XML 문서 검색을 위한 구조 기반 클러스터링

황 정 희[†] · 류 근 호[††]

요 약

웹에서 효율적인 정보 관리와 데이터 교환을 위해 XML의 중요성이 증가함에 따라 XML의 구조 통합과 구조 검색에 대한 연구가 진행되고 있다. 구조가 정의되어 있는 XML 문서의 구조 검색은 스키마 또는 DTD를 통해 가능하다. 그러나 DTD나 스키마가 정의되어 있지 않은 XML 문서에 대한 검색은 기존의 검색 방법을 적용할 수 없다. 그러므로 이 논문에서는 구조 정보가 주어지지 않은 많은 양의 XML 문서를 대상으로 구조를 빠르게 검색하기 위한 기반 연구로써 새로운 클러스터링 기법을 제안한다. 먼저 각 문서로부터 빈발한 구조의 특성을 추출한다. 그리고 추출된 빈발 구조를 문서의 대표 구조로 하여 유사 구조기반의 클러스터링을 수행한다. 이것은 서로 다른 구조의 전체 문서를 대상으로 검색하는 것보다 신속하게 구조 검색을 할 수 있도록 한다. 또한 유사한 구조들로 그룹화되어 있는 클러스터들을 기반으로 XML 문서에 대한 구조 검색을 수행한다. 아울러 구조 검색의 적용 방법을 기술하고, 그에 대한 결과의 예를 보여 제안 기법의 효율성을 증명한다.

Structure-based Clustering for XML Document Retrieval

Jeong Hee Hwang[†] · Keun Ho Ryu[††]

ABSTRACT

As the importance of XML is increasing to manage information and exchange data efficiently in the web, there are on going works about structural integration and retrieval. The XML document with the defined structure can retrieve the structure through the DTD or XML schema, but the existing method can't apply to XML documents which haven't the structure information. Therefore, in this paper we propose a new clustering technique as a basic research which make it possible to retrieve structure fast about the XML documents that haven't the structure information. We first extract the feature of frequent structure from each XML document. And we cluster based on the similar structure by considering the frequent structure as representative structure of the XML document, which makes it possible to retrieve the XML document faster than dealing with the whole documents that have different structure. And also we perform the structure retrieval about XML documents based on the clusters which is the group of similar structure. Moreover, we show efficiency of proposed method to describe how to apply the structure retrieval as well as to display the example of application result.

키워드 : 문서 클러스터링(Document Clustering), XML 클러스터링(XML Clustering), XML 문서(XML Document), 구조적 유사성(Structural Similarity), XML 구조검색(Structure Retrieval)

1. 서 론

인터넷은 누구나 정보에 쉽게 접근하고 정보를 게시할 수 있다는 장점으로 인해 사용자의 급격한 확산과 더불어 폭발적인 정보의 증가를 가져왔다. 그리고 여러 관련 문서의 확장, 통합 및 공유에 대한 어려움을 해결하고자 웹 데이터의 표준인 XML(External Markup Language)[1]이 제안되었다. XML은 사용자가 임의로 엘리먼트를 정의할 수 있고 엘리먼트는 여러 하위 엘리먼트를 가질 수 있는 계층적 구조를 형성하여 잘 정의된 구조화 문서(well-structured documents)의 형식을 갖는다. 이러한 XML의 구조적 특성으로 인해 XML 문서를 순서화된 레이블 트리로 표현하고 질의 트리나 패턴 매칭 트리에 대한 구조를 발견하기 위한 연구들[2-5]이 제시되고 있다. 이것은 XML의 구조적 특징이 정보 공유 및 통합, 정보 검색, 문서 관리시스템, 그리고 데이터 마이닝 등에 커다란 영향을 미치고 있기 때문이다.

정보 공유를 위한 XML 문서의 통합 및 구조 질의에 대한 검색의 실질적인 기초 과정은 탐색 대상의 구조를 만족하지 않는 데이터 구조를 미리 제거하는 필터링 과정을 포함한다. 이와 같이 검색 속도를 향상시킬 수 있는 방법의 하나로써 유사한 구조의 문서를 검색 이전에 미리 분류하는 문서 클러스터링에 대한 연구가 있다[6-8]. 유사 구조 문서의 클러스터링은 서로 다른 구조를 갖는 XML 문서들에 대해 구조적으로 유사한 문서들을 군집화하는 것으로써, 서로 다른 구조의 전

※ 이 논문은 2003년도 한국학술진흥재단의 지원에 의하여 연구되었음. (KRF-2003-002-D00280)
† 준 회 원 : 충북대학교 대학원 전자계산학과
†† 종신회원 : 충북대학교 전기전자 컴퓨터공학부 교수
논문접수 : 2004년 5월 3일, 심사완료 : 2004년 9월 8일

체 문서를 대상으로 검색하는 것보다 더 신속하고, 유연성 및 정확성을 제공하므로 기존 문서와의 통합 및 분류, 그리고 체계적인 문서 관리에 효과적이다.

한편 다양한 트리 구조에서 공통의 구조를 찾기 위해 빈발 패턴의 구조 발견을 목적으로 하는 연구들이 있다[9, 10]. 빈발 패턴의 공통 구조 추출은 유전체 데이터, 웹 마이닝, 반구조 문서의 구조 검색 등의 응용 분야와 매우 밀접한 관련을 갖으며, 효율적인 XML 문서의 관리 및 질의 최적화 등에 적용하여 관심 있는 유용한 정보에 대한 빠른 접근 및 구조 통합을 위한 기반을 제공한다.

기존 XML의 구조 검색[2, 8, 11]은 스키마 또는 DTD에 의한 구조 정보를 통해 검색 가능하다. 그러나 구조 정보가 없는 XML 문서에는 직접 적용하기 어렵다. 따라서 이 논문에서는 구조 정보가 주어지지 않은 많은 양의 XML 문서를 검색하기 위해 빈발 엘리먼트의 경로 구조를 기준으로 XML 문서를 클러스터링하고, 이를 기반으로 하는 구조 검색 방법을 제안한다. XML 문서는 문서의 특성을 구별할 수 있는 의미 있는 엘리먼트에 의한 계층적인 구조이므로 엘리먼트를 통해 문서의 내용을 예측할 수 있으며 문서의 종류에 대한 분류가 가능하다. 그러므로 XML 문서에서 엘리먼트의 순차적 구조를 중심으로 구조적 특성을 추출한다. 그리고 기존의 문서 클러스터링에서는 일반적으로 문서를 대표하는 객체들 간의 거리에 기반을 두어 유사성을 측정하였다. 그러나 거리 기반 클러스터링은 거리 측정을 위한 기준을 만들어야 하며 데이터의 특성을 고려한 거리 측정의 기준 생성이 쉽지 않으므로 정확한 측정이 어렵다는 문제점이 있다. 이러한 문제점을 개선하기 위해 이 논문에서는 새로운 방식의, 대량의 데이터에 유연하게 적용할 수 있는 트랜잭션 데이터를 위한 클러스터링 알고리즘을 적용한다. 또한 구조 검색에서는 사용자의 구조 질의에 대해, 전체 XML 문서를 대상으로 유사성을 비교하지 않고 구조적 유사성에 의해 구별되는 클러스터를 기반으로 검색을 수행하므로 써 검색의 범위를 줄인다.

논문의 구성은 다음과 같다. 먼저 2장에서는 이 논문의 연구 기반이 되는 XML의 구조 추출과 클러스터링에 대한 기존 연구 내용을 알아보고 3장에서는 XML 문서의 구조적 특성을 추출하는 방법을 기술한다. 그리고 4장에서는 추출된 구조에 대한 클러스터링 방법을 설명한다. 5장에서는 4장에서 제시한 클러스터링을 기반으로 하는 구조 검색 방법을 설명하고 6장에서는 제안한 클러스터링 알고리즘의 성능 및 구조 검색의 적용 결과를 통해 효율성을 검증한다. 7장에서는 이 논문의 내용을 요약하고 결론을 맺는다.

2. 관련 연구

XML은 기존의 비구조적 데이터와 다른 구조적 특성을 갖기 때문에 구조에 관련된 연구가 많은 부분을 차지한다. 이 논문에서는 트리 구조의 부모/자식 및 형제 노드간의 관계와 같은 상세한 구조 검색을 초점으로 하지 않고, 많은 XML 문서들에 대한 유사한 구조의 그룹화 및 이를 기반으로 하는 유사도 측정에 의한 구조 검색 방법을 제안한다. 그러므로 이와 관련된 기반 연구에 대해 기술한다. 특히 XML의 공통 구조 추출 및 문서의 구조 통합을 위한 유사 구조 문서의 클러스터링에 관한 연구들을 살펴보도록 한다.

[7, 12, 13]은 마이닝 기법을 이용한 XML 문서의 구조추출 방법으로 문서내의 엘리먼트들에 대한 구조 발견을 위한 Intra-Structured Mining과 문서간의 구조적 연관성 발견을 위한 Inter-Structured Mining으로 분류한다. 그러나 마이닝의 적용 가능성을 언급했을 뿐 구체적인 적용 방법은 제시하지 않았다. [14]는 목표 문서의 구조와 유사한 구조의 문서를 찾는 방식에 대해, XML를 트리로 표현하고 트리의 경로에 대한 공유 정도를 기준으로 문서 구조의 유사성을 측정할 수 있는 계산방식을 제안하였다. 이렇게 기준 문서와 비교 문서에 대한 구조적 유사성 측정 방법은, 일정한 기준을 두지 않고 유사한 구조의 문서를 자동으로 분류하는 클러스터링과는 차이가 있다. 그리고 [15]은 트리 구조의 변화를 탐색하기 위해 일반적으로 사용하는 edit distance를 사용하여 구조적 특성에 대한 매칭 기법을 이용하고 클러스터링하는 방법을 제안하였다. 그러나 많은 양의 문서에는 적용하기 어렵다. [4]는 XML의 구조적 유사성 측정을 위해 트리로 표현되는 노드의 매칭 조건에 대한 비용을 가지고 매트릭스를 구성하는 방법을 제시하였으나 유사성 측정이 복잡하다는 단점이 있다. [16]은 내용을 갖는 엘리먼트에 대한 ePath와 문서 id를 가지고 비트 맵 인덱스를 구성하여 XML 문서를 클러스터링 하는 방법을 제시하였다. 그러나 대량의 데이터에 대해서는 많은 공간을 차지한다는 문제점이 있다.

또한 공통의 구조 발견을 위한 연구로써, [8]은 서로 다른 DTD들을 통합할 때 적당한 통합 구조의 DTD를 찾기 위한 방법으로 DTD를 클러스터링 하는 방법을 제안하였다. 그러나 구조가 제공되지 않는 XML 문서에는 적용할 수 없다. [10]은 패턴 매칭 트리를 찾는 트리 마이닝 알고리즘을 사용하여 트리 구조에 중첩되어 있는 서브 트리의 상세한 구조발견을 위한 방법을 제안하였다. 이 알고리즘은 문서의 구조 특성에 의한 분류보다는 공통의 서브 트리 구조 추출을 목적으로 한다. [17]은 레이블화 되어 있는 트리구조의 집합에서 공통으로 발생하는 연관된 레이블의 쌍을 기반으로 연관규칙을 이용하여 그룹화하고 그것에 대해 최대 빈발 구조를 추출한다. 그러나 연관된 레이블의 쌍을 기반으로 하기 때문에 다중 관계의 트리 구조는 탐색되지 못하는 단점이 있다.

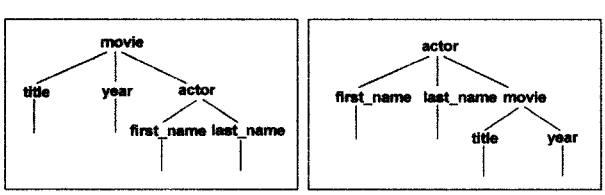
기존의 연구들에서 XML 문서의 구조 추출 및 유사 구조의 클러스터링을 위해 마이닝 기법을 일부 적용하고 있으나 다양한 구조의 다중 XML 문서에 대한 구조적 분류보다는 구조 정보인 DTD가 주어진 XML 문서 및 소수의 XML 문서에 대한 구조적 매칭에 적용 가능한 방법을 제시하고 있다. 또한 대

부분의 문서 클러스터링에 관한 연구에서는 문서의 특성을 벡터로 표현하고 이에 대한 거리 계산에 의해 유사성을 측정하여 그룹화 한다. 그러나 문서의 특성에 따라 유사성을 측정하기 위한 기준의 정의 및 정확성 측정에 어려움이 있다.

이러한 기존 연구의 문제점들을 해결하기 위하여 [22]에서 트랜잭션 데이터를 위한 알고리즘을 사용하는 새로운 방식의 XML 클러스터링을 제안하였으나 이를 구조 검색에 적용하기 위한 구체적인 방법은 제시하지 않았다. 따라서 이 논문에서는 많은 양의 XML 문서의 구조 검색에 효율적인 유사 구조 기반의 클러스터링과 이를 기반으로 구조 검색에 적용하는 방법을 제안한다. 제안하는 클러스터링 방법은 기존의 문서 클러스터링 방식을 탈피하여 대량의 데이터에도 유연하게 적용할 수 있는 [18]의 CLOPE 알고리즘을 이용한다. 이 알고리즘은 클러스터내의 공통 항목의 비율이 높을수록 유사항목의 밀도가 높은 양질의 클러스터를 생성한다는 개념을 이용한다. 그러나 클러스터에 속하는 트랜잭션의 개별 항목을 고려하지 않고 클러스터들의 공통항목만을 비교하기 때문에 클러스터간의 유사성이 높아질 수 있고 적정한 수의 클러스터 생성을 조절할 수 없다는 단점이 있다. 그러므로 우리는 이 문제점을 개선하기 위하여 [19]의 주요항목 개념을 추가적으로 이용하므로 써 클러스터링의 효율성을 높이고 신속한 구조 검색을 가능하게 하는 방법론을 제시한다.

3. XML의 구조 특성 추출

XML 문서의 엘리먼트 순서와 부여된 엘리먼트 그 자체는 XML 문서의 형태와 종류를 구분할 수 있게 해 주는 특징을 가지고 있다[9, 18]. (그림 1)은 XML 문서의 특성을 설명하기 위하여 두 개의 XML 문서의 일부에 대한 엘리먼트를 트리 구조로 표현한 것이다. (a)와 (b)의 두 문서는 같은 엘리먼트들로 구성되어 있지만 (a)는 영화에 대한 정보를 나타내는 문서이고 (b)는 영화배우의 정보를 나타내는 문서로써 엘리먼트의 구조에 따라 문서의 내용이 다르게 구분된다. 그러므로 XML 문서에 대한 분류에서는 의미적 구조를 고려해야 한다. 의미적 구조란 엘리먼트의 구조를 통해서 내용이 유사한 문서를 클러스터링 하는 것이다. 따라서 이 논문에서는 엘리먼트의 연관성뿐만 아니라 엘리먼트의 구조적 순서도 함께 고려할 수 있는 순차패턴 마이닝을 이용하여 문서의 대표 구조를 위한 엘리먼트를 추출한다.



(a) movie.xml (b) actor.xml

(그림 1) XML 문서의 트리 구조

XML 문서의 구조적 유사성은 XML 문서 구조 경로에 대하여 서로 다른 문서들이 같은 구조 경로를 얼마나 공유하고 있는지를 파악하고, 이 공유 구조의 정도에 따라 구조적 유사성을 판별하는 것이다. 그러므로 먼저 각 문서의 대표 구조를 추출한다. 다음의 예제문서는 문서의 대표 구조를 찾는 과정을 설명하기 위한 영화 정보 문서 <movie.xml>의 일부이고 레벨 1의 <movie>부터 레벨 4에 해당하는 <director>, <first_name>, <last_name>, <nationality> 등으로 구성되어 있다.

```
<movie>
  <title>Sunset Blvd.</title>
    <year> 1950 </year>
    <directed_by>
      <director>Billy Wilder</director>
    </directed_by>
    <cast>
      <actor>
        <first_name>Charles Dickens</name>
        <last_name>1812</born>
        <nationality>English</nationality>
      </actor>
    </cast>
</movie>
```

여러 가지 이름의 엘리먼트로 구성되어 있는 문서에서 문서의 특성을 나타내는 구조를 추출하기 위해서는 각각의 엘리먼트를 쉽게 구별할 수 있는 재명명의 절차가 필요하다. 그러므로 위 예제 문서에서 내용을 포함하는 엘리먼트를 중심으로 경로 구조를 추출하고 이들 엘리먼트를 쉽게 식별하기 위하여 내부 엘리먼트 매핑 테이블을 이용하고, (그림 2)와 같이 알파벳으로 재명명 한다.

X_id	Original paths	Transformed paths
1	movie/title	a/b1
2	movie/title/year	a/b1/c1
3	movie/title/directed_by/director	a/b1/c2/d1
4	movie/title/cast/actor/first_name	a/b1/c3/d2/e1
5	movie/title/cast/actor/last_name	a/b1/c3/d2/e2
6	movie/title/cast/actor/nationality	a/b1/c3/d2/e3

(그림 2) 엘리먼트의 경로에 대해 재명명된 구조 시퀀스

(그림 2)와 같이 실제적인 내용을 포함하는 의미 있는 엘리먼트들을 중심으로 재명명된 구조는 문서의 구조 특성을 추출하기 위한 시퀀스 입력 정보가 되며 하나의 경로에 포함되어 있는 엘리먼트들은 발생 시퀀스를 구성하는 하나의 항목으로 고려한다. 그리고 가장 빈발한 시퀀스 패턴을 발견하기 위하여 순차 패턴 마이닝을 이용하여 문서를 대표하는 엘리먼트 구조 정보를 찾는다. 순차 패턴 마이닝 알고리즘은 연관 규칙과는 달리 트랜잭션의 발생 횟수와 발생 순서를 고려하므로 XML 문서의 구조 추출에 적합하다[14, 20].

시퀀스들의 집합에 대한 빈발 구조를 추출하기 위하여 이 논문에서는 후보패턴을 생성하지 않는 [21]의 PrefixSpan 알고리즘을 이용한다. 이 알고리즘은 빈발 패턴 탐색을 위해 노드를 확장할 때, 그 노드가 나타내는 빈번한 시퀀스를 포함하고 있는 시퀀스만을 모은, 시퀀스(prefix) 이후의 부분만을 지정한 Project DB를 이용하며, 성능의 우수성이 [21]에서 증명되었다. 이 알고리즘을 이용하기 위해 먼저 빈발 구조 지지도(Frequent Structure Support)를 다음과 같이 정의한다.

[정의 1] 빈발 구조 최소 지지도 (Frequent Structure Minimum Support)

빈발 구조 최소 지지도란 한 문서의 전체 경로 중에서 빈발 구조 비율을 만족하는 최소 빈발도이며, 이를 만족하는 구조 경로를 빈발 구조라 한다[22]. 이것을 식으로 표현하면 다음과 같다.

FFMS = 빈발 구조 비율 * 문서 전체 경로의 수

(0 < 빈발 구조 비율 < 1)

각 문서마다 발견되는 전체의 경로 수는 다르다 그러므로 빈발 구조 최소 지지도는 모든 문서에 대해 동일한 최소 지지도를 적용하기 위한 것으로 빈발 구조 비율을 통해 최소 지지도를 산출하고 PrefixSpan(자세한 알고리즘은 [21]를 참고)에 적용한다.

최대 빈발패턴은 문서에서 가장 공통적으로 사용되는 구조로서 중요한 의미를 갖는다. 반면에 이 논문에서는 최대 빈발 패턴의 구조가 아니더라도 최대 빈발 구조에 대한 일정 비율 이상을 만족하는 구조(예, 최대빈발구조 길이 5 * 80% = 빈발 구조 길이 4)도 중요한 의미로 보고 클러스터링을 위한 기초 구조 항목에 포함한다. 이것은 최대 빈발 패턴의 구조만이 그 문서를 대표하는 유일한 구조가 아닌 경우(예, 하나의 문서에 여러 가지 주제가 함께 포함되어 있는 경우)를 고려하고, 또한 이 논문에서는 빈발 구조 자체를 클러스터링 하기 위한 구조 항목으로 간주하기 때문에 발생할 수 있는 빈발 구조의 손실을 피하기 위함이다.

다양한 경로의 XML 문서에서 의미 있는 구조를 추출하는 것은 문서의 주제어를 추출하는 것과 유사하며, XML문서의 엘리먼트에 대한 구조 경로를 통해 계층적 구조를 이루는 문서의 의미 있는 대표 구조를 추출하는 것이다. 그리고 순차 패턴 알고리즘을 통해 찾아진 빈발패턴은 문서에서 많은 하위노드를 포함하는 엘리먼트일수록 발생 빈도수가 많이 나타나게 되고 이것은 문서에서 그 엘리먼트가 차지하는 비중이 크다는 것을 의미한다.

4. 구조적 클러스터링

각 문서의 빈발 구조를 가지고 유사 구조의 문서를 클러스터링하기 위해서 우리는 XML 문서를 하나의 트랜잭션으로 가정하고 각 문서에서 추출된 빈발 구조들을 트랜잭션의 항목들로 취급하여 유사한 항목기준의 그룹으로 문서를 클러스터링 한다.

4.1 클러스터 할당 기준

클러스터링은 공통의 많은 항목을 하나의 클러스터에 많이 포함하도록 하여 응집도를 높이고 각 클러스터간에는 유사성이 거의 없도록 하여 전체적으로 각 클러스터가 구별되도록 하는 것이 목적이다. 먼저 클러스터링을 수행하기 위해서는 각 문서를 클러스터에 할당하기 위한 기준이 필요하다. 이 논문에서는 [18]에서 제시된 전체적으로 양질의 클러스터를 생성하도록 유도하는 할당 방식을 기준으로 하고, 좀 더 신속하고 효율적인 클러스터 관리를 위해 [19]의 주요항목(Large Item) 개념을 추가한다.

모든 트랜잭션에 포함되어 있는 빈발 구조 항목들의 집합 $I=(i_1, i_2, \cdots, i_n)$하고, 클러스터 집합 $C=(C_1, C_2, \cdots, C_m)$문서를 나타내는 트랜잭션 집합 $T=(t_1, t_2, t_3, \cdots, t_k)$이라 표기한다. 클러스터에 트랜잭션을 할당하기 위한 기준이 되는 클러스터 할당 이익을 다음과 같이 정의한다.

[정의 2] 클러스터 할당 이익

클러스터 할당 이익은 전체 클러스터에 대해 각 클러스터를 구성하는 고유 항목에 대한 누적 항목 비율의 합이다[22]. 이것을 식으로 표현하면 다음과 같다.

$$Gain(C) = \frac{\sum_{i=1}^{m} G(C_i) \times |C_i(T_r)|}{\sum_{i=1}^{m} |C_i(T_r)|} = \frac{\sum_{i=1}^{m} \frac{T(C_i)}{W(C_i)^2} \times |C_i(T_r)|}{\sum_{i=1}^{m} |C_i(T_r)|}$$

여기서 G는 각 클러스터에서 고요항목 W에 대한 누적 항목의 비례를 나타내는 H를 나타내는 것으로써, $H = T$(전체 항목 수) / W(고유항목수)이고 $G = T/W^2$이다.

클러스터 할당을 위한 기준 함수인 $Gain$은 전체적으로 각 클러스터에 대한 공통항목의 비율이 높게 구성될 때 큰 값의 클러스터 할당 이익이 산출되고, 최대의 $Gain$값이 되도록 하는 클러스터에 트랜잭션을 할당한다.

그러나 개별 항목의 누적 빈도를 고려하지 않고 공통 항목의 비율인 $Gain$만을 이용하면 기존의 클러스터와 새로운 클러

스터에 할당할 경우의 *Gain*을 비교하여 클러스터를 할당하게 될 때 새로운 클러스터에 대한 $Gain = \frac{\text{항목수}}{\text{항목수}^2}$ 가 되는 상당히 높은 값의 할당 이익 값이 산출된다. 따라서 이미 존재하는 C_1, C_2 클러스터에 할당하기보다는 새로운 클러스터를 생성하여 트랜잭션을 할당하게 되는 경우가 많아지므로 적당한 크기 이상의 많은 클러스터가 생성될 수 있다는 문제점이 있다. 우리는 이 문제를 개선하기 위해 클러스터내의 주요항목과 이를 이용하는 클러스터 참여도를 정의한다.

[정의 3] 주요항목

클러스터 C_i에 대한 항목의 지지도는 C_i에서 항목 i_j ($j<=n$)를 포함하고 있는 트랜잭션의 수이고, 사용자가 지정한 최소 지지도, θ ($0<\theta\leq1$)에 대해 C_i내에서 그 항목을 포함하고 있는 트랜잭션의 수가 항목의 지지도 $\text{Sup} = \theta * |C_i(T_r)|$ 이상이면 항목 i_j는 C_i의 주요항목이다[22].

$$C_i(L)i_j = |C_i(T_r)i_{j\in I}| >= \text{Sup}$$

이 때 $|C_i(T_r)|$는 클러스터 C_i에 포함된 전체 트랜잭션의 수이며, $|C_i(T_r)i_{j\in I}|$는 클러스터 C_i에 포함된, 항목 i_j를 포함하고 있는 트랜잭션의 수를 의미한다.

[정의 4] 클러스터 참여도

빈발 구조 항목으로 구성된 문서 t_k와 클러스터 C_j의 주요 항목과의 공통 항목비율이고[22] 이것은 문서 t_k가 C_j에 속할 가능성의 정도를 나타내며 다음과 같은 식으로 표현한다.

$$p\_Allo(t_k \Rightarrow C_j) = \frac{|t_k| \cap C_j(L)}{|t_k|} >= \omega\ (0<\omega<1 : \text{최소 참여도})$$

여기서 $|t_k|$는 삽입되는 문서 t_k에 포함된 항목의 수이다.

클러스터 참여도는 이 논문에서 두 번 사용된다. 첫 번째는 모든 클러스터에 대한 *Gain*을 산출하고, 이를 이용하여 할당 가능성을 비교하는 데 소요되는 시간을 줄이기 위하여 ω_1의 클러스터 참여도를 만족하는 클러스터에 대해서만 *Gain*을 산출하기 위해서 사용한다. 그리고 두 번째는 새로운 클러스터에 대한 *Gain*이 가장 클 때 새로운 클러스터를 생성하기 이전에 클러스터의 응집도를 고려한 최소 참여도 ω_2를 만족하는 기존 클러스터가 존재하는지 검사하기 위하여 사용되며, 만약 존재하면 새로운 클러스터를 생성하지 않고 최대의 참여도를 갖는 기존 클러스터에 해당 문서를 할당한다. 이것은 이미 존재하는 클러스터에 대한 할당 가능성을 검사하여 적정 수의 클러스터 생성을 유도한다. ω_2가 크면 클러스터 생성 가능성이 줄어들고, ω_2를 작게 하면 클러스터의 생성 가능성이 커지는 반면 클러스터의 응집도는 작아질 수 있다.

4.2 클러스터 갱신을 위한 차분 연산

새로운 문서의 삽입에 대하여 적당한 클러스터를 빠르게 발견하기 위한 방법으로 기존 클러스터의 할당 이익에 대한 차분 연산을 다음과 같이 정의한다.

[정의 5] 차분 연산

기존 클러스터에 대하여 삽입되는 트랜잭션의 항목들에 대한 클러스터 할당 이익의 변화 정도에 대한 연산을 차분 연산[22]이라 하고 식으로 나타내면 다음과 같다.

$$Diff\_Gain(\Delta^+) = New\_Gain(C_i) - Old\_Gain(C_i) \quad (1)$$
$$= \frac{T'(C_i)}{W'(C_i)^2} \times (|C_i(T_r)|+1) - \frac{T(C_i)}{W(C_i)^2} \times |C_i(T_r)|$$

기존 클러스터에 대한 새로운 트랜잭션의 삽입에 대하여 고유 항목의 수, 총 항목의 수, 해당 트랜잭션의 수를 W', T', $|C_i'(T_r)|$로 표시하고, 식 (1)에 의하여 기존 클러스터 할당 이익에 대한 차분을 계산한다.

(그림 3)은 각 XML 문서에서 추출된 구조적 특성을 기반으로 차분 연산을 이용하여 XML 문서를 클러스터링 하는 알고리즘을 보여준다. 삽입되는 트랜잭션에 대한 클러스터 할당은 주어진 클러스터 참여도(ω_1)를 만족하는 이미 존재하는 클러스터들 중 가장 큰 차분 값을 갖는 클러스터에 할당한다. 그러나, 만약 새로운 클러스터를 생성하는 경우에 대한 *Gain* 값이 더 클 경우에, 기존 클러스터에 대한 참여도가 클러스터의 응집도를 고려한 기준치(ω_2)를 만족한다면 새로운 클러스터를 생성하지 않고 기존 클러스터에 할당하여 생성되는 클러스터의 수를 조절한다.

```
• Insert transaction t
  extract representative structure using sequence pattern mining ;
  while not end of the existing cluster and p_Allo(C) >= w1//(w1=0.2)
    find a cluster(Ci) maximizing Diff_Gain(C) ;
    find a cluster(Cj) maximizing p_Allo(C) ;
    if new cluster Gain((Ck) > Diff_Gain(Ci)
      if p_Allo(Cj) >= w2 // (w2=0.5)
        allocate t to an existing cluster Cj ;
      else allocate t to an new cluster Ck ;
    else allocate t to an existing cluster Ci ;
```

(그림 3) 차분 연산에 의한 XML 문서 클러스터링 알고리즘

5. 클러스터링 기반 구조 검색

유사 구조 기반의 XML 클러스터링은 관심 있는 구조에 대해 유사도가 높은 클러스터로 검색 범위를 줄임으로써 결과의 정확성과 신속성을 기할 수 있으므로 XML 문서의 구조 검색에 효율적으로 적용할 수 있다.

주어진 질의 구조 Q와 가장 유사한 구조의 XML 문서를 검

색하기 위해서 유사도 측정이 필요하다. XML의 구조는 명시적으로 표현되는 엘리먼트의 계층 관계로 이루어져 있다. XML를 표현하는 트리 구조에서 계층 관계는 부모-자식 관계의 직접적으로 연결된 에지의 구성을 기초로 한다. 구조적 유사성을 측정하기 위해서는 에지(edge)의 단순한 연결과 더불어 특정 노드에 대해 루트 노드부터 대상 노드까지의 엘리먼트의 순차적 연관성도 함께 고려되어야 한다.

유사 구조의 XML 클러스터링을 기반으로 하는 구조 검색 과정은 세 단계로 구성된다. 첫 번째 단계는 사용자로부터 입력받은 질의 구조를 (그림 2)와 같이 구성되는 내부의 엘리먼트 매핑 테이블에 의해 엘리먼트를 단순화시킨다. 두 번째 단계는 재명명된 질의 구조 Q와 각 클러스터를 대표하는 구조와의 유사도를 측정하여 가장 유사한 구조의 클러스터를 발견한다. 그리고 세 번째 단계는 가장 유사한 구조의 해당 클러스터에서 질의 구조와 가장 근접한 구조를 포함하는 문서들을 유사도에 따라 사용자에게 제공한다.

이와 같은 구조 검색 과정에서 가장 중요한 것은 사용자가 입력한 구조와 XML문서와의 유사성 계산이다. XML 문서는 트리 모델(tree model)로 표현되는 계층적 구조이므로 이 논문에서는 트리의 에지와 경로를 동시에 고려하여 유사성을 측정한다. 구조적 유사성 측정을 위한 정의는 다음과 같다.

[정의 6] 에지 유사성

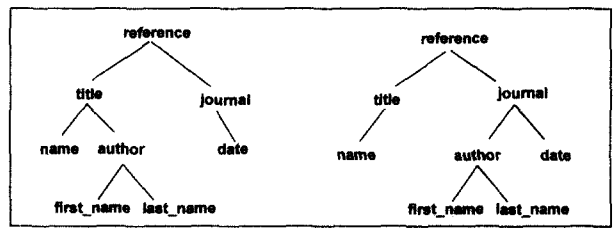
트리 구조의 서브 구조에 대한 에지의 수에 대하여 공통 에지의 수에 대한 정도를 에지 유사성이라 한다. 순서화된 엘리먼트로 구성된 XML 문서의 트리 구조 T는 부모 노드와 자식 노드로 구성되는 에지($\mu \rightarrow \nu$)로 구성되며, 주어진 질의 구조 Q의 에지($\mu' \rightarrow \nu'$)에 대해 $\mu = \mu'$, $\nu = \nu'$를 만족하면 공통의 에지라고 한다. 그리고 이러한 공통의 에지 수의 정도에 의해 유사성을 측정한다. 유사성 측정을 위한 식은 다음과 같이 표현한다.

$$EageSim(Q, T) = \frac{|E_Q \cap E_T|}{|E_Q \cup E_T|}$$

이 때 $|E_Q \cup E_T|$는 중복없이, 트리 구조 Q, T를 구성하는 전체의 에지 수를 의미하고, $|E_Q \cap E_T|$는 Q, T에 존재하는 공통의 에지 수를 의미한다. 그러므로 공통의 에지 수가 많을수록 유사도가 높다.

(예 1) (그림 4)는 에지의 유사성 측정 방법을 보이기 위해 예제 XML 문서의 구조를 트리로 표현한 T1, T2이다.

(그림 4)의 T1, T2는 다음과 같은 에지의 집합으로 구성된다(엘리먼트를 첫 알파벳으로 간단히 표기한다).



(a) T1 (b) T2

(그림 4) 유사 XML 문서의 트리 구조

E_{T1} = {r→t, r→j, t→n, t→a, j→d, a→f, a→l}
E_{T2} = {r→t, r→j, t→n, j→a, j→d, a→f, a→l}

그러므로 (정의 6)에 의해 T1, T2에 대한 에지의 유사성은 다음과 같다.

$$EageSim(T1, T2) = \frac{|E_{T1} \cap E_{T2}|}{|E_{T1} \cup E_{T2}|} = \frac{6}{8}$$

[정의 7] 경로 유사성

주어진 순서화된 엘리먼트로 구성된 XML 문서의 트리 T의 루트 노드부터 단계적으로 노드 길이가 증가된 경로($\nu_1 \rightarrow \nu_2$, $\nu_1 \rightarrow \nu_2 \rightarrow \nu_3$, $\nu_1 \rightarrow \nu_2 \rightarrow \nu_3 \rightarrow \ldots \rightarrow \nu_N$)에 대해, 질의 구조 트리 Q의 서브 구조의 공통 경로($\nu_1' \rightarrow \nu_2' \rightarrow \nu_3'$) 크기에 대한 공통 경로의 정도를 경로 유사성이라 하고, 다음과 같은 식에 의해 유사성을 측정한다.

$$PathSim(Q, T) = \frac{Max_{com}|P_Q \cap P_T|}{Max_{path}|P_Q, P_T|}$$

여기서 $Max_{path}|P_Q, P_T|$는 질의 구조 트리 Q와 비교 대상 트리 T의 경로에서 가장 큰 경로의 길이를 의미하며, $Max_{com}|P_Q \cap P_T|$는 가장 큰 공통 경로의 길이를 의미한다.

(예 2) (그림 4)의 T1, T2에서 경로 유사성 측정을 위한 경로의 집합은

P_{T1} = {r→t, r→t→n, r→t→a, r→t→a→f,
r→t→a→l, r→j, r→j→d}
P_{T2} = {r→t, r→t→n, r→j, r→j→a, r→j→d,
r→j→a→f, r→j→a→l}

이고 T1, T2에 대한 경로의 유사성은 다음과 같다.

$$PathSim(T1, T2) = \frac{Max_{com}|P_{T1} \cap P_{T2}|}{Max_{path}|P_{T1}, P_{T2}|} = \frac{3}{4}$$

그리고 위와 같이 정의된 에지 유사성과 경로 유사성을 동시에 고려하는 유사성 측정을 위한 계산식은 식 (2)과 같이 표현한다.

$$Sim(Q, T) = \alpha * EageSim(Q, T) + \beta * PathSim(Q, T) \quad (2)$$
$$(\alpha + \beta = 1, \alpha \geq 0, \beta \geq 0)$$

그러므로 (예제 1)과 (예제 2)에서의 에지 유사성과 경로 유사성을 식 (2)에 적용한 T1과 T2의 구조적 유사성 Sim(T1, T2) = $0.5 * \frac{6}{8} + 0.5 * \frac{3}{4} = 0.75$ 이다.

이렇게 에지 유사도와 경로 유사도를 함께 고려하는 유사도 측정을 통해 트리 구조의 구조적 매칭 정도를 판단할 수 있으며 유사도가 1에 가까울수록 유사성이 크다는 것을 의미한다.

질의 Q의 구조에 대해 가장 유사한 구조 T를 포함하는 클러스터를 탐색할 때 각 클러스터를 대표하는 주요 항목의 구조에 대한 비교를 통해 유사도를 계산하고, 가장 큰 유사도의 구조를 포함하는 클러스터내의 문서들에 대해, 문서의 대표 구조와 질의 구조를 비교하여 유사도 순위를 나타낸다.

6. 실험 및 적용

이 장에서는 제안하는 클러스터링 알고리즘의 성능을 평가한다. 그리고 구조 검색 인터페이스를 구현하여 클러스터링을 기반으로 하는 검색의 적용 결과를 보인다.

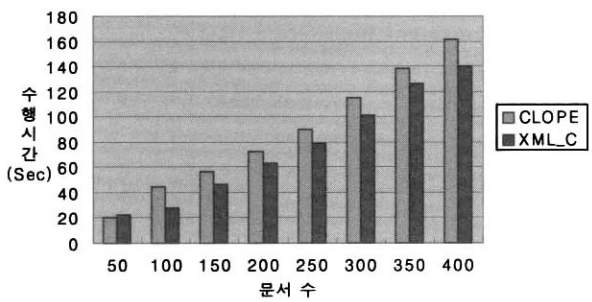
6.1 클러스터링 실험

제안 알고리즘 XML_C의 효율성을 측정하기 위해 기존 알고리즘 CLOPE[18]과의 실험을 하고 그에 대한 비교 결과를 기술한다. 실험을 위한 데이타는 위스콘스 XML 데이터 뱅크[23]에서 제공하는, 각 분야별 다른 DTD로 구성된 문서(책, 클럽, 연극, 경매, 회사, 학과, 배우, 영화 등)에 대해 총 400개의 문서를 가지고 클러스터링을 수행하였다.

우선 각 문서에서 대표 구조를 추출하기 위해 각 문서 전체 경로에 대한 빈발 구조 비율을 0.2로 하여 3장에서 설명하였던 순차패턴을 이용하였다. 대표 구조의 추출 결과는 최대 빈발 패턴의 평균 구조 길이는 5.4이었고 한 문서에서의 빈발 구조는 평균 4.9개의 빈발 패턴 구조가 추출되었다. 또한 중복되지 않도록 하면서 최대 빈발구조 길이의 80% 이상의 구조도 빈발 구조에 포함하여 차분 연산을 이용한 클러스터링을 수행하였다(클러스터 참여도에 대한 가중치로는 ω_1를 0.2, ω_2를 0.5로 하였다).

(그림 5)는 문서 수 증가에 따른 클러스터링의 평균 수행시간의 비교를 보여준다.

CLOPE은 제안 알고리즘보다 평균적으로 더 많은 수행시간이 소요되는 것을 알 수 있다. 그러나 실험 초기에는 XML_C이 CLOPE보다 약간의 차이를 보이며 더 많은 시간이 소요되는 것을 볼 수 있다. 이것은 클러스터의 주요항목을 구성하는 데 소요되는 시간이 주된 원인으로써, 데이터 량이 작을 때는 주요항목을 이용하는 것이 적절하지 않음을 의미한다. 반면에 문서의 수가 증가할수록 주요항목을 이용하는 것이 수행시간에 효과가 있다는 것을 보여주는 것이다. 그것은 주요항목을 이용하여 최소의 클러스터 참여도를 만족하는 클러스터에 대해서만 차분의 할당 이익을 비교하므로 클러스터의 할당을 위한 $Gain$의 비교 시간을 감소시킬 수 있기 때문이다.

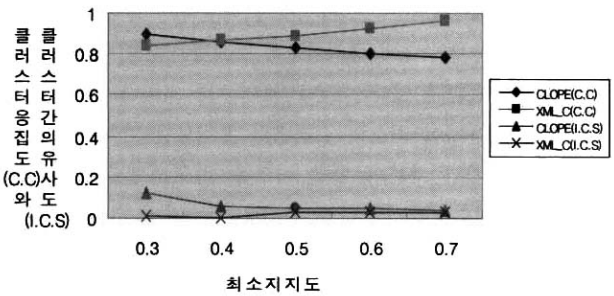


(그림 5) 문서 수의 증가에 따른 수행시간

CLOPE과 XML_C의 알고리즘에 의해 클러스터링을 수행한 후 클러스터의 정확도 및 유효성을 나타내는 클러스터의 응집도와 클러스터간의 유사도에 대한 비교 실험을 하였다. 이 실험에서 클러스터 C_i의 응집도 $Coh(C_i)$는 클러스터 C_i에 포함된 전체 항목 $T(C_i)$에 대한 주요항목이 차지하는 비율을 기준으로 계산한다. 따라서, 클러스터의 응집도를 계산하기 위한 식은 $Coh(C_i) = \frac{C_i(L)}{T(C_i)}$ 이고, 1의 값에 가까울수록 좋은 응집도를 나타낸다. 또한 클러스터 C_i, C_j의 클러스터간의 유사도 $Sim(C_i, C_j)$는 주요 항목 집합에 대한 공통의 주요항목의 비율을 기준으로 하며, 이에 대한 계산은 $Sim(C_i, C_j) = \frac{L(C_i \cap C_j) \times \frac{|L(C_i \cap C_j)|}{|L(C_i + C_j)|}}{C_i(L) + C_j(L)}$ 으로 하고, 0에 가까울수록 거의 유사성이 없는 좋은 클러스터를 나타낸다. 여기서 $L(C_i \cap C_j)$는 공통 항목에 대한 각 클러스터에서의 발생 횟수이고, $|L(C_i \cap C_j)|$은 공통 항목들의 누적 발생 횟수, $|L(C_i + C_j)|$는 주요항목의 전체 누적 횟수를 나타낸다.

이와 같이 클러스터 응집도와 클러스터간의 유사도를 측정하기 위해서는 클러스터의 주요항목을 이용하는 데 기존 알고리즘에서는 주요항목 개념을 사용하지 않으므로, 클러스터링 수행 결과의 클러스터들에 대해 지지도를 만족하는 주요항목을 같은 방식으로 추출하고 이것을 응집도와 유사도 공식에 적용하여 우리의 알고리즘과 비교하였다. (그림 6)은 최소 지

지도의 변화에 따른 클러스터의 응집도와 클러스터간의 유사도를 나타낸 것이다.



(그림 6) 클러스터의 응집도와 클러스터간의 유사도

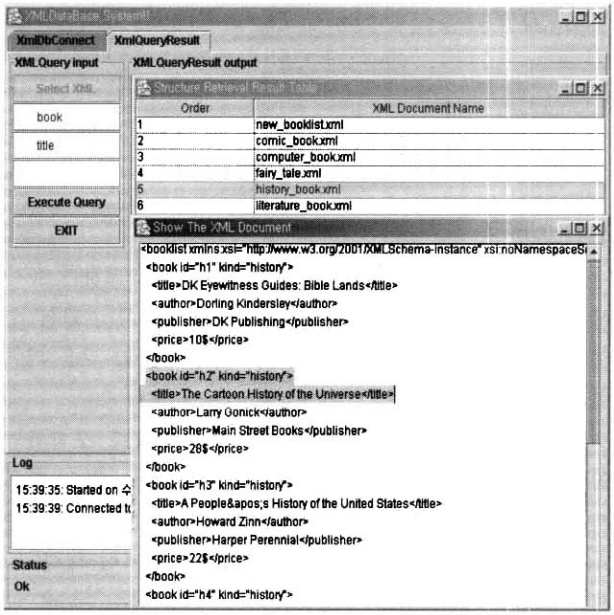
CLOPE과 XML_C에 대한 클러스터의 응집도와 클러스터간의 유사도를 비교해 보면 XML_C이 평균적으로 높은 응집도의 분포를 나타내고, 클러스터간의 유사도에서도 CLOPE보다 더 낮은 결과를 보이므로 클러스터간의 구별성이 좋으면서 유사 구조의 문서가 잘 밀집되어 있는 양질의 클러스터가 생성되었음을 확인할 수 있다.

한편 XML_C의 실험 결과를 살펴보면 클러스터 응집도와 클러스터간의 유사도 모두를 고려할 때 지지도 0.4이하에서 가장 좋은 결과를 보인다. 또한 클러스터의 응집도와 클러스터간의 유사도의 결과가 비례하는 것을 볼 수 있다. 이것은 지지도의 증가에 따라 클러스터를 유지하기 위해서 주요항목을 포함하는 더 많은 문서가 요구되므로 이를 만족하지 못하는 경우 적정 크기 이상의 클러스터 생성이 증가하고 그로 인해 클러스터간의 유사도가 증가하게 되는 것으로 생각된다. 그러므로 양질의 클러스터 생성을 위해서는 반복적인 실험을 통해 적절한 지지도의 선택이 중요하다.

6.2 구조 검색 적용

사용자 질의 구조에 대한 XML의 구조 검색을 위해 계층 구조의 세 개의 엘리먼트까지 입력이 가능한 사용자 인터페이스를 구현하였고 클러스터링 실험에 사용된 문서들을 오라클 9i의 XML DB에 저장하여 검색을 수행하였다. (그림 7)에서 화면의 왼쪽은 사용자가 질의 구조를 입력할 수 있는 창이고, 화면의 오른쪽은 그에 대한 검색 결과를 보여준다. (그림 7)에서는 엘리먼트 book/title의 구조로 구성된 질의 구조의 입력에 대한 검색 결과를 나타내며, 5장에서 제시한 구조적 유사성 측정을 통해 질의 구조와 가장 유사한 구조의 문서들을 유사도의 순위와 함께 보여준다. 그리고 문서 리스트에서 사용자가 특정 문서를 선택하면 그 문서의 전체 내용을 오른쪽 아래의 화면에 제공한다.

(그림 7)의 구조 검색은 유사 구조의 문서를 미리 클러스터



(그림 7) 사용자 구조 질의와 XML 문서의 검색 결과

링하고 그것을 기반으로 검색을 수행한다. 즉, 질의 구조에 대해 모든 문서와의 유사도 측정을 하지 않고 먼저 각 클러스터를 대표하는 주요 구조와의 유사성 측정을 통해 유사 구조의 클러스터를 발견하고 그 클러스터에 속하는 문서들에 대한 유사도를 측정하여 가장 유사한 구조의 문서를 검색한다. 따라서 비교 대상의 범위가 해당 클러스터로 축소되므로 전체의 문서를 대상으로 하는 것보다 검색속도가 빠르다. 이러한 클러스터링 기반의 구조 검색은 구조가 주어져 있지 않는 대량의 XML 문서에 대한 구조 검색이나, 상세한 구조보다는 전체적인 문서의 구조 및 형식을 발견하고자 하는 검색에 효율적으로 적용할 수 있다. 또한 질의 구조와 가장 높은 유사도의 클러스터내에 있는 관련된 구조의 문서에 대한 추천도 가능하다.

7. 결 론

최근 XML기반의 응용 범위가 확장되면서 XML 문서가 증가하고 있고 이에 따라 XML 문서를 대상으로 하는 검색의 필요성이 커지고 있다. 이 논문에서는 다양한 구조의 XML 문서에서 엘리먼트의 구조를 중심으로 문서를 클러스터링하고 이를 기반으로 하는 구조 검색 방법을 제안하였다. 제안 기법은 먼저, XML 문서를 구성하는 엘리먼트의 순서와 발생 빈도를 동시에 고려하여 유사 구조의 문서를 그룹화 하였다. 그리고 사용자의 구조 질의에 대한 구조 검색에서는, 구조적 유사성에 의해 구별되는 클러스터의 주요 구조와의 유사도 측정을 통해 검색의 범위를 해당 클러스터로 줄이고, 가장 유사한 구조의 XML 문서를 발견한다.

이 논문에서 사용된 클러스터링 알고리즘은 기존의 연구에

서 단지 공통 항목의 비율만을 가지고 클러스터링을 수행함에 따라 적정 크기 이상의 클러스터 생성과 클러스터간의 유사도가 낮아지는 문제점을 해결하기 위해 주요항목 개념의 클러스터 참여도를 이용하여 해결하고자 하였다. 제안하는 클러스터링 알고리즘은 기존 연구와의 비교 실험을 통해 평균적으로 더 높은 클러스터의 응집도와 더 낮은 클러스터간의 유사도의 결과를 얻을 수 있음을 확인하였다.

또한 클러스터링 기반의 구조검색에서는 구조의 유사성 측정을 위해 에지의 유사성과 경로의 유사성을 함께 고려하였고 검색 과정은 세 단계로 구성된다. 첫 번째 단계는 사용자로부터 입력받은 질의 구조를 단순화시킨다. 두 번째 단계는 단순화된 질의 구조 Q와 각 클러스터를 대표하는 구조와의 유사도를 측정하여 가장 유사한 구조의 클러스터를 발견한다. 그리고 세 번째 단계는 가장 유사한 구조의 해당 클러스터에서 질의 구조와 가장 근접한 구조를 포함하는 문서들을 유사도에 따라 사용자에게 제공한다.

이 연구는 문서의 구조적 특성을 엘리먼트에 의한 XML 문서의 경로 구조에 기반을 두므로 구조 중심의 문서 및 문서의 전체적인 형식에 의한 분류 등과 같은 유사 구조 검색에 효율적으로 적용할 수 있다.

향후 연구로는 XML 문서의 상세한 구조적 관계에 대한 질의가 가능하도록 하는 세분화된 구조 추출 및 검색의 정확도 측정을 위한 연구가 진행될 것이다.

참 고 문 헌

[1] W3C, Extensible Markup Language(XML) 1.1, http://www.w3.org/TR/xml11, W3C Working Draft. April, 2002.

[2] S. W. Kim, et al., "Indexing and Retrieval of XML-encoded Structured Documents in Dynamic Environment," Lecture Notes in Computer Science(LNCS) Vol.24, No.80, 2002.

[3] M. Garafalalos, A. G. R. Rastogi, S. Seshadri, K. Shim, "XTRACT : A System for Extracting Document Type Descriptors from XML Documents," Proceedings of the ACM SIGMOD, 2000.

[4] Z. Zhang, R. Li, S. Cao, Y. Zhu, "Similarity Metric for XML Documents," Workshop on Knowledge and Experience Management(FGWM) 2003.

[5] J. Madhavan, P. A. Bernstein, E. Rahm, "Generic Schema Matching with Cupid," Proceedings of VLDB., 2001.

[6] J. T. Wang, D. Shasha, G. J. S. Chang, "Structural Matching and Discovery in Document Databases," Proceedings of the ACM SIGMOD on Management of Data, 1997.

[7] R. Nayak, R. Witt, A. Tonev, "Data Mining and XML Documents," International Conference on Internet Computing, 2002.

[8] M. L. Lee, L. H. Yang, W. Hsu, X. Yang, "XClust : Clustering XML Schemas for Effective Integration," Proceedings of the ACM International Conference on Information and Knowledge Management, 2002.

[9] T. Asai, K. Abe, S. Kawasoe, H. Arimura, H. Sakamoto, "Efficient Substructure Discovery from Large Semi-structured Data," Proceedings of the SIAM International Conference on Data Mining, 2002.

[10] M. Zaki, "Efficiently Mining Frequent Tree in a Forest," Proceedings of the ACM SIGKDD International Conference, 2002.

[11] E. Kotasakis, "Structural Information Retrieval in XML Documents," ACM Symposium on Applied Computing (SAC), 2002.

[12] J. Widom, "Data Management for XML : Research Directions," IEEE Computer Society Technical Committee on Data Engineering, 1999.

[13] A. G. Buchner, M. Baumgarten, M. D. Mulvenna, R. Bohm, S. S. Anand, "Data Mining and XML : Current and Future Issues," Proceedings of WISE, 2000.

[14] J. W. Lee, K. Lee, W. Kim, "Preparation for Semantics-Based XML Mining," Proceedings of IEEE International Conference on Data Mining(ICDM), 2001.

[15] F. D. Francesca, G. Gordano, G. Manco, R. Ortale, A. Tagarelli, "A General Framework for XML Document Clustering," Technical report, n(8), ICAR-CNR, 2003.

[16] J. Yoon, V. Raghavan, V. Chakilam, "BitCube : Clustering and Statistical Analysis for XML Documents," Proceedings of the International Conference on Scientific and Statistical Database Management, 2001.

[17] A. Termier, M. C. Rouster, M. Sebag, "TreeFinder : A First Step towards XML Data Mining," Proceedings of IEEE International Conference on Data Mining(ICDM), 2002.

[18] Y. Yang, X. Guan, J You, "CLOPE : A fast and effective clustering algorithm for transaction data" Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2002.

[19] K. Wang, C. Xu, "Clustering Transactions Using Large Items," Proceedings of ACM CIKM-99, 1999.

[20] K. Winkler, M. Spiliopoulou, "Employing Text Mining for Semantic Tagging in DIAsDEM," KI, Vol.16, No.2, 2002.

[21] J. Pei, J. Han, B. M. Asi, H. Pinto, "PrefixSpan : Mining Sequential Pattern Efficiently by Prefix-Projected Pattern Growth," Proceedings of International Conference on Data Engineering(ICDE), 2001.

[22] J. H. Hwang, K. H. Ryu, "Incremental Clustering of XML Documents Based on Similar Structure," to be published in KISS.

[23] NIAGARA query engine, http://www.cs.wisc.edu/niagara/data.html.

황 정 희

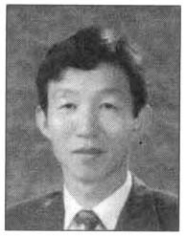
e-mail : jhhwang@dblab.chungbuk.ac.kr

1991년 충북대학교 전산통계학과(이학사)

2001년 충북대학교 대학원 전자계산학과 (이학석사)

2001년~현재 충북대학교 대학원 전자계산학과(박사과정)

관심분야 : XML, 데이터 마이닝, 능동 데이터베이스, 시공간 데이터베이스



류 근 호

e-mail : khryu@dblab.chungbuk.ac.kr

1976년 숭실대학교 전산학과(이학사)

1980년 연세대학교 공업대학원 전산전공 (공학석사)

1988년 연세대학교 대학원 전산전공 (공학박사)

1976년~1986년 육군군수 지원사 전산실(ROTC 장교), 한국전자통신연구원(연구원), 한국방송통신대 전산학과(조교수) 근무

1989년~1991년 Univ. of Arizona Research Staff(TempIS 연구원, Temporal DB)

1986년~현재 충북대학교 전기전자 컴퓨터공학부 교수

관심분야 : 시간 데이터베이스, 시공간 데이터베이스, Temporal GIS 및 지식기반 정보검색 시스템, 데이터 마이닝 및 데이터베이스 보안, 바이오 인포매틱스