XML 문서의 자동변환을 위한 효율적인 스키마 매칭 알고리즘

이준승° 이경호
연세대학교 컴퓨터과학과
jslee°@icl.yonsei.ac.kr khlee@cs.yonsei.ac.kr

An Efficient Schema Matching Algorithm for An Automated Transformation of XML Documents

Jun-Seung Lee Kyong-Ho Lee
Department of Computer Science, Yonsei University

요 약

본 논문에서는 XML 문서의 자동변환을 위해 2단계의 상향식 매칭 방법을 제안한다. 제안된 방법은 단말노드 사이의 유사도 비교를 통해 임계값을 넘는 후보 매칭집합을 결정하고, 단말노드가 포함되어 있는 경로의 유사도 비교를 통해 적절한 일대일 매칭을 추출한다. 특히, 노드 사이의 유사도 비교를 위해 축약어 사전, 일반 동의어 사전, 도메인 온톨로지를 적용한다. 실제 전자상거래용 XML 스키마를 대상으로 실험한 결과 제안된 방법은 평균적으로 97%의 정확률을 보였다.

1. 서 론

스키마 매칭에 대한 연구는 정보의 통합이나, 변환, 공통된 인터페이스의 적용등의 다양한 활용을 위해 오래전부터 여러 방법으로 연구되고 있다. 본 논문에서는 XML[1] 문서의 자동변환을 위해 소스와 타겟이 되는 두 스키마에서 의미적으로 연관이 있는 단말노드 사이에 일대일(one to one) 매칭을 찾아내는 방법을 제안하고자 한다.

대부분의 기존 연구 방향과는 달리 먼저 단말노드간의 매칭을 통해 후보매칭 집합을 찾고 이 중에서 구조적인 정보, 즉 경로의 유사도를 비교하여 가장 적절한 일대일 매칭을 찾는 상향식 방법을 사용한다. 이 방법은 후보매칭을 찾고 그 좁혀진 탐색범위 안에서 적절한 일대일 매칭을 찾는 것으로 다른 방법에 비해 효율적이고 정확한 결과를 나타낸다.

또한, 각 노드사이에 유사도를 계산하기 위해서 문자열 비교뿐 아니라, 축약어 사전 및 일반동의어 사전[2]을 활용한 시스템을 제안하였다. 특히 본 시스템은 도메인 온톨로지를 정의하여 특정한 도메인에 대해 어휘들간의 관계를 수치화 하여 적용시킬 수 있다.

본 논문에서는 알고리즘의 평가를 위해 5개의 실제 전자상거래에서 사용되고 있는 실험데이터를 가지고 여러 측면에서 실험하였다. 평균 97%의 정확률과 81%의 재현률로 높은 정확률을 보이고 있다. 본 논문의 구성은 2장에서의 간단한 관련연구와 3장에서는 전체적인 시스템의 구조와 각 단계별 매칭의 과정을 구체적으로 기술하였고, 4장에서는 실험 및 결과를, 5장에서는 결론과 향후연구 방향을 기술하였다.

2. 관련연구

스키마 매칭이란 두 스키마를 입력으로 받아서 의미적으로 상응하는 원소들 사이의 관계를 찾는 것이다. 스키마 매칭은 웹 기반 데이터 통합, 전자상거래, 스키마 통합, 스키마 진화 및 이동 등 많은 어플리케이션에서 중요한 역할을 차지함에 다양한 측면에서 다양한 방법으로 연구가 진행되고 있다[3].

Xtra[4]는 XML 문서의 자동변환을 위하여 최소 비용의 변환 연산에 해당하는 대응관계를 추출한다. 하지만, 변환 연산에만 치중하여 실제 그 노드가 속하고 있는 구조에 대한 정보를 고려하지 않고, 변환연산에 제한이 많아 실제 스키마에서 일어날 수 있는 삽입 연산 등은 찾지 못하게된다. 이외에도 학습기법에 기반한 LSD[5]는 소스 스키마 뿐만 아니라 XML문서에 대한 정보까지 이용하여 매칭을 시도하고 있다. 그럼에도 학습을 위해 필요한 사용자의 수동매칭에 대한 노력이나, 학습을 위한 충분하고 포괄적인 학습데이터에 대한 문제가 해결되야 한다. 이 밖에도 여러가지 방법을 콤포넌트화하여 다양하게 적용하고, 사용자의 피드백을 통해 정확성을 좀 더 향상 시킬 수 있는 시스템에 대한 연구[6]도 진행되고 있다.

3. 스키마 매칭 알고리즘

본 연구는 정보의 통합이나, 일반적인 용도의 스키마 매칭이 아니라 문서의 변환을 위한 스키마 매칭으로 일대일의 단말노드의 매칭관계를 찾아내고자 하는 것이다. 따라서 대부분의 정보를 포함하고 있는 단말노드를 중심으로 매칭을 시작하여, 그 중에 가장 타당한 매칭 관계를 선택하는 상향식 방법의 매칭을 제안한다.

3.1. 시스템 구조

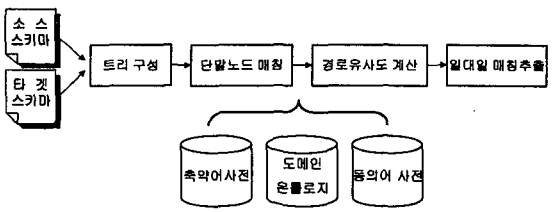

그림 1 시스템 구조

스키마 매칭 과정은 그림 1과 같이 크게 2단계로 이루어 진다. 먼저 트리로 구성된 두 스키마의 단말노드간의 매칭을 실시한다. 하지만 단말노드 매칭 과정은 n:m의 중복된 매칭관계를 보이게된다. 따라서 단말노드 매칭에서 생성된 후보매칭집합에서 단말노드가 형성하는 경로간의 유사도 비교를 통해 일대일의 최종 매칭을 선택하게 된다.

3.2. 단말노드 매칭

트리로 구성된 두 스키마의 모든 단말노드을 비교하여 유사도가 임계값 이상인 매칭관계를 찾는다. 단말노드의 유사도는 식(1)과 같이 노드의 이름을 비교하는 언어적인 유사도와 XML 스키마[7]에서 정의한 데이터 타입에 대한 유사도를 각각의 가중치를 주어 곱해진 값의 합으로 정의한다.

$$\text{단말노드유사도}(N_s, N_t) = w_l * \text{언어적유사도}(N_s, N_t) + w_t * \text{데이터타입유사도}(N_s, N_t) \quad (1)$$

• 언어적 유사도

주어진 두 노드의 이름의 유사도로 먼저 입력된 노드의 이름을 대문자나 특수기호를 기준으로 토큰화하고 시스템에서 제공하는 3가지 사전을 검색한다. 언어적 유사도는 각 토큰간의 유사도를 통해 전체 토큰중에 유사한 토큰의 유사도의 값으로 계산한다.

$$\text{언어적유사도}(Ts, Tt) = \frac{\sum \text{유사도}(Ts_i, Tt_j)}{|Ts| + |Tt|} \quad (2)$$

Ts_i: 소스 노드의 토큰, $1 \le i \le n$

Tt_j: 타겟 노드의 토큰, $1 \le j \le m$

먼저 각 토큰은 축약어 사전을 검색하여 축약어인 경우 원래 이름으로 대치되고, 각 토큰의 비교를 통해서 유사도를 계산한다. 똑같은 문자열의 토큰이면 1.0의 유사도를 반환하고, 똑같지 않다면 도메인에서 사용되는 특정한 언어간의 유사도가 정의되어 있는 도메인 온톨로지를 검색하여 정의된 유사도를 반환한다. 도메인 온톨로지에도 포함되어 있지 않은 토큰은 일반적인 동의어 사전[2]을 검색하여 동의어인 경우 0.8의 유사도를 반환한다. 일반적인 동의어 사전에도 없는 경우 토큰의 유사도는 0이 된다. 이렇게 구해진 각 토큰간의 유사도의 합을 전체 토큰의 수로 나누어 두 노드의 언어적인 유사도를 결정하게 된다.

• 데이터타입 유사도

XML 스키마 명세서에서 제공하고 있는 여러 데이터 타입들(예, string, date)들의 정보를 비교한 유사도로, 매칭시 단말노드가 포함하고 있을 정보의 손실의 정도에 따라 4단계로 나누어 유사도를 부여한다. 즉 동일한 데이터 타입의 변환의 경우(예, string → string) 가장 큰 유사도를 부여하고, 정보 비손실 변환가능(예,. int → string) , 정보 손실 변환가능(예, string → int), 변환 불가(예, date → boolean)의 단계의 따라 점점 낮은 유사도를 부여하게 된다.

3.3. 경로유사도 계산

한 단말노드가 여러 단말노드와 매칭관계를 갖을 경우 가장 유사한 매칭을 선택하기 위하여 대응관계를 갖는 단말노드의 부모노드로 부터 루트노드까지의 중간노드의 집합을 경로로 정의하고, 경로 유사도를 계산한다. 경로 유사도는 식(3)과 같이 비교 대상이 되는 두 경로에 포함되어 있는 전체 중간노드 중에 매칭되는 중간노드의 비율로 정의한다. 즉, 경로유사도 계산을 위해서 중간노드들 간의 비교가 필요하다.

$$\text{경로유사도}(Ps, Pt) = \frac{Ps\text{와 } Pt\text{사이에 대응관계를 갖는 중간노드의 수}}{|Ps| + |Pt|} \quad (3)$$

Ps: 소스 경로, Pt: 타겟 경로

경로를 따라서 포함되어 있는 중간노드간에 유사도를 계산하여 임계값 이상의 관계를 보이는 노드를 매칭되었다고 한다. 중간노드간의 유사도를 구하기 위해서 단말노드 매칭에서 사용했던 언어적인 유사도와 중간노드의 구조적인 정보를 이용하여 나타내는 구조적 유사도를 각 가중치를 곱하여 더한 값으로 나타낸다.

$$\text{중간노드유사도}(N_s, N_t) = w_l * \text{언어적유사도}(N_s, N_t) + w_s * \text{구조적유사도}(N_s, N_t) \quad (4)$$

• 구조적 유사도

중간노드의 하위 트리에 포함되어 있는 단말노드 중 매칭관계를 가지고 있는 전체 노드에 대한 두 중간노드의 하위트리 안에서 매칭되고 있는 단말노드의 비율로 나타낸다. 즉, 하위에 매칭되는 단말노드를 많이 포함하고 있는 중간노드가 그렇지 않은 중간노드보다 더 유사하다고 가정한다.

$$\text{구조적유사도}(Ns, Nt) = \frac{|\text{단말노드매칭}(LNs, LNt)|}{|LNs| + |LNt|} \quad (5)$$

LNs : Ns트리(Ns를 루트로 갖는 서브트리)의 대응관계를 갖는 단말노드의 집합

LNt : Nt트리(Nt를 루트로 갖는 서브트리)의 대응관계를 갖는 단말노드의 집합

3.4. 일대일 매칭 추출

일대일 매칭 추출과정에서는 단말노드 매칭에 의해 선택된 후보매칭의 집합에서 경로의 유사도가 가장 큰 매칭을 찾아낸다.

단말노드 매칭결과는 소스 단말노드와 타겟 단말노드를 행과 열로 하여 단말노드 사이의 유사도와 경로의 유사도를 가지고 있다. 먼저 한 소스노드와 단말노드 매칭관계를 보이고 있는 여러 타겟 노드중 가장 높은 경로 유사도의 매칭을 최종 매칭으로 선택한다. 모든 소스노드에 대해 가장 높은 경로유사도의 타겟노드를 찾는다. 하지만 소스노드를 중심으로 가장 유사한 타겟노드를 선택하다보면 중복되는 타겟노드가 선택될 가능성이 있다. 즉 n:1의 최종매칭이 나타나게 된다. 따라서, 모든 소스노드에 대해서 가장 적당한 타겟노드를 최종매칭으로 선택한 후, 타겟노드를 중심으로 다시 검색을 하여 n:1이 발생하는 최종 매칭관계를 확인하여, 한 타겟노드에 중복되어 매칭을 보이고 있는 소스노드의 매칭 중 가장 적절한 매칭만을 남겨두고 나머지 매칭은 제거하게 된다. 위의 두 단계의 과정을 거치면 일대일 관계를 나타내는 최종매칭이 선택되게 된다.

표 1. 일대일매칭추출 알고리즘

```
일대일매칭추출( 단말노드매칭테이블[ ][ ])
{
/// 하나의 소스에 대해 가장 적당한 타겟노드 선택
for( i < 소스의 단말노드의 수)
    최적경로유사도, 최적단말노드유사도 초기화;
    for( j < 타겟의 단말노드의 수)
    {
       if( 매칭[i][j].경로유사도 > 최적경로유사도 )
             매칭[i][j]를 최종매칭으로 선택;
       else if(매칭[i][j].경로유사도 = 최적경로유사도)
            if(매칭[i][j].단말노드유사도 > 최적단말노드유사도)
             매칭[i][j]를 최종매칭으로 선택;
    }
/// 위에서 선택된 최종매칭중에 여러 소스노드가 하나의 타겟노드와
/// 매칭되는 경우를 확인하여 중복된다면 가장 적당한 매칭만을
/// 판단하여 놔두고, 나머지 관계는 최종매칭에서 제외
for( j < 타겟의 단말노드의 수)
{
      for ( i < 소스의 단말노드의 수)
      if( 매칭[i][j]
      {
          if( 매칭[i][j].경로유사도 < 최적경로유사도 )
               매칭[i][j]를 최종매칭에서 제외;
          else if(매칭[i][j].경로유사도 = 최적경로유사도)
               if(매칭[i][j].단말노드유사도 < 최적단말노드유사도)
                    매칭[i][j]를 최종매칭에서 제외;
      }
}
```

하지만 경로유사도만을 기준으로 가장 적절한 매칭을 찾다보면, 동일한 경로유사도를 보이는 상황이 많이 발생하게 된다. 예를 들어 같은 하위트리에 포함되어 있는 이웃관계의 두 타겟 단말노드는 한 소드 노드에 대해 동일한 경로를 나타낸다. 따라서 경로유사도가 똑같은 경우엔 단말노드 유사도 비교를 통해 가장 적절한 매칭을 선택한다.

이러한 매칭결과는 경로유사도 임계값을 확인하여 최종결과로 사용자에게 보여지게 된다. 경로유사도 임계값이란, 단말노

드는 비슷해서 매칭 관계가 생성되었지만 이름만 유사할 뿐 실제 경로를 따져보면 전혀 다른 매칭 관계를 제거하기 위해 사용하는 값이다. 최종결과에 포함되었더라도 경로 유사도 임계값보다 낮은 경로유사도를 보이는 매칭관계이면 최종 결과에서 제거된다.

4. 실험 및 평가

제안된 알고리즘의 평가를 위해 5개의 구매요청서에 관한 스키마를 대상으로 실험을 하였다. 5개의 스키마를 가지고 각각의 조합으로 10번의 실험을 실시하여, 전문가가 수동으로 매칭한 결과와 시스템을 통해 찾아진 매칭결과를 가지고 정확률과 재현률을 계산하였다. 실험데이터는 Excel, CIDX, Noris, Apaturm, Paragon로 biztalk[7]에서 사용하고 있으면 평균적으로 77개의 노드를 포함한다.

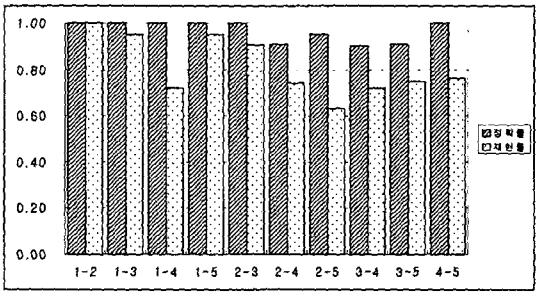

그림 2. 각 실험데이터에 따른 정확률과 재현률

그림 2. 각 실험 데이터에 따른 정확률과 재현률을 그래프로 나타낸 것이다. 평균 97%의 정확률과 81%의 재현율을을 보이고 있다. 특히 대부분의 실험에서 높은 정확률을 보임으로 찾아낸 결과는 거의 정확하다고 신뢰할 수 있다.

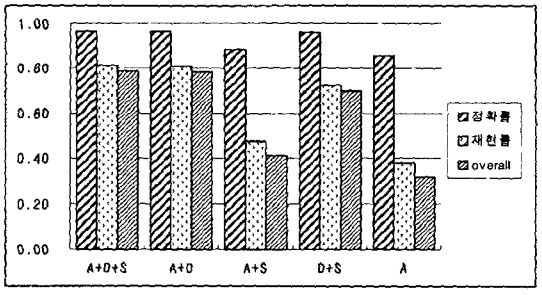

그림 3. 사용된 사전에 따른 결과비교

그림 3.은 언어적 유사도를 계산할 때 사용되는 어휘사전에 따라 나타나는 결과의 평균값을 비교한 것이다. 그림에서 A는 축약어 사전을, D는 도메인 온톨로지를, S는 일반 동의어 사전을 나타낸다. 여기서 overall값은 [8]에서 여러 스키마 매칭연구에 대한 결과를 비교할 때 소개된 값으로 *재현률*(2-(1/정확률))* 로 계산한다. 이 값은 잘못된 매칭결과를 제거하거나, 찾지못한 실제 매칭을 추가하는데 드는 노력의 값을 나타낸 것으로 값이 클수록 매칭결과에 대해 수정할 필요가 없다는 것을 의미한다.

실험 결과, 도메인 온톨로지의 역할이 정확성을 높히는데 중요하다는 것을 알 수 있다. 실험에서 사용한 도메인 온톨로지는 "head - header" 와 같은 일반동의어 사전에는 없지만 직관적으로 비슷한 단어의 관계와, "ship - deliver"와 같은 도메인에서 중요하게 작용하는 단어의 관계들을 포함한다.

그림 4. 단말노드 임계값에 따른 실험 결과를 보여주고 있다. 단말노드 임계값은 초기 후보매칭 집합을 결정하는 것으로 작으면 단말노드의 관계가 대부분 선택되어 정확성이 많이 떨어지게 되고 값이 너무 크다면, 후보 매칭 집합이 거의 없기 때문에 또한 정확한 결과예측이 힘들어진다. 본 실험에서는 위와 같은 실험을 통해 단말노드 임계값을 0.6으로 설정하여 실험하였다. 이밖에 경로유사도 임계값은 0.3으로 30%이하로 경로가 매칭되는 결과는 제거하였고, 언어적인 유사도에 대한 가중치는 0.8, 데이터 타입에 대한 가중치와 구조적인 유사도에 대한 가중치 모두 0.2의 값으로 설정하였다. 대부분 스키마가 많이 제약된 데이터 타입을 사용하는 것이 아니라 가장 보편적인 string같은 타입을 사용하고 있기 때문에 언어적인 부분에 더 많은 가중치를 부여하여 실험하였다.

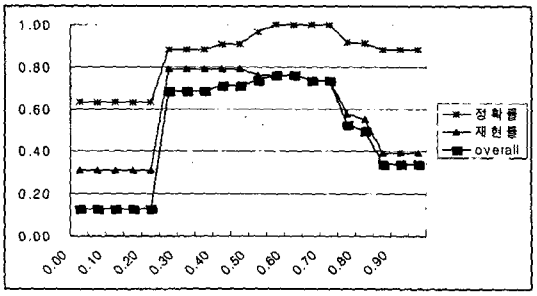

그림 4. 단말노드 임계값에 다른 결과

5. 결론 및 향후연구

본 논문에서는 XML 문서의 자동 변환을 위해 스키마 사이에서의 일대일 매칭관계를 찾는 효율적인 방법을 제안하였다. XML문서의 변환에 초점을 두고, 단말노드간의 정확한 일대일 매칭을 찾기위해 단말노드 부터 비교하고 그 중에 가장 적합한 것을 선택해 나가는 상향식 방법의 매칭 방법을 사용하였다. 이 방법은 모든 노드에 대한 비교의 필요없이 가능성있는 후보 집합만을 고려하여 되어 효율적이다. 하지만 초기 단말노드 매칭과정에서 걸러지는 정답 매칭에 대해서는 구조적인 정보를 비교해 보지도 못하고 버려지는 약점이 있다. 이는 실험을 통해 평균 81%의 재현률로 나타난다. 따라서 더욱 정확한 매칭을 위해 해당 도메인에 적합한 도메인 온톨로지가 요구된다.

앞으로 사용자 피드백에 의해 자동으로 도메인 온톨로지를 구축 및 수정할 수 있는 방법을 연구하여 기존 매칭 결과를 활용하고자 한다.

참고문헌

[1] XML "http://www.w3.org/TR/REC-xml"
[2] wordnet-a lexical database "http://www.cogsci.princeton.edu/~wn"
[3] Erhard Rahm and Philip A. Bernstein, "A survey of approaches to automatic schema matching," VLDB, vol. 10, issue 4, 2001.
[4] H. Su, H. Kuno, E. A. Rundensteiner, "Automating the transformation of XML documents," RIDE-DM, 2002.
[5] AnHai Doan, Pedro Domingos, and Alon Halevy, "Learning to Match Schemas of Data Sources : A Multistrategy Approach,"Machine Learning, vol. 50, pp.279~301, 2003.
[6] Hong-Hai Do and Erhard Rahm, "COMA-A system for flexible combination of schema matching approaches," VLDB, 2002.
[7] XML spec. "http://www.w3.org/TR/2000/REC-xml-20001006"
[8] biztalk "http://www.biztalk.org"
[9] Do, Hong-Hai, Melnik, Sergey., Rahm, Erhard, "Comparison of Schema Matching Evaluations," Proc. GI-Workshop "Web and Databases", LNCS 2593, 2002.