반복적 알고리즘을 이용한 온톨로지 매핑

안진현○, 최기선
KAIST 전산학과
jhahn@world.kaist.ac.kr, kschoi@cs.kaist.ac.kr

An iterative algorithm for Ontology mapping

Jinhyun Ahn○, Key-Sun Choi
Computer Science Department, KAIST

요 약

온톨로지 매핑은 서로 다른 온톨로지에 있는 클래스가 유사한 개념을 표현한 것인지 판단하는 문제이다. 클래스 유사도를 계산 하는 방법에는 클래스의 이름 어휘 유사도, 의미 유사도, 클래스 관계/속성 유사도 그리고 클래스 상하위 관계 유사도 등이 제안되었다. 본 논문에서는 이러한 클래스 유사도를 계산하기 위한 반복적 유사도 계산 알고리즘을 제안한다. 매 반복 단계마다 모든 클래스 쌍의 유사도를 전부 갱신 하는 방법과 유사도가 최대인 쌍만 선택적으로 갱신 하는 방법을 비교 실험하였다. 실험 결과 유사도가 최대인 쌍만 업데이트하는 방법의 성능이 좋았고 소요시간도 적었다.

주제어: 온톨로지 매핑, 시맨틱 웹, 단어 유사도

1. 서 론

온톨로지는 “특정 도메인에 공통적으로 받아들여지는 개념과 개념들 간의 관계를 표현”한 것이다 [9]. 온톨로지에 표현된 개념을 사용한 서로 다른 시스템을 통합하는 경우, 온톨로지의 각 개념 간의 대응 관계를 식별할 필요가 있다. 동일한 의미의 개념을 온톨로지에서는 다르게 표현되어 있을 수도 있기 때문이다. 동일한 의미의 개념 쌍을 식별함으로써 그 개념과 관련된 데이터를 서로 다른 시스템끼리 공유할 수 있다.

온톨로지에서의 개념은 클래스로 표현이 된다. 주어진 두 개의 온톨로지의 각 클래스 간의 대응 관계를 식별하는 작업을 온톨로지 매핑[1]이라고 한다[2]. 두 개의 클래스가 유사한 개념을 가리키고 있다면, 두 개의 클래스는 유사하고 따라서 대응 관계가 있다고 정의한다. 클래스 쌍이 개념적으로 유사한지 판단하는 여러 가지 방법이 제시되었다 [2]. 그 중 잘 알려진 방법으로는, 클래스 이름 어휘 유사도, 클래스 이름 의미 유사도, 클래스 관계/속성 유사도 그리고 클래스 상하위 관계 유사도 등이 있다 [5].

본 논문에서는 기존의 잘 알려진 클래스 유사도 계산 방법들을 사용하여 주어진 두 개의 온톨로지의 모든 클래스 쌍 간의 유사도를 계산하는 반복적 알고리즘을 제시한다. 클래스 상하위 관계 유사도의 경우 해당 클래스 쌍의 유사도가 상하위 클래스 쌍 간의 유사도에 영향을 받으므로, 클래스 간의 유사도를 반복적으로 갱신할 필요가 있다. 클래스 간의 유사도를 갱신하는 방법에는 두 가지 방법이 있다. 첫 째는 매 반복 단계마다 모든 클래스 쌍의 유사도를 갱신 하는 것이고, 둘째는 유사도가 최대인 클래스 쌍만 선택적으로 갱신 하는 방법이다. Ontology Alignment Evaluation Initiative[2]에서 제공하는 벤치마크 온톨로지를 사용하여, 두 방법의 소요시간과 성능을 비교 하였다. 실험 결과 유사도가 최대인 쌍만 업데이트하는 방법의 성능이 좋았고 소요시간도 적었다.

2. 관련 연구

Ehrig, M[5]는 온톨로지 매핑 시 클래스 간 유사도 계산에 사용되는 자질들을 제시하였다. 클래스 이름, URI (Uniform Resource Identifier)를 포함한 클래스 이름, 관계 이름, 상하위 개념, 인스턴스, 관계의 정의역(domain)과 치역(range) 등을 유사도에 기여하는 정도에 따라 우선순위를 정해서 유사도를 계산하는 방법을 제시하였다. 하지만, 우선순위에 대한 근거는 기술이 안 되어 있다.

Ehrig, M[4]은 온톨로지 매핑에 필요한 일반적인 작업들을 정의하고 반복적으로 수행함으로써 온톨로지 매핑 정확도를 점점 증가시키는 온톨로지 매핑 프레임 워크를 제시하였다.

Tang [6]은 온톨로지 매핑 문제를 위험 최소화(risk minimization) 문제로 정의하였다. 클래스간의 모든 대응 관계 경우의 수에서 잘못된 대응 관계들이 가장 적게 나타나는 경우를 찾는다. 사용된 자질은 이전 연구와 비슷하게 클래스 이름, 인스턴스, 속성 정보 그리고 상하위 개념 등이다.

3. 문제 정의

온톨로지는 클래스 이름, 클래스 상하위 관계, 관계 이름, 관계의 공역(domain)과 치역(range), 관계 상하위 관계, 속성 이름, 속성의 공역과 치역 그리고 데이터 타입으로 구성되어 있다[1]. 본 연구에서는 클래스와 인스턴

1) 본 논문에서는 (Ontology) alignment, mapping, matching 등을 구분하지 않고 모두 온톨로지 매핑이라 칭한다.

2) http://oaei.ontologymatching.org/

스를 구분하지 않고 모두 클래스라고 가정한다. 각 구성 요소에 대한 예제는 표 1에 기술되어 있다.

구성 요소	예제
클래스 이름	Thing, Device, Protocol, Mobile_Device
클래스 상하위 관계	(Thing, Device), (Device, Mobile_Device)
관계 이름	support,
관계의 공역과 치역	(support, (Mobile_Device, Protocol))
관계의 상하위 관계	
속성 이름	hasWeight
속성의 공역과 치역	(hasWeight,(Mobile_Device, Integer))
데이터 타입	Real Number

표 1. 온톨로지의 구성 요소 및 예제 [1]

본 연구에서는 온톨로지의 구성요소 중에서 클래스 이름, 클래스 상하위 관계, 관계 이름 그리고 속성 이름만 고려한다. 관계 이름과 속성 이름은 구분하지 않고 동일하게 취급한다.

온톨로지 매핑은 주어진 두 개의 온톨로지의 각 클래스간의 대응 관계를 식별하는 작업이다[3]. 매핑 하려는 원 온톨로지(source ontology)와 매핑 대상이 되는 대상 온톨로지(target ontology) 등 두 개의 온톨로지가 주어졌을 때, 클래스 간 대응 관계는 4-튜플로 정의된다 [1].

<c1, c2, mr, n>

c1 : 원 온톨로지의 클래스
c2 : 대상 온톨로지의 클래스
mr : 대응 관계 (동의어, 포함관계)
n : 대응에 대한 확신도 (0.0 ~ 1.0)

본 논문에서는 동의어 관계와 포함 관계를 구분하지 않는다. 동의어 관계는, 서로 같은 개념을 가리키지만 다른 이름으로 표현 된 경우를 가리킨다. 포함 관계는, 동의어 관계에 해당하는 개념이 대상 온톨로지에 존재하지 않지만 상위 개념은 존재하는 경우가 있다. 예를 들어, "Wi-Bro" 이라는 개념에 대한 동의어 개념이 대상 온톨로지에 없고 대신 "IEEE standard"라는 개념이 존재하는 경우가 있다. "Wi-Bro"라는 개념을 "IEEE standard"에 포함 관계로 대응 시킨다. 본 논문에서는 동의어 대응 관계와 포함 대응 관계를 구분하지 않고 둘을 포괄하는 유사 대응 관계만 고려한다. 따라서 위의 4-튜플에서 mr은 항상 같은 값을 가지고 유사 대응 관계에 대한 확신도는 n에 의해 표현된다. 클래스 간 대응 서수(cardinality)는 M 대 1로 제한한다. 즉, 대상 온톨로지의 하나의 클래스에 원 클래스의 여러 개의 클래스가 대응될 수 있다.

본 논문에서는 원 온톨로지와 대상 온톨로지가 주어졌을 때, 클래스 간 대응 관계를 계산하는 방법을 제시한다. 먼저 기존의 알려진 두 개의 클래스 사이의 유사도 계산 방법을 제시하고, 전체 클래스 사이의 유사도 계산 알고리즘을 제시한다.

4. 클래스 유사도

서로 다른 온톨로지에 있는 각각의 클래스의 유사성을 판단하는 방법 들[2] 중 본 논문에서는 잘 알려진 방법 네 가지를 사용한다. 각각에 대해 간단히 소개한다.

클래스 이름 어휘 유사도는 클래스의 이름의 어휘 단위의 유사도로서, 가장 간단한 방법은 Edit distance를 사용한 방법이다 [3]. 그림 1은 클래스 이름 어휘 유사도 공식이다.

$$\text{LabelLexicalSim}(c1,c2) = \frac{|\{(c1,c2) \mid \text{EditDist}(a,b) > h \text{ such that } a \in \text{tokens}(c1) \text{ and } b \in \text{tokens}(c2)\}|}{\max(|\text{tokens}(c1)|, |\text{tokens}(c2)|)}$$

where
$\text{tokens}(w)$ is the set of normalized tokens in w
h is the threshold
$\text{EditDist}(a,b)$ is a conventional edit distance function

그림 1. 클래스 이름 어휘 유사도

Edit distance 값이 일정 임계치 이하의 토큰 쌍을 유사 토큰이라고 정의하고, 유사 토큰의 비율을 계산하는 공식이다. 예를 들어, TouchPad와 TouchScreen의 이름 어휘 유사도는 1/2이다.

클래스 이름 의미 유사도는 이름의 표현에 관계없이 의미적인 유사도로서, 잘 알려진 방법은 WordNet을 사용한 방법이다 [8]. WordNet synset의 분류체계 상에서 각각의 클래스 이름에 해당하는 synset을 찾고 그 두 개의 synset의 공통 부모 synset을 찾는다. 그 공통 부모 synset의 정보량[7]이 두 개의 클래스 이름의 의미 유사도이다. 분류체계 상 상위에 있을 수록 정보량이 적으므로, 공통 부모 synset이 상위에 있을 수록 유사도가 작다.

클래스 관계/속성 유사도는 두 개의 클래스가 가지는 관계와 속성이 서로 겹치는 정도를 나타낸다 [5]. 그림 2는 두 개의 클래스의 관계 및 속성이 겹치는 정도를 계산하는 공식이다.

$$\text{PropertySim}(c1,c2) = \frac{|\{r' \mid (c1,r',c') \in O1\} \cap \{r \mid (c2,r,c') \in O2\}|}{|\{r \mid (c1,r,c') \in O2\}|}$$

where
O1 is the source ontology and O2 is the target ontology

그림 2. 클래스 속성 유사도 공식

r은 관계와 속성을 모두 지칭한다. 예를 들어, 다음과 같이 클래스와 그에 대한 관계/속성이 원 온톨로지와 대상 온톨로지에 있다고 가정하자.

원 온톨로지
PDA, support, Wi-Fi
PDA, runsOn, PalmOS

대상 온톨로지
SmartPhone, consistOf, LCD
SmartPhone, support, WiBro

SmartPhone, runsOn WindowsCE

클래스 관계/속성 유사도는 2/3이다. 왜냐하면 SmartPhone의 관계/속성의 개수는 3개이고 그 중에 PDA와 공유하는 것은 두 개이기 때문이다.

클래스 상하위 관계 유사도는 주어진 두 개의 클래스 각각의 상하위 클래스가 서로 유사한 정도를 나타낸다 [2]. 즉, 서로 유사한 상위 클래스를 가지거나 또는 서로 유사한 하위 클래스를 가지는 두 개의 클래스는 유사하다는 가정에 바탕을 두고 있다.

4. 클래스 유사도 계산 알고리즘

원 온톨로지와 대상 온톨로지에 있는 모든 클래스 쌍에 대해 클래스 유사도를 계산하는 방법을 설명한다. 본 연구에서는 행렬 기반의 반복적 알고리즘을 제안한다. 표 2에 제안된 알고리즘의 입력, 출력 그리고 과정을 기술하였다. 구체적인 과정은 다음과 같다. 먼저 원 온톨로지와 대상 온톨로지의 각 클래스를 열과 행으로 하는 행렬을 생성한다. 행렬의 각 셀은 해당 되는 클래스 쌍의 유사도를 의미한다. 처음에는 어휘_유사도, 의미_유사도 그리고 관계/속성 유사도의 선형 조합(linear combination) 사용해서 행렬의 각 셀을 초기화한다. 다음은 이 행렬을 반복적으로 갱신하는 과정이다. 이러한 과정을 수행하는 이유는 상하위 관계 유사도를 계산하기 위해서이다. 상하위 관계 유사도는 대상이 되는 두 개의 클래스 자체의 유사도 이외의 상위 클래스와 하위 클래스의 유사도를 고려하기 때문에, 유사도가 상하위 클래스의 유사도에 영향을 받는다.

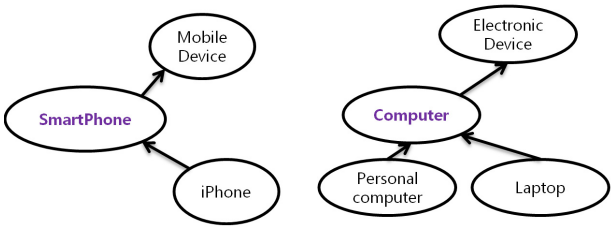

그림 3. 원 온톨로지와 대상 온톨로지의 분류체계의 예제

예를 들어, 그림 3에서 SmartPhone과 Computer 사이의 유사도는 Mobile Device와 Electric Device 사의의 유사도 값에 영향을 받는다. 또한 Mobile Device와 Electric Device 사이의 유사도는 그 상위 클래스 사이의 유사도에 영향을 받는다.

행렬의 각 셀을 갱신하는 과정에서 어떤 셀을 갱신할지 선택하는 두 가지 방법이 있다. 첫 째는 모든 셀을 갱신하는 것이고 둘 째는 각 열에 대해 최대값 만 갱신하는 것이다. 두 번째 방법의 경우, 예를 들어, 표 3에서는 굵게 표시된 부분만 갱신하는 것이다. 표 3의 경우 원 온톨로지의 MasterThesis라는 클래스는 대상 온톨로지의 MScThesis라는 클래스에 유사 대응 관계 확신도가 0.461538이라는 의미이다.

```
알고리즘 : 클래스 간 유사도 계산

입력 : 원 온톨로지 O1, 대상 온톨로지 O2
출력 : 클래스 간 유사도 행렬

과정 :
A by B 행렬 M 생성
        //A은 O1에 있는 클래스의 개수
        //B은 O2에 있는 클래스의 개수
M을 초기화
M(i, j) = C*(어휘_유사도(i,j) + 의미_유사도(i,j) + 관계/속성_
          유사도(i,j))
          C는 정규화 상수
//O1에 있는 i번째 클래스와 O2에 있는 j번째 클래스의 유사
도 계산하여 M의 각 셀에 저장

while(M의 값이 변하지 않을 때 까지)
 //업데이트할 대상 셀(i, j) 선택 (전부 or 최대값)
 M(i, j) = (M(i, j) + 상하위 관계 유사도(i, j))/2

M을 출력
```

표 2. 클래스간 유사도 계산 알고리즘

	MasterThesis	Misc	Page Range
JournalPaper	0	0	0.01
List	0	**0.01**	0
MScThesis	**0.461538**	0.005	0
PageInterval	0	0	**0.338333**

표 3. 첫 번째 반복 단계에서의 행렬 M의 일부

5. 실험

본 논문에서 제시한 온톨로지 매핑 알고리즘의 두 가지 방법을 비교하기 위해 실험을 하였다. 실험 데이터는 Ontology Alignment Evaluation Initiative에서 제공하는 2008년 벤치마크 온톨로지를 사용하였다. 벤치마크 온톨로지는 33개의 클래스, 24개의 관계, 40개의 속성으로 구성되었다. 원 온톨로지는 101 온톨로지를 사용하였고 대상 온톨로지는 클래스 이름을 동의어로 바꾼 205 온톨로지를 사용하였다[3]. 그림 4는 각 반복 단계 별 결과이다. Brute-force는 전체 셀을 선택하여 갱신하는 방법이고 best-only는 최대값을 가지는 셀만 선택하여 갱신하는 방법이다. 두 가지 방법의 소요 시간과 성능을 비교하였다.

그림 4는 각 반복 단계별 소비 시간이다. Brute-force 방법의 경우 매 단계를 수행하는데 평균 2682.125 mili-second가 걸린 반면, Best-only 방법의 경우 매 단계를 수행하는데 평균 1440.522 mili-second가 걸렸다. 약 2배 정도의 시간이 차이가 났다.

3) http://oaei.ontologymatching.org/2008/benchmarks/

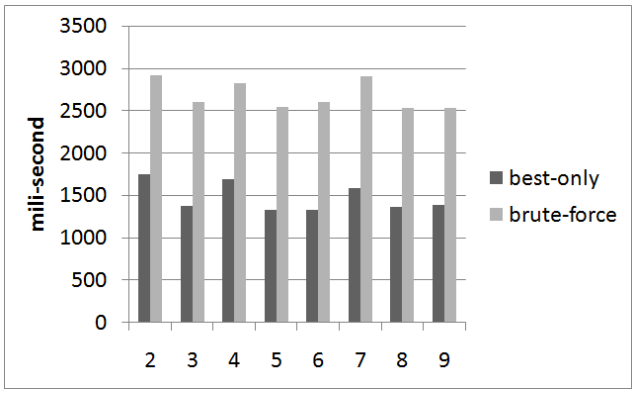

그림 4. 전체 클래스 쌍을 업데이트하는 전략과 유사도가 최대인 쌍만 업데이트하는 전략의 각 반복횟수별 소비 시간 비교. 첫 번째 반복단계의 경우 행렬을 초기화하는 단계이므로 소요시간이 동일하므로 제외하였다.

그림 5는 각 반복 단계 별 두 가지 방법에 대한 F-measure 비교이다. best-only방법의 경우 수렴하였고, brute-force는 점점 성능이 떨어졌다. best-only 방법의 경우 초기의 유사도 값에 큰 변화를 주지 않는 범위에서 갱신이 이루어졌기 때문에 성능에 큰 변화가 없었고, Brute-force 방법의 경우 매 반복 단계마다 유사도 값이 큰 변화를 겪기 때문에 초기 유사도 값의 영향이 점점 줄어들게 되어 성능이 점점 떨어졌다. 즉, 상하위 관계 유사도가 크게 도움이 안 되었다고 볼 수 있다. 이것은 2가지 이유가 있다. 첫 째, 본 논문에서 제시한 클래스 이름 유사도, 의미 유사도 그리고 관계/속성 유사도 자체의 문제점이다. 이 부분은 향후 연구를 통해 분석할 것이다. 실제로 다른 온톨로지 매핑 시스템이 같은 데이터에 대해 precision과 recall 모두 0.9 이상의 성능[10]을 보여주고 있기 때문에 클래스 유사도에 대한 성능 개선이 필요하다. 둘 째, 온톨로지의 분류체계와 상하위 관계 유사도간의 상관관계가 적을 수 있다는 점이다. 즉, 분류체계 상에서 크게 관련이 없는 클래스를 상하위 관계로 구성한 경우 상하위 관계 유사도가 도움이 안 될 것이다.

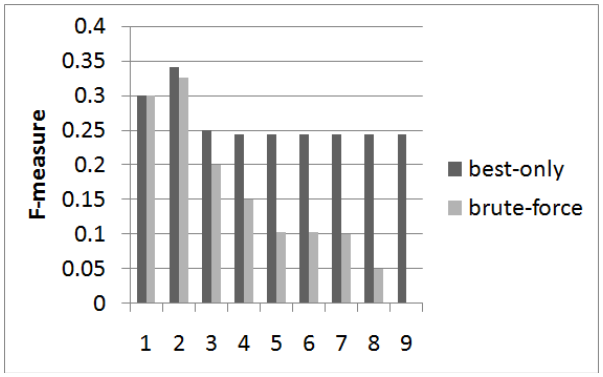

그림 5. 전체 클래스 쌍을 업데이트하는 전략과 유사도가 최대인 쌍만 업데이트하는 전략의 각 반복횟수별 F-measure 비교

brute-force 방법의 경우 best-only 방법에 비해 복잡도도 크고 성능도 좋지 않다는 것을 관찰 하였다. 하지만, 실험 데이터의 크기가 크지 않기 때문에 다른 경우에는 다른 결과가 있을 수 있다.

6. 결론

본 논문에서는 두 개의 온톨로지에 있는 각 클래스 간의 유사도를 계산하는 알고리즘을 제시하였다. 클래스 유사도는 클래스 이름 어휘 유사도, 의미 유사도, 클래스 속성 유사도 그리고 클래스 상하위 관계 유사도의 조합으로 정의했다. 서로 다른 온톨로지에 있는 각각의 클래스 쌍 간의 상하위 관계 유사도를 계산하기 위해 행렬을 이용한 반복적 유사도 계산 알고리즘을 제안하였다. 반복적으로 갱신하는 두 가지 방법의 성능과 복잡도를 비교하기 위해 실험을 하였다. 실험 결과 초기 유사도 값에서 최대값만 선택해서 갱신하는 방법의 성능이 더 좋았고 시간도 적게 들었다.

실험 데이터가 작기 때문에 일반화하기는 향후 좀 더 많은 클래스로 구성된 온톨로지를 대상으로 실험을 할 예정이다. 현재의 클래스 유사도는 선형 조합으로 정의가 되었는데 평가를 해서 각 클래스 유사도 측정 방법에 대한 문제점을 분석하고 개선할 예정이다.

Acknowledgements

본 논문은 지식경제부 및 정보통신연구진흥원의 정보통신선도기반기술개발사업과 2009년도 두뇌한국 21사업의 지원을 받아 수행되었습니다.

본 논문은 KAIST 연구원의 IT 융합 연구소의 지원을 받아 수행되었습니다.

참고문헌

[1] Cimiano, P.: Ontology Learning and Population from Text - Algorithms, Evaluation and Applications. Springer (2006)
[2] Euzenat, J., Shvaiko, Pavel Ontology Matching. Springer (2007)
[3] Jérôme Euzenat, Expressive alignment language and implementation, KWEB EU-IST-2004-507482, 2007
[4] Ehrig, M., Staab, S., Sure, Y.: Bootstrapping ontology alignment methods with APFEL. Special interest tracks and posters of the 14th international conference on World Wide Web, Chiba, Japan (2005) 1148--1149
[5] Ehrig, M., Sure, Y.: Ontology Mapping by Axioms (OMA). In Wissensmanagement (2005) 528-532
[6] Tang, J., Li, J., Liang, B., Huang, X., Li, Y., Wang, K.: Using Bayesian decision for ontology mapping. Journal of Web Semantics 4 (2006) 243-262
[7] Lin, D. An Information-Theoretic Definition of Similarity. In Proceedings of the Fifteenth international Conference on Machine Learning (July 24 - 27, 1998).
[8] Resnik, P. Semantic Similarity in a Taxonomy: An

Information-Based Measure and its Application to Problems of Ambiguity in Natural Language, Journal of Artificial Intelligence Research, Vol. 11, pp. 95-130. 1999
[9] Gruber, T.R.: Toward principles for the design of ontologies used for knowledge sharing. The International Journal of Human-Computer Studies 43 (1995) 907--928
[10] Yves R. Jean-Mary and Mansur R. Kabuka, ASMOV: results for OAEI 2008, The Third International Workshop on Ontology Matching, October 26, 2008, Karlsruhe, Germany