색인 기법을 사용한 XML 문서 검색 모델

이 순 미

◈ 목 차 ◈

1. 서 론

XML로 저장되고 교환되는 데이터의 양이 늘어남에 따라 XML 문서에 대한 검색의 필요성이 점점 커지고 있다. XML 문서의 검색에 사용되는 질의어는 비구조적 질의어와 구조적 질의어로 나눌 수 있다.

비구조적 질의어는 문서에 대한 정확한 구조 정보를 알 수 없거나 비정형적인 구조를 가진 경우에 사용되는 질의어로서 사용하기에 쉽고 정보에 접근이 용이하나 결과의 정확도는 다소 떨어지며 이러한 비구조적 질의어 사용의 가장 대표적인 방법이 키워드 검색이다.

구조적 질의어는 관계형 데이터베이스 상의 SQL과 같은 역할을 하는 질의어로서 XML 문서의 구조 정보를 활용하여 정확한 결과를 얻을 수 있고 표현능력도 뛰어난 장점이 있다. 그러나 구조적 질의어를 사용하여 엘리먼트의 구조나 내용 수준의 상세한 질의를 처리할 경우에 한 문서에서 처리해야 할 엘리먼트의 수가 수천 개 정도의 단위로 굉장히 많을 때에는 색인의 크기와 질의 처리 성능 등으로 인한 오버헤드가 증가하게 된다. 이러한 문제점을 해결하기위하여 본 논문에서는 XPath와 유사한 경로식을 이용하여 구조검색을 한 후에 질의문에 나타난 검색어를 사용하여 엘리먼트의 내용을 검색하는 XML 문서의 검색 모델을 제안하였다. 제안된 검색 모델에서는 우선 경로 식별자와 경로 색인을 이용한 구조 검색을 통하여 질의에 만족하는 XML 문서의 서브 트리를 생성하며 이 서브 트리에 대하여 엘리먼트의 내용 검색이 이루어지게 되는데 이 때에 엘리먼트 단위로 정보를 처리하면서도 큰 규모의 문서 집합에 적용될 수 있도록 하기 위하여 검색의 결과가 될 수 있는 엘리먼트들끼리 미리 분할하여 색인에 저장하여 처리되는 엘리먼트의 수를 줄이게 된다.

본 논문의 구성은 다음과 같다. 2장에서는 관련 연구를 서술하였고 3장에서는 구조 검색을 위한 경로 색인을 정의하고 질의 처리 과정을 기술하였으며 4장은 내용 검색 부분으로 분할 색인에 관하여 기술하였고 마지막으로 5장에서 결론을 맺었다.

2. 관련 연구

XML에 관한 표준이 제정되기 시작한 초기부터 XML 질의에 관한 연구가 활발하게 이루어졌다. 이런 초기의 연구들은 XML 문서에 대한 정확한 질의 표현을 목표로 이루어졌으며 XQL, XML-QL 등을 거쳐 XPath가 표준으로 제정되고 XQuery로 확장되고 있다.

* 경인여자대학 컴퓨터정보학부

구조적 검색 질의는 XPath와 같은 경로식을 이용하여 표현할 수 있으며 구조 검색의 처리 방법은 상향식 또는 하향식 트리 순회를 기반으로 하는 반복적 혹은 재귀적 처리를 필요로 한다[1,2]. 이 처리 방법은 트리 전체를 순회하여 처리 영역이 광범위하며 효율성이 떨어지는 단점이 있기 때문에 이를 보완하기 위하여[3,4]에서는 엘리먼트에 구조적 정보를 포함하는 특별한 식별자를 부여하여 트리의 순회 없이 구조적 검색 질의 처리가 가능하도록 설계하였다.

XML 문서는 그 유연성 때문에 단순한 데이터를 표현하는 한 방식이 될 수도 있지만 기본적으로 일반 텍스트를 구조화 한 것이기도 해서 데이터베이스 기반 기술과 정보 검색(IR) 기술의 융합이 필요하다. 그래서 최근에는 데이터베이스 기술과 정보 검색 기술을 융합한 시스템이 많이 연구되고 있다. XRANK[5], XSEarch[6,7] 모두 이러한 경향을 반영한 최근의 연구 결과이다. XRANK는 최소 공통 선조(least common ancestor)노드를 질의의 결과로 정의하고 이를 구하기 위한 DIL 색인구조와 질의 처리 알고리즘, 그리고 Top-k 질의를 효율적으로 처리할 수 있는 RDIL, 키워드 간의 상관관계가 적은 경우에도 효율적으로 동작할 수 있는 HDIL을 제안하였다. 본 논문에서는 질의 결과 생성의 기준으로 XRANK와 동일한 최소 공통 선조 노드를 사용하였다. XRANK에서는 질의 처리를 효과적으로 하기 위한 색인 구조나 알고리즘 등을 비교 분석했지만 처리해야 할 정보가 늘어나게 되어 큰 규모의 문서집합에 적용되기 힘든 문제점을 지닌다. [8]의 연구에서는 이러한 문제점을 해결하기 위하여 질의의 결과가 될 수 있는 엘리먼트들을 미리 분할하여 저장하는 방법을 제안하였으나 구조 정보는 고려하지 않고 키워드만을 가지고 검색이 수행된다.

본 연구에서는 [8]의 연구를 확장하여 DTD 정보를 활용한 구조 검색이 가능하며 큰 규모의 XML 문서 집합에 적합하도록 엘리먼트들을 분할하는 검색 모델에 관하여 연구하였다.

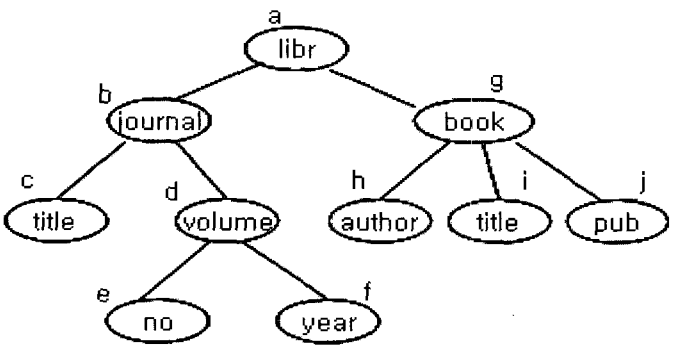
3. 구조 검색

3.1 경로 정보 테이블과 색인의 구조

본 논문에서는 XML 문서를 노드에 이름이 붙혀진 라벨된 트리(ordered node labeled tree)로 모델링을 하였다. 그림 1은 XML 문서 예제이며 그림 2는 그림 1의 문서 예제의 DTD 정보를 분석하여 작성된 구조 트리와 절대 경로를 바탕으로 작성된 경로 정보 테이블(Path Information Table)을 나타낸다. 경로 정보 테이블에서 PID(Path ID)는 각 경로마다 고유하게 할당된 값으로 해당 경로의 마지막 엘리먼트에 대한 유일한 식별자 값을 의미하게 된다.

```
<libr>
<journal>
   <title>ACM computing Surveys </title>
   <volume>
         <no> Volume37 Issue1 </no>
         <year> march 2005 </year>
   </volume>
</journal>
<journal>
   <title> Journal of ACM </title>
   <volume>
         <no> Volume52 Issue1 </no>
         <year> Jan 2005</year>
   </volume>
</journal>
<book>
   <author> kochmer </author>
   <title> JSP and XML </title>
   <pub> Addison Wesley </pub>
</book>
<book>
   <author> Silberschatz </author>
   <author> Abraham </author>
   <title> Database System Concepts </title>
   <pub> Macgraw-Hill </pub>
</book>
<book>
   <author> Date </author>
   <title> Introduction to Database Systems
   </title>
   <pub> Addison Wesley </pub>
</book>
</libr>
```

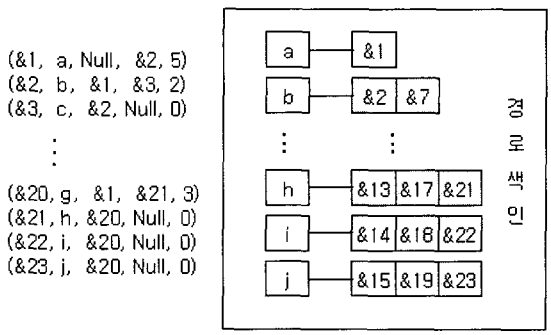
(그림 1) XML 문서 예제



경로정보 테이블	
단순절대경로	PID
/libr	a
/libr/journal	b
/libr/journal/title	c
/libr/journal/volume	d
/libr/journal/volume/no	e
/libr/journal/volume/year	f
/libr/book	g
/libr/book/author	h
/libr/book/title	i
/libr/book/pub	j

(그림 2) XML 문서 구조와 경로 정보 테이블

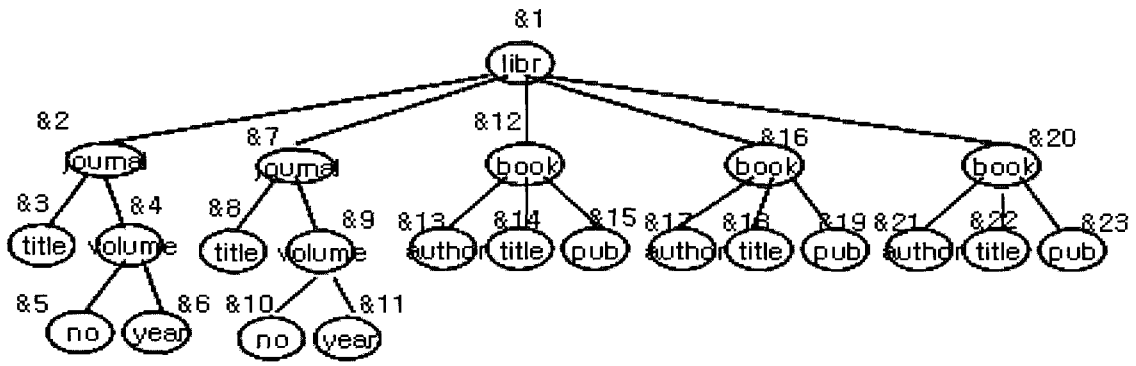
그림 3은 그림 1의 XML 예제 문서를 트리로 모델링한 그림이다. 트리의 각 노드에는 노드 번호가 붙어 있는데 이는 각 노드(엘리먼트)를 유일하게 식별하는 식별자이다. 각 엘리먼트 노드에 대하여 <UID, PID, PUID, FUID, NOC>로 구성된 엔트리가 생성된다. 여기서, UID는 엘리먼트의 식별자이며 PID는 경로 식별자, PUID와 FUID는 각각 부모 엘리먼트와 첫 번째 자식에 대한 식별자 값을 나타내며 NOC는 자식 엘리먼트의 개수를 나타낸다. 색인을 통하여 생성된 엔트리들의 위치 정보를 얻을 수 있게 되는데 본 논문에서는 경로 식별자(PID) 값을 키(key)로 하는 경로 색인을 정의하여 구조 검색을 처리한다. 그림 4는 각 엘리먼트 노드에 대한 엔트리와 경로 색인의 구조를 나타낸다.



(그림 4) 엘리먼트 노드 엔트리와 경로 색인 구조

3.2 구조적 검색을 위한 질의 처리

구조적 검색을 위한 질의 처리 방법은 주어진 질의에 포함된 엘리먼트 이름과 타입 정보를 이용



(그림 3) XML 문서 트리

하여 트리의 순회를 최소화하여 질의를 처리할 수 있도록 설계하였다. 질의 처리 과정은 우선 사용자 질의를 분석하여 질의에 포함된 엘리먼트들의 경로를 적절한 스트링 형태로 표현하는데 본 논문에서는 이것을 질의 경로라고 정의하였다. 생성된 질의 경로를 그림 2의 경로 정보 테이블에 있는 단순절대경로 필드의 값들과 스트링 매칭을 통해 최종 결과에 포함될 PID(경로 식별자) 값을 추출하여 목표 대상만을 직접 접근할 수 있게 하였다. 스트링 매칭을 위한 표현에 사용된 표기 중 와일드 카드 문자인 '*'는 0개 이상의 문자를 의미한다. 각 질의의 최종 반환 값은 UID의 집합으로 한다. 그림 1의 XML 문서에 대하여 다음과 같은 질의가 주어진다고 가정하자.

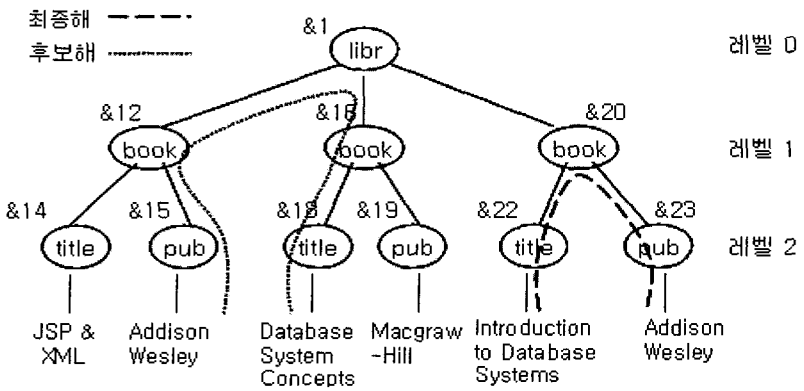
질의 1 : 'Addison Wesley' 출판사에서 발행한 'Database'에 관한 책을 검색하라.

위의 질의에서 검색 기준 엘리먼트는 'pub'와 'title'이며 검색 대상 엘리먼트는 'book'이 다. 먼저 사용자 질의를 분석하여 질의 경로인 '*book/pub'와 '*book/title'을 생성한다. 생성된 질의 경로를 그림 2의 경로 정보 테이블의 단순절대경로 필드 값과 스트링 매칭을 수행하여 '/book/pub'와 '/book/title'로 끝나는 PID 집합을 추출하면 {i, j}가 된다. 추출된 각 PID를 키 값으로 하여 경로 색인을 이용하여 해당되는 UID의 집합을 구하면 {&14, &15, &18, &19, &22, &23}이 된다. 여기서 검색 대상은 'pub'나 'title'이 아니라 이들의 부모 엘리먼트에 해당하는 'book'이므로 UID의 집합에 속하는 각각의 UID에 대하여 부모 엘리먼트인 PUID를 구하여 얻게 되는 {&12, &16, &20}이 구조 검색의 결과이다. 본 논문에서 검색의 결과로 반환된 노드는 그 노드를 루트 엘리먼트로 갖는 서브 트리를 의미한다. 즉, 서브 트리와 서브 트리의 루트 노드는 서로 같은 의미로 사용한다. 이 상에서 살펴본 바와 같이 구조 검색 단계에서 경로 색인을 사용함으로써 주어진 질의와 다른 경로 상에 있는 엘리먼트들에 대한 접근이나 필터링 과정이 요구되지 않음을 알 수 있다.

4. 내용 검색

4.1 색인 분할의 필요성

앞에서 언급한 구조 검색 단계를 통하여 질의와 관련이 있는 경로만이 선택되었으나 그림 5에서 알 수 있듯이 결과로 선택되어진 경로 상에도 정확하지 않은 정보들이 포함되어 있음을 알 수 있다. 예로 들은 XML문서에서 처리해야 할 book 엘리먼트의 수가 굉장히 많을 때에는 부정확한 엘리먼트의 내용을 처리하기 위하여 소모되는 색인의 크기증가와 질의 처리 성능 저하 등의 오버헤드가 증가하게 된다. 이러한 문제점을 해결하기 위하여 내용 검색 단계에서는 검색의 결과가 될 수 있는 엘리먼트들



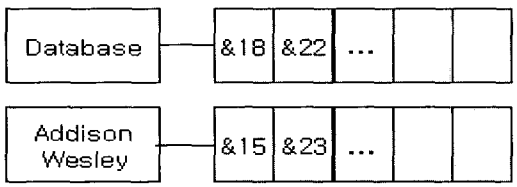
(그림 5) 구조 검색 결과로 생성된 결과 트리

끼리 미리 분할(partition)하여 색인에 저장하여 질의 처리 시에 분할된 정보만을 이용함으로써 처리되는 엘리먼트의 개수를 줄이는 방법을 적용하였다.

그림 5에서 'libr'은 루트 노드이며 레벨은 0으로 정의한다. 레벨이 루트에 가까울수록 '낮다'고 하고 루트에서 멀어질수록 '높다'고 한다. 즉 레벨이 높을수록 루트에서 더 멀리 떨어진 노드임을 의미한다. 본 논문에서는 엘리먼트와 애트리뷰트의 구별을 하지 않았으며 엘리먼트와 애트리뷰트 모두 노드라 지칭한다. 본 논문에서 검색의 결과로 반환된 노드는 그 노드를 루트 엘리먼트로 갖는 서브 트리를 의미한다. 결과는 질의의 조건을 만족하는 전체 문서 트리의 서브 트리이고 결과의 어떤 서브 트리도 조건 모두를 만족하지 않는다고 정의한다. 이는 최소 공통 선조 노드(LCA, least common ancestor)를 결과로 정의하는 것으로 높은 레벨에 있는 결과일수록 정확한 답으로 보는 것이다.

예제 XML 문서에 대한 질의 "Addison Wesley 출판사에서 발행한 Database에 관한 책을 검색하라."에서 조건은 출판사(pub)가 'Addison Wesley'인 것과 제목(title)에 'Database'가 포함되어 있는 것이다. 이 질의에 대한 가장 적절한 결과는 그림 5에서 노드 &20으로 표현되는 서브 트리이다. 이 노드는 하위 서브 트리에 'Addison Wesley'와 'Database'를 모두 포함하고 있으며 이 노드의 후손 노드 중 어느 노드도 이 두 키워드를 모두 포함하고 있지 않다. 노드 &1이 루트인 서브 트리도 결과로 볼 수 있으나 하위에 결과 될 수 있는 다른 노드(&20)를 포함하고 있으며 레벨이 &20보다 낮으므로 덜 정확한 결과이다.

정보검색 시에 질의 처리를 위해 사용되는 가장 일반적인 구조가 역색인(inverted index)이다. 본



(그림 6) 그림 5 예제에 대한 역색인

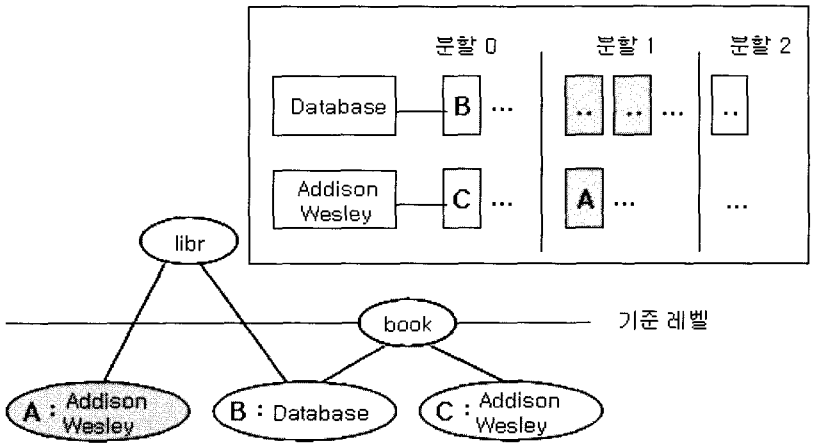
논문의 내용 검색 단계에서도 역색인을 사용한다. 역색인을 통하여 특정 단어를 입력하면 해당 단어가 존재하는 위치 정보의 리스트를 얻을 수 있다. XML 문서 검색의 경우 이 위치는 문서 단위가 아니라 엘리먼트 단위의 정보를 가지도록 구성된다.

이 역색인을 이용하여 키워드 검색을 하는 가장 기초적인 알고리즘은 각 키워드 별 위치 정보들의 조합을 검사라는 방법이다. 이 두 단어를 모두 포함하는 쌍은 (&15, &18), (&15, &22), (&23, &18), (&23, &22)이고 이들 조합의 결과가 (&1, &1, &1, &20) 이다. 이 중 가장 정확도가 높은 것으로 추정되는 것이 &20이므로 &20을 출력하게 된다. 이러한 질의 처리는 여러 번 리스트를 스캔해야 하므로 비효율적이며 엘리먼트의 집합이 커지게 되면 역색인 리스트의 크기도 커져서 많은 압축과 리스트 구성 기법이 필요하게 된다. 따라서 본 논문에서는 내용 검색 시에 색인을 분할하여 질의 처리 시에 역색인 리스트의 일부분만을 대상으로 처리할 수 있도록 하는 방법을 제안하였다.

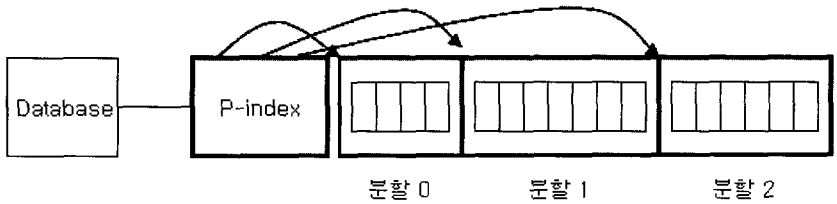
4.2 색인 분할

색인 분할의 개념은 의미있는 결과를 생성할 수 있는 가능성이 있는 노드들끼리 분할하여 색인에 저장함으로써 처리해야하는 엘리먼트의 개수를 줄이는 것이다. 분할 방법은 기준이 되는 레벨을 정하여 이 레벨보다 낮은 레벨에서 공통 선조를 갖는 노드들은 결과로 보지 않고 기준 레벨 이상의 레벨에서 공통 선조를 갖는 노드들만을 의미있는 결과로 보는 것이이다. 색인 분할 과정에서 중요한 점은 정확한 해를 얻기 위한 기준이 되는 기준 레벨을 설정하는 방법인데 본 논문에서는 3.2절에서 언급한 질의 경로의 가장 상위 엘리먼트가 위치한 레벨을 기준 레벨로 정의하였는데 그 이유는 질의의 조건을 만족하는 해를 주어진 경로 안에서 구할 수 있도록 하기 위함이다.

그림 7에서 "Addison Wesley 출판사에서 발행한 Database에 관한 책을 검색하라."라는 질의에 대한 질의 경로는 '*book/pub'와 '*book/title'이므



(그림 7) 분할의 개념



(그림 8) 분할 색인의 구조

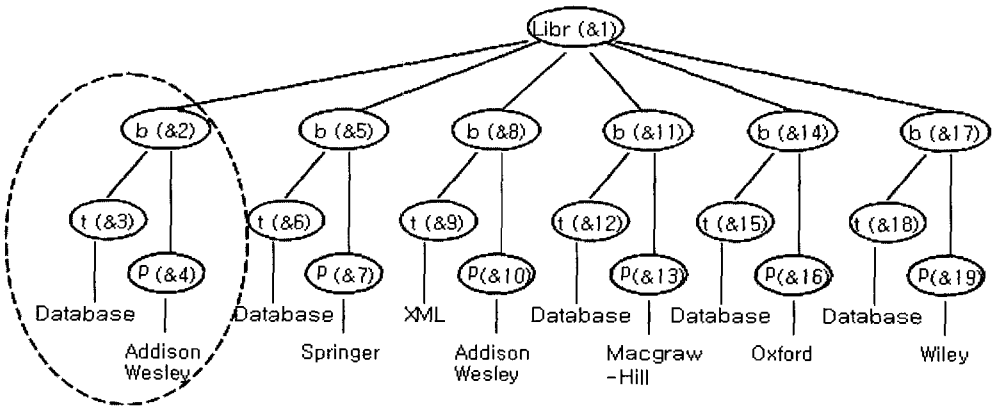
로 이 질의 경로의 가장 상위 엘리먼트인 'book'이 위치하고 있는 레벨이 기준 레벨이다. 그림 7에서 B와 C의 공통 선조 노드는 'book'인데 'book'이 기준 레벨이므로 의미있는 해가 될 수 있고 A와 B는 기준 레벨보다 낮은 레벨에 있는 'libr'가 공통 선조 노드이므로 해에서 제외된다. 따라서 B와 C를 같은 분할에 포함시키고 A와 B는 다른 분할에 속하도록 저장한다면 서로 다른 분할에 속해 있는 A와 B는 비교하지 않게 된다.

4.3 분할 색인의 구조

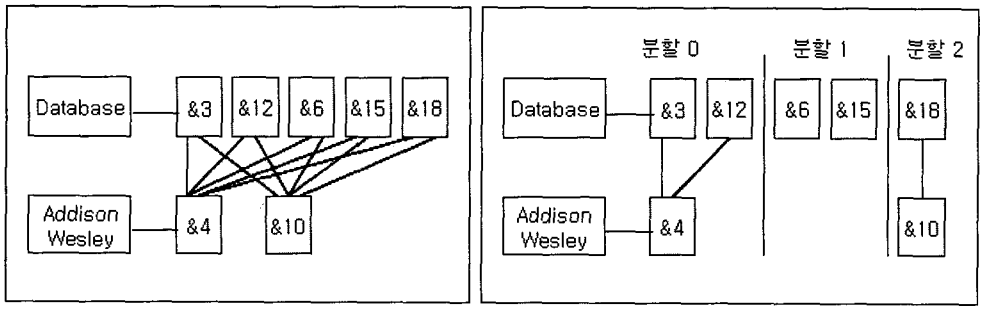
분할된 노드들은 각 분할별로 그룹핑되어 엘리먼트 내용의 역색인 리스트 내에 저장된다. 분할 색인의 구조는 그림 8과 같다.

역색인 리스트 내의 각 분할들에 효율적으로 접근하기 위하여 리스트의 처음 부분에 분할 색인 정보 P-index를 가진다. 이 색인 정보에는 전체 분할 수, 각 분할들의 분할 값(P-value)과 리스트 내의 오프셋 정보, 포함하는 노드 수에 관한 정보로 구성되어 있다. 질의 처리 시에는 우선 질의 조건절에 포함된 키워드 단어별로 P-index 정보를 검색하여 같은 P-value를 가진 분할들만을 선별하여 같은 분할에 속하는 엘리먼트들을 대상으로 최소 공통 선조 노드를 구하는 알고리즘을 적용하여 질의를 처리한다.

그림 9는 그림 5의 XML 문서 트리를 확장하여 간략하게 표시한 것이다. 그림 9의 문서 예제에 대해 3.2절의 질의 1을 처리한다고 가정하자. 우선 질의 경로에 근거하여 기준 레벨을 레벨 1로 설정한다. 이 기준 레벨에 속하는 노드(b 노드)들의 순서를 3으로 나누어 나머지가 0이면 첫 번째 분할(Partition 0), 1이면 두 번째 분할(Partition 1), 2이면 세 번째 분할(Partition 2)에 넣는 방법으로 분할한 경우의 역색인 리스트 구조는 그림 10의 B와 같다. A의 일반적인 역색인의 경우에 두 역색인



(그림 9) 확장된 문서 트리 예제



A. 일반 역색인 B. 분할 역색인

(그림 10) 역색인 리스트 비교

리스트의 모든 노드들은 최소한 한 번 이상씩 처리되는 데에 비하여 B의 분할 역색인을 사용하면 같은 분할에 속하는 노드들끼리만 처리하면 된다.

(&3, &12)와 (&10)과 같이 다른 분할에 속하는 엘리먼트들은 서로 매치가 되어도 공통 선조 노드가 &1이 되어 기준 레벨보다 낮은 레벨에 존재하므로 해에서 벗어나게 된다. 따라서 다른 분할에 속하는 노드들끼리는 비교할 필요가 없다. 같은 분할에 속하는 노드들만을 처리하면 분할 0에 속하는 (&3, &4), (&12, &4) 그리고 분할 2에 속하는 (&18, &10)의 조합이 생성된다. 각 순서쌍의 최소 공통 선조 노드를 구하면 (&2, &1, &1)이 되며 이 중에서 공통 선조 노드가 기준 레벨 이상인 것은 &2이므로 최종 결과는 &2가 된다. 위의 질의 처리과정에서 알 수 있듯이 일반 역색인에 비하여 분할 역색인을 사용하면 모든 엘리먼트 노드를 처리할 필요가 없어 질의 처리의 효율성이 증가하며 색인의 크기도 감소한다.

5. 결 론

본 논문에서는 질의 처리의 효율성을 높여주는 XML 문서 검색 모델을 제안하였다. 제안된 검색 방법은 구조 검색과 내용 검색의 두 단계로 진행된다. 우선 첫 번째 단계에서는 경로 식별자를 이용하여 문서 트리에 대한 순회를 최소화시키는 색인 구조를 설계하였으며 이 단계의 결과로 XML 문서 중에서 질의와 관련이 있는 엘리먼트들로만 구성된 서브 트리가 반환된다. 두 번째 내용 검색 단계는 질의문에 나타난 검색어를 사용하여 엘리먼트 내용을 검색하는 단계인데 이 때에 빠른 검색을 위하여 가능성이 있는 엘리먼트들끼리 미리 분할을 한 후

저장하여 처리 대상이 되는 엘리먼트의 수를 줄이는 방법을 제안하였다. 구조 검색 단계에서 경로 정보 테이블과 경로 색인을 사용함으로써 다른 경로 상에 있는 엘리먼트들에 대해 접근할 필요없이 주어진 질의와 관련이 있는 엘리먼트들만 검색된다. 내용 검색 시에도 분할 역색인을 사용하여 처리해야 할 엘리먼트 노드의 수를 줄여줌으로써 질의 처리의 효율성을 증가시키며 색인의 크기도 감소시켰다.

추후의 연구과제로 제안한 XML 문서 검색 모델을 구현하여 성능을 비교하는 연구가 진행되어야 한다.

참 고 문 헌

[1] J. McHugh et. al., "Indexing Semistructured Data, Technical Report", Stanford Univ., 1998.

[2] J. McHugh, J. Widom, "Query Optimization for XML", VLDB 1999.

[3] D. Shin et al,. "BUS:An Effective Indexing and Retrieval Scheme in Structured Documents", ACM Digital Libraries, 1998.

[4] Q. Li, B. Moon, "Indexing and Querying XML Data for Regular Path Expressions", VLDB 2001.

[5] L. Guo, et al. "XRANK: Ranked Keyword Search over XML Documents", SIGMOD '03.

[6] S. Cohen, J. Mamou, Y. Kanza, Y. Sagiv, "XSEarch: A Semantic Search Engine for XML", VLDB 2003.

[7] D. Carmel, Y. S. Maarek, M. Mandelbrod, Y. Mass, A. Soffer, "Searching XML Documents via XML Fragments", SIGIR 2003.

[8] 이홍래, 이형동, 유상원, 김형주, "PIX: XML 문서 검색을 위한 색인 분할 기법", 한국정보과학회논문지, vol.31 No.06, 2004.

[9] 김성완, 정헌석, 이재호, 임해철, "XML 문서에서 엘리먼트 타입을 이용한 구조적 검색 기법의 설계", 정보과학회학술발표논문집, vol.30 No.01, 2003.

[10] R Goldman and J. Widom, DataGuides: Enabling Query Formulation and Optimization in Semistructured Databases, VLDB 1997.

[11] http://www.w3.org/XML/

[12] XPath: XML Path languge, Nov. 1999. http://www.w3.org/TR/xpath

[13] J. Robie, J. Lapp, and D.S Schach. XML query language(XQL). The Query Languages Workshop. W3c, Dec. 1998. http://www.w3.org/TrandS/QL98/pp/xql.html

[14] XQuery: A query language for XML, Feb. 2001. http://www.w3.org/XML/Query

◐ 저 자 소 개 ◑

이 순 미

1984년 이화여자대학교 수학과 졸업(학사)
1986년 이화여자대학교 대학원 수학과 전산전공 졸업(석사)
1997년 홍익대학교 대학원 전산학과 졸업(박사)
1998년~현재 경인여자대학 컴퓨터정보학부 부교수
관심분야 : 데이터베이스, 객체지향 데이터베이스, 분산시스템