtering system. However, beyond this simple assumption, a filtering system can provide a service that allows data publishers to have their own schema; hence, the data sources will become heterogeneous. The number of data sources is expected to be large in a filtering system and the data sources are frequently published, updated, and disappeared, that is, dynamic. In this paper, we introduce implementing a simple and efficient XPath query translation method for such a dynamic environment. The method is especially targeted for a query which is composed based only on users' knowledge and experience without a graphical guidance of the global schema. When a user queries a large number of heterogeneous data, there is a high possibility that the query is not consistent with the same local schema assumed by the user. Our query translation method also supports a function for this problem. Some experimental results for query translation performance have shown that our method has reasonable performance, and is more practical than the existing method.

Key words : Semantic XML filtering system, Heterogeneous data, Data integration, XPath query translation

· 본 연구는 한국과학재단 특정기초연구(R01-2006-000-10609-0) 지원으로 수행되었음

† 정 회 원 : 서강대학교 컴퓨터공학과 교수
jhkimygk@gmail.com
†† 종신회원 : 서강대학교 컴퓨터공학과 교수
spark@dblab.sogang.ac.kr

논문접수 : 2008년 3월 31일
심사완료 : 2008년 6월 3일

정보과학회논문지: 컴퓨팅의 실제 및 레터 제14권 제7호(2008.10)

1. 서 론

과거 몇 년 동안 효율적이며 확장 용이한(scalable) XML 필터링 시스템을 만들려는 다양한 노력이 있었다. 예로, XFilter[1], YFilter[2], AFilter[3], PosFilter[4]등을 살펴 볼 수 있다. 하지만 필터링 서비스에는 의미적 상호 운용성(semantic interoperability)의 해결되어야 할 또 다른 중요한 문제가 있다. 즉, 같은 주제를 담고 있는 몇몇 데이타 소스들이 각기 다른 스키마를 가질 때, 특정 데이타 소스에 대한 XPath 질의는 다른 데이타 소스와 매치되지 않으므로 적절한 질의 번역 기능이 요구된다.

이러한 문제와 관련하여 본 연구에서 조사한 바에 의하면 Chen et al.[5], Kanza et al.[6], Amer-Yaiha et al.[7] 등의 연구를 살펴 볼 수 있다. Kanza et al.과 Amer-Yaiha et al.의 방법은 XPath 질의의 구조적 이질성의 문제를 고려한 질의 번역 방법이고, Chen et al.의 방법은 본 연구와 유사하게 XPath 질의의 구조적 이질성뿐만 아니라 의미적 이질성의 문제도 함께 고려한 질의 번역 방법이다. 하지만 이러한 방법들은 XML 메시지 필터링과 같이 매우 다이내믹한 환경을 고려한 방법은 아니다. XML 필터링 시스템에서는 매우 방대한 수의 XML 문서가 수시로 출판, 갱신, 삭제되며, 또한 매우 방대한 수의 XPath 질의가 수시로 등록 및 취소된다. 따라서 보다 신속한 질의 번역의 실질적인 방법이 요구된다.

본 논문에서는 제안하는 방법이 큰 흐름에서는 Kanza et al.과 Amer-Yaiha et al.의 구조적 변형처럼 XPath 질의 내의 질의 노드들의 순서를 바꾸거나, 생략, XPath 경로 축 (예로, '///', '/', '*')의 변형을 고려하는 것, Chen et al.의 방법처럼 같은 의미의 다른 표현(즉 'buyer'에 대하여 'bidder'로 표현될 수 있음)을 위하여 온톨로지 맵핑 기술을 사용하는 것과 같이 이론적으로는 유사하지만, 이러한 이론적 개념의 보다 실질적인 구현 방법을 소개하고자한다. 제안 방법은 먼저 1) 방대한 수의 XPath 질의 노드의 번역을 위하여 XML 구조 및 온톨로지 맵핑 정보를 관계형 데이타베이스로 저장하며 질의 번역도 시스템에 의하여 자동 생성된 SQL문으로 수행된다. 2) 방대한 수의 XML 문서의 온톨로지 맵핑을 보다 단순화시킨다. 3) XPath 질의의 효율적 스키마 매칭을 위하여 기존 XML 트리 레이블링 기법[8,9]을 활용한다. 6.4절의 Chen et al. 방법과의 비교 실험에 있어서는 이러한 구현 방법이 매우 효율적이라는 것을 보여 주었다.

본 논문의 구성은 다음과 같다. 먼저 2장에서는 연구 동기 및 XPath 질의에 대한 몇 가지 가정을 소개한다. 3장에서는 기반 기술로 사용된 온톨로지 맵핑에 대하여 소개하며, 4장과 5장에서는 XML 트리 레이블링을 활용한 XPath 질의 번역 구현을 소개한다. 6장에서는 다양한 실험 결과를 제시하며, 7장에서는 기존 XPath 질의 번역 방법과의 차이점을 언급한다. 8장에서는 본 연구의 결론을 맺는다.

2. 연구 동기 및 가정

그림 1(a)(b)(c)의 XML 문서들에 대해 다음 XPath 질의를 고려하자.

'//buyer/person//name'

비록 위의 XPath 경로 표현식은 사이트 B로부터 출판된 XML 문서와 정확히 매칭되지만, 사실 사이트 A와 C 또한 의미적으로 같은 질의 결과를 포함하고 있다. 즉, 어휘 'buyer'와 'to' 사이, 어휘 'name'과 'bname' 사이, 그리고 어휘 'name'과 'bidder_name' 사이에 동의어 (synonym) 관계가 존재한다. 따라서 사이트 A와 C에 대하여 다음 번역된 질의 '//to/bname', '//bidder_name'을 고려할 수 있다. 이러한 의미적 질의 번역을 위한 다음의 일반적인 프레임워크를 생각할 수 있다.

- 전역(global) 질의를 지역(local) 질의들로 번역하는 방법(G2L): 본 방법에서는 이질적인 지역 데이타 스키마들 사이에서 자주 공유되는 개념들을 가지고 전역 스키마를 새로이 구성한다. 그리고 전역 스키마의 비주얼(visual)한 데이타 가이드를 사용자에게 제시한다. 사용자는 이러한 데이타 가이드에 기반하여 전역 XPath 질의를 작성한다. 작성된 XPath 질의는 지역 스키마에 준하는 지역 XPath 질의들로 번역된다.

  하지만 이러한 G2L 프레임워크는 다음과 같은 단점을 갖는다.

  (1) 우선 사용자는 자신에게 익숙한 지역 어휘 (vocabulary)로 이루어진 지역 XPath 질의를 등록할 기회를 잃어버린다. 예로, 어떤 XML 필터링 시스템을 통하여 사이트 A의 XML 문서들을 자주 구독하였던 사용자는 사이트 A의 XML 태그(즉 어휘)에 매우 친숙할 것이다: '//to/bname'. 하지만 사용자는 그림 1(d)의 전역 스키마의 비주얼 데이타 가이드를 참조하여 '//buyer/name'만을 작성하여야 한다. 이러한 방법은 또한 전역 스키마의 크기가 커질 경우 사용자의 비주얼 데이타 가이드 참조 인터페이스가 복잡해질 문제점을 갖는다.

  (2) 전역 스키마에 포함되지 않은 자주 공유되지 않는 지역 개념들에 대한 질의 요구 사항이 존재할 수 있다. 예로, 그림 1(c)의 XML 태그 'auction_news'는 모든 지역 스키마들에서 공유되는 공통 개념이 아니기 때문에 그림 1(d)의 전역 스키마에 정의되지 않았다. 따라

```
<closed_transaction>
 <from>
  <sname>Youngsoo</sname>
  <email>Youngsoo@samsung.com</email>
 </from>
 <to>
  <bname>Jaehoon</bname>
  <email>Jaehoon@gmail.com</email>
 </to>
 <items>
  ::          ::          ::
</closed_transaction>
```

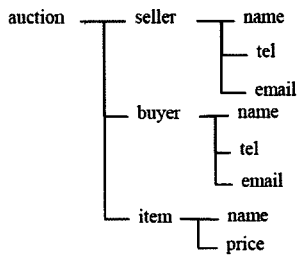
(a) From site *A*

```
<auctions>
 <auction>
  <seller>
   <person>
    <name>Youngsoo</name>
    <email>Youngsoo@samsung.com</email>
   </person>
  </seller>
  <buyer>
   <person>
    <name>Jaehoon</name>
    <email>Jaehoon@gmail.com</email>
    <phone>02-333-4444</phone>
   </person>
  </buyer>
  <item>
      <name>toy</name>
      <price>1000</price>
  </item>
  <item>
      <name>pencil</name>
      <price>1000</price>
  </item>
  ::          ::          ::
</auctions>
```

(b) From site *B*

```
<order_information>
 <seller_name>Youngsoo</seller_name>
 <seller_contacts>
  <email>Youngsoo@samsung.com</email>
 </seller_contacts>
 <bidder_name>Jaehoon</bidder_name>
 <bidder_contacts>
  <email>Jaehoon@gmail.com</email>
  <tel>02-333-4444</tel>
 </bidder_contacts>
 <items>
  <item>
   <item_name>toy</item_name>
   <value>1000</value>
  </item>
  <item>
    <item_name>pencil</item_name>
    <value>1000</value>
  </item>
 </items>
     ::          ::          ::
 <auction_news>
   <description>a bargain sale</description>
   <description>a special discount
   </description>
          ::          ::          ::
</order_information>
```

(c) From site *C*



(d) 전역 스키마의 비주얼 데이터가이드

그림 1 경매 관련 샘플 XML 문서 스트림

서, 비록 사용자는 사이트 C에 친숙하고 관심이 많지만, 지역 XPath 질의 '//auction_news/description'을 필터링 시스템에 등록할 수 없다.

본 연구에서는 의미적 XML 필터링을 위한 L2L의 새로운 프레임워크를 고려한다.

- 지역(local) 질의를 지역(local) 질의들로 번역하는 방법(L2L): 본 방법에서는 어떤 비주얼한 전역 데이타 가이드 없이, 단지 사용자 자신의 지식과 이전 경험에 의존하여 XPath 질의를 작성한다. 예로, 사용자가 그림 1(b)의 구독한 XML 문서에 매우 익숙하다고 하자. 그러면 사용자는 단지 이전에 친숙했던 XML 태그들과 구조를 생각하며 사이트 B에 부합한 지역 질의를 작성할 수 있을 것이다. 작성된 질의는 시스템 내부적으로 사이트 A와 C에 적합한 질의로 번역된다. 하지만 이러한 방식의 중요한 문제점은 사용자의 기억에 의해 작성된 질의가 실제 스키마와 불일치할 수 있다는 것이다. 즉, 사용자의 잘못된 기억과 다수의 지역 스키마 정보의 혼재함으로 인해 불일치되는 (inconsistent) 지역 XPath 질의를 작성할 수 있다: 예로, '//auction/buyer/bname', '//person/buyer/name', '//to//bidder_name' 등. 이러한 문제는 질의와 데이타 스키마 불일치 문제라고 불리어 진다[5]. 따라서 본 프레임워크에서는 부분적으로 불일치되는 지역 XPath 질의를 올바르게 번역할 수 있는 기능이 요구된다. 본 연구에서 고안한 질의 번역 기법은 이러한 질의와 데이타 스키마 불일치 문제를 다룬다.

사실 이러한 의미적 XPath 질의 번역 문제는, 만약 이질적 데이타 소스의 수가 많지 않다면, 중요치 않은 문제일 것이다. 하지만 XML 필터링 시스템은 데이타 소스의 수가 매우 많으며, 또한 데이타 소스들로부터 출판되는 XML 문서가 빈번히 생성되며, 갱신되고, 삭제되는 매우 다이내믹한 환경을 갖는다. 또한 질의 번역이 매우 빠르게 이루어 져야 한다. 본 연구의 목적은 기존의 의미적 XPath 질의 번역 방법들과 비교하여 다이내

믹 XML 필터링 환경에 보다 실질적인 L2L 질의 번역 방법을 개발하는데 있다.

2.1 XPath 질의에 대한 가정

여기서는 SemFilter 시스템에서 사용 가능한 XPath 질의 유형에 대하여 언급한다.

(1) 비록 XPath 질의[10]에는 다양한 탐색 축(axis)이 정의되어 있지만, 본 연구에서는 L2L 방식에서 사용자들은 '/', '//', '*'의 보다 단순한 탐색 축만을 사용하여 XPath 질의를 쉽게 작성하는 경향이 있음을 가정한다. 따라서 $(n_1, n_2, ..., n_i)^{/, //, *}$의 XPath 경로 표현식만을 고려한다. 여기서, $n_1, n_2, ..., n_i$는 질의 경로 노드(query path node)이며, 예로 '//buyer/person//name'는 질의 경로 노드 (buyer, person, name)과 {/, //}의 탐색 축으로 이루어져 있다.

(2) XPath 트리 질의에 관하여서는, YFilter[2]에서 제안된 질의 경로 분해(query decomposition) 기법을 이용한 필터링 방법을 가정한다. 예로, 질의 '$/n_1[n_2]//n_3[n_4/n_5]/n_6$'는 하나의 주경로(main path) '$/n_1//n_3/n_6$'와 두 개의 프레디킷 경로(predicate path) '$/n_1/n_2$', '$/n_1//n_3/n_4/n_5$'로 분해될 수 있다. 세 개의 단일 경로 질의가 XML 필터링 시스템에 각각 등록되고, 만약 유입되는 XML 문서에 대해 세 개의 단일 경로 질의가 모두 매치될 때, 그 문서는 필터링된다. 따라서 본 연구에서는 단일 경로 질의의 의미적 질의 번역에 초점을 맞춘다.

(3) L2L에서 등록되는 XPath 질의의 주요 유형은 주로 부분 매칭 경로 질의(partial matching path query) [4]임을 가정한다. 이러한 질의는 조상 후손 경로 탐색 축(descendant axis) '//'으로 시작하며 또한 중간 경로 표현식에 '//'을 많이 포함하는 질의이다: 예로, '//buyer //name', '//name', '//bidder_name' 등.

3. 다수의 지역 스키마를 고려한 온톨로지 맵핑의 단순화

본 연구의 질의 번역 기법은 먼저 온톨로지 맵핑 기술에 기반한다. 각 지역 XML 스키마의 엘리먼트와 속성 이름이 온톨로지에서의 어휘(vocabulary)로 간주되며, 지역(local) 어휘들은 필터링 시스템에서 관리되는 전역(global) 어휘들과 매칭된다. 이러한 온톨로지 관점에서의 어휘 매칭은 XML 스키마의 복잡한 태그 구조를 무시하도록 한다. 즉, 관리자는 XML 스키마에 나타나는 어휘들만을 고려하여, 스키마 구조에 상관없이 전역 어휘에 매칭한다. 이것은 다이내믹하며 실시간적인 XML 필터링 환경에서 스키마 매칭 작업을 신속히 하기 위함이다.

지금까지 더욱 정확한 온톨로지 맵핑을 위하여 온톨로지 개념들 사이에 발생할 수 있는 다양한 의미적 관계들이 연구되었다. 예로, 표준 웹 온톨로지 언어인 RDF[11] 및 OWL[12]에서 그러한 다양한 관계를 살펴볼 수 있다(예로, equivalentClass, equivalentProperty, sameAs, unionOf, complementOf, intersectionOf 등). 하지만 본 연구에서는 동의어 (synonym)와 상/하위의 (hypernym/hyponym)의 관계만을 고려할 것이다. 이것은 OBSERVER [13]에 언급된 것처럼, 다수의 이질적인 스키마들 사이에서의 어휘 관계를 정확하고 구체적으로 정의하는 것은 매우 복잡하고 어려운 일이기 때문이다. 더욱이 다수의 데이타 소스로부터 끊임없이 XML 문서가 출판되고, 갱신되고, 삭제되는 다이내믹한 XML 필터링 환경에서는 이러한 작업이 더더욱 어려울 것이다. 따라서 동의어와 상/하위어 관계에 기반한 단순한 온톨로지 맵핑 기술을 활용한다.

정의 1 (동의어, 상/하위어 관계).

- 동의어 관계는 각기 다른 온톨로지의 두 어휘가 어휘 그 자체에서 같은 의미를 가지는 것을 나타낸다. 예로, 그림 1(a)의 지역 어휘 'to'는 그림 2(a)에서의 전역 어휘 'buyer'와, 비록 다른 XML 스키마 구조와 다른 온톨로지 계층 구조에 속하지만, 같은 의미를 가진다.
- 비슷하게, 상/하위어 관계는 어떤 어휘가 다른 어휘보다 어휘 그 자체에서 덜 일반적일 경우 성립한다. 예로, 그림 1(a)의 지역 어휘 'closed_transaction'은 그림 2(a)에서의 전역 어휘 'auction'보다 덜 일반적인 개념이다.

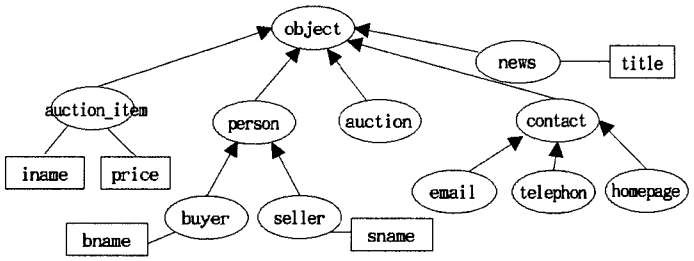
본 연구에서는 또한 다이내믹한 환경에서의 단순 온톨로지 맵핑을 위하여, 일대일 맵핑과 복수 맵핑을 고려한다.

정의 2 (일대일 맵핑).

- 일대일(one-to-one) 맵핑은 어떤 단일 어휘가 다른 단일 어휘와 일대일의 개념적 관계를 갖는 것을 표현한다. 반대로, 다대다(one-to-many) 맵핑은 어떤 단일 어휘가 다수의 단일 어휘들로 표현되는 개념과 일대일의 개념적 관계를 갖는 것을 표현한다. 예로, 그림 1(c)의 지역 어휘 'bidder_contacts'는 그림 2(a)의 두 전역 어휘 'email', 'telephone'의 합집합과 동의어 관계를 갖는다: bidder_contacts ≡ (email ∪ telephone).

정의 3 (복수 맵핑).

- 단수 맵핑은 어떤 어휘의 맵핑이 한번만 이루어지는 것을 의미한다. 반대로, 복수 맵핑은 어떤 어휘의 맵핑이 여러 번 허용되는 것을 의미한다. 예로, 그림 1(a)의 지역 어휘 'to'는 그림 2(a)의 전역 어휘



* 위의 그림에서 원은 클래스이며, 사각형은 속성이며,
화살표는 subclassof, subpropertyof 관계를 나타낸다.

(a) 계층적 그래프로 표현된 전역 온톨로지

id	gterm	lterm_path	lterm
1	buyer	/auctions/auction/buyer	buyer
2	buyer	/closed_transaction/to	to
3	bname	/auctions/auction/buyer/person/name	name
4	bname	/closed_transaction/to/bname	bname
5	sname	/auctions/auction/seller/person/name	name
6	sname	/closed_transaction/from/sname	sname
7	sname	/order_information/seller_name	seller_name
8	bname	/order_information/bidder_name	bidder_name
9	auction	/auctions/auction	auction
10	auction	/closed_transaction	closed_transaction
11	auction	/order_information	order_information
12	news	/oder_information/auction_news	auction_news
13	email	/auctions/auction/buyer/person/email	email
14	person	/auctions/auction/buyer/person	person
15	person	/auctions/auction/seller/person	person
...	...	...	...

(b) 단순화된 온톨로지 맵핑

그림 2 IMT 설계

'buyer'와도 맵핑되고, 동시에 전역 어휘 'person'과도 맵핑될 수 있다.

4. 질의 경로 매칭에 의한 비분절적 XPath 질의 번역 구현

먼저 L2L XPath 질의 번역 기법의 하나로 주어진 질의의 전체 질의 경로 단위로 번역을 수행하는 비분절적(inariculate) XPath 질의 번역 방법을 소개한다. 비분절적 방법은 6장의 실험에서 자세히 소개하겠지만, 매우 신속한 질의 번역 성능을 가진다. 하지만 L2L 질의 번역 방법이 제공해야 될 질의와 데이타 스키마 불일치 문제를 해결하지는 못한다.

4.1 온톨로지 맵핑 테이블(IMT)

IMT(Interontology Mapping Table)는 앞장에서 소개한 지역 온톨로지와 전역 온톨로지 사이에서의 동의어 및 상/하위어 관계를 저장한다. 본 연구에서 IMT는 그림 2(b)와 같이 관계형 데이타베이스의 테이블로 구현되었다. IMT의 컬럼 gterm은 그림 2(a)의 전역 온톨로지의 표준 어휘들을 저장한다. 컬럼 lterm은 전역 어휘와 매칭되는 각 지역 스키마의 지역 어휘들을 저장한다. 컬럼 lterm_path는 지역 매칭 어휘의 전체 노드 경로 정보를 저장한다. 예로, 그림 1(a)의 지역 어휘 'to'(= '/closed_transaction/to')는 전역 어휘 'buyer'의 동의어로 저장되었다. 또한 그림 1(c)의 지역 어휘 'auction_news'('/order_information/auction_news')는 전역 어휘 'news'의 하위어로 IMT에 저장되었다. 여기서 IMT는 동의어 및 상/하위어 관계를 별도로 식별하지는 않는다. 이는 뒤에서 소개될 L2L 질의 번역을 보다 단순하게 처리하여 질의 번역 성능을 높이기 위함이다.

본 연구에서 IMT의 전역 표준 어휘는 시스템 관리자

에 의하여 점진적으로 추가될 수 있음을 가정한다. 예로, 새로운 XML 문서가 출판되는 것을 가정하자. 시스템 관리자는 새로운 XML 문서로부터의 지역 어휘와 매치될 전역 어휘를 검색한다. 만약 새로운 어휘와 매치되는 표준 전역 어휘가 존재하지 않는다면, 새로운 지역 어휘를 전역 어휘로 저장하고 이후부터는 이에 매칭한다. 사실 방대한 양의 지역 어휘와 전역 어휘를 매칭하는 작업은 매우 힘든 일이 될 수 있다. 하지만 이러한 문제는 본 논문의 주제를 넘어서는 것이며, 본 논문에서는 지역 어휘와 전역 어휘의 효율적 매칭을 위하여, GLUE[14]에서 제안된 것 같은 반-자동적인(semi-automatic) 온톨로지 맵핑 도구를 사용할 수 있음을 가정한다.

4.2 스키마 구조 테이블(SST)

SST(Schema Structure Table)는 각 지역 스키마의 구조 정보를 저장한다. 이러한 정보는 다음의 경우에 활용된다. 앞서 소개한 온톨로지 맵핑 정보에 기반하여 XPath 질의를 번역할 경우, 다수의 번역 가능한 질의들을 얻을 수 있다. 이러한 질의중 가장 근사하게 번역된 하나의 질의를 선택하기 위하여 스키마 구조 정보가 활용된다.

본 연구에서는 스키마 구조 정보에 대한 경로 매칭을 효율적으로 평가하기 위하여, XML 트리 레이블링중의 하나인 Dietz 방법[8]을 활용한다. Dietz는 트리상의 두 노드 u, v가 주어질 경우, 만약 $(pre(v) < pre(u)) \wedge (post(v) > post(u))$ 이면 u는 v의 후손 노드임을 증명하였다: 여기서 $pre()$는 전위 순회(preorder traversal)에 의한 노드 레이블링 값이며, $post()$는 후위 순회(postorder traversal)에 의한 노드 레이블링 값이다. 또한 같은 부모 노드를 공유하는 형제 노드를 구별하기 위하여, 각 노드 v는 부모 노드의 $pre(v)$ 값을 저장하고 같은 $pre(v)$ 값을 가지는 노드들을 형제 노드로 판별할 수 있다. 그림 3(a)는 그림 1(b)의 지역 스키마에 대하여 pre/post 트리 순회에 의하여 할당된 레이블링 값을 보여준다. 할당된 레이블링 값은 그림 3(b)의 관계형 테이블에 저장된다. 컬럼 pre, post, par, 그리고 site는 각각 전위 순회 값, 후위 순회 값, 부모 노드의 전위 순회 값, 스키마 식별자 값을 저장한다. 특별히 pid 컬럼은 해당 지역 어휘의 정보를 저장하고 있는 IMT에서의 id 값을 참조한다. 이러한 id/pid 참조 관계에 의하여, 어떤 등록된 XPath 질의와 매칭되는 전역 어휘를 효율적으로 발견할 수 있다. 예로, XPath 질의 '//auction/buyer//name'를 위한 IMT와 SST에 대한 다음 SQL문을 보자.

Q1: select *lterm*, *pre*, *post*, *par*, *site* from *SST* where *lterm* = '*auction*'

Q2: select *lterm*, *pre*, *post*, *par*, *site* from *SST* where *lterm* = '*buyer*'

Q3: select *lterm*, *pre*, *post*, *par*, site, *pid* from *SST* where *lterm* = '*name*'

Q4: select *pid* from *Q1*, *Q2*, *Q3*
where *Q1.pre* = *Q2.par* and *Q1.site* = *Q2.site* and
Q2.pre < *Q3.pre* and
Q2.post > *Q3.post* and *Q2.site* = *Q3.site*

Q5: select distinct *gterm* from *IMT*, *Q4* where *IMT.id* = *Q4.pid*

SST로 부터의 질의 Q4에 대한 pid 값은 3이고, Q5에 의하여 IMT로부터 해당 지역 어휘의 전체 경로 정보는 '/auctions/auction/buyer/person/name'이며 해당 전역 어휘는 'bname'임을 알 수 있다. 따라서 IMT와 SST를 이용한 비분절적 질의 번역 방법은 다음과 같이 쉽게 이루어 질 수 있다. 어떤 주어진 XPath 질의에 대하여 SST를 이용하여 gterm 값을 구하고, IMT에서 같은 gterm 값을 가지는 모든 lterm_path 값들이 각 사이트에 대한 번역 질의가 된다. 예로, 'bname'을 갖는 lterm_path는 사이트 A에 대하여 '/closed_transaction/to/bname', 사이트 C에 대하여 '/order_information/bidder_name'이다.

위에서 제안하는 질의 번역 방법은 관계형 데이타베이스 시스템을 이용하여 구현되었다. 그 이유는 1) 제안 방법이 IMT와 SST의 조인 질의와 같은 테이블 연산을 많이 포함하기 때문이다. 2) 또한 XML 필터링 시스템에서 다수의 이질적인 지역 스키마로 인하여, IMT와 SST의 크기는 매우 커지기 때문이다. 3) 관계형 데이타베이스 시스템은 XML 필터링에서의 IMT와 SST의 다이내믹한 관리를 효율적으로 지원할 수 있다.

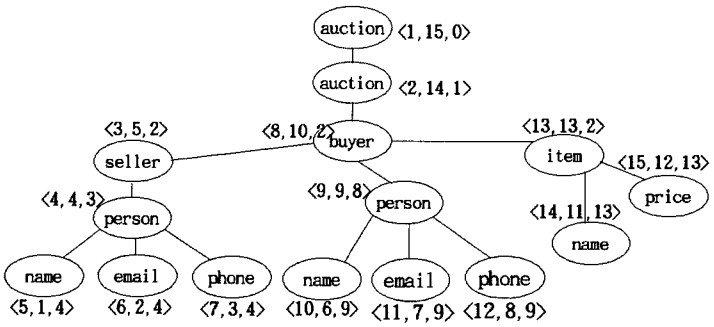
4.3 번역된 질의의 NFA 머신 등록

의미적 XPath 질의 번역에 의해 번역된 질의는 원래의 질의와 함께 PosFilter[4]의 NFA와 같은 오토마타에 최종적으로 등록된다. 그림 4(b)의 NFA(Nondeterministic Finite Automaton)는 그림 4(a)의 등록 질의에 대하여 원래 질의 및 번역 질의의 오토마타 구성을 보여준다.

5. 어휘 매칭에 의한 분절적 XPath 질의 번역 구현

비분절적 XPath 질의 번역과 비교하여, 제안하는 분절적(articulate) XPath 질의 번역 방식은 전체 질의 경로 단위가 아닌 주어진 XPath 질의의 질의 노드 단위의 번역을 수행한다. 이러한 분절적 방식은 L2L 질의 번역에서의 질의와 데이타 스키마 불일치 문제를 지원한다.

질의와 데이타 스키마 불일치 문제의 효율적 해결을 위해 제안하는 분절적 XPath 질의 번역 방법은 다음의



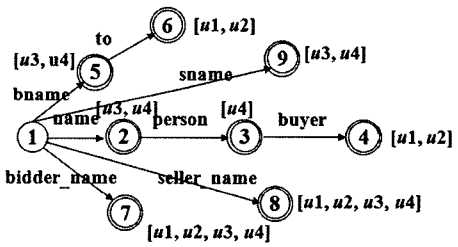
(a) <pre, post, par>

lterm	pre	post	par	site	pid
buyer	8	10	2	B	1
to	5	6	1	A	2
name	5	1	4	B	5
name	10	6	9	B	3
auction	2	14	1	A	9
bname	6	4	5	A	4
sname	3	1	2	A	6
closed_transaction	1	11	0	A	10
bidder_name	6	5	1	C	8
seller_name	2	1	1	C	7
person	4	4	3	B	15
person	9	9	8	B	15
order_information	1	15	0	C	11
auction_news	14	14	1	C	12
...	...	...	...	...	...

(b) 지역 스키마 구조 정보의 효율적 저장

그림 3 SST 설계

[$u1$] //buyer/person//name
[$u2$] //to/*/bname
[$u3$] //seller_name
[$u4$] //person/name



(a) 등록 XPath 질의

(b) PosFilter NFA

그림 4 번역된 질의의 NFA 필터링 머신 등록

주요 가정에 기반한다. 이러한 가정들은 가장 근사하게 번역된 질의를 찾기 위한 탐색 공간을 현저히 줄이기 위함이다. 본 연구에서는 이러한 가정들이 실질적이며 수용할 만한 가정이라고 생각한다.

가정 1. 사용자가 어떤 지역 스키마에 대한 XPath 질의를 작성할 경우, 그 사용자는 질의에 사용되는 지역 어휘들에 대하여 주로 동의어, 혹은 상/하위어에 대하여 혼돈할 가능성이 높다. 예로, 그림 1(b)의 '//auction/buyer//name'에 대하여, 'buyer'를 그림 1(a)의 지역 어휘 'to'와 혼돈하여 '//auction/to//name'을 작성하는 경우이다.

가정 2. 주어진 XPath 질의의 모든 질의 노드들은 마

지막 목표 질의 노드(정의 4를 참조)의 의미를 분명히 하기위해 사용되어지며, 따라서 각 질의 노드의 IMT상의 매칭 어휘들은 각 지역 스키마에서 유사한 조상-후손 노드 순서를 가지게 된다. 예로, 질의 '//auction/buyer//name'에서 'auction'과 'buyer'는 목표 질의 노드 'name'이 보다 구체적인 경매에서의 구매자의 이름임을 나타내기 위해 사용된 것이다. 따라서 IMT 상의 매칭 어휘 {'auction', 'closed_transaction', 'order_information'}은 {'buyer', 'to'}와 {'name', 'bname'}의 조상 노드들이고, 매칭 어휘 {'buyer', 'to'}는 {'name', 'bname'}의 조상 노드들이다. 만약 어떤 주어진 XPath 질의에 대하여 다수의 번역 가능한 질의들이 존재할 경우, 가장 근사하게 번역된 질의는 모든 질의 노드의 매칭 어휘를 가지며, 또한 그러한 매칭 어휘들이 원래의 질의의 조상-후손 관계를 최대한 만족하는 것이다.

정의 4 (목표 질의 노드).

- 목표 질의 노드(target query node)는 어떤 주어진 XPath 질의의 궁극적 질의 결과가 되는 노드이다. 따라서 번역된 질의에는 목표 질의 노드의 매칭 어휘들이 반드시 포함되어야 한다. 예로, '//auction/buyer//name'의 번역 질의들에는 목표 질의 노드인 'name'의 매칭 어휘들이 반드시 포함되어야 한다.

가정 3. 사용자들은 적어도 조상-후손 관계의 질의 노드 순서를 정확히 기억할 가능성이 높다. 왜냐하면, 조상 노드들은 후손 노드들보다 더 일반적인 개념의 어휘이기 때문이다. 예로, 사용자가 지역 어휘 'auction', 'buyer', 'name'을 기억할 경우, 쉽게 '//auction/buyer//name'의 질의를 작성한다. 이것은 온톨로지에서의 어휘 개념이 'auction' > 'buyer' > 'name'의 순서를 가지기 때문이다.

가정 4. 또한 사용자들은 기억하기 어려운 부모-자식 노드 관계('/' 탐색 축)를 정확히 기억할 수 있다. 따라서 번역 가능한 질의 중 원래와 질의와 비교하여 '/' 탐색 축을 가장 많이 만족시키는 질의가 가장 근사하게 번역된 질의이다.

지금부터는 위의 주요한 가정에 기반한 분절적 XPath 질의 번역 방법을 소개한다. 분절적 방법 또한 IMT와 SST를 사용한다.

(단계 1) 어떤 주어진 단일 경로 XPath 질의 (n_1, n_2, ..., n_i)$^{/,\ //,\ *}$의 질의 노드 n_1, n_2, ..., n_i에 대하여 IMT로부터의 매칭 어휘를 계산한다. 이것은 가정 1을 반영한다.

예로, '//auction/buyer//name'의 질의 노드 'auction', 'buyer', 'name'에 대하여 다음 SQL문이 수행된다.

*Q*1: select distinct *gterm* from *IMT* where *lterm* = '*auction*'

*Q*2: select *lterm* from *IMT*, *Q*1 where *IMT.gterm* = *Q*1.*gterm*

***Q*3**: select *lterm*, *pre*, *post*, *par*, *site* from *SST*, *Q*2
where *SST.lterm* = *Q*2.*lterm* /* μ('*auction*') */

*Q*4: select distinct *gterm* from *IMT* where *lterm* = '*buyer*'

*Q*5: select *lterm* from *IMT*, *Q*4 where *IMT.gterm* = *Q*4.*gterm*

***Q*6**: select *lterm*, *pre*, *post*, *par*, *site* from *SST*, *Q*5
where *SST.lterm* = *Q*5.*lterm* /* μ('*buyer*') */

*Q*7: select distinct *gterm* from *IMT* where *lterm* = '*name*'

*Q*8: select *lterm* from *IMT*, *Q*7 where *IMT.gterm* = *Q*7.*gterm*

***Q*9**: select *lterm*, *pre*, *post*, *par*, *site*, *pid* from *SST*, *Q*8
where *SST.lterm* = *Q*8.*lterm* /* μ('*name*') */

(단계 2) 단일 경로 질의 상의 모든 탐색 축 ('/', '//', '*')을 조상-후손 탐색 축 '//'으로 대체한다. 예로, '//auction/buyer//name' → '//auction//buyer//name'.

(단계 3) 단일 경로 질의 노드 순서의 역순으로 매칭 어휘들 $\mu(n_i)$, $\mu(n_{i-1})$, ..., $\mu(n_1)$의 왼쪽 외부 조인 (Left Outer Join) 연산을 수행하는 SQL문을 구성한다. 이러한 과정은 가정 2와 3을 반영한다.

가정 2와 3을 효율적으로 평가하기 위하여, 본 단계에서는 왼쪽 외부 조인 연산을 사용한다. 예로, '//auction//buyer//name'을 고려하자. 정의 4에 따라 목표 질의 노드는 번역 질의에 항상 포함되어야 하기 때문에, μ('name')과 μ('buyer') 사이의 왼쪽 외부 조인 연산을 먼저 고려한다. 그림 5는 이러한 연산 과정을 보여준다. 만약 μ('name')에 해당되지 않는 μ('buyer')가 있을 경우, 이는 null 값을 갖는다. 때문에 아래의 Q11 SQL 문장에서 pre, post, par 컬럼이 null 값을 갖는 경우 이는 μ('name')에서의 pre, post, par 값으로 대체된다.

*Q*10: select *Q9.lterm* as *q9_lterm*, *Q9.pre* as *q9_pre*, *Q9.post* as *q9_post*, *Q9.par* as *q9_par*, *Q9.site* as *q9_site*, *Q9.pid* as *q9_pid*, *Q6.lterm* as *q6_lterm*, *Q6.pre* as *q6_pre*, *Q6.post* as *q6_post*, *Q6.par* as *q6_par*
from *Q9* left outer join *Q6*
on *Q9.pre* > *Q6.pre* and *Q9.post* < *Q6.post* and *Q9.site* = *Q6.site*

*Q*11: select *q9_lterm*, *q9_pre*, *q9_post*, *q9_par*, *q9_site*, *q9_pid*, *q6_lterm*,

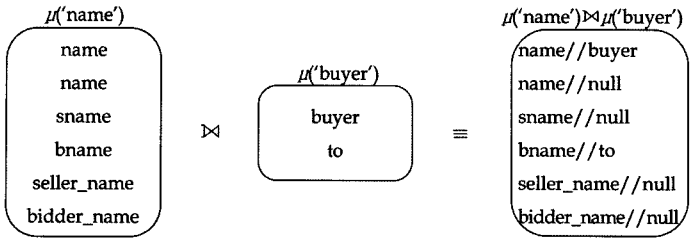


그림 5 LEFT OUTER JOIN 연산 활용

case when *q6_pre* is null then *q9_pre* else *q6_pre* end as '*q6_pre*',
case when *q6_post* is null then *q9_post* else *q6_post* end as '*q6_post*',
case when *q6_par* is null then *q9_par* else *q6_par* end as '*q6_par*' from *Q*10

계속해서 아래와 같이 다음 역순에 의해 {μ('name') ⋈ μ('buyer')}와 μ('auction')의 왼쪽 외부 조인 연산이 수행된다.

*Q*12: select *q9_lterm*, *q9_par*, *q9_pid*, *q6_lterm*, *q6_pre*, *q6_par*, *Q3.lterm*, *Q3.pre*
from *Q*11 left outer join *Q*3
on *Q*11.*q6_pre* > *Q3.pre* and *Q*11.*q6_post* < *Q3.post* and *Q*11.*q9_site* = *Q3.site*

비록 이러한 매칭 어휘들의 왼쪽 외부 조인 연산을 사용하는 것이 $\mu(n_i)$, $\mu(n_{i-1})$, ..., $\mu(n_1)$의 $(i-1)$ 개의 조인 연산 비용을 초래할 수 있지만, 2.1절에서 가정한 의미적 XML 필터링에서의 부분 매칭 경로 질의의 사용에 근거하여 경로 질의의 깊이 i가 작음을 기대할 수 있다.

(단계 4) 단계 3에서 구성된 SQL문을 수행하며, 질의 결과중 가정 2와 4에 근거하여 가장 근사하게 번역될 수 있는 레코드를 선택한다. 가정 2와 4에 근거한 선택 평가 기준은 다음과 같다.

Score := #{lterm} + #{'/'}

여기서 #{lterm}은 널 값을 갖지 않는 lterm 컬럼의 수이며, #{'/'}는 원래 질의의 부모-자식 탐색 축을 만족하는 pre, par 컬럼의 수이다. SST에서 par 컬럼은 부모 노드의 전위 순회 노드 값을 가지고 있으므로, pre is not null ∧ par is not null ∧ pre = par 이면 탐색 축 '/'를 만족한다. 예로, 그림 6의 SQL 질의 결과 중 두 번째 다섯 번째 레코드가 가장 최선의 번역 가능 질의로 선택된다. 왜냐하면, q9_lterm, q6_lterm, lterm 컬럼이 널 값을 갖지 않기 때문에 #{lterm}의 값은 3이다. 또한 원래 질의의 부모-자손 축 'auction/buyer'에 대하여 q6_par = pre를 만족하기 때문에 #{'/'}의 값은 1이다. 따라서 가장 높은 점수 4 (= 3+1)을 갖는다.

(단계 5) 비분절적 방법에서처럼, 가장 최선의 번역으로 선택된 레코드의 pid 값을 이용하여 IMT에서의 같은 gterm 값을 갖는 lterm_path 값을 선택한다. 예로, 그림 6에서 q9_pid의 값 3, 4와 일치하는 geterm 값은 'bname'이며, 각 사이트에 대하여 번역된 질의는 lterm_path 'auctions/auction/buyer/person/name', '/closed_transaction/to/bname', '/order_information/bidder_name'이다.

	q9_lterm	q9_par	q9_pid	q6_lterm	q6_pre	q6_par	lterm	pre
1	name	4	5	NULL	5	4	auction	2
2	name	9	3	buyer	8	2	auction	2
3	name	13	20	NULL	14	13	auction	2
4	sname	2	6	NULL	3	2	closed_transaction	1
5	bname	5	4	to	5	1	closed_transaction	1
6	bidder_name	1	8	NULL	6	1	order_information	1
7	seller_name	1	7	NULL	2	1	order_information	1
8	iname	9	22	NULL	10	9	closed_transaction	1
9	item_name	11	23	NULL	12	11	order_information	1

그림 6 가장 근사하게 번역될 질의의 선택

제안된 분절적 XPath 질의 번역은 사용자의 잘못된 기억 혹은 지식에 근거한 부분적으로 불일치되는 XPath 질의를 옳게 번역할 수 있는 기능을 제공한다. 예로, 다음의 경우들을 살펴보자.

• 다른 지역 스키마의 어휘를 혼돈하는 경우:

등록된 XPath 프로파일 '//buyer/person//bname'을 고려하자. 그림 1(b)의 사이트 B에 대하여 사실은 어휘 'name'이 맞다. 하지만 이러한 오류는 단계 1에 의하여 'bname'이 'name'으로 대치될 수 있다.

• 다른 지역 스키마의 스키마 구조를 혼돈하는 경우:

- 등록된 XPath 프로파일 '//buyer/name'을 고려하자. 사실은 그림 1(b)에서 'buyer//name'의 조상-후손 탐색 축이 맞는 것이다. 이것은 단계 2에 의하여 '/' 탐색 축이 '//' 탐색 축으로 대치된다.
- 등록된 XPath 프로파일 '//person/buyer//name을 고려하자. 사이트 B에 대하여 질의 노드 경로 'buyer/person' 맞다. 사실 이러한 오류는 가정 3에 위배되는 것이며, 본 연구의 질의 번역 방법에서는 고려되

지 않은 것이다. 하지만 제안된 질의 번역 방법은 단계 3, 4에 의하여 부분적인 질의 번역을 수행할 수 있다. 즉, 단계 3, 4에 의하여 가장 최선의 번역 가능한 질의 '//buyer//name'과 '//person//name'이 구해진다. 단계 5에 의하여 부분적인 정보에 근거하여 번역된 질의 {'/auctions/auction/buyer/person/name', '/closed_transaction/to/bname', '/order_information/bidder_name', '/auctions/auction/seller/person/name', '/closed_transaction/from/sname', '/order_information/seller_name'}을 구할 수 있다.

6. 실험 및 분석

본 장에서는 제안된 L2L XPath 질의 번역 방법에 대한 몇 가지 실험 결과를 소개한다. 먼저 직관적으로 분절적 질의 번역 방법은 비분절적 질의 번역 방법보다 더 많은 질의 번역 시간을 요구함을 알 수 있다. 이는 더 많은 조인 연산을 수행하기 때문이다. 6.2절에서는 비분절적 질의 번역 방법과 분절적 질의 번역 방법의 질의 번역 수행 시간을 비교한다. 다음으로 6.3절에서는 이러한 비분절적 질의 번역 방법과 분절적 질의 번역 방법을 결합하여 운영하는 보다 효율적인 방법을 소개한다. 6.4절에서는 제안된 질의 번역 방법과 기존의 Chen이 제안한 질의 번역 방법과의 수행 시간을 비교한다.

6.1 실험 환경

표 1은 제안된 질의 번역 성능에 영향을 미칠 수 있는 주요한 실험 인자를 보여준다. 먼저 실험 인자 N과 S는 SST의 저장 크기를 결정한다. 예로, N = 100, S = 1,000 일 때, SST는 100,000 스키마 노드를 저장한다 (= $N * S$). 실험 인자 N, S, R, T는 IMT의 저장 크기를 결정한다. N = 100, S = 1,000, R = 90%, T =30% 일 때, IMT는 27,000 (= $N * T * S * R$)개의 매칭 어휘를 저장한다. 여기서, R은 전체 지역 스키마 수에 대한 온톨로지 맵핑 관계를 가지는 지역 스키마 수의 비율이다. 실험 인자 T는 하나의 지역 스키마 내에서 온톨로지 맵핑 관계를 가지는 스키마 노드의 점유율이다. 다른 중요한 실험 인자로 주어진 XPath 질의의 질의 깊이를 고려할 수 있다. 이것은 질의 깊이가 왼쪽 외부 조인의 수를 결정하기 때문이다. 2.1절에서의 가정 3에 근거하여 질의 깊이 5까지 실험하였다.

모든 실험은 2.66 GHz 펜티엄 IV CPU와 1GB의 메모리를 가진 윈도우 XP 컴퓨터에서 실험되었으며, 자바로 구현되었다. 사용된 관계형 DBMS는 오라클 10g이며 데이타베이스 버퍼 캐시 크기는 디폴트인 113MB를 그대로 사용하였다.

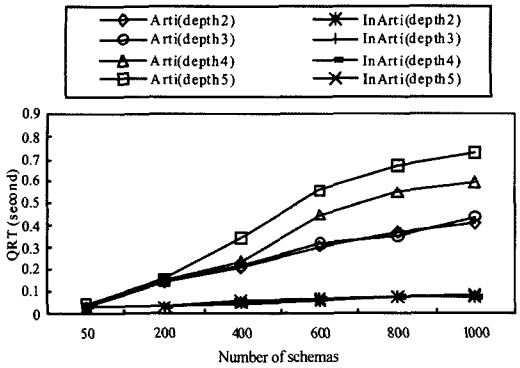
6.2 비분절적 방법 vs. 분절적 방법

그림 7(a) 그래프는 N = 100, R = 30%, T =30%일 때 S의 변화에 따른 각 질의의 질의 번역 시간 (QRT)을 보여준다. 그림에서 볼 수 있듯이, 분절적 방법이 분절적 방법에 비해 더욱 긴 질의 번역 시간을 가짐을 알 수 있다. 하지만 대체적으로 1초 이내이며, 이는 실질적으로 허용될 수 있는 질의 번역 시간이라고 생각한다. 지역 스키마의 수가 50개 이내일 때에는 0.1초 이내이다. 비분절적 방법은 빠른 L2L 질의 번역 시간을 보장할 수 있지만, 서두에서 언급한 질의와 스키마 불일치 문제를 해결할 수 없다.

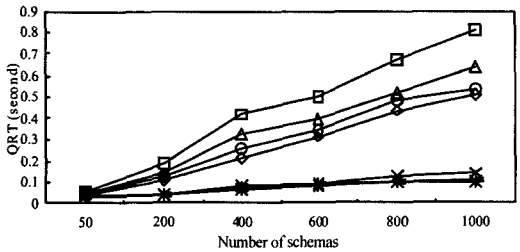
그림 7(b)는 R = 90%로 증가되었을 때의 질의 번역 시간을 보여준다. 각 질의에 대하여 약 0.1초의 증가분을 관찰할 수 있다. 각각의 그래프에서 질의 깊이에 따른 질의 번역 시간을 관찰할 수 있는데, 분절적 방법에서는 질의 깊이에 따라 중요한 차이가 있음을 관찰할 수 있다.

표 1 실험 인자

실험인자	값의 범위	설명
N	50, 100	하나의 지역 스키마에서의 태그 수
S	50, 200 400, 600 800, 1,000	이질적 지역 스키마의 수
R	30%, 50% 70%, 90%	= (a-b)/a×100 a: 이질적 지역 스키마의 수 b: 어떤 다른 지역 스키마와 온톨로지 맵핑 관계를 갖지 않는 지역 스키마의 수
T	30%, 60%	= (a-b)/a×100 a: 하나의 지역 스키마에서의 태그 수 b: 하나의 지역 스키마에서의 온톨로지 맵핑 관계를 갖지 않는 스키마 태그 수
Q	2, 3, 4, 5	XPath 질의의 질의 깊이 - 깊이 2: '//aaa//eee' - 깊이 3: '//aaa/kkk//ccc' - 깊이 4: '//aaa//sss/bbb/ddd' - 깊이 5: '//aaa/fff//ggg//ttt/hhh'



(a) N=100, R=30%, T=30%



(b) N=100, R=90%, T=30%

그림 7 비분절적 방법과 분절적 방법의 질의 번역 시간 비교

6.3 비분절적 방법과 분절적 방법의 조합에 의한 L2L 질의 번역기 구현

모든 등록된 XPath 질의가 불일치되는 질의는 아니기 때문에, 매번 분절적 질의 번역 방법을 수행할 필요는 없다. 따라서 그림 8과 같은 비분절적 방법과 분절적 방법의 조합에 의한 L2L 질의 번역기 구현을 생각할 수 있다. 즉, 어떤 XPath 질의가 등록될 경우 비분절적 방법에 의한 질의 번역을 수행한다. 만약 번역된 질의가 존재한다면, 이것은 불일치되는 질의가 아니다. 하지만 번역된 질의가 존재하지 않을 경우, 이것은 불일치의 질의이므로 분절적 질의 번역 방법을 수행한다.

그림 9의 그래프는 위와 같은 조합의 방법이 순수한

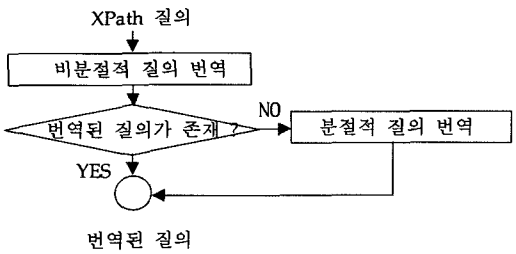


그림 8 분절적 방법과 비분절적 방법의 조합 운영

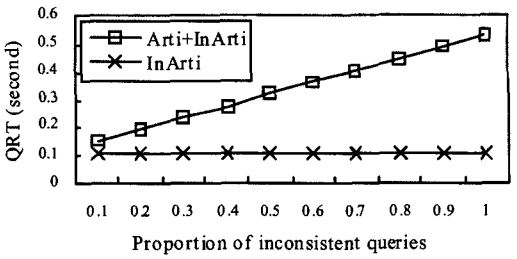


그림 9 비분절적 방법과 조합 방법의 질의 번역 시간 비교

분절적 방법에 비교하여 어는 정도의 개선 효과가 있는지를 보여 준다. 비교 기준은 가장 빠른 질의 번역 성능을 보여 주는 비분절적 방법의 시간이다. 불일치 질의의 수가 적을 때는 조합 방법과 비분절적 방법이 비슷함을 알 수 있다. 또한 50%의 불일치 질의일 경우(0.3초) 100%의 순수한 분절적 방법이 사용되는 것(0.55초) 보다 2배의 개선 효과가 있는 것을 관찰할 수 있다.

6.4 기존 방법과의 비교 분석

본 절에서는 XPath의 구조적, 의미적 특성을 고려한 질의 번역에 관한 기존 연구로서 Chen et al.[5]과의 비교 실험 결과를 기술한다. Kanza et al.[6]과 Amer-Yaiha et al.[7]과의 비교 실험에 관하여서는 두 방법이 XML의 구조적 이질성만을 고려한 질의 번역 방법이기 때문에 Chen et al. 방법을 비교 대상으로 선정하였다. Chen et al. 방법에 관하여서는 참고문헌[5]의 알고리즘 1을 구현하여 실험하였다. 특별히 본 논문의 제안 방법이 온톨로지 맵핑에 대하여 IMT 테이블을 사용하기 때문에, Chen et al. 방법 역시 WordNet 대신에 IMT를 사용하도록 하였다. 따라서 참고문헌[5]의 정의 3.4의 $sim(w_1, w_2)$는 만약 어휘 w_1, w_2가 IMT 테이블 상에서 동의어 관계를 가지면 1이고, 그렇지 않으면 0의 값을 가지도록 하였다. 실험 인자 값은 N = 50, R = 30%, T = 30%으로 설정하였으며, Chen et al. 방법의 Dewey 코드 레이블링은 Java String 클래스로 코딩하였다. XML DTD 파싱을 위해서는 Wutka DTD 파서(parser)를 사용하였다.

그림 10의 그래프에서 볼 수 있듯이 제안하는 분절적 방법(Our)은 앞서의 실험 결과와 같이 1초 이내의 질의 번역 시간을 보여 주지만, Chen의 방법(Existing)은 35초부터 크게는 약 17분까지의 질의 번역 시간을 보여 준다. 이것은 Chen의 방법이 어떤 XPath 질의 번역을 수행할 때마다 매번 모든 지역 XML DTD를 파싱하고 파싱된 트리를 순회하면서 가장 최적의 질의 번역 경로를 찾는 방식이기 때문이다. 즉, 다수의 XML 문서에 대하여 다수의 XPath 질의를 번역해야한다면 이러한

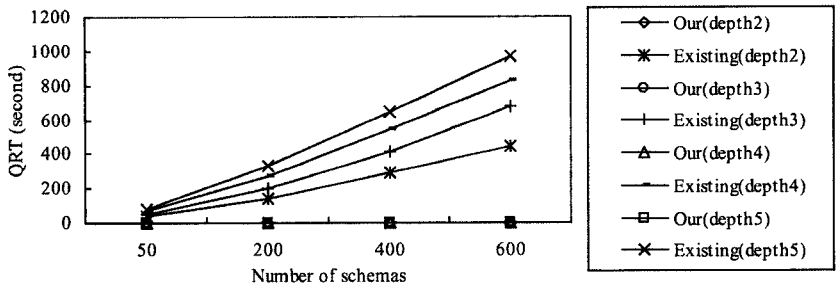


그림 10 기존 Chen의 제안 방법과의 질의 번역 시간 비교

방법은 비효율적이다. 실험에서는 모든 지역 XML DTD의 총 파싱 시간이 1초 내외로 적었으며, 대부분의 시간이 파싱된 트리에 대한 질의 경로 매칭에 많은 시간을 사용함을 알 수 있었다. 즉, 아래와 같은 수식에 있어 (DTD_Parsing + Dewey_Coding)의 시간은 미미하며, Path_matching 시간이 현저히 크다.

QRT_{byChen} = DTD_Parsing + Dewey_coding + Path_matching,

QRT_{byChen} : Chen et al. 방법의 질의 번역 시간,

DTD_Parsing: XML DTD에 대한 파싱 트리 생성 시간,

Dewey_coding: 파싱된 DTD 트리에 대한 Dewey 코드 레이블링 시간

Path_matching: DTD 트리와 질의 트리의 경로 매칭에 의한 최적의 번역 가능한 경로를 찾는 시간.

또한 Chen et al. 방법은 부모/자식 노드의 효율적 판별 및 트리 레벨(level) 값을 효율적으로 얻기 위하여 Dewey 코드 레이블링을 사용하는데, Chen et al. 방법과 같은 메인 메모리상에서의 트리 매칭시의 트리 레이블링 사용은 본 논문에서의 데이타베이스 활용 방식에서의 트리 레이블링 사용보다 큰 효과가 없음을 확인할 수 있었다.

이러한 실험 결과는 다수의 지역 XML DTD에 대한 질의 번역을 신속히 수행해야하는 매우 다이내믹한 XML 필터링 환경에서 본 연구에서 제안하는 관계형 데이타베이스 상의 IMT, SST를 사용하는 SQL 질의 번역 방법이 매우 효과적임을 보여주는 것이다.

7. 관련 연구

OBSERVER 시스템[13]은 소스(source) 온톨로지로부터의 어휘들로 작성된 질의를 다른 타깃(target) 온톨로지의 질의로 번역하는 방법을 소개하였다. 하지만 사용 질의는 XPath 질의가 아닌 온톨로지에서의 서술 논리(Description Logics, DL) 언어이다. 구성 요소 IRM (Interontology Relationship Manager)은 두 서론 다른 지역 온톨로지 어휘들의 동의어, 상/하위어 관계를 저장하며, 질의 번역은 임의의 지역 DL 질의의 어휘를 IRM을 사용하여 다른 지역 온톨로지의 어휘로 대치하는 것이다. 본 연구에서는 이러한 기본 아이디어를 다이내믹한 XML 필터링 환경 및 XML 스키마 구조를 갖는 XPath 질의의 번역으로 확장하였다.

Chen et al.[5]은 본 연구와 유사하게 XML 태그 어휘 사이에서의 개념 관계 및 XML 트리 구조 정보를 활용하는 의미적 XPath 질의 번역 방법을 제안하였다. 하지만 제안된 방법은 본 연구의 방법과 비교하여 다음과 같은 주요한 차이점을 갖는다. (1) 먼저 Chen의 방법은 다이내믹한 XML 필터링 환경을 고려한 것이 아니기 때문에, 6.4절에서 언급한 것처럼 질의 번역시마다 매번 XML DTD 문서를 파싱한다. 따라서 그림 3의 SST와 같이 방대한 수의 XML 구조 정보를 사전에 데이타베이스화할 필요가 있다. (2) Chen의 방법은 2.1절의 세 번째 가정에서의 XML 필터링 환경에서 빈번히 발생할 수 있는 부분 매칭 경로 질의를 처리할 수 없다. 하지만 이러한 유형의 질의는 L2L에서 매우 빈번하다.

Kanza et al.[6]은 반구조적(semistructured) 데이타에 대한 경로 질의의 질의 번역 방법을 소개한다. 하지만 제안된 연구의 주안점은 PTIME에 이루어 질 수 있는 경로 질의의 질의 번역 방법이 존재할 수 있음을 증명하는데 있다. 그러한 예가 될 수 있는 구체적 알고리즘은 제시하지 않는다. 또한 DAG (Directed Acyclic Graph) 데이타베이스, 혹은 재귀적 순환 (Cyclic)이 존재하는 데이타베이스에 대한 질의 번역은 NP-Complete인데, DAG 혹은 Cyclic 데이타베이스를 트리(Tree) 데이타베이스로 변환하는 방법을 제안한다. 트리 데이타베이스에 대한 질의 번역은 PTIME이다. 또한 본 연구와의 주된 차이점은 Kanza는 XML의 구조에 기인한 질의 변형만을 고려하지, 본 연구에서와 같은 어휘들의 의미적 혼동을 고려하지는 않는다.

Amer-Yahia et al.[7] 또한 XPath 질의의 구조에 근거한 질의 번역 방법을 소개한다. 하지만 XML 태그 어

휘의 동의어, 상/하위어와 같은 의미적 번역 방법을 다루진 않는다. 본 연구와 유사하게 주어진 XPath 질의 표현식에서 (1) '/' 경로 축을 '//' 경로 축으로 대치하는 방법('edge generalization relaxation'이라고 불림), (2) 임의의 하위 트리를 조상 노드에 붙여 버리는 방법('subtree promotion relaxation' 이라고 불림), (3) 단말 질의 노드(leaf query node)를 지우는 방법('leaf node deletion relaxation'이라고 불림) 등을 통하여 다양한 질의 매칭을 수행한다. 번역 가능한 질의 중 최적의 질의 번역의 선택은 여러 XML 스키마 문서들에 대하여 가장 특정적으로 매칭되는 경로 질의중(IDF의 개념) 한 문서에서 많은 매칭이 일어나는(TF의 개념) 경로를 선택하는 것이다. 하지만 이러한 TF/IDF의 계산은 다이내믹한 XML 필터링 환경에서는 부적합하다.

8. 결론 및 향후 연구

본 논문에서는 다이내믹한 XML 필터링 시스템을 위한 L2L 질의 번역 방법을 소개하였다. L2L XPath 질의 번역은 어떤 비주얼한 데이타 가이드가 제공되지 않는 상황에서의 단지 사용자의 기억과 지식에 근거하여 작성된 질의를 각 지역 스키마에 적합한 질의로 번역하는 개념이다. 이러한 질의 번역의 주요한 특징은 작성된 질의가 부분적으로 스키마에 모순되는 질의일 수 있다는 것이다. 본 연구에서는 이러한 질의와 스키마 불일치 문제를 처리할 수 있는 L2L 질의 번역 방법을 구현하여 실험해 보았다. 실험 결과를 통하여서 제안된 방법은 실제적으로 받아들일만한 질의 번역 성능을 보여주었으며, 기존의 다른 질의 번역 방법과 비교하여 더 나은 성능을 보여 주었다. 이것은 제안하는 질의 번역 방법이 다이내믹한 XML 필터링 환경에서 빠른 질의 성능을 보장하기 위해, 더욱 단순화된 온톨로지 맵핑 기술 및 고성능 데이타베이스 시스템에 저장된 XML 트리 레이블링 값을 활용하기 때문이다.

하지만 현재의 방법은 재귀 구조(recursion)를 갖는 XML 문서의 질의 번역 방법은 고려하고 있지 않다. 또한 프레디킷을 갖는 XPath 질의의 질의 번역(즉, 단일 경로가 아닌 트리 형태의 질의 번역)에 관하여, 비록 YFilter[2]에서의 질의 경로 분해 기법을 가정하였지만, 이의 보다 효율적 기법이 요구된다. 본 연구에서는 이러한 내용들에 대하여 현재 계속적으로 연구 진행중이다.

참 고 문 헌

[1] M. Altinel and M. J. Franklin, "Efficient filtering of XML documents for selective dissemination of information," Proc. 26th VLDB, Cairo, Egypt, pp. 53-64, Sept. 2000.

[2] Y. Diao, M. Altinel, M. J. Franklin, H. Zhang, and P. Fischer, "Path sharing and predicate evaluation for high-performance XML filtering," ACM Transactions on Database Systems, Vol.28, No.4, pp. 467-516, 2003.

[3] K. S. Candan, W. Hsiung, S. Chen, J. Tatemura, D. Agrawal, "AFilter: Adaptable XML filtering with prefix-caching and suffix-clustering," Proc. 32th VLDB, Seoul, Korea, pp. 559-570, 2006.

[4] J. Kim and S. Park, "PosFilter: An efficient filtering technique of XML documents based on postfix sharing," Proc. 24th BNCOD, Glasgow, Scotland, pp. 70-81, 2007.

[5] C. X. Chen, G. A. Mihaila, S. Padmanabhan, and I. M. Rouvellou, "Query translation scheme for heterogeneous XML data sources," Proc. 7th WIDM, pp. 31-38, Nov. 2005.

[6] Y. Kanza and S. Sagiv, "Flexible queries over semistructured data," In Proc. 20th Symposium on Principles of Database Systems, pp. 40-51, May 2001.

[7] S. Amer-Yahia, N. Koudas, A. Marian, D. Srivastava, and D. Toman, "Structure and Content Scoring for XML," In Proc. of 31th Inter. Conf on Very Large Data Bases (VLDB'05), pp. 361-372, 2005.

[8] Q. Li and B. Moon, "Indexing and querying XML data for regular path expressions," In Proc. of 27th Inter. Conf. on Very Large Data Bases (VLDB'02), pp. 361-370, 2001.

[9] R. Agrawal, A. Borgida, and H. V. Jagadish, "Efficient management of transitive relationships in large data and knowledge bases," In Proc. of the SIGMOD Inter. Conf. on Management of Data, pp. 253-262, 1989.

[10] XPath Version 1.0, http://www.w3.org/TR/xpath

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

[12] OWL Web Ontology Language Overview, W3C Recommendation, http://www.w3.org/TR/owl-features/

[13] E. Mena, A. Illarramendi, V. Kashyap, A. Sheth, "OBSERVER: An approach for query processing in global information systems based on interoperation across pre-existing ontologies," International journal on Distributed And Parallel Databases (DAPD), 8(2), pp. 223-271, 2000.

[14] A. Doan, J. Madhavan, P. Domingos, A. Halvey, "Learning to map between ontologies on the semantic web," In Proc. of the 11th International Conference on World Wide Web, pp. 662-673, 2002.

김 재 훈

1997년 건국대학교 전자계산학과 공학사 1999년 건국대학교 컴퓨터·정보통신공학과 공학석사. 2005년 서강대학교 컴퓨터공학과 공학박사. 2005년 3월~2006년 9월 삼성전자 정보통신총괄 통신연구소 책임연구원. 2006년 9월~현재 서강대학교 컴퓨터공학과 BK 21 연구교수. 관심분야는 웹 데이타베이스, 데이타베이스 보안, 데이타 프라이버시, 시맨틱 웹 임

박 석

1978년 서울대학교 계산통계학과(이학사) 1980년 한국과학기술원 전산학과(공학석사). 1983년 한국과학기술원 전산학과(공학박사). 1983년 9월~현재 서강대학교 컴퓨터공학과 교수. 1989년~1991년/2002년~2003년 University of Virginia 방문교수. 1997년 2월~현재 한국정보보호학회 이사. 2005년 한국정보과학회 부회장. 2004년 1월~2005년 12월 한국정보과학회 편집위원장. 1999년~2007년 DASFAA Steering Committee 멤버. 관심분야는 데이타베이스 보안, 실시간 시스템, 트랜잭션 관리, 데이타웨어하우스, 웹 데이타베이스 임