RDF 메타 데이터를 이용한 인덱스 기반의 XML/SGML문서 검색 방법에 관한 연구

오 동 현*, 김 규 태*, 정 회 경**, 이 수 연*
*광운대학교 컴퓨터공학과 정보공학 연구실
**배재대학교 컴퓨터공학과 멀티미디어 정보공학연구실
snail92@explore.kwangwoon.ac.kr

A Study of a Method of Index-based XML/SGML Document Retrieval Using RDF Metadata

Oh Dong-Hyun, Kim Gyu-Tae, Jung Heoi-kyung, Lee Soo-Youn
Information Engineering Lab of Computer Engineering, Kwangwoon University

요 약

인터넷이 급속히 성장함에 따라 대량의 SGML/XML문서를 보다 효과적으로 다룰 필요성이 증대하고 있다. SGML/XML문서를 데이터베이스에 저장하는 경우에 문서를 파싱하여 파싱된 결과를 모두 분리하여 저장하고 서로의 연관관계를 모두 구분하는 경우 구조화 정보를 최대한 이용할 수 있는 등 여러 가지 장점을 지니게 된다. 하지만, 이 경우 분할단위의 폭발적인 증가로 인한 시스템 성능 저하와 내용중복으로 인한 색인저장 오버헤드가 문제이다. 이런 문제점을 해결방안의 하나로서 본 논문에서는 RDF 메타데이타를 통하여 검색시 의미가 있는 단위로 분할 단위를 축소 지정하고 이 축소된 정보를 기반으로 인덱스를 생성하여 내용중복을 방지하는 방법을 제안하였다.
이방법은 RDF메타데이타를 통해 이루어짐으로서 웹기반에서 자동으로 이루어질 수가 있으며, 이를 통해서 기존의 방법보다 자동화된 검색을 할 수 있다.

1. 서론

인터넷이 급속히 성장함에 따라 대량의 SGML/XML문서를 보다 효과적으로 다룰 필요성이 증대하고 있다. SGML/XML문서를 데이터베이스에 저장하는 경우에 문서를 파싱하여 파싱된 결과를 모두 분리하여 저장하며 서로의 연관관계를 모두 구분하는 경우 구조화 정보를 최대한 이용하여 여러 가지 장점을 지니게 된다. 하지만, 이 경우 분할단위의 폭발적인 증가로 인한 시스템 성능 저하와 내용중복으로 인한 색인저장 오버헤드가 문제이다.
RDF는 1999년 3월 3일 현재 W3C의 PR(Proposed Recommendation)로서 웹 자원에 대하여 메타데이타를 기술하여 자동처리를 가능하게해주는 표준이다.[1]
본 논문에서는 RDF 메타데이타를 통하여 검색시 의미가 있는 단위로 분할 단위를 축소하고 이 축소된 정보를 기반으로 인덱스를 생성하여 내용중복을 방지하는 방법을 제안하였다.

이 논문은 1998년 과학재단의 특정기초연구과제 연구비에 의하여 연구되었음

본 논문은 2장 RDF 메타데이타와 구조화 문서 SGML/XML를 위한 인덱스 구조를 기술하고, 3장에서는 RDF메타데이타와 인덱스 테이블의 자동 변환, 4장 결과 및 고찰, 5장 결론 순으로 구성하였다.

2. RDF 메타데이타와 구조화 문서 SGML/XML를 위한 인덱스 구조

2.1 RDF 메타데이타

일반적으로 HTML과 XML같은 웹문서는 데이터베이스안에 적합한 자료모델과는 차이가 있다.
웹문서는 태그명을 통해 문서안의 다른 정보와 구분되어 전체 문서 구조를 이루게 된다. 이때 모든 태그명이 문서 원 의미구조와 일치하지 않으므로 모든 태그명을 대상으로 문서를 다루게 되면 너무 많은 분할 단위로 인한 시스템의 성능저하와 인덱스 정보의 심한 중복의 문제를 일으킨다.

메타데이타는 "데이타에 대한 데이터"로서 원 문서에 대한 응용 처리에 대한 부가적인 정보를 기술하는데 사용한다. W3C의 RDF는 XML형식으로 웹자원의 부가적인 정보를 기술하는데 사용한다. 이 권고안을 이용하면 원 문서에 대하여 원 문서의 효과적인 관리를 위해 필요한 부가적인 정보를 기술할수 있다.
더욱이 이러한 부가적인 정보는 응용 프로그램 모듈간에 자동 해석될수 있어 프로그램간의 연동을 쉽게 하는 장점이 있다.

2.2 SGML/XML를 위한 구조화 문서 인덱스 구조

SGML/XML같은 전자문서를 데이터베이스에서 각 구성 요소 수준에서 정확한 검색을 하려면 각 구성요소까지 추출할 수 있는 인덱스 구조를 가져야 한다.
전체 문서 단위로 인덱싱을 하게되면 데이터의 중복은 없으나, 구조화 문서로서의 장점, 재구성, 재사용, 정확한 검색을 하기 어렵고, 모든 엘리먼트에 대한 인덱싱을 하게 되면 구조화 문서로서의 모든 장점을 이용할수 있지만 분할단위의 폭발적인 증가로 인한 시스템 성능 저하와 내용 중복으로 인한 색인저장 오버헤드가 문제를 일으킨다.

따라서 기본적으로 구조화 문서의 장점을 살리기 위한 엘리먼트 단위 또는 인덱스 단위로의 인덱싱을 하지만, 인덱싱 되는 수준은 사용자에게 맡기거나 임의로 정하는 연구가 있었다[5].

3. RDF 메타데이타와 인덱스 테이블의 자동 변환

이 장에서는 XML/SGML 문서 검색을 위한 인덱스 테이블을 구성하는 방법에 대하여 설명한다. 인덱스 테이블은 검색의 효율을 높이기 위해서 최소의 의미있는 정보 블럭 단위로 구성되어져야 하며, 이를 위해 그 XML/SGML 문서의 DTD에 대한 의미 있는 엘리먼트의 메타데이타를 작성하고, 이 메타데이타를 이용하여 문서내의 최소 정보 블럭 단위의 인덱스 테이블을 구성하게 된다.

3.1 문서의 관리 정보를 표현한 메타데이타

XML/SGML 문서의 DTD를 통해서 의미 있는 정보 블럭 즉, 의미있는 엘리먼트의 범위를 기술한다.

다음은 위의 DTD에서 의미 있는 정보 블럭에 대한 메타데이타 (RDF Schema) 이다.

```
<!ELEMENT Bib (Book+) >
<!ELEMENT Book (Author+, Title, Publisher) >
<!ATTLIST  Book     Year - CDATA >
<!ELEMENT Publisher (Name, Address) >
<!ELEMENT Author (Firstname?, LastName) >
<!ELEMENT Title          #PCDATA >
<!ELEMENT Name           #PCDATA >
<!ELEMENT Address        #PCDATA >
<!ELEMENT Firstname      #PCDATA >
<!ELEMENT LastName       #PCDATA >
```

Figure1. Sample DTD (Bib.DTD)

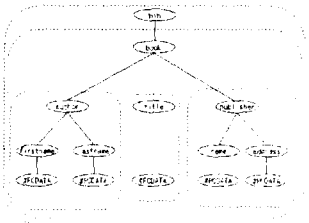

Figure2. Sample DTD의 정보 블럭 단위

```
<rdf:RDF xml:lang="en"
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:rdfs="http://www.w3.org/TR/1999/PR-rdf-schema-19990303#">

<rdfs:class ID="bib">
<rdf:type resource="http://www.w3.org/TR/1999/
                    PR-rdf-schema-19990303#Class" />
<rdfs:subClassOf rdf:resource="http://www.w3.org/TR/1999
                    /PR-rdf-schema-19990303#Resource"/>
</rdf:class>
<rdf:class ID= book >
<rdf:type resource="http://www.w3.org/TR/1999/PR-
                                    rdf-schema-19990303#Class"/>
<rdfs:subClassOf rdf:resource="#bib"/>
</rdf:class>
<rdf:property ID="year">
<rdf:type resource="http://www.w3.org/1999/02/22-
                                    rdf-syntax-ns#Property"/>
<rdfs:domain rdf:resource="#book"/>
<rdfs:range rdf:resource="#Integer"/>
</rdf:property>
<rdf:class ID="author">
<rdf:type resource="http://www.w3.org/TR/1999/PR-
                                    rdf-schema-19990303#Class"/>
<rdfs:subClassOf rdf:resource="#book"/>
</rdf:class>
<rdf:class ID="title">
<rdf:type resource="http://www.w3.org/TR/1999/PR-
                                    rdf-schema-19990303#Class"/>
<rdfs:subClassOf rdf:resource="#book"/>
</rdf:class>
<rdf:class ID="publisher">
<rdf:type resource="http://www.w3.org/TR/1999/PR-
                                    rdf-schema-19990303#Class"/>
<rdfs:subClassOf rdf:resource="#book"/>
</rdf:class>
</rdf:RDF>
```

3.2 메타데이타의 정보를 효과적으로 검색할 수 있는 인덱스구조

메타데이타를 이용하여 엘리먼트 인덱스 테이블, 속성 인덱스 테이블, 내용 인덱스 테이블, 구조 인덱스 테이블 등을 구성한다.

3.3 메타데이타에서의 인덱스구조로 변환 방법

엘리먼트 인덱스 테이블

Element Name	Element ID (Eid)	Document ID (Did)	Unique ID (Uid)

속성 인덱스 테이블

Attribute Name	Element ID (Eid)	Document ID (Did)	Unique ID (Uid)	datatype	Value

구조 인덱스 테이블

Document ID	K	Node ID	Element ID (Eid)

내용 인덱스 테이블

Keyword	Unique ID (Uid)	Document ID List (DidList)	Element ID List (EidList)

Figure3. 인덱스 테이블

위의 DTD에서 의미있는 정보 블럭 단위(element) 는 <book>, <author>, <title>, <publisher>등이다. <author> 엘리먼트는 <firstname> 엘리먼트와 <lastname> 엘리먼트를 포함하고 있는 정보 블럭이 된다. 즉, RDF Schema 에서는 이런 의미있는 정보 블럭 단위만 기술함으로써 인덱스 테이블을 생성할 수 있게 한다. RDF Schema 의 <rdf:type>엘리먼트 안에서 각 엘리먼트를 클래스라 하고, 하위 엘리먼트는 서브클래스, 그리고 각 엘리먼트의 속성은 property를 이용하여 표현하고, 속성 에서 <rdf:domain>은 속성의 해당 엘리먼트를 <rdf:range>는 datatype을 표현한다. 이러한 방법으로 DTD를 참조하는 RDF Schema를 구성하고 이를 이용하여 'Figure3'과 같은 인덱스 테이블을 구성한다.

4. 결과 및 고찰

위의 XML Sample Document Instance에 대한 인덱스 베이블 생성 예를 살펴보면 Figure 6와 같다.

```
<Bib>
 <Book year="1995">
  <Title> An Introduction to Database Systems </Title>
  <Author><LastName> Date </LastName></Author>
  <publisher><Name> Addison-Wesley </Name></publisher
 </Book>

 <Book year="1998">
  <Title> Foundation for Object/Relational Database </Title>
  <Author><LastName> Date </LastName></Author>
  <Author><LastName> Darwen </LastName></Author>
  <publisher><Name>Addison-Wesley </Name></publisher
 </Book>
</Bib>
```

Figure4. Sample Document Instance

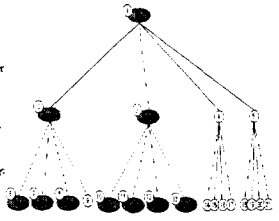

Figure 5. Node Index Tree

Element

Element Name	Element ID	Document ID	Unique ID
Bib	1	1	1
Book	2	1	2
Book	3	1	3
Title	4	1	4
Author	5	1	5
Publisher	6	1	6
Title	7	1	7
Author	8	1	8
Author	9	1	9
Publisher	10	1	10

Attribute

Attribute Name	Element ID	Document ID	Unique ID	Data Type	Value
Year	2	1	1	Integer	1995
Year	3	1	2	Integer	1998

Node

Node ID	K	Element ID	Document ID
1	4	1	1
2	4	2	1
3	4	3	1
6	4	4	1
7	4	5	1
8	4	6	1
10	4	7	1
11	4	8	1
12	4	9	1
13	4	10	1

K 차 완전 이진 트리

i 번째 노드의 부모
: Parent () = (i−2)/k + 1

i 번째 노드의 j 번째 자식
: Child(i,j)=k(i−1)+j+1

Figure 6. Sample DI 에 대한 Index Table

위의 인덱스 테이블을 이용한 실제 검색 예를 보면 Figure 7 과 같다.

검색 질의어는 "Book 엘리먼트의 자식 엘리먼트인 Author 엘리먼트를 찾아라" 이다.

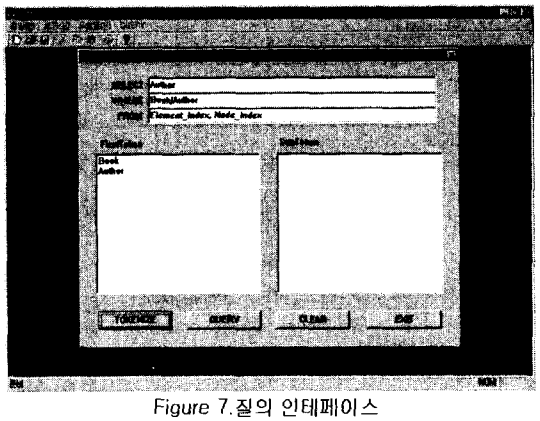

Figure 7.질의 인테페이스

검색된 결과는 Figure8 과 같다.

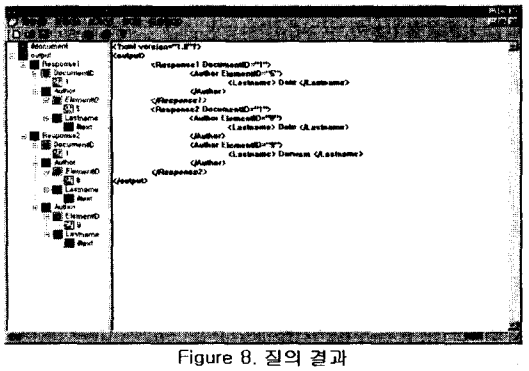

Figure 8. 질의 결과

XML/SGML DTD의 의미있는 정보 블럭을 RDF Schema 형태로 재구성함으로서 XML/SGML 문서 검색을 위한 최소 의미 단위 정보 블럭에 대한 인덱스 테이블을 구성 할 수 있다. 그리고, 이 인덱스 테이블을 이용한 검색은 기존의 의미가 부여되지 않은 정보 검색보다 효율을 높일 수 있다.

5. 결론

SGML/XML문서를 데이터베이스에 저장하는 경우에 문서를 파싱하여 파싱된 결과를 모두 분리하여 저장하며 서로의 연관관계를 모두 구분하는 경우 구조화 정보를 최대한 이용하여 여러 가지 장점을 지니게 된다. 하지만, 이 경우 분할단위의 폭발적인 증가로 인한 시스템 성능 저하와 내용중복으로 인한 색인저장 오버헤드가 문제이다.

본 논문에서는 메타데이타를 통하여 검색시 의미가 있는 단위로 분할 단위를 축소하고 이 축소된 정보를 기반으로 인덱스를 생성하여 내용중복을 방지하는 방법을 제안하였다.

이방법은 RDF 메타데이타를 통해 이루어짐으로서 웹기반에서 자동으로 이루어질 수가 있으며, 이를 통해서 기존의 방법보다 보다 최적화된 검색을 할 수 있었다.

6. 참고 문헌

[1] RDF(Resource Description Framework), http://www.w3.org/TR/PR-rdf-schema/,

[2] T.Dao and R.Sacks-Davis, Indexing Structured Text for Queries on Containment Relationships, In Preceedings of the 7th Australasian Database Conference, Melborune, 1996

[3] Brian E. Travis Dale C. Waldt, "The SGML Implementation Guide", Springer, Germany, 1995

[4] Eric Van Herwijnen, "Practical SGML Second Edition", KLUWER ACADEMIC PUBLISHERS, Stevens Printing, 1994

[5] 장재우 외 4인, "SGML 정보검색을 위한 인덱스 관리자의 설계 및 구현", 정보과학회 논문지 제5권 2호 p135-146, 1999년 4월

[6] 정회경, XML가이드", 그린 1998