# XML 문서의 변경을 고려한 XML 전문 검색 역인덱스

권국봉*, 홍동권*, 김권양**
*계명대학교 정보통신대학
**경일대학교 컴퓨터공학과
e-mail:{gbkwon,dkhong}@kmu.ac.kr, kykim@bear.kyungil.ac.kr

# Update conscious and depth insensitive inverted indexes for XML full-text queries

Guk-Bong Kwon*, Dong-Kweon Hong*, Kweon-Yang Kim**
*College of Information and Communication, Keimyung Univ.
**Dept of Computer Engineering, Kyungil University

요 약

XML 문서는 관계형 테이블과는 달리 문서의 구조가 매우 복잡하고 불규칙하여 부분적인 정보를 최대한 활용하는 전문 검색이 일반적인 구조적 검색보다 더 중요한 역할을 한다. XML 문서는 계층이 있으므로 계층을 사용하는 전문 검색 연산은 계층을 제공함으로써 검색 공간을 줄여서 검색의 정확성과 효율성을 훨씬 더 높일 수 있다. 전문 검색 연산을 효과적으로 지원하기 위한 방법으로는 역인덱스를 (inverted index) 사용하는 것이 가장 일반적인 방법이다. 지금까지의 전문 검색을 위한 XML 문서의 구조 정보를 표현, 저장하는 방법들은 문서의 내용이 변경되지 않는 정적 문서(static documents)만을 고려하여 왔다. 이 방법들은 문서가 동적으로 변화할 경우 저장된 문서의 구조 정보 중에서 많은 부분을 다시 표현해야 하는 비효율적인 면이 있다. 본 논문은 XML 문서의 동적인 변화를 지원하면서 동시에 복잡한 XML 전문 검색을 지원하기 위한 방법으로 경로 스트링을 사용하는 효율적인 역 인덱스 구축 기법을 제안하고 제안하는 방법이 복잡한 문서의 검색과 문서의 동적인 변화를 효율적으로 검색할 수 있음을 보인다.

## 1. 서론

인터넷 상의 디지털 정보 교환을 위한 표준 형식인 XML의 등장과 함께 XML을 효과적으로 사용하기 위한 여러 가지 질의어가 연구되어 왔다[1]. 지금까지의 연구 결과로 2003년 W3C에서는 XML 질의어(Query)의 표준으로 XQuery를 발표하였다[2]. 초기에 발표된 XQuery 표준은 상당히 구조적인 XML 데이터의 데이터 검색만을 언급하고 있었으며 그 이후 전혀 구조적이지 않은 XML 문서의 전문 검색 (Full-text retrievals)과 XML 데이터 또는 XML 문서의 변경 기능에 대한 표준의 시도도 이루어지고 있다[3,4].

XML 질의에서의 전문 검색 기능은 매우 중요한 부분을 차지한다. XML 문서는 관계형 테이블과는 달리 문서의 구조가 매우 복잡하고 불규칙하여 부분적인 정보를 최대한 활용하는 전문 검색이 일반적인 구조적 검색보다 더 중요한 역할을 한다. XML 문서는 계층이 있으므로 계층을 사용하는 전문 검색 연산은 계층을 제공함으로써 검색 공간을 줄여서 검색의 정확성과 효율성을 훨씬 더 높일 수 있다. 전문 검색 연산을 효과적으로 지원하기 위한 방법으로는 역인덱스를 (inverted index) 사용하는 것이 가장 일반적인 방법이다. 하지만 기존의 일반 문서를 위한 역인덱스와는 달리 XML 문서와 XML 전문 검색을 위한 역인덱스는 키워드에 대한 정보 외에 XML 문서의 구조 정보를 효과적으로 저장할 수 있어야 한다. 지금까지의 전문 검색을 위한 XML 문서의 구조 정보를 표현, 저장하는 방법들은

※ 본 연구는 한국과학재단 목적기초연구 (R01-2003-000-10001-0)지원으로 수행되었음.

문서의 내용이 변경되지 않는 정적 문서(static documents)만을 고려하여 왔다. 이 방법들은 문서가 동적으로 변화할 경우 저장된 문서의 구조 정보 중에서 많은 부분을 다시 표현해야 하는 비효율적인 면이 있다. 이러한 문제를 해결하는 방안으로 본 논문은 XML 문서의 동적인 변화를 지원하면서 동시에 복잡한 XML 전문 검색을 지원하기 위한 방법으로 효율적인 역 인덱스 구축 기법을 제안한다.

## 2. 관계연구

전문 검색을 지원하기 위한 방법으로 정보검색(Information Retrieval) 분야에서 널리 사용하고 있는 방법은 역인덱스 방법이다[5]. 텍스트 문서에서 키워드를 발췌하여 역인덱스를 만들고, 키워드 검색은 역인덱스를 사용하여 검색하려고 하는 키워드가 존재하는 문서와 그 문서에서의 위치, 정확도 등을 쉽게 찾아낼 수 있게 한다. 전문 검색 연산을 효과적으로 지원하기 위한 방법으로는 B-tree를 이용한 역인덱스가 (inverted index) 가장 일반적인 방법이다. 하지만 기존의 일반 문서를 위한 역인덱스와는 달리 XML 문서와 XML 전문 검색을 위한 역인덱스는 키워드에 대한 정보 외에 XML 문서의 구조 정보를 효과적으로 저장할 수 있어야 한다. 지금까지의 전문 검색을 위한 XML 문서의 구조 정보를 표현, 저장하는 방법들은 문서의 내용이 변경되지 않는 정적 문서(static documents)만을 고려하여 왔다. 이 방법들은 문서가 동적으로 변화될 경우 저장된 문서의 구조 정보 중에서 많은 부분을 다시 표현해야 하는 비효율적인 면이 있다[5].

이진 테이블 (Binary table)[6] 방식은 키워드와 엘리먼트 또는 엘리먼트와 엘리먼트의 관계를 표시하기 위하여 XML 문서에서 부모-자식 관계를 형성하고 있는 모든 구성 원소들에 대해서 부모-자식 정보를 표현한다. 이 방법은 XML 변경이 발생하더라도 XML 문서의 다른 원소들의 관계에 영향을 주지 않는다. 하지만 XML 문서의 경로가 깊어져서 질의에서 확인해야 하는 경로가 길어질 경우 경로에 존재하는 엘리먼트에 대한 연속적인 부모-자식 관계 검색에 많은 부담이 생기게 된다.

번호 부여 방식 (Numbering scheme)[7]은 XML 문서의 각각의 구성 원소에 적절한 번호를 부여하여 그 번호를 이용하여 구성 원소들 사이의 계층 관계를 파악하게 한다. 하지만 XML 변경이 발생하여 새로운 구성 성분이 XML 문서의 중간에 삽입됨에 따라 직렬화된(serialized) 문서의 관점에서는 삽입 위치 다음에 존재하는 모든 내용들의 위치는 전부 이동 된다. 정적인 번호 부여 방식을 이용한 위치가 변화된 구성 원소에 새로운 번호를 부여하기도 어려울 뿐 아니라 적절한 번호를 모두 소모하여 XML 문서의 모든 구성 원소에 새로 번호를 부여해야 하는 일도 발생할 수 있다.

아직까지 XML 변경 구문 및 기능에 대한 널리 인정되는 표준안은 없다. 하지만 본 논문에서는 기본적인 제안(working draft)[8]을 바탕으로 XML 문서 변경 기능을 고려한다. 특히 기본적인 제안[8]에 있는 기능 중에서 다음의 5가지 핵심 기능을 고려하였다.

| 기능 | 설명 |
|---|---|
| xupdate:insert-before | 조건으로 선택된 노드의 앞 형제 노드로 입력 |
| xupdate:insert-after | 조건으로 선택된 노드의 뒤 형제 노드로 입력 |
| xupdate:append | 조건으로 선택된 노드의 자식 노드로 입력 |
| xupdate:update | 조건으로 선택된 노드의 내용을 변경 |
| xupdate:remove | 조건으로 선택된 노드의 삭제 |

[표 3] XML 변경의 기능

| 삽입(insert) 전 | 삽입 후 |
|---|---|
| <books><br><book><br><title> Data on the Web </title><br><author><br><family>Kim</family><given>Young Chul</given><br><family>Lee</family><given>Eun Suk</given><br></author><br><summary><br>This book mainly mentions<br><keyword>semistructured data</keyword><br><keyword>database</keyword><br><keyword>XML data</keyword><br></summary><br></book><br>....<br></books> | <books><br><book><br><title> Data on the Web </title><br><author><br><family>Kim</family><given>Young Chul</given><br><family>Lee</family><given>Eun Suk</given><br><family>Hong</family><given>Gil Dong</given><br></author><br><summary><br>This book mainly mentions<br><keyword>semistructured data</keyword><br><keyword>database</keyword><br><keyword>XML data</keyword><br></summary><br></book><br>....<br></books> |

[그림 1] 도서목록 예제 XML 문서

### 3. XML 문서의 변경을 고려한 XML 전문 검색 역인덱스

XML 문서로 작성되어 있는 도서목록에서 요약 부분에 키워드 XML을 포함하고 있는 책을 찾으려고 한다. XML 문서의 구조와 엘리먼트에 맞게 "<summary>에 XML을 포함하고 있는 <book>은?"의 형태로 질의를 작성하게 되며 XML 질의어는 형식에 따라 서로 다른 언어 형식을 가지게 될 것이다. <그림 2>는 본 논문에서 사용하는 도서목록의 내용 일부분을 보여주고 있다. <그림 2>를 사용하여 XML 질의어는 여러 가지 실행 계획 (Execution plan)을 만들어낼 수 있으며 그 중에서 가장 최적 실행 계획이 실제로 수행된다. 몇 가지 실행 계획을 보면 다음과 같다.

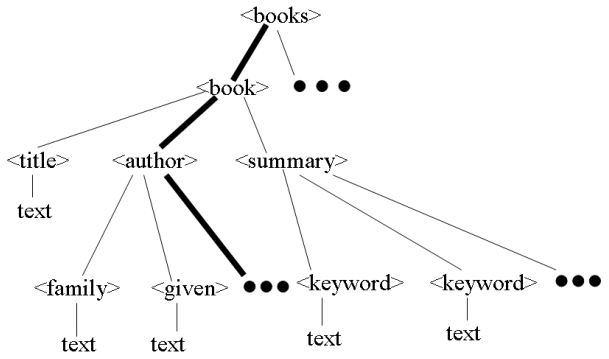
<실행 계획 1>
1. XML 키워드를 포함하고 있는 엘리먼트를 찾고
2. 찾은 엘리먼트가 <summary> 이거나 또는 조상 엘리먼트에 <summary>가 있는 것을 확인
3. 찾은 <summary>의 조상으로 <book> 엘리먼트를 찾아낸다.

<실행 계획 2>
1. 문서에 있는 path 중에서 <book>.. <summary> .. 에 맞는 2개의 path를 찾고
   (<books><book><summary><keyword>)
   (<books><book><summary>)
2. path 속에 XML이라는 키워드가 있는지 확인하여 <book>을 찾아낸다.

앞의 실행 계획을 효과적으로 실행하기 위해서 역인덱스는 키워드를 쉽게 찾을 수 있어야 하고 또 키워드가 존재하는 위치, 경로 등에 대한 정보를 쉽게 확인할 수 있게 구성되어야 한다. 또 문서의 일부분이 변경되더라도 역인덱스가 많이 변경되지 않도록 해야 한다.

앞의 실행 계획을 바탕으로 살펴보면 XML 문서의 역인덱스에는 키워드, 엘리먼트, 에트리뷰트가 반드시 포함되어야 한다. 키워드, 엘리먼트 이름 등을 추출하기 위하여 XML 문서의 분석이 미리 이루어져야 하며 이때 각 구성 원소들의 계층 관계를 같이 분석할 수 있어야 한다. 본 논문에서 제안하는 방법은 키워드, 엘리먼트 등의 XML 문서 구성 원소들과 함께 계층 정보를 표현하기 위하여 XML 문서의 루트에서부터 엘리먼트까지 만들어지는 경로 스트링(path string)을 이용한다. 본 논문에서는 편의상 에트리뷰트는 엘리먼트와 같은 방법으로 지원될 수 있으므로 엘리먼트만 고려한다. 경로 스트링은 XML 문서의 루트에서부터 만들어지므로 경로를 쉽게 파악하기 위하여 [그림 1]의 XML 문서를 Infoset으로 표현하면 [그림 2]의 형태로 구성된다.

키워드, 엘리먼트, 경로를 추출하기 위한 추출 작업에서는 키워드, 엘리먼트를 추출하면서 각각의 구성 원소들이 어느 경로의 자식으로 연결된 것인지 키워드 위치 정보를 같이 추출한다. 이 추출 과정은 경로와 키워드, 키워드의 위치를 단일 패스(single pass)의 문서 읽기로 처리할 수 있으므로 경로 스트링과 키워드의 위치를 구하기 위한 추가적인 부담은 없다.




[그림 2] XML 문서를 Inforset으로 표현한 그림

[그림 2]에서 키워드 및 경로 추출 프로그램을 통하여 추출해 낸 경로는 다음의 [그림 3]과 같다. 이들 경로 스트링 중에서 주어진 XML 질의의 답을 만들어낼 수 있는 경로는 "/books/book/summary"와 "/books/book/summary/keyword" 밖에 없다. 이들 경로 스트링을 이용하여 만들어진 XML 질의의 실행 계획을 살펴보자.

```
/books
/books/book
/books/book/title
/books/book/author
/books/book/author/family
/books/book/author/given
/books/book/summary
/books/book/summary/keyword
```

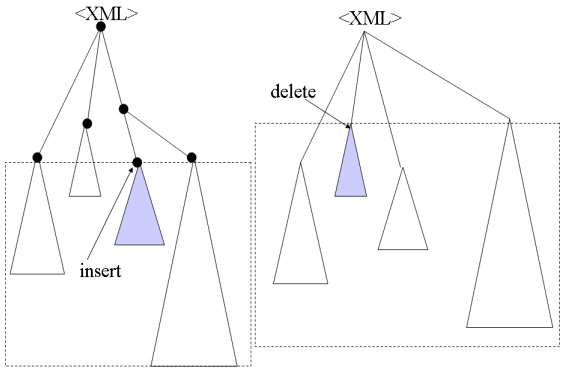
[그림 3] XML 문서에서 만들어지는 경로

<실행 계획>
1. XML이라는 키워드를 찾고, 경로 스트링에 book*summary가 포함되어 있는 경로들을 찾는다.
2. 그 다음 키워드의 위치를 사용하여 2개의 결과를 조인하여 XML 질의에 대한 답을 찾을 수 있다. 이때 질의 최적화를 위한 조인 방법은 키워드의 개수

와 경로 스트링의 빈도수에 따라 달라진다.

## 4. 분석

앞에서 경로 스트링과 키워드, 그리고 키워드가 어느 경로 스트링의 자식으로 존재하는지 추출하는 방법과 XML 전문 검색 질의를 실행하기 위하여 추출된 정보를 어떻게 사용하는지 알아보았다. 제안한 방법은 이진 테이블 (Binary table)[6] 방식과는 달리 검색 하는 정보의 문서에서의 깊이에 상관없이 스트링 연산으로 관련 경로를 찾아낼 수 있다. 이진 테이블 방식의 반복적인 조인 방법과는 달리 문서 깊이에 무관하게 항상 일정한 검색 성능을 유지 할 수 있다. 번호 부여 방식 (Numbering scheme)[7]은 XML 문서의 구성 원소에 적절한 번호를 부여하여 각 원소의 번호를 이용하여 원소들 사이의 계층 관계를 계산한다. 하지만 이 방법은 정적 문서에는 매우 좋은 성능을 보이지만 문서의 변경에 취약점을 가지고 있다. 반면에 본 논문에서 제안한 경로 스트링을 이용하는 방식은 [표 1]의 XML 변경 연산이 발생할 때 삽입 또는 삭제가 이루어지는 부분 주변에는 전혀 영향을 주지 않는다. 따라서 XML 문서의 변경이 발생할 경우 변경되어야 할 역인덱스가 최소화 되어 XML 문서 변경이 쉽게 이루어질 수 있다. 다음 [그림 4]는 XML 문서에 [표 2]의 xupdate:append 와 xupdate:remove 기능을 사용하여 XML의 입력, 삭제가 국소적으로 이루어짐을 도식화한 것이다.




[그림 4] XML 문서에서의 입력과 삭제

## 5. 결론 및 향후 연구 방향

지금까지 관계형 데이터베이스에서 SQL이 차지하는 역할처럼 XML에서는 XML 질의어가 데이터의 검색, 입력, 변경을 맡을 것으로 기대하고 있다. 특히 SQL과는 달리 반구조적인 XML의 특성 때문에 XML 질의어에서의 전문 검색 기능은 매우 중요한 부분을 차지하고 있다. 현재 W3C에서도 XML Query의 전문 검색 기능을 고려하여 표준화 작업이 진행 중에 있다. 최근 XML 전문 검색과 함께 많은 응용에서 XML 변경 기능도 XML 질의어가 반드시 가져야 하는 기능으로 인식되면서 XML 변경에 대한 기능과 형식에 대한 연구가 진행되고 있다.

본 논문에서 제안하는 방법은 복잡한 구조를 가진 XML 문서의 변경과 전문 검색을 효과적으로 지원하기 위하여 역인덱스에 키워드와 함께 구조적인 정보를 경로 스트링 (path string)으로 저장한다. 제안하는 방법은 문서의 깊이에 영향을 받지 않고 항상 일정한 검색 성능을 보이며 XML 문서의 변경이 발생해도 역인덱스의 변경을 최소화할 수 있다. 현재 XML 질의를 처리하기 위한 연구로 관계형 모델을 사용한 방법과 XML native DBMS에서의 인덱스에서 제안한 역인덱스 기법을 효과적으로 표현할 수 있는 방법을 연구하고 있으며 다양한 XML 문서를 사용한 성능 평가를 진행 중에 있다.

## 참고문헌

[1] Don Chamberlin, J. Robie, and D. Florescu, "Quilt: An XML query language for heterogeneous data sources" in Lecture Notes in Computer Science, Springer-Verlag, 2000.

[2] XQuery 1.0: An XML Query Language W3C Working Draft 12 Nov 2003. http://www.w3.org/TR/Query.

[3] XQuery and XPath Full-Text Requirements W3C Working Draft 02 May 2003. http://www.w3.org/TR/xquery-full-text-requirementsXQuery

[4] I. Tatarinov, Z. Ives, A. Halevy, D. Weld, "Updating XML" in Proceedings of ACM SIGMOD May Santa Barbara, CA 2001.

[5] Ricardo Baeza-Yates, Berthier Riberiro-Neto,: Modern Information Retrieval, Addison Wesley (1999).

[6] D. Florescu, D. Kossmann, and I. Manolescu, "Integrating keyword search into XML query processing" WWW9/Computer Networks, 33(1-6) 2000.

[7] C. Zhang, J. Naughton, D. DeWitt, Q. Luo, G. Lohman, "On supporting Containment Queries in Relational Database Management Systems" in Proceedings of ACM SIGMOD, May Santa Barbara, CA 2001.

[8] XML:DB Andreas Laux and Lars Martin. XUpdate Working Draft-2000-0914. http://www.xmldb.org