Journal of Korea Multimedia Society Vol. 7, No. 9, September 2004(pp. 1195-1207)

XML 문서의 자동변환을 위한 스키마 매칭 알고리즘

이준승[†], 이경호[††]

요 약

스키마 매칭은 XML 문서의 자동 변환을 위한 전처리 과정으로서 필수적이다. 스키마 매칭에 관한 기존 연구는 의미적으로 대응 가능한 모든 매칭관계를 고려하기 때문에 다대다의 대응관계를 추출한다. 이에 명확한 매칭관계를 필요로하는 XML 문서의 자동변환에는 적합하지 않다. 본 논문에서는 스키마 사이의 일대일 대응관계를 추출할 수 있는 효율적인 스키마 매칭 알고리즘을 제안한다. 제안된 알고리즘은 두 단계로 구성된다. 먼저 단말노드 사이의 언어적 유사도와 데이터타입 유사도를 이용하여 후보매칭을 계산한다. 계산된 후보매칭의 경로유사도 비교를 통해 일대일 매칭을 추출하게 된다. 특히 제안된 방법은 보다 정교한 수준의 스키마 매칭을 위하여 축약어 사전, 동의어 사전, 그리고 도메인 온탈로지에 기반한다. 제안된 알고리즘의 성능을 평가하기 위해서 전자상거래 분야에서 사용 중인 스키마를 대상으로 실험한 결과, 평균적으로 97%의 정확률을 보여 기존 연구보다 우수하였다.

A Schema Matching Algorithm for an Automated Transformation of XML Documents

Jun-Seung Lee[†], Kyong-Ho Lee[††]

ABSTRACT

Schema matching is prerequisite to an automated transformation of XML documents. Because previous works about schema matching compute all semantically-possible matchings, they produce many-to-many matching relationships. Such imprecise matchings are inappropriate for an automated transformation of XML documents. This paper presents an efficient schema matching algorithm that computes precise one-to-one matchings between two schemas. The proposed algorithm consists of two steps: preliminary matching relationships between leaf nodes in the two schemas are computed and one-to-one matchings are finally extracted based on a proposed path similarity. Specifically, for a sophisticated schema matching, the proposed algorithm is based on a domain ontology as well as a lexical database that includes abbreviations and synonyms. Experimental results with real schemas from an e-commerce field show that the proposed method is superior to previous works, resulting in an accuracy of 97% in average.

Key words: XML, Schema Matching(스키마 매칭), Automated Transformation(자동 변환), One-To-One Matching(일대일 매칭), Path Similarity(경로유사도)

1. 서 론

XML(eXtensible Markup Language)[1] 문서는 논리적 구조정보를 표현할 수 있으며 플랫폼에 독립적이라는 장점 때문에 다양한 분야에서 정보의 공유 및 교환을 위한 표준으로 널리 사용되고 있다. XML

※ 교신저자(Corresponding Author) : 이준승, 주소 : 서울 서대문구 신촌동 134(120-749), 전화 : 02)2123-3878, FAX : 02)365-2579, E-mail : jslee@icl.yonsei.ac.kr
접수일 : 2003년 11월 7일, 완료일 : 2004년 1월 29일
[†] 준회원, 연세대학교 대학원 컴퓨터과학과 석사과정
[††] 정회원, 연세대학교 컴퓨터산업공학부 조교수
(E-mail : khlee@cs.yonsei.ac.kr)
※ 이 논문은 2003년도 한국학술진흥재단의 지원에 의하여 연구되었음.(KRF-2003-003-D00429)

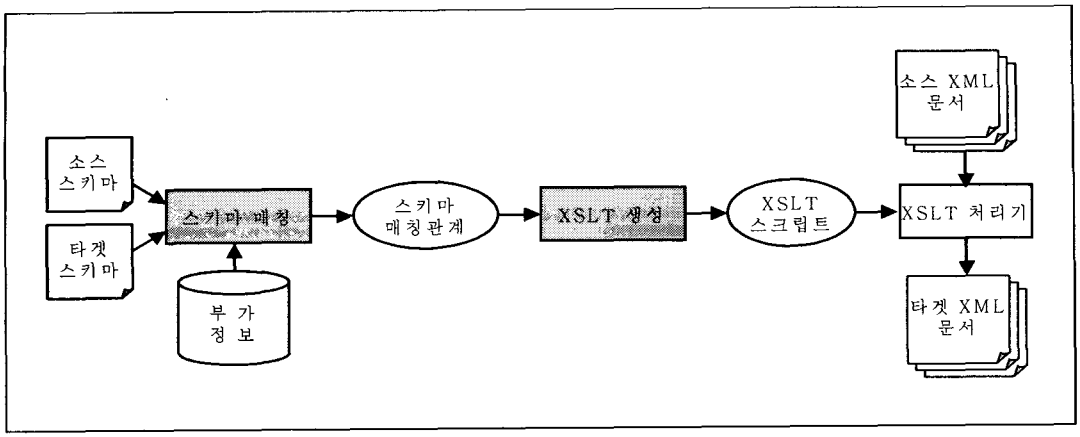

그림 1. XML 문서의 변환과정

문서가 보편적으로 사용됨에 따라 XML 문서의 구조 정보를 정의한 XML 스키마[1] 역시 급증하고 있다.

XML 문서는 사용자가 원하는 태그(tag)를 정의하여 사용할 수 있는 특징 때문에, 사용자 그룹에 따라 다른 구조로 정의된 스키마를 사용하게 된다. 다른 스키마를 이용하여 작성된 XML 문서를 사용하기 위해선 두 스키마 사이에 의미적 관계를 찾아 원하는 정보로 변환시켜야 한다. 예를 들어, 전자상거래에서 사용되는 구매요청서를 기업마다 다른 스키마를 이용하여 작성한 경우, 두 기업의 구매요청서를 서로 이용하기 위해선 두 기업의 구매요청서의 스키마 사이에 의미적 연관관계를 찾아 이것을 이용하여 XML 문서를 변환시켜야 한다. 즉, 서로 다른 스키마에 따라 작성된 XML 문서를 교환 및 공유하기 위해서는 XML 문서간의 변환과정이 수행되어야 한다. 특히 XML 스키마의 종류와 수가 급증함에 따라 XML 문서간의 자동변환이 중요한 이슈로 떠오르고 있다.

일반적으로 XML 문서의 변환 과정은 그림 1과 같이 스키마 매칭과 변환용 XSLT(eXtensible Stylesheet Language Transformations)[3]를 이용한 변환 스크립트 생성의 두 단계로 구성된다. 스키마 매칭 단계에서는 소스스키마와 타겟스키마를 입력으로 받아 의미적인 매칭관계를 생성하고, 변환 스크립트 생성 단계에서는 전 단계에서 계산된 매칭관계를 이용하여 XML 문서를 변환시킬 수 있는 스크립트를 생성한다.

1) 본 논문에서 XML 스키마는 DTD(Document Type Definition) [1]와 XML Schema [2]를 모두 포괄하는 의미로 사용된다.

기존에 XML 문서의 변환을 위하여 제안된 방법[4-6]은 주로 사용자에 의하여 수작업으로 입력된 매칭관계로부터 변환용 XSLT 스크립트를 생성한다. 한편, 크기가 수 메가 바이트에 이르는 스키마의 경우, 수동 방식의 스키마 매칭은 지나치게 많은 비용을 필요로 한다. 따라서 XML 문서의 변환을 효과적으로 지원하기 위해서는 스키마 매칭의 자동 계산이 선행되어야 한다.

스키마 매칭에 대한 연구는 데이터 통합(data integration)[7], 스키마 클러스터링(schema clustering)[8] 등 다양한 분야에서 진행되고있다[9]. 그러나 기존 연구의 대부분은 스키마 매칭의 결과로 다대다(many-to-many)의 매칭관계를 계산하기 때문에 일대일(one-to-one)의 매칭관계를 필요로 하는 XML 문서 변환에는 적합하지 않다. 변환 스크립트 작성을 위해선 소스스키마의 어떤 요소의 내용이 타겟스키마의 어떤 요소의 내용으로 변환되는지 명확히 구분해주어야 한다. 따라서, 제안된 방법은 기존의 연구와는 다르게 두 단계의 매칭과정을 거쳐 더욱 명확한 일대일 매칭관계를 추출한다.

예를 들면, 그림 2는 두개의 구매요청서용 스키마의 일부를 트리형태로 표현한 것이다. 소스스키마인 S_1은 물품이 배달되는 주소정보를, 타겟스키마인 S_2는 배달 정보와 함께 청구자에 대한 정보를 가지고 있다. 이때 기존 연구의 대부분은 S_1의 'Name'을 S_2의 두개의 'Name'과 모두 연결하여 두개의 매칭관계 ①과 ②를 생성한다. 즉, 의미적으로 유사한 일대다(예에선 1:2)의 매칭관계를 결과로 계산한다. 그러나 문서의 변환을 위해선 보다 정확한 매칭인 ②만을

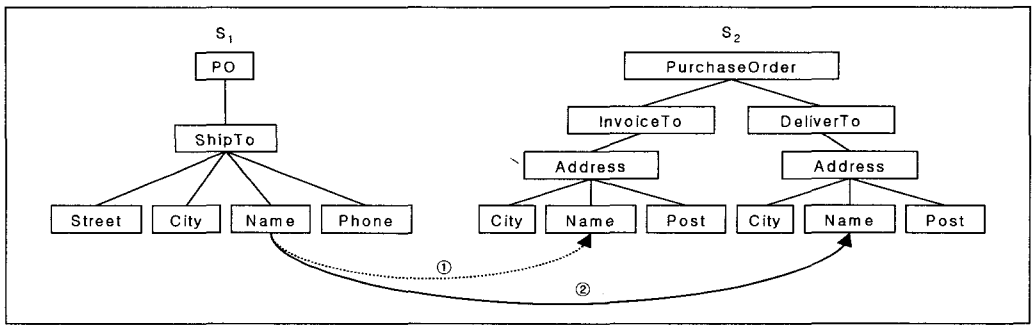

그림 2. 소스 및 타겟스키마의 예

선택하여야 할 것이다. 따라서 본 논문에서는 XML 문서의 자동변환을 위해서 스키마를 구성하는 단말 노드 사이의 일대일 대응관계를 효율적으로 추출하는데 목적을 둔다.

제안된 방법은 어휘적 유사성에 기반하여 다대다 관계의 후보 매칭 집합을 계산하고, XML 스키마의 구조적 정보를 이용하여 최종 일대일 매칭관계를 선택하는 두 단계로 구성된다. 또한, 정확률을 향상시키기 위하여 축약어 사전, 동의어 사전, 그리고 도메인 온톨로지의 부가정보를 적용하는 방법을 제안한다. 높은 정확률은 사용자의 간섭을 최소화시킴으로 시스템의 자동화 수준을 높힐 수 있다. 제안된 알고리즘의 성능을 평가하기 위해서 전자상거래 분야에서 사용 중인 스키마를 대상으로 실험한 결과, 평균적으로 97%의 정확률(precision)과 81%의 재현률(recall)을 보여 같은 실험 데이터를 사용한 기존의 연구보다 우수하였다.

본 논문의 구성은 다음과 같다. 2절에서 XML 스키마를 효과적으로 표현할 수 있는 문서 모델을 제안한다. 또한 스키마 매칭에 관한 기존연구를 간략히 기술하고, XML 문서변환 측면에서의 문제점을 기술한다. 3절에서는 제안된 방법을 단말노드 매칭과 경로유사도에 기반한 일대일 매칭의 두 단계로 구분하고, 각 단계에 대한 자세한 설명을 기술한다. 4절에서는 실험 결과를 통하여 제안된 방법을 기존 연구와 비교 및 분석한다. 마지막으로 5절에서는 결론과 향후 연구 방향을 기술한다.

2. 문서 모델 및 관련 연구

본 절에서는 XML 스키마를 효과적으로 표현할 수 있는 문서 모델을 제안한다. 또한 스키마 매칭에 관한 기존 연구 결과를 간략히 소개하고 이의 문제점을 기술한다.

2.1 문서 모델

본 논문에서는 XML 스키마를 구성하는 요소(element)와 속성(attribute)을 효과적으로 표현하기 위한 문서 모델을 제안한다.

XML 스키마는 XML 문서에 포함될 요소의 이름과 구조, 속성의 이름 등 문서 형식을 정의한다. 논리적 구성요소의 이름과 계층구조는 물론이고, 각각의 요소가 포함할 수 있는 정보의 데이터 타입을 정의한다. 그림 3은 전자 상거래 분야에서 물품의 구매요청을 위해 사용중인 XML 스키마의 예이다. 여기서 요소 'ShipTo'는 하위 요소로서 'PostalCode', 'Country', 'City', 그리고 'Street'을 포함하며 'Street'은 내용으로 문자열(string type)을 포함한다.

본 논문에서는 XML 스키마를 표현하기 위하여 뿌리 노드(root node)를 포함하며 형제 노드간에 순서가 존재하는 순서 트리(ordered tree)에 기반한 문서 모델을 제안한다. 문서 모델은 XML 스키마에 정의된 요소와 속성을 노드로 갖는다. 각각의 노드는 레이블(label)과 값(value)를 갖는다. 레이블은 XML 스키마 문서에서 정의된 요소와 속성의 이름이고 값은 XML 스키마에 정의된 요소나 속성의 데이터 타입으로 단말노드만 값을 갖게 된다.

그림 4는 그림 3을 제안된 문서 모델로 표현한 결과이다. 한편, 본 논문에서는 노드의 부모노드로부터 뿌리노드까지의 노드의 순차적인 집합을 해당 노드의 경로라고 정의한다. 예를 들어, 그림 4에서 노드 'UOM'의 경로는 'Item-Line-PO'에 해당한다.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
    <xs:element name="PO">
        <xs:complexType>
            <xs:sequence>
                <xs:element ref="POHeader"/>
                <xs:element ref="ShipTo"/>
                <xs:element ref="BillTo"/>
                <xs:element ref="Line"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="POHeader">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="Number" type="xs:string"/>
                <xs:element name="Date" type="xs:string"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="ShipTo">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="PostalCode" type="xs:string"/>
                <xs:element name="Country" type="xs:string"/>
                <xs:element name="City" type="xs:string"/>
                <xs:element name="Street" type="xs:string"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="BillTo">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="PostalCode" type="xs:string"/>
                <xs:element name="Country" type="xs:string"/>
                <xs:element name="City" type="xs:string"/>
                <xs:element name="Street" type="xs:string"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="Line">
        <xs:complexType>
            <xs:sequence>
                <xs:element ref="Item"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="Item">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="UOM" type="xs:string"/>
                <xs:element name="UnitPrice" type="xs:string"/>
                <xs:element name="QTY" type="xs:string"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
</xs:schema>
```

그림 3. 구매요청서를 표현하기 위한 XML 스키마의 예

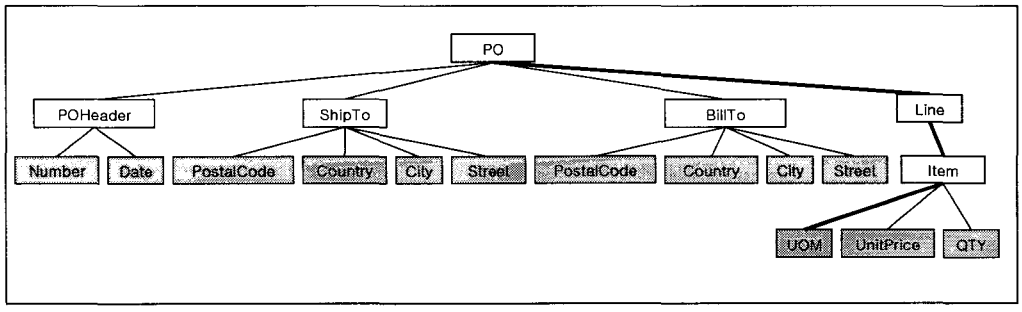

그림 4. 그림 3의 XML 스키마를 문서 모델로 표현한 결과

2.2 관련 연구

전술한 바와 같이 본 논문에서는 XML 문서의 자동 변환을 위한 스키마 매칭 알고리즘을 제안한다. 스키마 매칭에 관한 연구는 다양한 분야에서 진행되어 왔다. 표 1은 과거에 수행되었거나 현재 진행중인 스키마 매칭에 관한 연구결과를 간략하게 정리한 것이다.

Bergamaschi 등[7]이 제안한 ARTEMIS는 관계형 데이터 베이스 등에서 사용되는 스키마를 통합하기 위해 제안된 스키마 매칭 방법으로 동의어 및 유의어 사전을 사용하며 구조적인 유사도를 이용하여 매칭을 계산하지만, XML 스키마와 같이 계층형 스

표 1. 스키마 매칭에 관한 연구

저자	연도	명칭	특 징	대상	적용분야
Bergama-schi 등 [7]	1998	ARTE-MIS	동의어 사전과 데이터 타입을 이용하여 각 요소의 언어적 유사성과 구조적 유사성을 구하여 이에 의해 스키마들을 통합하는 통합도구를 제시	관계형, OO, ER	스키마 통합
Lee 등 [8]	2002	XClust	트리로 표현된 스키마의 언어적 유사도와 구조적 유사도를 계산하여 두 스키마 사이의 유사도 계산. 스키마 사이의 유사도를 이용하여 클러스터링하는 방법 제안	XML	클러스터링
Su 등 [10]	2001	Xtra	XML문서의 변환을 지원하기 위하여 스키마 매칭을 수행하며 이로부터 XSLT 스크립트를 생성하며 이를 위하여 변환연산과 비용모델을 제안	XML	변환
Milo 등 [11]	1998	Trans-Scm	그래프를 이용하여 두 개의 스키마에서 추출된 SGML 태그의 이름과 구조를 이용하여 스키마를 매칭시키는 규칙기반의 알고리즘 제안	SGML OODB	변환
Li와 Clifton [12]	1994	SemInt	신경회로망을 이용한 학습 기법을 적용하여 매칭을 계산하는 방법 제안	ER-DB	스키마 매칭
Doan 등 [13]	2001	LSD	기계 학습을 이용하며 샘플 스키마에 대한 학습을 통하여 새로운 문서에 대한 매칭 관계를 계산하는 방법 제안. 특히, 다양한 측면에서 학습시키고 종합시킬 수 있는 방법 제안	XML	스키마 매칭
Lerner [14]	2000	Tess	스키마가 변화하더라도 그 변화를 인식하고 새로운 스키마와 비교한 후 스키마간에 갱신이 가능한 변환을 지원할 수 있는 방법 제안	ER-DB	스키마 매칭
Miller 등 [15]	2001	Clio	이기종간의 스키마를 통합 및 관리하기 위하여 부가정보에 기반하여 매칭 관계를 계산하고 사용자 상호작용을 지원하는 방법 제안	XML ER-DB	스키마 매칭
Madhavan 등 [16]	2001	Cupid	원소 중심의 매칭과 구조 중심의 매칭을 적절히 혼합한 스키마 매칭 알고리즘 제안	XML ER-DB	스키마 매칭
Do와 Rahm [17]	2002	COMA	여러 가지 매칭 알고리즘을 조합하여 적용할 수 있는 시스템 제안. 사용자의 피드백과 기존의 매칭결과를 이용할 수 있는 방법 제안	XML	스키마 매칭
Melnik 등 [18]	2002	SF	유사도가 주변노드로 전파된다는 가정과 방향 그래프에 기반한 스키마 매칭 알고리즘 제안	ER-DB	스키마 매칭

키마에는 적합하지 않다. Lee 등[8]의 XClust는 스키마의 클러스트링을 위한 스키마 매칭 방법을 제안한다. XClust는 유사도를 이용하여 매칭 관계를 찾고, 매칭되는 정도에 따라 두 스키마 사이의 전체 유사도를 계산한다. 계산된 유사도는 스키마를 클러스트링하는데 사용한다.

기존의 XML 문서의 자동변환을 위한 방법으론 Su 등[10]이 제안한 Xtra가 있다. Xtra는 소스DTD를 타겟DTD로 변환하는데 필요한 최소 비용의 변환 연산을 계산한다. 또한 계산된 변환 연산를 이용하여 변환용 XSLT 스크립트를 생성한다. Xtra는 스키마의 구조 정보를 고려하지 않으며 최소 비용의 변환 연산에 해당하는 매칭을 계산하기 때문에 정확한 매칭결과를 계산하는데 있어서 한계를 갖는다. 정확한 매칭을 계산하기 위해서는 최소비용의 변환 연산보다 의미적 및 구조적으로 유사한 노드 사이의 매칭관계를 계산하여야 한다. 다른 문서 변환을 위한 연구로 Milo 등[11]이 제안한 TransScm이 있다. TransScm은 정의한 변환규칙에 따라 제안된 그래프 모델로 표현된 이종의 스키마 사이에 매칭관계를 계산하여 문서를 변환시킨다. TransScm에서는 SGML의 스키마로 생성된 문서를 OODB의 스키마로 생성된 문서로 변환시키는 방법을 제안하고 있다.

스키마 매칭에 관한 연구는 Li와 Clifton[12]가 제안한 SemInt와 Doan 등[13]이 제안한 LSD와 같은 학습기법에 기반한 방법도 있다. 먼저 제안된 SemInt는 관계형 스키마를 대상으로 저장되어 있는 데이터를 여러 측면에서 학습시킨 후, 이것을 이용하여 스

키마 사이의 매칭관계를 찾는다. 최근 연구된 LSD 역시 많은 양의 XML 문서를 이용한 학습과정을 통해 스키마 사이의 매칭을 찾는다. LSD는 전문가에 의해 결과가 계산되어 있는 스키마와 그것에 포함된 XML 문서를 이용하여 학습과정을 거쳐야 한다. 사용된 어휘, 기술된 형태, 빈도수 등 다양한 측면에 대해 학습시킨다. 따라서 본 방법은 학습을 위하여 많은 시간과 노력을 필요로 하며 학습을 위한 데이터 역시 사전에 준비되어야 한다. 특히 학습 데이터가 새로운 스키마를 포괄할 수 있을 만큼 충분해야 정확한 매칭을 찾을 수 있다.

Lerner[14]가 제안한 Tess는 기존의 스키마를 수정하여 새로운 스키마를 설계할 경우 기존의 스키마와 새로운 형태의 스키마 사이의 매칭관계를 계산하게 된다. 이름을 분석하고 구조 정보를 이용하여 계산된 유사도를 활용하여 매칭관계를 찾는다. 또, Miller 등[15]이 제안한 Clio는 여러가지 부가정보를 활용하여 다대다 관계의 매칭을 찾아내는 시스템으로 사용자의 상호작용을 지원하고 스키마를 통합 및 관리하는데 사용된다.

Madhavan 등[16]이 제안한 Cupid는 다양한 스키마를 대상으로 하며 복합적인 방법을 사용하여 매칭을 계산한다. XML 스키마와 데이터베이스 스키마에 적용 가능한 방법으로서 제안된 Cupid는 어휘 정보와 구조정보를 이용하여 매칭관계를 계산한다. 특히 XML 스키마의 계층적인 특징을 반영하여 구조적인 유사도를 계산하고 있다.

한편, 최근 진행되는 연구로서 Do와 Rahm[17]이 제안한 COMA와 Melnik 등[18]이 제안한 SF가 있다. COMA는 여러 가지 방법들을 모듈화하여 적용할 수 있는 방법을 제안한다. 또한, 사용자의 피드백과 기존 매칭결과의 재사용 등을 통해 정확성을 좀 더 향상시킬 수 있는 방법을 제안하였다. 그러나 매칭결과가 다대다의 매칭결과를 생성하기 때문에 문서의 자동변환에 적용하는데 부적절하다. SF는 그래프로 나타낸 문서모델에서 한 노드의 유사도가 주위의 다른 노드의 유사도에도 영향을 준다는 아이디어를 이용한 스키마 매칭 방법이다. 즉, 유사도를 주위 노드로 전파시킴으로 전체적으로 스키마 사이의 구조적 유사도를 반영시킬 수 있는 방법을 제안하고있다.

3. 스키마 매칭 알고리즘

제안된 방법은 XML 스키마간에 일대일의 매칭관계를 계산한다. 제안된 방법은 그림 5와 같이 언어적 및 데이터 타입 유사도에 기반한 단말노드 매칭과 경로유사도에 기반한 일대일 매칭 추출의 두 단계로 구성된다. 특히, 제안된 방법은 보다 정교한 수준의 매칭을 계산하기 위해서 축약어 사전과 도메인 온톨로지 등 부가적인 정보를 적용한다.

3.1 단말노드 매칭

두 스키마 트리의 모든 단말노드들을 비교하여 유사도가 임계값 이상인 매칭관계를 찾는다. 단말노드

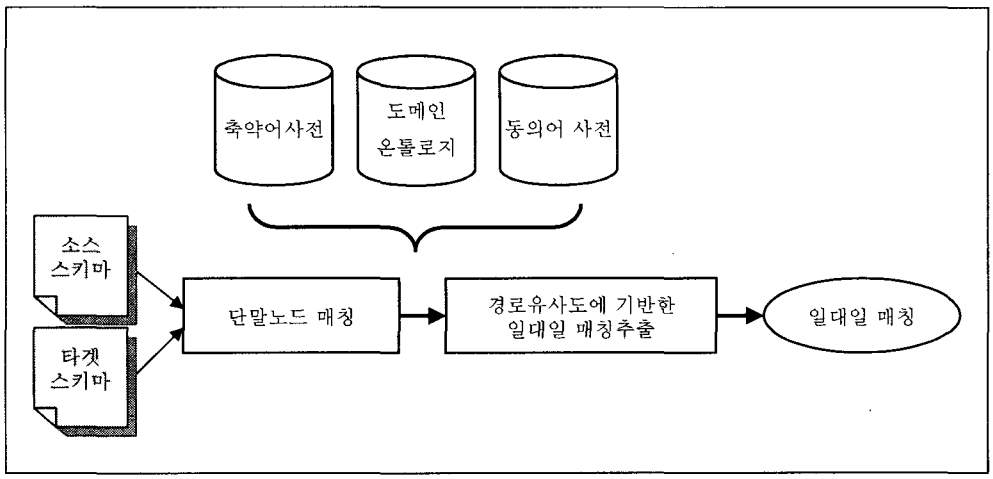

그림 5. 시스템의 구조

의 유사도는 식(1)과 같이 노드 레이블 사이의 언어적 유사도와 데이터 타입 유사도의 합으로 정의한다. 여기서 가중치 w_l과 w_t는 각각 언어적 유사도와 데이터 타입 유사도의 가중치를 의미한다. 언어적 유사도와 데이터 타입 유사도에 대한 자세한 설명은 다음과 같다.

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

3.1.1 언어적 유사도

주어진 두 노드 레이블 사이의 어휘적 유사도를 나타낸다. 먼저 입력된 노드의 이름을 대문자나 특수기호를 기준으로 토큰화하여 각 토큰 사이의 유사도를 계산한다. 이때 언어적 유사도는 식(2)와 같이 토큰 사이의 유사도의 합을 전체 토큰수로 나눈 값으로 정의한다.

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

Ts_i:소스 노드의 토큰, $1 \le i \le n$ Tt_j:타겟 노드의 토큰, $1 \le j \le m$

토큰 사이의 유사도를 구하기 위해서 축약어 사전, 도메인 온톨로지, 그리고 동의어 사전의 세 가지 부가 정보를 사용한다. 먼저 축약어 사전은 각 토큰들이 축약어인지 아닌지를 판단하여 축약어인 경우 원래의 이름으로 바꿔주는 역할을 한다. 축약어 사전은 축약어와 그 축약어의 전체이름을 가지고 있는 간단한 형태의 테이블 구조로 입력된 토큰과 같은 축약어가 있다면 입력 토큰을 전체이름으로 대체하게 된다. 축약어 검색이 끝난 토큰들은 문자열을 비교하여 동일한 토큰인 경우 1.0의 유사도를 반환한다. 동일하지 않은 토큰들은 도메인 온톨로지를 검색하게 된다.

도메인 온톨로지는 특정 도메인에서만 관계를 보이는 토큰사이의 관계 정도를 표현하고 있는 테이블로 각 축에 해당되는 단어의 검색을 통해 두 단어사이의 관계정도를 찾을 수 있다. 만약 두 토큰이 도메인 온톨로지에 포함되어 있다면 토큰 사이의 유사도는 도메인 온톨로지에 따라 부여된다. 도메인 온톨로지 테이블에는 각 토큰간에 1.0부터 -1.0사이의 유사도가 정의되어 있다. 특히, 음의 값은 두 토큰이 반의어 관계임을 나타낸다.

도메인 온톨로지에도 포함되어 있지 않다면 일반 동의어 사전[19]을 확인한다. 일반 동의어 사전에 포함되어 있는 두 토큰의 관계는 기본적인 유사도(0.8)를 적용시켜 두 토큰 사이의 유사도를 계산한다. 최종적으로 두 노드 사이의 언어적 유사도는 모든 토큰 사이의 유사도의 합을 전체 토큰의 수로 나눈 값으로 계산한다. 언어적 유사도를 계산하는 과정은 그림 6과 같다.

3.1.2 데이터타입 유사도

스키마 트리를 구성하는 단말노드는 다양한 종류의 데이터 타입을 값으로 갖는다. 특히 데이터 타입이 서로 다른 노드간의 변환은 노드가 포함하는 정보에 손실을 가져올 수 있다. 데이터타입 유사도는 이와 같이 서로 다른 데이터 타입을 갖는 노드간의 변환에 의하여 발생 가능한 정보 손실을 표현하기 위해서 제안되었다.

본 논문에서는 노드의 데이터 타입간의 유사한 정도를 해당 노드가 포함하는 정보의 손실 정도에 따라 '동일', '비손실 변환가능', '손실 변환가능', 그리고 '변환불가'의 4단계로 구분한다. 동일한 데이터 타입의 변환의 경우, 가장 큰 유사도를 부여하고, 정보 비손실 변환가능, 정보 손실 변환가능, 그리고 변환 불가의 단계의 따라 점점 낮은 유사도를 부여한다. XML 스키마에서 지원하는 44개의 데이터 타입 중에서 7개의 주요 데이터 타입간의 유사도는 표 2와 같다.

3.2 경로유사도에 기반한 일대일 매칭 추출

단말노드 매칭 과정을 통해 얻어진 매칭관계는 다수의 소스노드와 타겟노드가 연결된 다대다 관계일 수 있다. 제안된 방법은 경로 유사도에 기반하여 다대다 관계로부터 일대일 매칭을 추출한다. 특히 서로 대응하는 경로사이의 유사도를 경로유사도라고 하며 이에 대한 정의는 식(3)과 같다. 먼저 서로 대응되는 두 경로에 포함되어 있는 중간노드를 비교하며 매칭되는 중간노드를 찾는다. 경로유사도는 두 경로에 포함된 매칭관계를 갖는 중간노드의 비율로 나타낸다.

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

Ps:소스 노드의 경로, Pt:타겟 노드의 경로

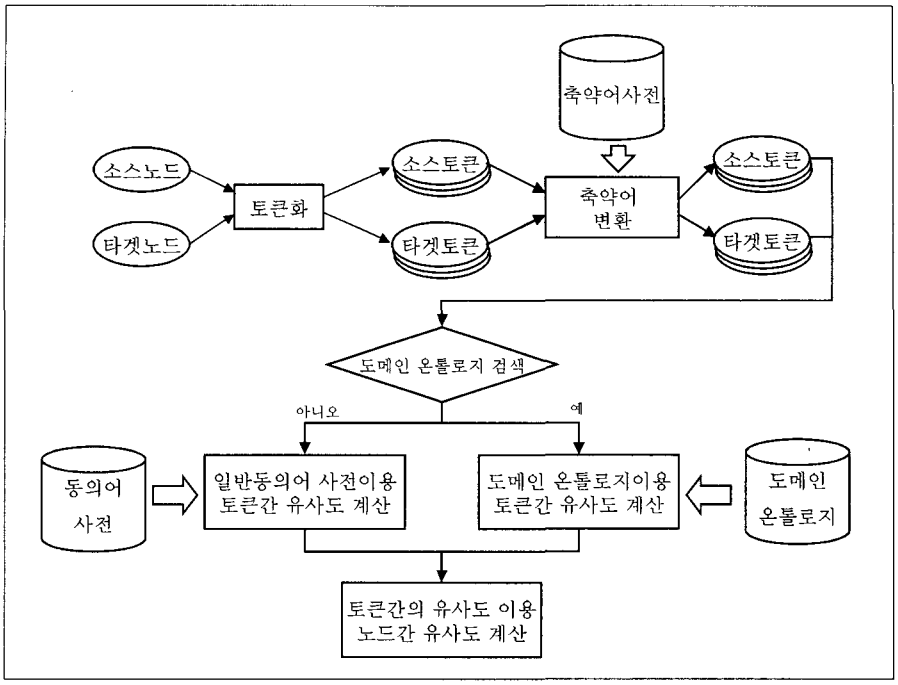

그림 6. 언어적 유사도 계산과정

표 2. 데이터 타입간의 유사도

타겟노드의 데이터 타입 / 소스노드의 데이터 타입	string	decimal	double	float	boolean	duration	dateTime
string	3.0	1.0	1.0	1.0	0.0	1.0	1.0
decimal	2.0	3.0	2.0	2.0	1.0	0.0	0.0
double	2.0	1.0	3.0	1.0	0.0	0.0	0.0
float	2.0	1.0	2.0	3.0	0.0	0.0	0.0
boolean	2.0	2.0	0.0	0.0	3.0	0.0	0.0
duration	2.0	0.0	0.0	0.0	0.0	3.0	1.0
dateTime	2.0	0.0	0.0	0.0	0.0	1.0	3.0

두 중간노드 사이의 유사도가 임계값보다 클 경우 해당 노드는 서로 매칭된다. 여기서, 중간노드 유사도는 식(4)와 같이 중간노드사이의 언어적 유사도와 구조적 유사도의 합으로 정의한다. 언어적 유사도는 단말노드 유사도를 계산할 때와 동일한 방법을 적용하고, 구조적 유사도는 중간노드를 뿌리노드로 갖는 서브트리(subtree)간의 구조적 유사도에 기반한다.

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

한편 중간노드 간의 구조적 유사도는 식 (5)와 같이 해당 서브트리에 포함된 단말노드 매칭의 비율로 정의한다. 즉, 서브트리를 구성하는 단말노드 사이에 매칭관계가 많을수록 해당 중간노드의 유사도가 증가한다.

$$\text{구조적 유사도}(Ns, Nt) = \frac{LNs\text{과 } LNt \text{ 사이에 매칭된 노드의 수}}{|LNs| + |LNt|} \quad (5)$$

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

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

제안된 방법은 일차적으로 경로유사도를 계산하여 경로 유사도가 임계값 이하의 매칭은 제거한다. 예를 들어, 그림 7은 단말노드 매칭 결과에 따라 경로

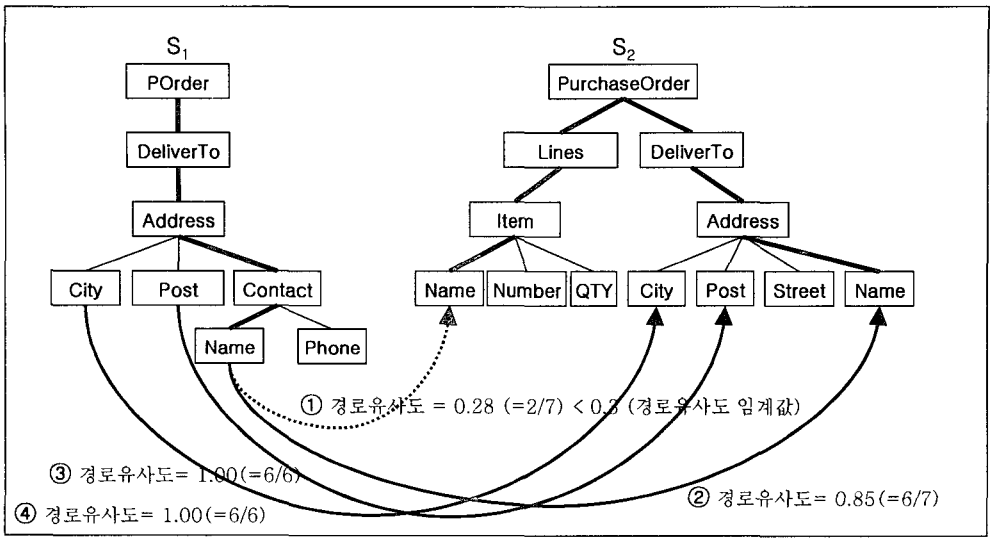

그림 7. 단말노드 매칭에 따른 경로유사도의 계산

유사도가 계산되는 과정을 그린 것이다. 단말노드 매칭결과 4개의 매칭이 계산되고, 각 매칭에 대해 경로유사도를 그림과 같이 계산할 수 있다. 특히, 매칭 ①은 동일한 'Name'이라는 동일한 어휘가 사용되어 단말노드 매칭과정에서 선택이 되었지만, 계산된 경로유사도(=0.28)는 임계값(=0.3)보다 작아 후보 매칭 집합에서 제거된다.

계산된 경로유사도를 이용하여 다대다 관계의 매칭 중에서 일대일 매칭을 계산하는 과정은 그림 8과 같이 두 단계로 구성된다. 먼저 소스트리로부터 일대다(one-to-many) 매칭을 검색하여 가장 유사한 매칭을 선택한다. 즉, 한 소스노드가 여러 타겟노드와 대응관계를 갖는 경우, 경로유사도가 가장 높은 타겟노드를 찾는다. 일대다 관계의 모든 소스 노드에 대하여 이 과정을 반복하여 적용한다.

일대다 매칭관계를 제거한다고 해서 일대일 매칭만 남는 것은 아니다. 하나의 타겟노드에 대해 여러 소스노드가 매칭되는 다대일(many-to-one) 매칭관계가 존재할 수 있다. 따라서 이전 단계에서 추출된 매칭 중에 다대일 매칭을 검색하고 이를 제거하는 과정을 적용한다. 즉, 임의의 타겟노드와 매칭관계를 갖는 소스노드가 다수 존재한다면 해당 매칭 중에서 경로유사도의 값이 가장 큰 매칭을 선택한다. 한편, 경로유사도를 비교하여 가장 유사한 매칭을 찾는 과정에서 동일한 값의 경로유사도를 갖는 매칭이 다수 존재할 수 있다. 이러한 경우, 가장 적절한 매칭의 선택을 위해서 단말노드 유사도가 높은 것을 선택한다.

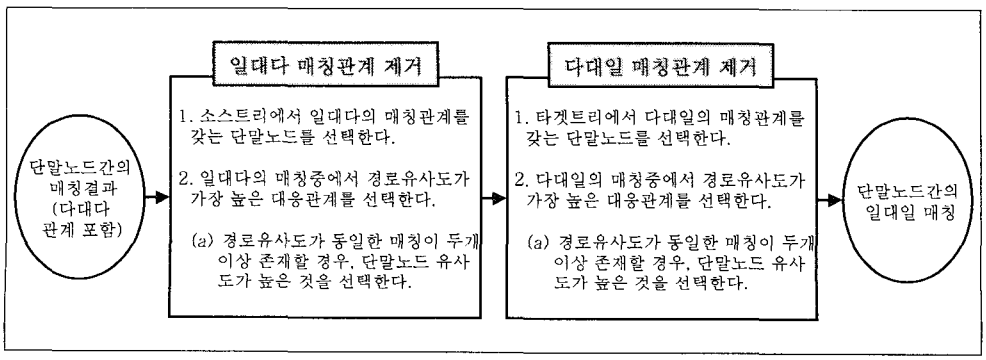

그림 8. 일대일 매칭추출 과정

예를 들어, 그림 9에서 S_1의 요소 'Street'은 S_2의 두개의 요소와 매칭관계를 갖는다. 이때 단말노드의 정보만을 가지고는 매칭 ①과 ② 중에서 보다 정확한 매칭을 판단할 수 없다. 일대일 매칭을 추출하기 위해서 경로유사도가 높은 매칭을 선택한다. 매칭 ①의 두 경로를 구성하는 5개의 중간노드 중에서 'PO'와 'PurchaseOrder'의 2개의 노드가 서로 대응하여 0.4 (=2/5)의 경로유사도를 갖는다. 반면에 ②는 'ShipTo'와 'DeliverTo'가 추가로 매칭되어 0.8(=4/5)의 경로유사도를 갖는다. 따라서 제안된 방법은 매칭 ②를 선택한다.

4. 실험 결과 및 성능 분석

제안된 방법의 성능을 평가하기 위하여 표 3과 같이 전자상거래 분야에서 사용 중인 구매요청서 용 스키마 5개를 대상으로 실험하였다. 실험에 사용된 스키마는 평균적으로 77개의 노드를 포함한다. 실험 결과 및 기존 연구와의 비교에 대한 자세한 기술은 다음과 같다.

4.1 성능 분석

본 논문에서는 5개의 스키마를 가지고 각각의 조합으로 10번의 실험을 수행하였다. 특히, 정확률과 재현률의 두 가지 측면에서 제안된 방법의 성능을 전문가에 의한 수동 매칭결과와 비교하였다. 제안된 방법의 성능을 정량적으로 평가한 결과는 그림 10과 같다.

표 3. 실험에 사용된 XML 스키마

번호	스키마 이름	노드수	URL
1	CIDX	40	http://www.cidx.org
2	Excel	54	http://www.biztalk.org
3	Noris	65	http://www.lcs.cz
4	Paragon	80	http://www.retailtrade.net
5	Apertum	145	http://www.greatplans.de

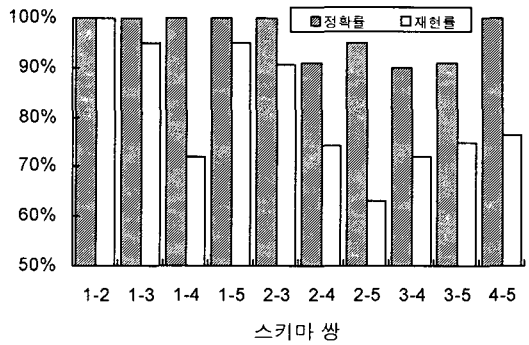

그림 10. 각 실험데이터에 따른 정확률과 재현률

제안된 방법은 평균적으로 97%의 정확률과 81%의 재현률을 보였다. 특히 대부분의 실험에서 높은 정확률을 보여 제안된 방법이 매우 정확하다고 신뢰할 수 있다. 한편, 그림 11은 언어적 유사도를 계산할 때 사용되는 부가정보의 유무에 따른 실험 결과를 나타낸 것이다. 그림에서 A는 축약어 사전, D는 도메인 온톨로지, 그리고 S는 일반 동의어 사전을 의미한다. 여기서 overall은 관련연구인 [20]과 마찬가지로 식(6)과 같이 정의한다. Overall은 전체비용에서 잘못된 매칭결과를 제거하거나, 찾지 못한 실제 매칭을

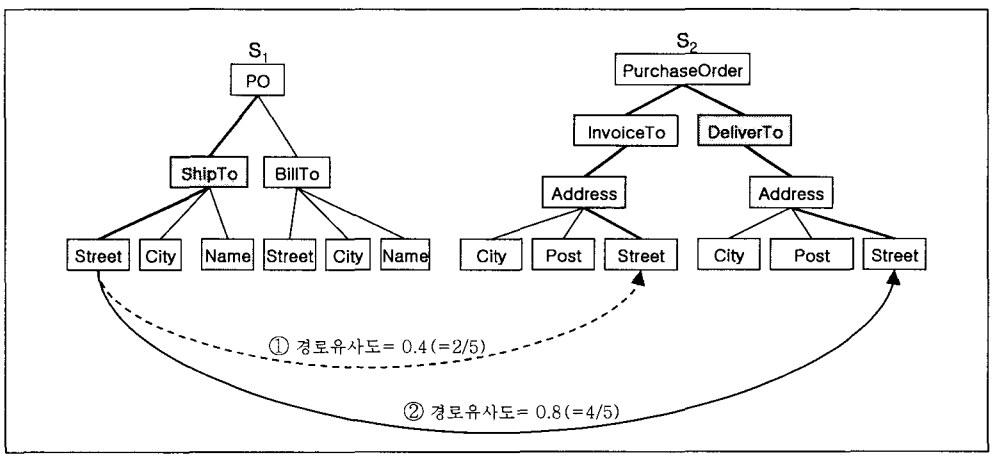

그림 9. 일대일 매칭 추출결과의 예

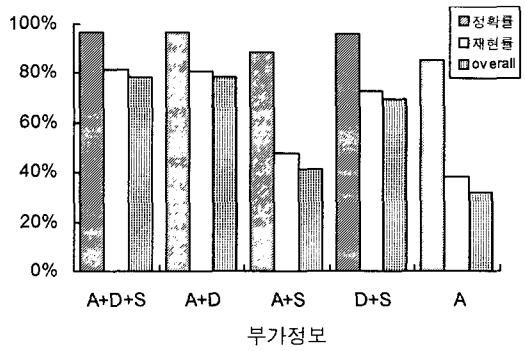

그림 11. 부가정보의 유무에 따른 결과비교

추가하는데 드는 비용을 뺀 것으로, 값이 작을수록 매칭결과에 대해 수정할 필요가 많다는 것을 의미한다.

$$Overall = 재현률 \times \left(2 - \frac{1}{정확률}\right) \quad (6)$$

실험 결과, 도메인 온톨로지의 역할이 정확률을 높히는데 중요하다는 것을 알 수 있다. 실험에서 사용한 도메인 온톨로지는 'head-header'와 같은 일반동의어 사전에는 없거나, 'ship-deliver'와 같이 도메인에서 중요하게 작용하는 단어간의 관계를 포함한다.

한편, 그림 12는 단말노드 임계값의 변화에 따른 실험 결과를 나타낸다. 단말노드 임계값은 초기 후보 매칭 집합을 결정하는 것으로 작으면 단말노드의 관계가 대부분 선택되어 정확성이 많이 떨어진다. 반면에 값이 너무 크면 후보 매칭 집합이 줄어들어 정확한 결과예측이 힘들어진다. 본 논문에서는 다양한 실험을 통해 단말노드 임계값을 overall이 더 이상 개선되지 않는 0.6으로 설정하였다.

이밖에 언어적 유사도에 대한 가중치, w_s,는 0.8, 그리고 데이터 타입 유사도에 대한 가중치, w_t,와 구조적 유사도에 대한 가중치, w_s,는 모두 0.2로 설정하였다. 대부분의 스키마가 string과 같은 보편적인 데

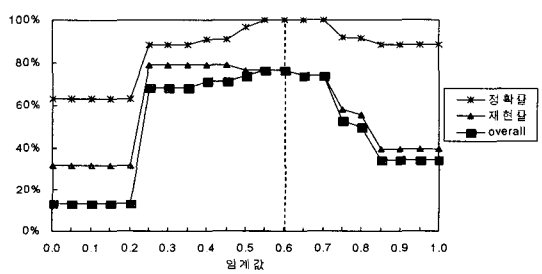

그림 12. 단말노드 임계값의 변화에 따른 실험 결과

이터 타입을 사용하기 때문에 언어적인 부분에 더 많은 가중치를 부여하여 실험하였다. 한편 경로유사도 임계값은 0.3으로 설정하였다. 즉, 경로에 포함된 중간노드 중 매칭되는 노드의 수가 30% 미만이면 의미가 없는 매칭으로 간주하여 제거하였다.

4.2 기존 연구와의 비교

실험 결과, 제안된 방법은 기존의 스키마 매칭 방법과 비교하여 보다 우수한 성능을 보였다. 다수의 스키마 매칭 알고리즘을 비교한 연구 [20]에 의하면 본 논문과 동일한 실험데이터를 대상으로 실험한 COMA가 93%의 정확률과 89%의 재현률을 나타내어 비교 대상 알고리즘 중에서 가장 우수한 것으로 나타났다. COMA는 다대다의 매칭 관계를 허용하기 때문에 제안된 방법보다 많은 수의 매칭관계를 추출하여 높은 재현률을 보인 것으로 분석된다.

자동화된 문서 변환을 위한 연구인 Xtra는 결과로 노드사이의 일대일 매칭을 계산하고, 매칭결과를 이용하여 작성된 변환 스크립트를 제시하고 있다. Xtra는 변환 연산과 각 연산에 따른 비용모델을 이용하여 최소 비용의 연산을 보이는 노드 사이의 관계를 찾는다. 하지만 노드가 포함되어 있는 구조적인 정보를 이용하지 않고 한 소스노드와 한 타겟노드 사이의 변환 과정만 고려하기 때문에 정확한 매칭을 계산하는데 한계가 있다.

LSD에서 제안한 시스템은 매칭결과를 알고 있는 스키마에 적합하게 작성된 많은 양의 XML 문서를 이용하여 학습시키고, 학습 결과를 이용하여 새로운 스키마에 대해 매칭을 찾는다. 학습 기법을 사용한 방법은 학습을 위해 수동으로 학습데이터에 대해 매칭을 수행해야하고, 학습데이터가 다른 스키마의 내용을 포괄할 수 있을 만큼 충분히 풍부해야 하는 단점이 있다. 본 논문의 실험데이터와 매칭의 적용대상이 다르기 때문에 직접적인 비교는 어렵지만 LSD의 실험결과 약 80%의 정확률과 재현률을 보이고 있다.

제안된 방법은 기존 연구보다 높은 수준의 정확률을 보였다. XML 문서의 자동변환 측면에서 높은 정확률은 자동 변환 시스템에 대한 보다 높은 신뢰도를 부여할 것으로 사려된다. 정확률이 낮은 시스템은 사용자가 결과를 다시 확인하는 등의 불필요한 노력이 필요하기 때문에 자동화의 목적에 적합하지 않다. 이런 측면에서 본 논문에서 제안하고 있는 방법은 기존

연구보다 XML 문서의 자동 변환에 적합하다.

5. 결론 및 향후연구

본 논문에서는 XML 문서의 자동변환을 위한 스키마 매칭 알고리즘을 제안하였다. 스키마 매칭에 관한 기존 연구는 다대다의 매칭결과를 계산하기 때문에 문서변환에 직접적으로 이용하기 어려우며 전반적으로 낮은 정확률을 보여 XML 문서의 자동변환에 적용하는데 있어 한계를 갖는다. 제안된 방법은 단말노드의 정보를 이용하여 다대다의 후보 매칭을 계산하고, 단말노드가 포함된 경로유사도에 기반하여 변환에 가장 적합한 일대일 관계의 최종 매칭을 찾는다. 계산된 결과는 문서 변환 스크립트를 생성하는데 바로 사용될 수 있다.

또한, 제안된 방법은 매칭결과의 정확성을 높히기 위하여 축약어 사전, 동의어 사전, 그리고 도메인 온톨로지의 부가정보를 적용하였다. 제안된 방법의 우수성을 입증하기 위하여 전자상거래 분야에서 사용중인 스키마를 대상으로 실험한 결과, 기존 연구보다 높은 정확률과 재현률을 보였다. 자동 변환 시스템에서 정확률은 시스템의 신뢰도를 좌우하기 때문에 성능을 평가하는 중요한 요소이다. XML 문서의 자동변환 시스템은 사용자가 일일이 해야하는 매칭과정을 최대한 줄여주기 위한 것이기 때문에 신뢰도가 낮은 시스템은 사용자의 노력을 줄여줄 수 없다. 따라서 본 연구에서 제안한 스키마 매칭 알고리즘은 XML 문서의 자동 변환 시스템에 적합하다.

제안된 방법은 문서 변환시 노드사이의 일대일 상황만 고려하고 있다. 하지만 스키마 매칭과정에서 분할(spilt)과 병합(merge)에 해당하는 다대일 또는 일대다 관계가 존재할 수 있다. 예를 들어, 소스스키마가 단말노드 'name'을 포함하며 타겟스키마에 요소 'firstName'과 'lastName'이 포함되어 있다고 가정하자. 이러한 경우, 'name'은 'firstName'과 'lastName'에 모두 매칭되어야 할 것이다. 즉 일대다의 매칭을 형성한다. 반대의 경우라면, 'firstName'과 'lastName'이 'name'과 다대일의 관계를 형성할 것이다. 이와 같은 예는 변환과정에서 정보가 분할 또는 병합되는 경우에 해당한다. 이런 경우, 두 문서의 변환을 위해서는 매칭 관계는 물론이고 변환과정에서 분할 또는 복사 연산 중 어느 것을 적용할 것인지 판단해야한다. 다대일 혹은 일대다의 매칭결과만으론 문서를 변환시킬 수 없다. 따라서, 본 연구에서는 일대일의 관계만을 추출하는데 초점을 맞추었다.

향후 연구로서 더욱 자동화된 문서변환을 위하여 일대일 관계의 매칭뿐만 아니라, 분할과 병합에 해당하는 일대다 또는 다대일 관계를 구분하여 추출할 수 있는 방법을 연구할 것이다. 또한, 부가 정보에 따른 정확률 분석결과, 도메인 온톨로지가 정확률을 향상시키는데 중요한 역할을 하고 있음을 알 수 있다. 따라서 기존의 매칭결과나 사용자의 피드백을 이용하여 도메인 온톨로지를 자동으로 구축하고 갱신할 수 있는 방법을 연구할 것이다.

참 고 문 헌

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

[2] World Wide Web Consortium, XML Schema 1.0, W3C Recommendation, http://www.w3.org/TR/xmlschema-0/, 2001.

[3] World Wide Web Consortium, XSL Transformations (XSLT) 1.0, W3C Recommendation, http://www.w3.org/TR/1999/REC-xslt-19991116, 1999.

[4] Eila Kuikka, Paula Leinonen, and Martti Penttonen, "Toward Automating of Document Structure Transformations," *Proc. ACM Symposium on Document Engineering*, pp 103-110, 2002.

[5] MicroSoft biztalk mapper, http://www.microsoft.com/biztalk/.

[6] XSLWiz, http://www.induslogic.com/.

[7] Sonia Bergamaschi, Silvana Castano, Sabrina De Capitani di Vimercati, S. Montanari, and Maurizio Vincini, "An Intelligent Approach to Information Integration," *Proc. Int'l Conf. on Formal Ontology in Information Systems*, pp. 253-267, 1998.

[8] Mong Li Lee, Wynne Hsu, LiangHuai Yang, and Xia Yang, "XClust: Clustering XML Sche-

mas for Effective Integration," *Proc. 11th Int'l Conf. on Information and Knowledge Management*, pp. 292-299, 2002.

[9] Erhard Rahm and Philip A. Bernstein, "A Survey of Approaches to Automatic Schema Matching," *VLDB Journal*, Vol. 10, No. 4, pp. 334-350, 2001.

[10] Hong Su, Harumi Kuno, and Elke A. Rundensteiner, "Automating the Transformation of XML Documents," *Proc. 3rd Int'l Workshop on Web Information and Data Management (WIDM)*, pp. 68-75, 2001.

[11] Tova Milo and Sagit Zohar, "Using Schema Matching to Simplify Heterogeneous Data Translation," *Proc. 24th Int'l Conf. on VLDB*, pp. 122-133, 1998.

[12] Wen-Syan Li and Chris Clifton, "Semantic Integration in Heterogeneous Databases Using Neural Networks," *Proc. 20th Int'l Conf. VLDB*, pp. 1-12, 1994.

[13] 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.

[14] Barbara Staudt Lerner, "A Model for Compound Type Changes Encountered in Schema Evolution," *ACM Transactions on Database Systems*, Vol. 25, No. 1, pp. 83-127, 2000.

[15] Renee J. Miller, Laura M. Haas, Mauricio A. Hernandez, Lingling Yan, C. T. Howard Ho, Ronald Fagin, and Lucian Popa, "The Clio Project: Managing Heterogeneity," *SIGMOD Record*, Vol. 30, No. 1, pp. 78-83, 2001.

[16] Jayant Madhavan, Philip A. Bernstein, and Erhard Rahm, "Generic Schema Matching with Cupid," *Proc. 27th Int'l Conf. VLDB*, pp. 49-58. 2001.

[17] Hong-Hai Do and Erhard Rahm, "COMA - A System for Flexible Combination of Schema Matching Approaches," *Proc. 27th Int'l Conf. VLDB*, pp. 610-621, 2002.

[18] Sergey Melnik, Hector Garcia-Molina, and Erhard Rahm, "Similarity Flooding - A Versatile Graph Matching Algorithm," *Proc. 18th Int'l Conf. on Data Engineering*, pp. 117-128, 2002.

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

[20] Hong Hai Do, Sergey Melnik, and Erhard Rahm, "Comparison of Schema Matching Evaluations," *Lecture Notes in Computer Science*, Vol. 2593, pp. 221-237, 2002.

이 준 승

2003년 연세대학교 컴퓨터과학과 졸업(학사)
2003년~현재 연세대학교 컴퓨터과학과 석사과정

관심분야 : XML 문서 처리

이 경 호

1995년 연세대학교 전산과학과 졸업(학사)
1997년 연세대학교 컴퓨터과학과 졸업(석사)
2001년 연세대학교 컴퓨터과학과 졸업(박사)
2001년 National Institute of Standards and Technology(NIST) 객원연구원
2002년~현재 연세대학교 컴퓨터산업공학부 조교수
관심분야 : XML 기반 멀티미디어 문서처리