XML 스키마 클러스터링을 위한 효율적인 알고리즘

임태우[0] 이경호
연세대학교 컴퓨터과학과
twrhim@icl.yonsei.ac.kr khlee@icl.yonsei.ac.kr

An Efficient Algorithm for Clustering XML Schema

Tae-Woo Rhim[0] Kyong-Ho Lee
Dept. of Computer Science, Yonsei University

요 약

최근 웹상에 산재한 정보들의 효율적인 검색과 이용을 위하여 정보의 구조를 정의하는 스키마들의 통합이 중요시되고 있다. 본 논문에서는 XML 스키마들을 클러스터링하기 위한 방법을 제안한다. 제안된 방법은 두 스키마를 통합하는데 드는 비용이 적을수록 스키마간의 유사도가 높다는 가정하에 스키마 사이의 공통된 구조의 크기를 계산한다. 이를 위해서 경로사이에 서로 대응하는 요소의 합이 최대가 되는 경로간의 일대일 매칭을 추출한다. 또한 계산된 유사도값에 기반하여 계층적 클러스터링 방법을 적용한다. 제안된 방법의 성능을 평가하기 위해서 다수의 XML 스키마를 대상으로 실험한 결과, 91%의 정확율과 93%의 재현율로서 기존의 알고리즘보다 우수하였다.

키워드 : XML 스키마, 클러스터링, 스키마 통합

1. 서 론

XML(eXtensible Markup Language) [4] 은 문서와 데이터를 구조적으로 표현할 수 있는 메타 언어이며 플래폼에 독립적이라는 특징 때문에 인터넷을 비롯한 다양한 분야에서 정보 교환을 위한 표준으로 널리 사용되고 있다. 웹상에 분포되어 있는 XML 문서들 간의 통합 검색과 변환의 필요성이 증대되면서 유사한 도메인의 스키마를 하나로 묶는 스키마 통합(schema integration)에 대한 관심이 증가하고 있다. 특히 스키마 클러스터링은 통합 스키마를 생성하기 위한 전단계로서 매우 중요하다.

XML 클러스터링에 관한 연구는 이미 여러 방법으로 진행되어 왔다. <표 1>은 클러스터링 연구결과를 간략히 요약한 것이다.

표 1. XML 문서 및 스키마 클러스터링 방법

이 름	연도	특 징	대 상
Charnyote과 Hammer [1]	2003	XML 문서 사이의 거리함수를 정의하고 거리 값에 기반한 MCM 클러스터링 방법 제안	XML 문서
Nierman과 Jagadish [5]	2002	한 XML 문서를 다른 문서로 변환하는데 필요한 변환연산의 비용을 이용하여 XML 문서들을 클러스터링 하는 방법을 제안	XML 문서
Lee 등 [2]	2001	트리로 표현된 스키마간의 언어적 유사도와 구조적 유사도를 계산하여 두 스키마 사이의 유사도 계산. 스키마 사이의 유사도를 이용하여 클러스터링 하는 방법 제안	XML DTD
Jeong과 Hsu [10]	2001	각 스키마가 통합 스키마로 통합되기 위한 변환연산의 양에 기반하여 스키마간 유사도 계산, 유사도에 기반한 계층적 클러스터링 방법 제안	XML DTD

기존 연구의 대부분은 엘리먼트 사이의 구조적 및 어휘적 유사도만을 위주로 하여 스키마 통합에 있어서 한계를 갖는다. 따라서 본 논문에서는 스키마 통합을 위한 클러스터링 방법을 제안한다. 제안된 방법은 두 스키마의 통합에 필요한 변환연산의 양에 의해 계산된 유사도 계산 알고리즘에 기반한다. 제안된 스키마 통합 과정은 <그림 1>과 같다.

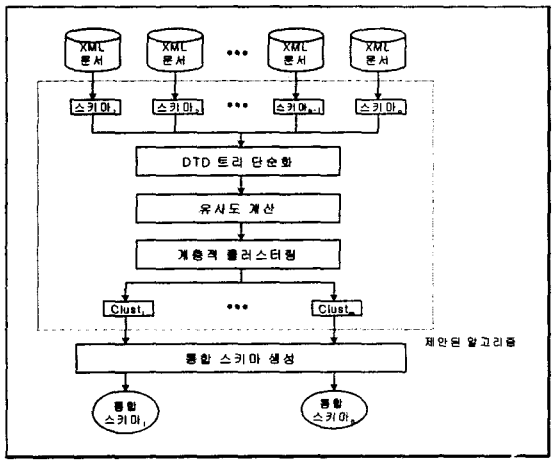

그림 1. 스키마 통합 과정.

제안된 방법은 각 스키마의 경로들을 매칭시켜 두 스키마의 공통구조를 찾아낸다. 이때 공통부분은 통합과정에서 변환이 필요하지 않은 부분이므로 그 비율이 스키마 사이의 유사도가 된다. 또한 보다 정교한 수준의 엘리먼트간 매칭을 위하여 축약어 사전과 동의어 사전을 이용한다. 제안된 알고리즘의 성능을 평가하기 위해서 4개의 카테고리에서 뽑은 실제 사용중인 스키마를 대상으로 실험한 결과, 평균적으로 91%의 정확률(precision)과 93%의 재현률(recall)을 보여 같은 실험 데이터를 사용한 기존의 연구보다 우수하였다.

한편, 본 논문에서는 XML 스키마를 표현하기 위해 트리구조에 기반한 문서모델을 제안하며 제안된 문서 모델에 따라 표현된 트리구조를 스키마 트리라고 정의한다. 스키마 트리를 구성하는 각각의 노드는 스키마의 엘리먼트를 표현한 것으로서 엘리먼트의 이름을 레이블(label)로 갖고 노드의 속성값으로 타입(type)과 빈도수를 갖는다. 타입 속성값은 해당 엘리먼트나 애트리뷰트의 데이터 타입으로 단말노드만 타입 속성값을 갖는다. 빈도수 속성은 빈도 연산자(none, *, +, ?) 중 한 개 혹은 (최소값, 최대값)의 쌍을 값으로 갖는다.

2. 제안된 클러스터링 알고리즘

본 절에서는 제안된 XML 스키마 클러스터링 알고리즘을 자세히 기술한다. 제안된 방법은 단순화, 유사도 계산, 그리고 클러스터링의 세 단계로 구성된다

2.1 단순화

XML 스키마간의 유사도 비교를 위해서 먼저 스키마를 스키마 트리로 모델링한다. 그런데 스키마에서 선택 연산자(choice operator)로 연결되어 있는 부분은 스키마 트리에서 많은 후보 트리들을 생성한다. 그러므로 구조적 정보의 손실을 최소화하는 범위에서 선택 연산자를 순서(sequence) 연산자로 변환한다. 이를 위하여 제안된 방법은 Lee 등이 제안한 단순화 규칙을 적용한다.

2.2 유사도 계산

본 논문에서 제안된 스키마 사이의 유사도는 두 스키마를 포괄하는 통합 스키마를 생성하는데 필요한 비용에 기반한다. 즉, 임의의 두 스키마를 통합 스키마로 변환할 때 변환 비용이 적게 들수록 유사하다고 간주한다. 따라서, 제안된 유사도는 식 (1)과 같이 두 스키마간의 공통된 구조의 비율로 정의한다.

$$\text{스키마 트리간 유사도} = \frac{\text{공통구조}}{|T_s|+|T_t|-\text{공통구조}} \quad (1)$$

$|T_s|$: 소스스키마트리의노드들의집합
$|T_t|$:타겟스키마트리의노드들의집합

스키마 트리간의 유사도를 구하기에 앞서 먼저 두 노드 Ns와 Nt에 대한 언어적 유사도(Ns, Nt)를 정의한다. 언어적 유사도는 두 노드의 레이블간의 유사도를 정의하며, 수식 (2)와 같이 노드의 이름과 빈도수 및 타입 속성의 유사도 값을 이용해 계산한다. 유사도 계산에 사용되는 스키마가 XML DTD일 경우 노드간의 데이터 타입이 존재하지 않으므로 w_3은 0가 된다.

$$\begin{aligned}\text{언어적 유사도(Ns, Nt)} = {} & w_1 \times \text{이름 유사도(Ns, Nt)} \\ & + w_2 \times \text{빈도 유사도(Ns, Nt)} \\ & + w_3 \times \text{타입 유사도(Ns, Nt)} \\ & \text{(단, } w_1 + w_2 + w_3 = 1)\end{aligned} \quad (2)$$

이름 유사도는 노드의 레이블간의 어휘적인 유사도를 나타낸다. 먼저, 각 노드명을 대문자나 특수문자를 기준으로 토큰화하여 각 토큰 사이의 유사도를 계산한다. 노드명 사이의 이름 유사도는 수식 (3)과 같이 토큰 사이의 유사도의 합을 전체 토큰 수로 나눈 값으로 정의한다.

$$\text{이름 유사도}(Ns,Nt) = \frac{2\times \sum \text{유사도}(Ns_i,Nt_j)}{|Ns|+|Nt|} \quad (3)$$

Ns_i:소스노드의 토큰,$1\le i\le n$ Nt_j:타겟노드의 토큰,$1\le j\le m$

토큰 사이의 유사도를 계산하기 위해 축약어 사전과 동의어 사전[8]을 이용한다. 비교할 토큰명이 축약어 사전에 포함되 있을 경우, 해당 축약어의 전체이름으로 대체하여 비교한다. 또한 동일하지 않은 두 토큰에 대하여는 동의어사전을 검색하여 동의어 관계가 존재할 경우, 두 토큰은 0.8의 유사도값을 갖는다.

두 노드의 빈도 유사도는 각 노드의 빈도수 사이의 유사도로 나타낸다. 각 빈도 지시자의 최소, 최대치에 기반하여 0.7에서 1 사이의 유사도를 할당한다.

타입 유사도는 스키마에서 단말 노드가 가지는 여러 데이터 타입간의 유사도를 변환에 의한 정보 손실의 정도에 기반하여 정의한다. 제안된 방법은 타입 속성간의 유사도를 정보의 손실 정도에 따라 '동일', '비손실 변환가능', '손실 변환가능', '변환불가'의 4단계로 구분하여 3.0, 2.0, 1.0, 0.0의 값을 할당한다.

한편, 두 트리의 유사도를 구하기 위해 먼저 두 트리의 루트 노드로부터 각 트리의 단말 노드들로 가는 경로들을 추출한다. 또한, 경로들 간의 LCS(Longest Common Subsequence) [9]를 계산한다. LCS는 두 경로상의 최장 공통경로를 의미하며 각 경로의 노드 사이의 1:1 매칭을 통하여 추출한다.

제안된 알고리즘은 두 스키마간의 공통구조를 계산하기 위해서 LCS의 합이 최대에 해당하는 경로간의 일대일 매칭을 찾는다. 이때 계산된 LCS의 합을 GCS(Greatest Common Subset)라고 정의한다. 제안된 알고리즘에서는 GCS를 계산하기 위해 그래프 G(V,E)를 모델링한다. 그래프 G(V,E)에서 V는 노드의 집합으로서 소스 및 타겟 트리의 각 경로를 의미하며 E는 노드간의 간선으로 경로간의 LCS의 크기를 값으로 갖는다. 그래프의 노드는 소스트리 및 타겟트리에 해당하는 두 종류로 구분된다. 즉, n개의 소스 경로와 m개의 타겟 경로로 구성된 그래프 G(V,E)는 가중치 이분 그래프(weighted bipartite graph) $K_{n,m}$에 해당한다. 즉, GCS를 계산하는 문제는 $K_{n,m}$에서 최대 이분 매칭(maximal bipartite matching)을 찾는 문제에 해당한다 [9]. 제안된 알고리즘에 대한 자세한 기술은 <그림 2>와 같다.

```
Algorithm FindGCS
입 력 : 스키마 Trees - 소스 스키마 Ts와 타겟 스키마 Tt
        Threshold t
출 력 : Ts 와 Tt 사이의 GCS
1. Ts와 Tt의 루트로부터 각 단말노드로의 경로 SPathi, 0≤i≤
   n, 와 TPathj, 0≤j≤m, 를 추출한다.
2. 각 경로 SPathi와 TPathj간의 LCS를 계산한다.
3. 각 경로 SPathi와 TPathj를 트리의 노드로, 각 경로간 LCS
   의 크기를 그 경로를 나타내는 노드간 간선의 웨이트로 갖도
   록 가중치 이분 그래프 Kn,m를 모델링한다.
4. Kn,m에서 최대 이분 매칭을 찾는다. 이때의 LCS의 합이
   GCS가 된다.
```

그림 2. 제안된 GCS 계산 알고리즘.

이렇게 구해진 GCS의 크기를 두 트리의 전체 노드의 수로 나눈 것이 두 트리의 유사도 값이 된다. 수식 (1)을 제안된 GCS에 따라 표현하면 다음 수식 (4)와 같다.

$$\text{스키마트리간 유사도} = \frac{|T_{gcs}|}{|T_s| + |T_t| - |T_{gcs}|} \quad (4)$$

2.3 클러스터링

본 절에서는 스키마간의 유사도에 기반한 계층적 클러스터링 방법[7]을 제안한다. 제안된 방법은 유사도 행렬을 이용해 각 스키마들의 클러스터를 나타내는 클러스터 배열을 생성한다.

```
Algorithm Clustering
입 력 : 스키마 Tree set S
        스키마 Similarity matrix S[n][n]
        Threshold Thcluster
출 력 : Cluster array C[n]
1 :  입력된 XML 스키마들을 single 클러스터로 할당한다.
2 :  스키마 유사도 행렬 S[n][n] 중에서 최대값 max을 선택한다.
3 :  If (max > Thcluster)
4 :  {
5 :    if ( (C[row] == C[col] )          // 같은 클러스터에 포함
6 :      유사도 행렬을 수정
7 :    else                              // 다른 클러스터
8 :    {
9 :      colth 클러스터에 속한 스키마들을 rowth 클러스터로 옮김
10 :     유사도 행렬과 클러스터 배열을 수정
11 :   }
12 : }
13 :  2-12 과정을 유사도 행렬중 최대값이 Thcluster보다 작아질
      때까지 반복한다
```

그림 3. 제안된 클러스터링 알고리즘

먼저 입력된 XML 스키마들을 각각 단일 클러스터로 할당한다. 그리고 주어진 스키마간의 유사도 행렬에서 가장 높은 유사도를 지니는 두 개의 스키마를 검색한다. 유사도가 $Th_{cluster}$보다 클 경우에만 클러스터링 과정을 적용한다. 적용되는 두 개의 스키마 모두 같은 클러스터에 포함되어 있을 경우, 두 스키마는 이미 클러스터링이 된 경우이므로 유사도값을 0으로 변경한다. 만일, 두 스키마의 클러스터 배열이 서로 다른 경우엔 다른 클러스터에 포함되어 있을 경우이므로 각 스키마를 포함된 두 개의 클러스터를 병합하고 두 스키마 사이의 유사도값을 0으로 초기화한다.(<그림 3>의 9~10번째줄 참조)

클러스터링 과정에서 유사도 행렬을 변경하게 되는데 유사도 행렬의 최대값이 $Th_{cluster}$보다 작아질 때까지 위의 과정을 반복한다. 유사도 행렬의 모든 유사도 값이 $Th_{cluster}$보다 작아지면 클러스터링 작업을 끝내고 클러스터 배열을 결과값으로 반환한다. *제안된 계층적 클러스터링 알고리즘은 <그림 3>과 같다.*

3. 실험 결과

본 절에서는 제안된 방법의 정확도와 시간 복잡도를 분석하였다. 제안된 방법의 성능을 평가하기 위해 <표 2>와 같이 Lee 등의 실험 데이터중 일부를 이용하여 실험하였다. 데이터는 100여개의 DTD로 구성되어 있으며 4개의 도메인에 속해있다.

표 2. 실험 데이터

Domain	No. of DTDs	Domain	No. of DTDs
Travel	43	Publication	20
Patient	20	Hotel	26

본 논문에서는 <표 4>에서 나타낸 데이터를 대상으로 실험을 수행하였다. 각 스키마간의 유사도에 기반하여 계층적으로 클러스터링한 후 그 결과를 실제 스키마들의 도메인과 비교하였다. XML DTD를 대상으로 한 실험으로서 DTD에서는 타입 속성이 존재하지 않으므로 가중치는 $(w_1, w_2, w_3) = (0.7, 0.3, 0)$으로 설정하였다. 클러스터링에 사용되는 유사도의 임계값을 바꿔가며 4번의 실험을 진행하였다. 임계값의 변화에 따라 클러스터링의 정확률과 재현율은 <그림 4>와 같다.

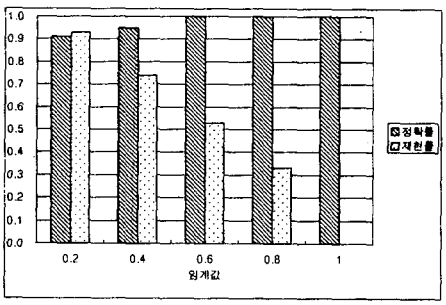

그림 4. 임계값에 따른 알고리즘의 정확률과 재현율

제안된 방법은 평균적으로 91%의 정확율을 보이고 있다. 정확률은 잘못 클러스터링된 스키마들을 제외한 비율로서 정확율이 높을 경우, 스키마 통합 시스템은 더 적은 비용으로 통합 스키마를 생성할 수 있다. 또한 재현율은 스키마들이 처음에 속한 그룹으로 클러스터링 되는 스키마들의 비율로 정확율과 함께 클러스터링 알고리즘이 얼마나 잘 작동하는지를 나타내는 수치이다.

실험 결과, 제안된 방법은 기존의 클러스터링 방법과 비교하여 보다 우수한 성능을 보였다. <표 3>에서 나타나듯이 각각의 임계값에 대해서 제안된 알고리즘이 Lee 등의 방식보다 같거나 높은 수준의 정확도를 보였다. 또한 재현율 면에서도 기존 방식보다 향상되었다.

또한, 각 방법에 대한 시간 복잡도를 <표 4>에 나타내었다. 제안된 알고리즘은 단말노드로부터 경로를 추출하는 방법을 사용하여 $O(n^2e^3)$의 복잡도를 가지며 특성상 깊이(depth)가 깊은 트리에서 더 나은 성능을 보인다. 또한 경로간의 매칭을 통해 두 스키마간의 공통부분을 구함으로서 스키마 통합에 필요한 변환의 양을 정확하게 구해낼 수 있다. 그러므로 본 논문에서 제안하는 방법은 기존 연구보다 스키마 통합에 적합하다.

표 3. 각 방법론의 잘못 클러스터링된 스키마의 수 및 재현률

임계값		0.2	0.4	0.6	0.8
제안된 알고리즘	mis-clustering	4	2	0	0
	재현률	0.93	0.74	0.53	0.33
Lee 등	mis-clustering	5	3	0	0
	재현률	0.90	0.61	0.47	0.23

표 4. 각 방법의 시간 복잡도

	Lee 등	Jeong과 Hsu	제안된 알고리즘
유사도 계산	$O(n^2m^2e^4)$	$O(n^2Km^2)$	$O(n^2e^4)$
계층적 클러스터링	O(n^3)		

n : 실험에 사용된 스키마의 수, m : 스키마가 포함한 노드의 수, e : 스키마의 단말 노드수, K : 각 노드의 평균 자식수

4. 결론 및 향후 연구방향

기존의 클러스터링에 관한 연구는 스키마의 노드의 언어적, 구조적 정보를 이용하여 유사도를 계산하였으며 클러스터링 후 실제 통합 스키마 형성에 직접 사용하기에는 한계가 있었다.

제안된 방법은 경로간의 공통 순서를 찾아 스키마간의 최대 공통 구조를 계산함으로서 가장 변환이 용이한 스키마 통합을 위한 클러스터를 생성한다. 또한 노드가 아닌 각 스키마 경로간의 매칭을 수행함으로서 시간 복잡도를 줄인다. 또한 매칭결과의 정확성을 높이기 위해 축약어 사전과 동의어 사전을 적용하였다. 실험 결과, 제안된 방법은 기존 방법보다 높은 정확률을 보였다.

향후 연구로는 더욱 정확하고 효율적인 클러스터링을 위해 경로간 뿐만 아니라 서브트리간의 매칭관계를 추출할 수 있는 방법을 연구할 것이다. 또한 이렇게 클러스터링 과정에서 추출된 정보를 이용하여 그룹핑된 여러 스키마들을 통합 스키마로 변환하는 방법을 연구할 것이다.

5. 참고문헌

[1] Charnyote Pluempitiwiriyawej and Joachim Hammer, "Element Matching across Data-oriented XML Sources Using a Multi-strategy Clustering Model", Data & Knowledge Engineering, Sep, 2003,

[2] M. Lee, L. Yang, W. Hsu and X. Yang, "XClust : Clustering XML Schemas for Effective Integration", Proc. 11th Int. conf. on Information and knowledge management, pp. 292 - 299, Nov, 2002

[3] Damien Guillaume and Fionn Murtagh, "Clustering of XML Documents", Computer Physics Communications 215-227, May 2000

[4] World Wide Web Consortium, Extensible Markup Language (XML) 1.0 (Second Edition), W3C Recommendation, http://www.w3c.org/TR/REC-xml, 2000.

[6] Euna Jeong, Chun-Nan Hsu, "Induction of Integrated View for XML Data with Heterogeneous DTDs", CIKM 2001: 151-158

[5] Andrew Nierman and H. V. Jagadish, "Evaluate Structural Similarity in XML Documents", WebDB, 61-66, 2002

[7] E. Gose, R. Johnsonbaugh and S. Jost, "Pattern Recognition and Image Analysis", Prentice Hall

[8] George A. Miller, "WordNet: A Lexical Database for English," Communications of the ACM, Vol. 38, No. 11, pp. 39-41, 1995.

[9] T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Strin, "Introduction to Algorithms", 2nd edition, The MIT Press. 2001