XML 스키마간의 복합매칭 추출을 위한 대화형 기법

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

Interactive Approach to Discover Complex Matchings between XML Schemas

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

요 약

본 논문은 온톨로지를 활용한 스키마 매칭 알고리즘을 제안한다. 기존의 대부분의 스키마 매칭 방법은 단순매칭을 대상으로 하는 반면, 제안된 방법은 계층적 구조의 온톨로지에 기반하여 복합매칭을 계산할 수 있다. 특히, 제안된 온톨로지는 이전의 매칭결과에 대한 사용자의 피드백을 이용하여 자동으로 갱신됨에 따라 적절한 도메인 정보를 유지할 수 있다. 성능평가를 위한 실험결과, 온톨로지의 적용이 매칭 성능을 향상시킴을 확인할 수 있었다.

1. 서론

스키마 매칭이란 특정 정보를 정의하고 있는 메타정보인 스키마 사이에 의미적인 관계를 계산하는 것으로, 시스템 통합이나 정보의 상호운용에 중요한 역할을 한다. 과거에는 주로 데이터베이스 스키마를 대상으로 연구가 진행되었고 최근에는 계층적 형태인 XML 스키마를 대상으로 많은 연구가 진행되고 있다.

기존의 대부분의 스키마 매칭에 관한 연구는 단순매칭만을 대상으로 한다. 단순매칭이란 복사 연산을 통해 정보를 변환할 수 있는 일대일 매칭관계를 의미한다. 하지만 실제 매칭 과정에서는 단순매칭 뿐만 아니라 다대일 또는 일대다의 관계로 매칭이 형성되는 복합매칭 관계도 발생하게 된다. 복합매칭은 단순매칭과는 다르게 변환을 위해 병합 혹은 분할 등의 연산이 적용되어야 한다[1].

또한, 기존의 연구 대부분은 매칭 성능 향상을 위해 유의어 사전[2]과 같은 부가 정보를 활용한다. 하지만 대부분 부가정보는 정적인 형태로 변화되는 어휘나 추가되는 어휘를 적용할 수 없고, 초기 구축에 많은 비용이 소요된다.

따라서, 본 논문에서는 스키마 사이의 복합매칭 계산을 지원할 수 있는 온톨로지를 제안한다. 또한, 적절한 도메인 정보의 유지를 위해 이전 매칭 결과에 대한 사용자의 피드백을 분석하여 자동으로 온톨로지를 갱신하는 방법을 제안한다. 제안된 방법은 기존 연구에 비해 온톨로지 구축 비용을 줄일 수 있으며 항상 적절한 정보를 유지할 수 있는 장점이 있다.

성능 평가를 위해 실제 사용되는 스키마를 사용하여 실험 결과, 제안된 방법은 기존 연구에 비해 정확도 면에서 매우 우수하였다. 특히, 온톨로지를 적용한 경우 적용하지 않았을 경우에 비해 재현율이 13% 향상됨을 확인 할 수 있었다.

본 논문의 구성은 다음과 같다. 2절에선 복합매칭을 지원하는 스키마 매칭 연구에 대해 간략히 기술하고, 3절에서는 제안된 온톨로지 구조와 연산을 자세히 기술한다. 4절에서는 전체적인 스키마 매칭 알고리즘에 대해 간략히 기술하고, 5절에서는 제안된 방법을 평가하기 위한 실험 결과를 정리한다. 끝으로 6절에서는 결론과 향후 연구방향을 기술한다.

2. 관련연구

기존에 스키마 매칭에 관한 다양한 연구가 진행되고 있다. 본 절에서는 그 중 복합매칭 계산을 지원하는 연구에 대해 간략히 정리한다.

Xu [3]는 온톨로지를 활용하여 복합매칭을 찾을 수 있는 방법을 제안한다. Xu가 제안한 온톨로지는 특정 도메인에 포함될 모든 개념(concept)들의 관계를 방향 그래프(directed graph)로 표현하며 각 온톨로지를 구성하는 개념에 대응하는 어휘리스트를 포함한다. 그러나 제안된 온톨로지는 전문가에 의해 수동으로 설계되어 어휘목록을 구축하는데 많은 비용이 소요된다.

Dhamankar 등 [4]이 제안한 iMAP은 학습기법을 활용하여 복합매칭을 찾는 방법으로 미리 학습된 분류기에 해당 스키마에 따라 작성된 문서를 입력하여 스키마 사이의 매칭을 찾는다. 특히 iMAP은 소스와 타겟문서에서 반복되는 데이터에 대한 정보를 이용하여 복잡한 연산에 해당하는 매칭을 계산할 수 있다. 하지만, iMAP은 관계형 데이터베이스를 대상으로 제안되어 계층형 구조의 스키마의 특성을 고려하지 못한다. 또한 학습 데이터가 다양한 입력 스키마를 포괄할 수 있어야 정확한 매칭결과를 얻을 수 있다.

본 논문에서는 기존 연구의 문제점을 해결하기 위해서 동적으로 갱신 가능한 온톨로지를 제안한다. 제안된 온톨로지는 도메인에 사용되는 어휘들의 계층적 구조를 기술하기 때문에 복합매칭을 지원하고, 이전 매칭 결과를 분석하여 자동으로 갱신됨으로 더욱 향상된 매칭결과를 기대할 수 있다.

3. 온톨로지

본 절에서는 온톨로지의 구조와 자동 갱신을 위한 연산에 대해 기술한다. 제안된 온톨로지는 이전 매칭 결과에 대한 사용자의 보정 작업을 분석하여 자동으로 갱신된다. 또한, 개념 사이에 *PartOf* 또는 *IsA*와 같은 다양한 관계를 기술함으로 복합매칭을 지원한다.

3.1. 온톨로지 구조

제안된 온톨로지는 개념노드와 간선으로 이루어진 트리의 집합으로 나타낸다. XML 스키마의 요소와 속성의 이름이 각 개념노드의 레이블로 표현되고, 간선은 노드 사이의 관계를 나타낸다. 특히, 간선은 *IsA*, *PartOf* 그리고 *Similar*의 3개의 관계로 표현되며, 각각의 관계는 일반화, 부분합 그리고 유사관계를 의미한다.

또한, 온톨로지는 노드 사이에 연결강도를 기술하고 있다. 연결강도는 두 노드 사이의 관계 정도를 나타내는 것으로 연결강도가 높을 수록 노드 사의의 관계가 정확함을 의미한다. 연결강도는 사용자 피드백에 따라 증가하거나 감소한다. 예를들어, 사용자가 매칭관계를 제거하면 연결강도는 감소하고 사용자가 관계를 추가시키면 연결강도는 증가한다.

제안된 온톨로지는 사용자 피드백 형태에 따라 적용되는 갱신 연산을 정의하고 있다. 다음절에서 온톨로지 연산에 대해 구체적으로 기술한다.

3.2. 온톨로지 연산

온톨로지 연산은 매칭결과에 대한 사용자 피드백에 따라 선택된다. 사용자 피드백은 다음과 같이 정의한다.

FeedBack = (*SourceNodeName, TargetNodeName, Relationship*).
Relationship = {Similar | IsA | PartOf | Remove}.

예를들어, 시스템의 매칭결과에 잘못된 매칭이 포함된 경우 사용자는 해당 매칭을 제거한다. 이러한 사용자의 후처리는 (HomeAddress, CompanyAddress, Remove) 와 같이 나타낸다. 반대로, 시스템이 찾지 못한 매칭을 추가한 경우 (lastName, familyName, Similar) 와 같은 사용자 피드백이 입력된다. 이러한 사용자 피드백에 따라 다음과 같은 온톨로지 연산이 이루어진다.

Adding Concepts. 사용자 피드백으로 입력된 노드 중 하나는 이미 온톨로지에 포함되어 있고, 다른 노드는 온톨로지에 포함되어 있지 않은 경우, 새로운 개념노드를 이미 존재하는 개념노드에 추가된다. 두 노드 사이의 관계는 사용자가 입력한 관계를 적용한다. 예를들어, 그림 1의 (a)와 같이 사용자 피드백이 (*PHONE, MobilePHONE, IsA*) 이면 *MobilePHONE* 개념노드가 추가된다.

Adding Trees. 사용자의 피드백으로 입력된 두 노드 모두 온톨로지에 포함되어 있지 않은 경우, 두 노드를 이용하여 하나의 트리를 형성하여 온톨로지에 추가한다. 그림 1의 (a)에서 처럼, 사용자 피드백이 (*Telephone, HomePhone, IsA*)인 경우 새로운 트리가 추가된다.

Merging Trees. 사용자 피드백으로 입력된 노드가 온톨로지의 다른 트리에 포함된 경우, 두 트리는 하나의 트리로 병합된다. 예를들어, 그림 1의 (a)의 온톨로지에 사용자 피드백이 (*Telephone, PHONE, Similar*) 로 입력되면 그림 1의 (b)와 같이 두 트리가 병합되게 된다.

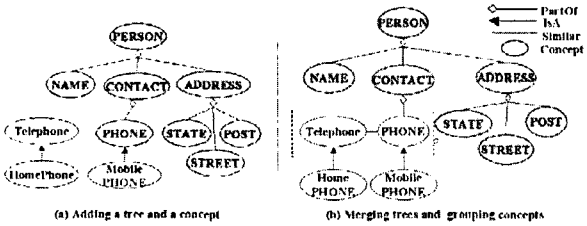


그림 1. 온톨로지 연산의 예

Grouping Concepts. Adding concepts 이나 merging trees 연산이 발생한 경우 사용자 피드백의 관계가 *Similar* 이면 동일한 관계를 같은 개념들은 그룹화된다. 그룹화된 개념들은 다른 개념들과 동일한 관계를 갖는다. 예를들어, 그림 1의 (b)와 같이 *Similar* 관계로 두 트리가 병합된 경우 *Telephone*과 *PHONE*은 그룹화가 되어 다른 개념들과 동일한 관계를 형성하게 된다. 즉, *Telephone*은 *CONTACT*와 *PartOf* 관계를 갖고 *PHONE*은 *HomePHONE*과 *IsA* 관계를 형성하게 된다. 이때, 연결강도는 두 관계의 연결강도의 평균값으로 표현된다.

Deleting Relationships. 시스템이 잘못된 매칭 결과를 계산한 경우, 사용자는 *Remove* 관계의 사용자 피드백을 입력하게 된다. *Remove* 관계의 피드백이 입력된면 온톨로지에 해당 관계가 존재할 경우 관계강도를 감소시킨다. 만약 관계강도가 임계값 이하로 내려간다면 두 개념 사이의 관계는 온톨로지에서 제거된다.

4. 스키마 매칭 알고리즘

제안된 스키마 매칭 알고리즘은 단말노드 비교를 통한 후보매칭 계산과, 경로 비교를 통한 최종매칭 추출의 두 단계로 구성된다. 두 단계로 구성된 매칭과정은 더욱 정교한 매칭을 가능케 한다.

4.1. 단말노드 사이의 후보매칭 생성

후보매칭 생성을 위한 단말노드 매칭과정에서는 소스스키마와 타겟스티마의 모든 단말노드를 비교하여 적절한 매칭을 선택한다. 단말노드 매칭은 온톨로지에 포함된 여부에 따라 두가지 방법으로 나누어 계산된다.

온톨로지에 비교하고 있는 단말노드가 포함어 있으면, 두 단말노드는 후보매칭으로 선택하고 단말노드 유사도는 온톨로지의 관계강도로 나타낸다. 특히, 동일한 서브트리의 단말노드가 상대의 하나의 단말노드에 매칭이 되고, 관계가 IsA나 PartOf인 경우, 하나의 복합매칭을 형성하여 후보매칭에 추가한다. 이 과정을 통해 최종 매칭 추출 과정에서 복합매칭 역시 하나의 매칭으로 다룰 수 있다.

단말노드의 관계가 온톨로지에 포함되어 있지 않다면 단말노드 유사도 계산을 통하여 임계값 이상의 관계를 후보매칭에 포함시킨다. 단말노드 유사도는 노드의 레이블이 포함하고 있는 언어적인 유사도와 데이터 타입 유사도를 이용하여 계산되는 값으로 각 어휘를 토큰화하고, 축약어 사전이나 일반동의어 사전을 이용하여 계산한다. 자세한 방법은 [5]에 기술한다.

그림 2는 단말노드 매칭의 예를 나타내고 있다. 소스스키마 S1의 Name은 S2와 ①과 ②의 매칭을 이루고, S1의 Phone은 동의어 사전을 이용했기 때문에 S2의 Telephone과 ③의 매칭이 가능하다. 특히, 온톨로지를 이용한 매칭 ④는 City와 Post는 Address와 PartOf 관계로 이루어진 하나의 복합매칭을 나타낸다.

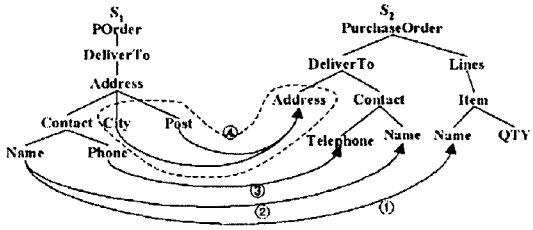


그림 2. 단말노드 매칭의 예.

4.2. 경로유사도에 기반한 최종매칭 추출

이전 과정에서 계산된 후보매칭은 예에서와 같이 다대다의 관계를 나타내기 때문에 문맥정보인 경로유사도를 비교하여 가장 적합한 매칭을 선택해야 한다. 경로유사도는 후보매칭의 경로에 포함된 중간노드의 유사도의 평균으로 계산할 수 있다. 이 단계에서는 후보매칭의 모든 경로유사도를 계산하고 경로유사도를 비교하여 가장 적절한 최종 매칭을 선택한다.

경로유사도 계산을 위해서는 먼저 중간노드의 유사도를 계산해야 하는데, 중간노드 유사도는 단말노드 유사도와 비슷하게 언어적인 유사도와 중간노드의 하위에 포함된 서브트리의 유사정도를 나타내는 구조적인 유사도를 이용하여 계산할 수 있다. 자세한 방법은 [5]에 기술한다.

매칭성능 향상을 위해 경로유사도 임계값을 설정하여 일정 값이하의 경로유사도는 최종매칭 선택 이전에 제거한다. 이 과정은 단말노드만 유사한 의미없는 후보매칭을 제거함으로 매칭 성능을 향상 시킨다.

경로유사도가 계산되면 차례로 모든 소스노드에 대해 일대다의 매칭관계를 검색하여 경로유사도가 가장 높은 매칭을 선택

하고 타겟노드에 대해서도 같은 방법으로 다대일의 매칭관계를 검색하여 가장 경로유사도가 높은 매칭을 선택한다. 만약 경로유사도가 동일한 경우가 여러개 발생한 경우, 단말노드 유사도가 높은 것을 우선적으로 선택한다.

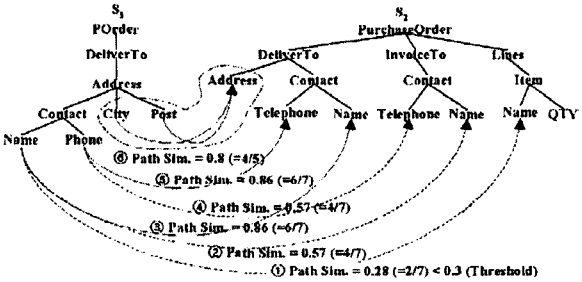
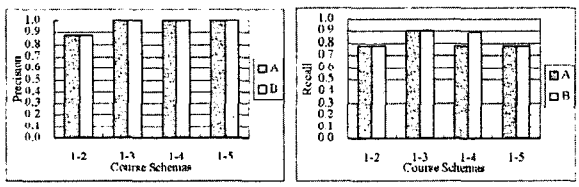


그림 3. 최종 매칭 추출의 예.

예를 들어, 그림 3은 최종 매칭 추출 과정을 도식한 것으로 각 매칭 ①에서 ⑥은 단말노드 매칭에서 선택된 후보매칭이다. 먼저, 매칭 ①은 경로유사도가 임계값보다 낮아 제거된다. ①이 제거되어도 S1의 Name은 S2와 1:2의 관계로 매칭되어 있다. 가장 적절한 매칭 선택을 위해 경로유사도를 비교하여 최종적으로 경로유사도가 더 높은 ③이 선택된다. ④와 ⑤도 비슷한 방법으로 ⑤가 선택되게 된다. ⑥은 복합매칭으로 충돌되는 매칭이 없어 그대로 선택되게 된다. 하지만 복합매칭 역시 다른 매칭과 다대일 일대다 관계로 매칭된다면 경로유사도 비교를 통해 선택된다. 즉, 복합매칭 역시 하나의 후보매칭으로 가정하고 최종매칭을 선택하게 된다.

5. 실험결과

제안된 방법의 성능을 평가하기 위해 두 도메인의 스키마를 대상으로 실험을 수행하였다. 사용된 데이터는 대학들의 수업리스트에 관한 것과 부동산 리스트에 관한 것으로 Doan[6]의 실험에 사용했던 것을 사용하였다.



(a) Precision of course schemas (b) Recall of course schemas

(c) Precision of house schemas (d) Recall of house schemas

그림 4. 실험 결과

수업리스트 스키마는 단순매칭만을 포함하고 있고, 부동산 리스트는 복합매칭을 포함하고 있다. 그림 4의 (a)와 (b)는 수업리스트에 대한 실험결과를 (c)와 (d)는 부동산 리스트에 대한 실험결과를 재현율과 정확률로 나누어 도식하고 있다. 특히, 실험 A는 부가정보로 동의어사전과 축약어 사전만을 사용하였고, 실험 B는 본 논문에서 제안한 온톨로지를 사용하여 실험하였다. 실험 B의 경우 이전의 실험결과를 이용하여 온톨로지를 업데이트하고, 업데이트된 온톨로지를 이용하여 입력된 새로운 스키마에 대해 실험하였다. 그림 4 에서 처럼 제안된 온톨로지의 사용은 매칭 성능을 향상시킴을 확인할 수 있었다.

수업리스트에 대한 실험에서는 평균 99%의 정확률과 85%의 재현율을 보였다. 하지만 단순매칭만 포함되어 있기 때문에 온톨로지 사용이 성능향상에 큰 도움을 주지는 못했다. 부동산 리스트의 경우, 온톨로지의 적용으로 찾지 못했던 복합매칭을 계산하여 재현율을 46%에서 72%로 향상시킴을 확인할 수 있었다. 하지만, 정확률에 비해 재현율이 낮은 것은 스키마를 기술하는 어휘가 매우 상이하여 초기 단말노드 매칭과정에서 찾지 못하는 경우가 발생했기 때문이다.

6. 결론 및 향후연구

본 논문에서는 온톨로지에 기반하여 복합매칭을 계산할 수 있는 스키마 매칭 알고리즘을 제안하였다. 제안된 온톨로지는 도메인에서 사용되는 어휘 사이의 *PartOf*나 *IsA*와 같은 다양한 관계를 계층적으로 표현함으로 복합매칭 계산을 가능하게 한다. 특히, 온톨로지는 이전 매칭결과를 활용하여 자동으로 갱신됨으로 항상 적절한 도메인 정보를 포함할 수 있고, 온톨로지 구축 및 갱신 비용을 줄일 수 있다. 매칭 방법은 단말노드 비교를 통한 후보매칭 계산과 경로유사도를 비교한 최종 매칭 선택의 두 단계로 구성됨으로 정확한 매칭을 계산할 수 있다.

성능 평가를 위한 실험결과, 제안된 방법은 기존 방법에 비해 매우 정확한 매칭결과를 나타내었고, 온톨로지의 적용으로 복합매칭을 계산함으로 매칭 결과를 향상시킴을 확인 할 수 있었다.

향후에는, 현재 문제되고 있는 단말노드 매칭 과정을 개선하기 위해 다양한 어형에도 높은 유사도를 계산할 수 있는 방법에 대한 연구를 진행할 것이다. 또한, 온톨로지 갱신이 사용된 레이블 단위로 갱신되기 때문에 다른 스키마에서 그대로 사용될 수 있는 확률이 떨어져 전체적으로 온톨로지 사용 효과를 저하 시키기 때문에, 온톨로지를 레이블 단위가 아닌 토큰 단위로 갱신할 수 있는 방법에 대한 연구를 진행할 예정이다.

참고문헌

[1] Erhard Rahm and Philip A. Bernstein, "A survey of approaches to automatic schema matching," VLDB, vol. 10, no. 4, pp. 334-350, 2001.

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

[3] Li Xu, David W. Embley, "Discovering direct and indirect matches for schema elements," Proceedings. 8th Conference on DASFAA, pp. 39-46, 2003.

[4] Robin Dhamankar, Yoonkyong Lee, AnHai Doan and Alon Halevy, "iMAP: Discovering Complex Semantic Mappings between Database Schemas," Proc. Int'l Conf. SIGMOD on Management of Data, 2004.

[5] 이준승, 이경호, "XML 문서의자동변환을 위한 스키마 매칭 알고리즘," 한국멀티미디어학회 논문지(in press), 2004.

[6] AnHai Doan, Pedro Domingos, and Alon Halevy, "Learning to Match Schemas of Data Sources: A Multistrategy Approach," Machine Learning, vol. 50, no. 3, pp. 279-301, 2003.