논문 19-45-02-26 The Journal of Korean Institute of Communications and Information Sciences '20-02 Vol.45 No.02
https://doi.org/10.7840/kics.2020.45.2.418

온톨로지 인스턴스 및 Linked Open Data 자원의 관계를 활용한 온톨로지 스키마 정렬 방법

김 민 환◆, 김 종 모*, 손 미 애°, 박 규 동**

Ontology Schema Alignment Method Using Relationship among Ontology Instances and Resources in Linked Open Data

Minhwan Kim◆, Jongmo Kim*, Mye Sohn°, Gyudong Park**

요 약

본 논문에서는 LOD 클라우드를 외부 지식으로 활용해 온톨로지 스키마를 정렬할 수 있는 방법을 제안한다. 이를 위해, 온톨로지 클래스와 인스턴스의 관계, 인스턴스와 LOD 자원의 관계 및 LOD 자원들 간의 계층 구조를 식별하였다. 온톨로지 클래스의 인스턴스와 개념적으로 연관성이 있는 LOD 클라우드의 자원을 찾기 위해 술어(predicate)를 활용한 One-class 클러스터링을 수행하였으며, Latent Semantic Analysis(LSA) 기반의 유사도 계산을 통해 클래스와 의미적 연관성이 가장 큰 LOD 자원을 결정하였다. 본 논문에서 제안한 프레임워크의 우수성을 입증하기 위해, 세 가지 온톨로지를 개발해 실험을 수행하였다. 실험 결과, 온톨로지 클래스와 자원간의 동치관계 식별과 이종 온톨로지 클래스 간의 동치관계 식별에서 우수한 성능을 확인했다.

키워드 : 온톨로지 스키마 정렬, LOD, One-class 클러스터링, Latent Semantic Analysis(LSA)
Key Words : Ontology schema alignment, LOD, One-class clustering, Latent Semantic Analysis(LSA)

ABSTRACT

In this paper, we propose a method to perform ontology schema alignment using LOD cloud as external knowledge. To do so, we attempted to align the ontology schema by identifying the relationship between the ontology classes and the instances, the relationship between the instances and the LOD resources, and the hierarchical structure among the LOD resources. We performed one-class clustering using the predicates to find the resources in the LOD cloud that are conceptually related to the instances of the ontology classes. We also determined the LOD resources with the most semantic association with the classes through the similarity calculation based on Latent Semantic Analysis (LSA). In order to prove the superiority of the proposed framework, three ontologies were developed. Experimental results show that the identification of equivalence relationships between ontology classes and resources and the identification of equivalence relationships between heterogeneous ontology classes are excellent.

※ 본 연구는 국방과학연구소의 국방 지휘통제 통합·연동기반기술 특화연구실 과제의 지원을 받아 수행되었습니다(UD180014ED).
◆ First Author : Sungkyunkwan University Department of Industrial Engineering, kmh3178@skku.edu, 학생회원
° Corresponding Author : Sungkyunkwan University Department of Industrial Engineering, myesohn@skku.edu, 정회원
* Department of Industrial Engineering, Sungkyunkwan University, dignityc@skku.edu
** 2nd R&D Institute, Agency for Defense Development, iobject@add.re.kr, 정회원
논문번호 : 201911-311-0-SE, Received November 12, 2019; Revised December 24, 2019; Accepted December 24, 2019

Ⅰ. 서 론

최근 컴퓨터 기술의 급속한 발전으로 인해 기존의 텍스트 위주의 사용자 환경에서 벗어나 이미지, 그래픽, 오디오 및 비디오 데이터 등을 제공하는 멀티미디어 사용자 환경으로 변화하고 있다.

온톨로지는 이질적인 도메인에서 사용되는 데이터들을 의미적으로 통합·공유하는 도구이다[1]. 그러나 데이터의 통합·공유에 활용할 수 있는 완성도 높은 온톨로지를 개발하기 위해서는 많은 시간과 노력이 소요된다[2]. 이를 극복하기 위해 등장한 방법 중의 하나가 온톨로지 정렬(ontology alignment)이다[3]. 온톨로지 정렬은 도메인 교차지식(cross-domain knowledge)을 내포하고 있는 다수의 도메인 온톨로지가 존재한다는 전제하에, 도메인 온톨로지에 포함된 클래스들 간의 "동치관계 (equivalence)" 또는 "상하관계 (subsumption)"를 식별하는 것으로 정의된다. 이러한 정의에서 따르면, 온톨로지 정렬의 궁극적인 목표는 다수의 도메인 온톨로지를 하나의 완성도 높은 온톨로지로 통합하거나 또는 일부 개념들을 활용해 새로운 온톨로지를 구축하는 것이라 할 수 있다.

온톨로지 정렬 연구는 스키마 기반 온톨로지 정렬(schema-based schema alignment)과 인스턴스 기반 온톨로지 정렬(instance-based schema alignment)로 분류된다. 전자는 대상 온톨로지의 구조가 상이하고 온톨로지별 클래스나 속성(property)을 표현하는 어휘가 다를 수 있기 때문에 매우 제한적인 수준에서의 온톨로지 정렬만을 수행한다. 후자의 경우, 두 개의 강력한 가정을 전제로 온톨로지 정렬을 수행한다. 첫 번째 가정은 온톨로지가 계층적인 구조를 갖고 있어야 한다는 것이고, 두 번째 가정은 사전에 온톨로지 인스턴스들 간의 정렬이 수행되어 있어야 한다는 것이다. 대부분의 온톨로지는 계층적 구조를 가지므로 첫 번째 가정을 만족시키는 것은 어렵지 않으나, 두 번째 가정을 만족하는 온톨로지를 찾는 것은 현실적으로 불가능하다. 이러한 문제를 해결하기 위해, 온톨로지에 선언된 요소들뿐만 아니라 WordNet이나 Wikipedia와 같은 온톨로지의 외부 정보를 활용해 두 온톨로지의 클래스들 간의 유사도 계산을 하고자 하는 연구가 수행되기도 했다[4]. 그러나 WordNet은 오직 영어로만 이루어진 어휘의 데이터베이스이기 때문에 적용 범위가 매우 제한적이며, Wikipedia는 다양한 정보가 많이 있어 적용 범위에는 문제가 없으나 문서들이 단순히 링크로만 연결되어 있기 때문에 온톨로지 정렬을 위한 참조 정보를 얻기가 매우 어렵기 때문에 온톨로지 정렬을 위한 외부 자원으로 활용하는 데 한계가 있다.

앞서 언급한 바와 같이, 온톨로지 정렬의 가장 큰 전제 조건은 대상 온톨로지들이 교차지식(cross-domain knowledge)을 내포하고 있어야 한다는 것이다. 이러한 맥락에서 온톨로지 정렬에 외부 지식(background knowledge)도 온톨로지들과 관련된 도메인 교차지식을 반드시 포함하고 있어야 한다. 이러한 도메인 교차지식을 가장 포괄적으로 내포하고 있는 것이 분산 웹 데이터베이스들을 연결하고 있는 Linked Open Data(LOD) 클라우드이다. LOD 클라우드를 외부 자원으로 사용한다면, WordNet의 적용 범위 문제와 Wikipedia의 참조 정보 획득 문제를 동시에 해결하는 것이 가능하다.

본 논문에서는 LOD 클라우드를 외부 지식으로 활용해 온톨로지 스키마를 정렬할 수 있는 방법을 제안한다. 제안 방법은 클래스의 인스턴스와 개념적으로 연관성이 있는 LOD 클라우드의 자원(resource)들을 찾은 후 해당 LOD 자원의 관계 정보를 활용해 온톨로지 클래스와 LOD 자원 간의 의미적 동치 관계를 찾아 이를 기반으로 온톨로지 스키마 정렬을 수행하는 것이다. 이 때, 클래스의 인스턴스와 개념적으로 연관성이 있는 LOD 클라우드의 자원을 찾기 위해 술어(predicate)을 활용한 one-class 클러스터링을 수행한다. 또한 Latent Semantic Analysis(LSA) 기반의 유사도 계산을 통해 클래스와 의미적 연관성이 가장 큰 LOD 자원을 결정한다. 이상의 과정을 모든 클래스에 대해 수행하면, 의미적으로 연관된 "클래스-LOD 자원"을 찾을 수 있고 이들 관계를 기반으로 클래스들 간의 관계를 유추하는 것이 가능해 진다.

본 논문은 다음과 같이 구성된다. 2장에서는 예제 시나리오를 사용해 온톨로지 정렬에 대해 설명하고, 3장에서는 전체 프레임워크 및 세부 기능에 대해 설명한다. 4장에서는 제안한 방법의 우수성을 입증하기 위해 실험 결과를 그리고 5장에서는 관련 연구에 대해 기술하였다. 마지막으로 6장에서는 결론 및 향후 연구 방향을 제시하였다.

Ⅱ. 예제 시나리오

본 절에서는 예제를 통해 LOD 기반 온톨로지 정렬 과정에 대한 이해를 돕고자 한다. 설명의 편의를 위해 온톨로지 정렬 대상 온톨로지를 두 개(O_1, O_2)로 가정했으며, 대상 온톨로지의 부분 구조를 그림 1에 도식

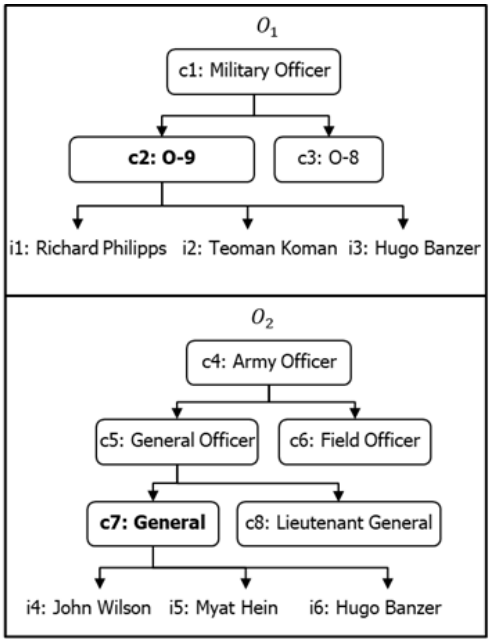

그림 1. 예제 온톨로지
Fig. 1. Illustrative ontologies

화하였다. 첫 번째 대상 온톨로지인 O_1는 2개의 최하위 클래스($c2$,$c3$)를 가지며, 이 중 $c2$는 3개의 인스턴스를 갖는다. 마찬가지로 두 번째 대상 온톨로지 O_2는 2개의 최하위 클래스($c7$,$c8$)를 가지며 $c7$는 3개의 인스턴스를 갖는다. O_1과 O_2는 군 구조(계급)와 관련된 온톨로지임에도 불구하고 이들의 구조나 클래스 명칭 등이 상이하기 때문에 두 온톨로지의 클래스들 간의 동치관계를 식별하기가 용이하지 않다.

이러한 어려움을 해결하기 위해, 대상 온톨로지의 말단 클래스가 갖는 인스턴스(그림에서 i_1-i_6)들과 관계가 있는 LOD 자원을 찾은 후, 이를 활용해 온톨로지 클래스와 가장 유사한 LOD 자원과의 동치관계를 식별한다. O_1 클래스와 LOD 자원간의 동치관계를 식별하는 과정을 그림 2에 도식화하였다. 먼저 O_1의 c_2에 속하는 인스턴스(i_1-i_3)들과 관계가 있는 LOD 자원(r_6-r_8)를 발견하고, 이를 토대로 LOD의 관계를 해석하여 c_2와 가장 유사한 LOD 자원인 r_2를 식별한다. 마지막으로, 판별을 통해 동치관계, 즉 "owl:sameAs"를 술어(predicate 혹은 property)로 활용한다.

이상의 과정을 정렬 대상인 모든 온톨로지와 그들의 클래스에 대해서 수행한 후, 이를 활용해 온톨로지 클래스들 간의 동치관계를 식별한다. 그림 3은 클래스

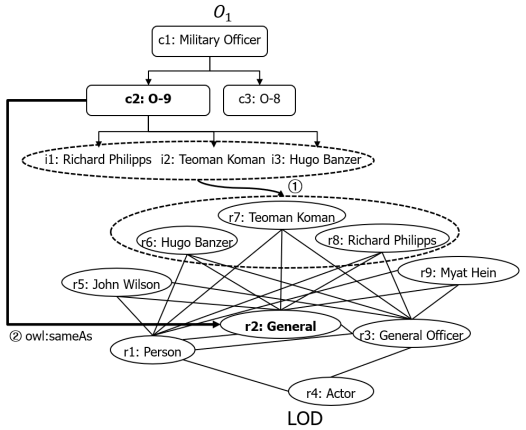

그림 2. O2 클래스와 LOD 자원간의 관계
Fig. 2. Relationships between O2 classes and LOD resource

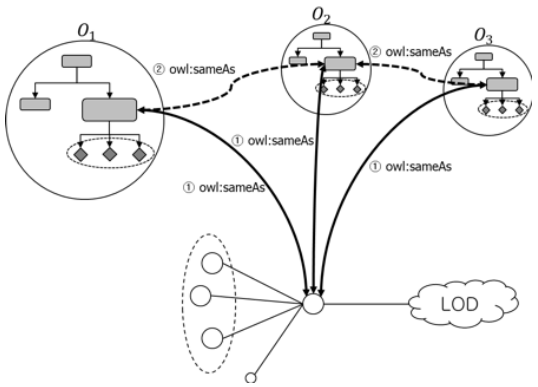

그림 3. 동치관계(owl:sameAs) 식별
Fig. 3. Identification of the equivalence relation (owl:sameAs)

들 간의 존재하는 동치 관계를 식별한 결과이다. 결론적으로, 온톨로지 클래스와 LOD 자원의 동치관계 발견을 통하여 간접적으로 이종의 온톨로지 클래스간의 동치관계를 발견할 수 있고, 이를 통해 온톨로지 정렬이 가능해 짐을 알 수 있다.

Ⅲ. LOD 자원과 온톨로지 인스턴스를 활용한 온톨로지 스키마 정렬 프레임워크

본 연구의 목표는 LOD 자원을 활용하여 복수 개의 온톨로지에 포함되어 있는 클래스들 간의 의미적 동치관계를 찾는 것이다. 이 과정은 3단계로 수행된다. 1단계에서는 온톨로지 클래스의 인스턴스와 의미적으로 유사한 LOD 자원을 식별한다. 식별된 LOD 자원들을 이용해 클래스와 의미적으로 동치관계("owl:sameAs")가 있는 자원을 찾는 것이 2단계이다.

복수개의 온톨로지에 존재하는 모든 클래스들을 대상으로 동치관계에 있는 LOD 자원을 찾은 후, 이 LOD 자원을 중심으로 이종의 온톨로지간의 동치관계를 갖는 클래스를 발견하여 클래스들 간의 스키마 정렬을 최종적으로 수행한다. 이러한 LOD 기반의 온톨로지 스키마를 정렬하는 과정을 도식화하면 그림 4와 같다.

본격적인 논의에 앞서, 본 논문에서 온톨로지 스키마 정렬의 대상인 목표 클래스를 정의한다.

정의 1 목표 클래스 (tc_{ij}) tc_{ij}는 온톨로지 정렬의 대상인 i^{th} 온톨로지에 포함된 j^{th}클래스로써, 온톨로지 정렬의 대상이다. tc_{ij}는 클래스 명을 나타내는 레이블(label)과 인스턴스를 원소로 갖는다. tc_{ij}는 다음과 같이 표현된다.

$$tc_{ij} = \{l_{ij}, i_{ij1}, i_{ij2}, \cdots, i_{ijk}\} \quad (1)$$

이때 인스턴스는 tc_{ij}와 "rdf:type"으로 직접 연결되어 있을 수도 있고 tc_{ij}의 하위 클래스(subclasses)를 통해 간접적으로 연결되어 있을 수도 있다. l_{ij}는 tc_{ij}와 "rdf:label"로 연결되어 있는 리터럴(literal) 값으로 tc_{ij}의 클래스 명을 나타내는 레이블이다. i_{ijk}는 tc_{ij}와 직 간접적으로 연결된 k번째 인스턴스의 레이블이다.

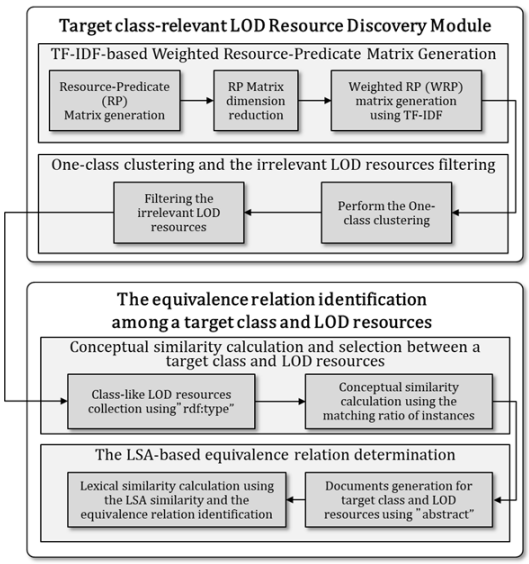

그림 4. 인스턴스와 LOD 자원 기반 온톨로지 정렬의 전체 프레임워크
Fig. 4. Overall framework of ontology schema alignment using LOD and ontology instances

3.1 목표 클래스와 연관된 LOD 자원 군집 발견

이 모듈은 온톨로지 정렬의 대상인 tc_{ij}의 인스턴스 ($i_{ijk}, \forall k$)와 매치되는 LOD 자원의 군집을 발견하기 위해 수행된다. 전술한 바와 같이, 온톨로지만을 활용하여 클래스들 간의 동치관계를 찾는 것이 어렵기 때문에, 외부 지식베이스로 도메인 교차지식이 풍부한 LOD를 활용한다. 그러나 LOD를 활용하기 위해서는 온톨로지와 LOD간의 관계 식별이 선행되어야 한다. 이를 위해, LOD가 특정 도메인에 관련된 인스턴스들 간의 관계를 중심으로 모델링되어 있다는 점에 착안해 목표 클래스의 인스턴스들과 연관된 LOD 자원의 군집을 발견한 후, 이를 이용해 목표 클래스와 LOD 자원들 간의 동치관계를 유추하고자 한다. 이때, 목표 클래스의 인스턴스들의 레이블을 활용하여 LOD 자원을 쿼리하는 것은 본 연구의 범위가 아니기 때문에, 인스턴스의 레이블과 매칭되는 LOD 자원이 SPARQL 쿼리를 통해 획득되었다고 가정한다. 획득된 Resource Description Framework(RDF) 트리플 집합은 다음과 같이 정의한다.

정의 2 쿼리된 LOD 자원의 RDF 트리플 집합 (R_l^{ij}) 목표 클래스의 모든 인스턴스 ($i_{ijk}, \forall k$)의 레이블을 활용하여 쿼리한 l번째 자원의 RDF 트리플 집합은 다음과 같이 표현된다.

$$R_l^{ij} = \{\cdots, t_{lh}^{ij}, \cdots\} \quad (2)$$

이때, t_{lh}^{ij}는 RDF 트리플 패턴으로 subject(s), predicate(p) 및 object(o)를 원소로 갖는 패턴 $< s,p,o >$로 구성된다. s,p는 URI를 값으로 가질 수 있으며 o는 URI 및 리터럴을 값으로 가질 수 있다.

3.1.1 TF-IDF 기반의 Weighted Resource-Predicate 행렬 생성

인스턴스 레이블을 활용한 SPARQL 쿼리 결과인 R_l^{ij}에는 tc_{ij}와 문법적 연관성(syntactic relation)은 있으나 의미적 연관성(semantic relation)이 없는 동음이의어(homonym)나 다의어(polysemy) 등이 포함되어 있을 수 있다. 이러한 LOD 자원들은 tc_{ij}와 개념적 연관성이 낮기 때문에 목표 클래스와 LOD 자원간의 정확한 동치관계를 찾을 때 방해될 수 있기 때문에 tc_{ij}와 개념적 연관성이 낮은 LOD 자원을 필터링해야 한다. 필터링을 위해 특정 데이터 셋에서 노이즈나 아웃

라이어를 제거하기 위해 활용되는 one-class 클러스터링 수행하고자 한다[5].

(1) RP 행렬 생성

One-class 클러스터링은 정형화된 행렬을 입력으로 활용한다. 이를 위해, 본 논문에서는 LOD 자원을 행(row)으로 술어의 어휘(term)를 열(column)로 갖는 Resource-Predicate(RP) 행렬을 사용한다. 이 RP 행렬의 원소는 특정 술어가 특정 LOD 자원에 등장하는 지 여부를 나타내는 이진변수이다. 이때, 트리플 패턴 중에서 술어만을 활용해 RP 행렬을 생성한 이유는 RDF 트리플 패턴 중에서 술어가 LOD 자원의 속성(attributes)이나 특징(characteristics)을 추상적으로 나타내는 스키마(schema)와 유사한 특징을 가진 요소이기 때문이다[6]. RP 행렬은 다음과 같이 정의된다.

정의 3 Resource-Predicate (RP) 행렬 RP 행렬의 행(row)은 LOD 자원을 그리고 열(column)은 술어로 구성된다. R_l^{ij}는 q번째 술어의 어휘(pt_q)를 포함하는 여부를 평가하는 이진변수인 rp_{lq}를 원소로 갖는다.

$$RP=\begin{bmatrix} rp_{11} & \cdots & rp_{1m} \\ \vdots & rp_{lq} & \vdots \\ rp_{n1} & \cdots & rp_{nm} \end{bmatrix} \tag{3}$$

이때, pt_q는 q번째 술어의 어휘이며 ($pt_q \in PT, PT=\bigcup_{lh}^{all} p_{lh}$), rp_{lq}의 값은 다음과 같이 도출된다 ($l=1,2,...,m, q=1,2,...,n$).

$$rp_{lq}=\begin{cases} 1 & \text{if } pt_q \; is \; \in R_l^{ij} \\ 0 & otherwise \end{cases} \tag{4}$$

(2) RP 행렬의 차원 축소

OWL 온톨로지와는 다르게 LOD 클라우드에 게시되는 자원들은 스키마의 제약을 받지 않기 때문에 (schema-free) 술어의사용에 제한이 없으며, 이는 RP 행렬의 차원을 크게 하는 원인이 된다. 또한 RP 행렬의 원소는 특정 술어가 특정 LOD 자원에 등장하는지 여부를 나타내는 이진변수이기 때문에 데이터 희소성(sparsity) 문제가 야기될 수 있다. 고차원이며 동시에 데이터 희소성 문제를 내포하고 있는 RP 행렬을 이용해 클러스터링을 수행하게 되면 결과의 정확도는 낮아지고 계산 복잡도는 커지게 된다[11]. 이러한 문제를 해결하기 위해, 본 논문에서는 정보량은 적으나 계산에 부담을 주는 술어, 즉 등장 빈도가 '0'인 값이 많이 갖는 술어의 열벡터를 RP 행렬에서 제거한 후 클러스터링을 수행하고자 한다. RP 행렬에서 불필요한 열벡터는 제거하는 과정은 다음과 같다.

1단계 RP 행렬의 모든 열벡터에 대해, 각 열벡터에 포함된 '0'의 빈도 즉 특정 술어를 포함하고 있지 않는 LOD 자원의 수를 산출한다.

2단계 이를 모든 술어에 대하여 산출한 후, 술어를 포함하지 않는 LOD 자원 개수의 평균을 도출한다(식 (5).

$$averg_{RP}=\frac{1}{n}\sum_{q=1}^{n}\sum_{l=1}^{m}|\{\{\text{'0'}\}\cap\{rp_{lq}\}\}| \tag{5}$$

3단계 모든 열벡터에 대해, 식 (5)의 평균값과 각 열벡터에 포함된 '0'의 빈도를 비교한다. 비교 결과 평균값에 비해 '0'의 빈도가 큰 열벡터는 정보량이 적은 술어로 판단하고 RP에서 제거한다. 이 과정을 통해 남은 q'번째 열 벡터 $RP_{q'}$는 다음과 같이 도출된다.

$$RP_{q'}=RP_q\left(\sum_{l=1}^{m}|\{\{\text{'0'}\}\cap\{rp_{lq}\}\}|>averg_{RP}\right) \tag{6}$$

식 (6)를 이용해 정보량은 적고 동시에 계산 부담을 유발하는 열벡터를 제거하면 다음과 같이 차원이 축소된 RP' 행렬을 도출할 수 있다.

$$RP'=\begin{bmatrix} rp_{11} \cdots rp_{1m'} \\ \vdots \;\; \ddots \;\; \vdots \\ rp_{n1} \cdots rp_{nm'} \end{bmatrix}, m' < m \tag{7}$$

(3) TF-IDF를 이용한 가중 RP 행렬 생성

축소된 RP' 행렬의 도출을 통해 RP 행렬의 고차원 문제와 데이터 희소성 문제를 해결하였다. 그러나 RP' 행렬의 원소는 이진변수이기 때문에 개별 술어의 중요도는 반영되어 있지 않다. 그러나 술어에 따라 출현 빈도는 높으나 LOD 자원의 특징을 설명·구분할 수 있는 정보가 부족할 수도 있고 반대로 출현 빈도는 낮으나 특정 LOD 자원만의 뚜렷한 특징을 드러내기 때문에 설명력이 클 수도 있다. RP' 행렬을 이용한 클러스터링 결과의 정확도를 높이기 위해서는

이러한 술어의 특징, 즉 술어의 정보량(informativeness)을 고려하는 것이 필요하다. 본 논문에서는 술어의 정보량을 도출하기 위해 TF-IDF 점수를 가중치로 활용한다. pt_q의 TF-IDF 점수(w_q)는 다음과 같이 산출된다.

$$w_q = \left\{ \log(f(pt_q)) \times \log\left(\frac{m'}{\sum_{l=1}^{m'} \{pt_q \in R_l^{ij}\}}\right) \right\} \quad (8)$$

이때, $f(pt_q)$는 모든 LOD 자원에 대한 pt_q의 출현 빈도이다. 또한, 편향(bias)을 피하기 위해 w_q는 Min-Max 스케일링을 수행하였다.

식 (8)을 활용하여 모든 q에 대해 w_q를 산출한 후, 이 값을 원소로 하는 벡터 $W=[w_1 w_2 \cdots w_{m'}]$를 생성한다. 이 벡터는 술어의 정보 중요도가 반영된 가중 RP' 행렬 (Weighted Resource-Predicate, WRP)을 도출하는 데 활용한다(식 (9)).

$$WRP = RP' \times W^T \quad (9)$$

3.1.2 tc_{ij}와 연관성이 낮은 LOD 자원 필터링

LOD 자원의 필터링은 WRP 행렬을 입력 데이터로 활용한 one-class clustering을 통해 달성한다. 이를 통해, R_l^{ij}중에서 문법적 연관성은 있으나 의미적인 연관성이 없는 동음이의어나 다의어와 같은 LOD 자원을 발견해 R_l^{ij}로부터 제거할 수 있다.

본 연구에서는 One-class information ball (OC-I) 알고리즘을 활용한 one-class 클러스터링 방법을 적용하였다[5]. 하나의 클러스터가 도출된 후, 해당 클러스터의 경계(boundary) 내에 속하지 않는 모든 LOD 자원을 의미적 연관성이 없는 자원으로 간주하고, 이를 R_l^{ij}에서 제거하여 $R_l^{ij'}$을 획득한다. 이를 알고리즘으로 표현하면 그림 5와 같다.

Input	$\boldsymbol{WRP}$, $\boldsymbol{R_{ij}}$
Output	$\boldsymbol{R'_{ij}}$
1	For all r in $\boldsymbol{R_{ij}}$:
2	$\boldsymbol{c} \xleftarrow{append} r$
3	$\boldsymbol{s} = [1, \cdots, n]$
4	For all k, l:
5	$k_{opt}, l_{opt} = \underset{k,l}{\text{argmin}}(cost(\boldsymbol{c}[k], \boldsymbol{s}[l]))$
6	$c_{opt} = \boldsymbol{c}[k_{opt}]$
7	$s_{opt} = \boldsymbol{s}[l_{opt}]$
8	$\boldsymbol{R'_{ij}} = filtering(\boldsymbol{R_{ij}}, c_{opt}, s_{opt})$
9	Return $\boldsymbol{R'_{ij}}$

그림 5. 연관성이 없는 LOD 자원 필터링 알고리즘
Fig. 5. Algorithm for filtering of irrelevant LOD resources

3.2 tc_{ij}와 LOD 자원간 의미적 동치관계 식별

이 모듈은 tc_{ij}와 의미적으로 유사한 LOD 자원의 집단인 $R^{ij'}(R_l^{ij'} \in R^{ij'}, \forall l)$을 이용해, tc_{ij}와 의미적 동치관계에 있는 LOD 자원을 식별하고 이들 간의 "owl:sameAs" 관계를 맺기 위해 수행한다. 그러나 LOD의 태생적인 한계인 개념 표현의 비일관성과 복잡성으로 인해 이들 간의 관계를 직접적으로 발견하는 것은 매우 어렵다. 이러한 어려움을 극복하기 위해 다음과 같은 방법을 제안한다.

3.2.1 tc_{ij}와 도메인 범위가 유사한 LOD 자원 발견

tc_{ij}의 개념을 포괄할 수 있는 LOD 자원을 찾아 이들 간의 개념적 유사성(conceptual similarity)을 평가하고 개념적으로 가장 유사한 LOD 자원을 선택한다.

(1) "rdf:type" 관계를 활용한 LOD 자원 획득

LOD 클라우드에서 $R^{ij'}$의 술어 "rdf:type" 관계를 활용하여 LOD 자원 중에 온톨로지의 클래스와 같은 역할을 수행할 수 있는 LOD 자원들을 찾는다. $R^{ij'}$와 "rdf:type"의 관계를 활용하여 발견한 의사 클래스(pseudo-class)에 해당하는 LOD 자원의 집합은 다음과 같이 정의한다.

정의 4 의사 클래스 기능을 하는 LOD 자원 집합 (CR) $R^{ij'}$와 "rdf:type" 관계를 통해 발견된 온톨로지 클래스와 같이 추상적인 개념을 나타내는 LOD자원들의 집합으로서 식 (10)과 같이 표현된다.

$$CR = \{cr_1, cr_2, \cdots, cr_r, \cdots\} \quad (10)$$

이때, cr_r은 r번째 의사 클래스 기능을 하는 LOD 자원의 URI이다.

(2) tc_{ij}와 LOD 자원간의 도메인 범위의 유사성 산출

CR의 원소는 $R^{ij'}$에 속한 개별 LOD 자원으로부터 "rdf:type" 관계를 활용하여 발견했기 때문에, CR의 모든 원소들이 $R^{ij'}$의 집합 전체를 대표한다고 보장할 수 없다. 즉, CR 의 일부는 tc_{ij}와 개념적 유사성이 없을 수도 있다. 대표성을 학보하기 위해 tc_{ij}와

CR의 모든 원소들 간의 개념적 유사도 평가를 통해 $R^{ij'}$ 집합을 대표하는 $cr_{r'}$을 선택한다. 이때, tc_{ij}와 cr_r 개념적 유사도($ConSIm_r^{ij}$) 는 식 (11)과 같이 산출된다.

$$ConSIm_r^{ij} = \frac{R^{ij'} \cap IR_r}{|R^{ij'}|} \quad (11)$$

이때, IR_r은 cr_r를 object, "rdf:type"을 술어로 갖는 LOD 자원들의 URI를 원소로 갖는 집합이다.

식 (11)을 이용해 모든 r에 대한 $DoSim_r$과 이들의 평균값 $DoSim_{averg}$을 산출한 후, $DoSim_{averg} \leq DoSim_r$인 cr_r을 원소로 갖는 LOD 자원의 집합 CR^{ω}($CR^{\omega} = \{cr_1^{\omega}, cr_2^{\omega}, \cdots, co_r^{\omega}, \cdots\}$)를 획득한다.

3.3 tc_{ij}와 어휘적으로 유사한 LOD 자원의 발견

CR^{ω}에는 tc_{ij}와 개념적으로 유사한 LOD 자원뿐만 아니라 이의 상위 개념(subsumption relation)이 존재할 수도 있다. 이러한 상위개념들을 동치관계에서 배제한 후, 어휘적 유사도(lexical similarity)를 활용한다.

(1) tc_{ij}와 LOD 자원의 주석 문서 생성

온톨로지의 클래스나 LOD 자원에는 이들의 정의나 의미를 설명하는 주석(annotation)이 붙어 있다. 주석의 대표적인 예는 "rdf:comment"와 "dbo:abstract"이다. 본 논문에서는 이러한 주석들로 구성된 문서를 활용해 tc_{ij}와 CR^{ω}의 어휘적 유사도 판별하고자 한다. 이 문서는 다음과 같이 정의된다.

정의 5 tc_{ij} 및 CR^{ω}의 문서집합(D) tc_{ij} 및 CR^{ω}과 "rdf:comment" 혹은 "dbo:abstract"의 관계를 통해 연결되어 있는 주석을 활용하여 생성한 문서들의 집합은 다음과 같이 표현된다.

$$D = \{d_{ij}, d_1, d_2, \cdots, d_r, \cdots\} \quad (12)$$

(2) tc_{ij} 및 LOD 자원의 어휘적 유사도 산출

문서의 유사도를 계산하기 위해, 일반 문서(plain text)를 효과적으로 처리할 수 있는 방법론인 Latent Semantic Analysis(LSA)를 활용한다[12]. LSA 알고리즘을 활용해 D에 포함된 주석의 차원을 축소한 후, tc_{ij}와 CR^{ω} 원소 쌍에 대한 코사인 유사도를 산출한다. 코사인 유사도 값을 사용해 임계값을 만족하는 cr_r^{ω}를 선택한 후, 해당 LOD 자원의 URL와 tc_{ij}간의 동치관계를 식별하고 "owl:sameAs" predicate을 활용하여 그들의 동치관계를 표현할 수 있는 RDF 트리플 패턴을 생성한다.

3.4 동치 LOD 자원 기반 온톨로지간 동치관계 생성

복수개의 온톨로지에 존재하는 모든 클래스들을 대상으로 동치관계에 있는 LOD 자원을 찾은 후, LOD 자원을 이용해 이종의 온톨로지간의 동치관계를 갖는 클래스를 발견하여 클래스들 간의 스키마 정렬을 수행한다.

Ⅳ. 실험 및 성능평가

본 논문에서 제안한 프레임워크의 우수성을 입증하기 위해 수행한 실험 환경과 실험 결과에 대해 상술한다.

4.1 실험 환경

군사용으로 개발된 온톨로지를 획득하는 것이 불가능하기 때문에 웹에 공개되어 있는 군 관련 자료로부터 도메인 지식을 수집해 미군, 한국군, NATO 군의 지상군 장교 계급체계와 관련된 3종의 온톨로지를 개발하였다. 3개국의 지상군 계급체계는 유사해 보이나 구조적인 차이와 실제로 사용하는 어휘 측면에서 차

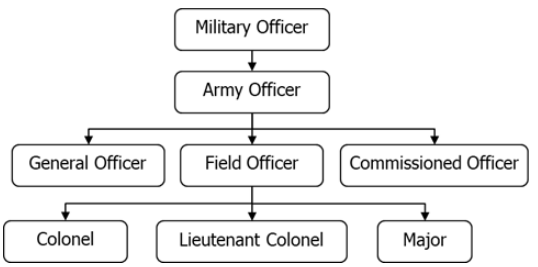

그림 6. O_1의 부분 온톨로지
Fig. 6. Partial ontology of O_1

표 1. 지상군 계급 온톨로지(O_1, O_2, O_3) 요약
Table 1. Summary of three kinds of army rank ontologies

	O_1	O_2	O_3
Nation	U.S.	Korea	NATO
Num. of classes	16	16	12
Num. of intances	237	284	215
depth	3	3	3

이가 있다. 그림 6은 미군의 지상군 계급체계 온톨로지를 도식화한 것이며, 세 가지 온톨로지에 대한 요약 정보는 표 1에 기술하였다.

4.2 실험 및 결과

4.2.1 온톨로지 클래스와 자원간의 동치관계 식별 평가

본 프레임워크를 통해 식별된 온톨로지 클래스와 LOD 자원간의 동치관계의 정확도를 평가하기 위해, 전문가들이 평가한 온톨로지 클래스와 LOD 자원간의 동치관계와의 비교를 수행하였다. 정확도 비교 결과를 요약하면 그림 7과 같다.

실험 결과, 온톨로지 클래스의 깊이가 깊을수록 동치관계의 정확도는 전문가와 유사한 것을 알 수 있다. 반면, 클래스의 깊이가 낮을수록, 즉 온톨로지의 상위 클래스일수록 동치관계의 정확도는 현저하게 낮아지며 최대 30%까지 낮아지는 것을 확인하였다. 이러한 결과는 온톨로지의 상위 클래스일수록 클래스가 표현하고자 하는 개념의 모호성이 커지기 때문에 개념들 간에 존재하는 명확한 동치관계 식별이 어려운 것으로 판단된다. 그러나 온톨로지의 클래스가 개념적으로 명확하거나 구체적일 경우 본 논문에서 제안한 방법이 도메인 전문가와 거의 유사한 정확도로 동치관계를 발견할 수 있다는 점에서 온톨로지 정렬을 성공적으로 수행하였다고 판단된다.

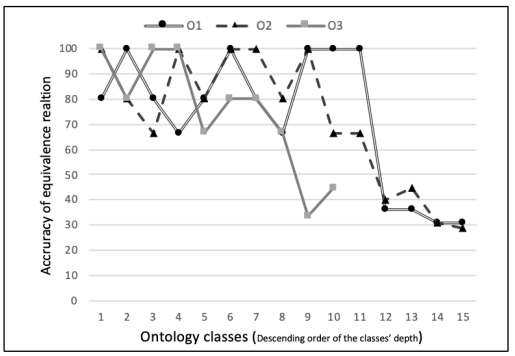

그림 7. 온톨로지 클래스와 LOD 자원간 동치관계 정확도
Fig. 7. Accuracy of equivalence relation between ontology classes and LOD resources

4.2.2 이종 온톨로지 클래스간 동치관계 식별 평가

식별된 이종 온톨로지 클래스간의 동치관계의 정확도를 평가하기 위해, 전문가들이 평가한 이종 온톨로지 클래스간의 동치관계와의 비교를 수행하였다. 실험을 위해, 우리가 찾아야 하는 동치관계를 갖는 개념을 표 2에 나타냈다.

표 2. 세가지 유형의 온톨로지 클래스 비교
Table 2. Comparison among the ranks in three ontologies

O_1	O_2	O_3
Captain	대위	OF-2
Colonel	대령	OF-5
Lieutenant General	중장	OF-8

전문가들이 식별한 이종 온톨로지 간의 동치관계와 본 프레임워크가 식별한 결과를 요약하면 표 3와 같다. 표 3에 요약된 바와 같이, 본 프레임워크는 전문가가 식별한 동치관계 총 35개 중 30개를 정확히 식별하였고, 5개는 찾지 못했다. 이러한 빈 연결은 온톨로지 클래스의 주석과 LOD 자원의 주석이 상이한 수준에서 작성되었기 때문에 두 문서의 어휘적 유사도가 낮아 동치관계를 찾지 못한 것으로 판단된다.

표 3. 온톨로지 클래스간의 동치 관계 비교
Table 3. Comparison of equivalence relation between ontologies

	O_1-O_2	O_2-O_3	O_3-O_1
Num. of equivalence identified by domain experts	15	10	10
Correct relation	13	9	8
Empty relation	2	1	2

V. 관련 연구

온톨로지 정렬 연구는 크게 스키마 기반 온톨로지 정렬과 인스턴스 기반 온톨로지 정렬로 분류된다. 스키마 기반 온톨로지 정렬에는 어휘 기반, 구조 기반 온톨로지 정렬이 있다. 온톨로지 정렬 연구 초기에는 COMA와 같은 스키마 기반 온톨로지 정렬 연구가 주를 이루었다[7]. 그러나 스키마 기반 온톨로지 정렬 방법은 스키마 수준의 정보에만 의존해 정렬하기 때문에 온톨로지 구조가 상이하거나 클래스와 속성의 레이블이 다른 경우 정렬을 수행하는 것이 제한적이다. 이러한 한계를 극복하기 위해서 인스턴스 기반 온톨로지 연구가 등장했다. IUT는 서로 다른 두 온톨로지가 공유하고 있는 인스턴스의 포함관계를 이용한 온톨로지 정렬 방법이다[8]. IUT와 같은 인스턴스 기반 온톨로지 정렬은 인스턴스들 간의 동치관계가 사전에 연결되어 있어야 한다는 전제조건을 만족해야한다. 그러나 인스턴스들의 레이블이 온톨로지별로 매우 상

이하기 때문에 이 전제조건을 만족시키는 것이 매우 어렵다. 요약하면, 스키마 기반 온톨로지 정렬과 인스턴스 기반 온톨로지 정렬과 같이 내부에 선언된 요소(클래스, 속성, 인스턴스)를 이용한 정렬 방법에는 한계가 있다. 이런 한계를 극복하고자 온톨로지 내부에 선언된 요소가 아닌 외부 지식을 활용한 온톨로지 정렬 연구가 수행되었다[4]. 외부 지식을 활용한 온톨로지 정렬은 정렬 대상 온톨로지를 각각 외부 지식에 연결한 후, 이를 기반으로 두 온톨로지의 관계를 발견한다. WikiMatch는 Wikipedia를 외부 지식으로 사용해 서로 다른 언어로 만들어진 온톨로지들 간의 정렬을 수행했고[9], AML은 WordNet을 외부 지식으로 사용해 온톨로지 정렬을 수행하였다[10]. 최근에는 랜덤 포레스트(Random Forest) 기법을 온톨로지 정렬에 적용시킨 연구도 수행되었다[13].

Ⅵ. 결론 및 추후 연구

본 연구의 목표는 LOD 자원을 활용하여 복수 개의 온톨로지에 포함되어 있는 클래스들간의 의미적 동치관계를 찾는 것이다. 이를 위해 다음과 같은 세 가지 단계로 구상된 프레임워크를 제안하였다. 1단계에서는 온톨로지 클래스의 인스턴스와 의미적으로 유사한 LOD 자원을 식별하며, 2단계에서는 식별된 LOD 자원들을 이용해 클래스와 의미적으로 동치관계에 ("owl:sameAs") 있는 자원을 찾는다. 마지막 단계는 복수개의 온톨로지에 존재하는 모든 클래스들을 대상으로 동치관계에 있는 LOD 자원을 찾고 이 LOD 자원을 중심으로 이종의 온톨로지 간의 동치관계를 갖는 클래스를 발견하여 클래스들 간의 스키마 정렬을 수행한다.

본 연구가 온톨로지 스키마 정렬 연구에 기여한 바는 다음과 같다. 첫째, 온톨로지 정렬에 필요한 교차지식을 LOD로부터 획득함으로써 온톨로지 정렬의 대상을 확대했다. 둘째, 온톨로지간의 관계를 직접적으로 정렬하는 것이 아니라 LOD 클라우드를 가교로 활용하여 간접적으로 정렬하기 때문에 새로운 온톨로지가 등장할 때 필요한 추가적인 관계 해석의 횟수의 부담을 크게 줄이는 게 기여하였다. 마지막으로, LOD와의 새로운 온톨로지가 추가되어도 기존에 정렬된 온톨로지와 유연하게 통합할 수 있는 프레임워크를 제안한 것도 매우 의미있는 일이다.

본 연구 결과를 발전시키기 위해 다음과 같은 연구를 추가적으로 수행할 계획이다. 첫째, LOD의 품질에 따라 온톨로지 정렬의 결과가 크게 영향받기 때문에, 이러한 문제를 피하기 위한 LOD 정제 및 LOD 내에서의 동치관계 발견 과정을 연구할 것이다. 둘째, 클러스터링 과정에서 활용된 비용 함수(cost function) 및 전역 탐색(global search) 방법을 RDF의 데이터 특징 및 속성에 맞게 개선하여 클러스터링의 성능을 향상시킬 것이다. 셋째, 어휘적 유사도를 산출할 때 주석만이 아니라 웹 도큐먼트를 추가적으로 활용하고, word embedding과 같은 방법을 적용함으로써 온톨로지 정렬의 적용 대상을 확장 및 정확도를 향상시킬 것이다.

References

[1] M. H. Khan, S. Jan, I. Khan, and I. A. Shah, "Evaluation of linguistic similarity measurement techniques for ontology alignment," in *IEEE 2015 Int. Conf. Emerging Technol. (ICET)*, pp. 1-6, Peshawar, Pakistan, Dec. 2015.

[2] M. Li, D. Wang, X. Du, and S. Wang, "Ontology construction for semantic web: A role-based collaborative development method," in *Asia-Pacific Web Conf. Springer*, pp. 609-619, Shanghai, China, Mar. 2005.

[3] J. Han, H. Jung, and D.-K. Baik, "Discrete cuckoo search based ontology alignment algorithm," in *Proc. KIPS Conf.*, pp. 664-667, 2014.

[4] I. G. Husein, S. Akbar, and F. N. Azizah, "Review of ontology matching with background knowledge," in *ICoDSE*, pp. 1-6, Denpasar, Indonesia, Oct. 2016.

[5] G. Gupta and J. Ghosh, "Robust one-class clustering using hybrid global and local search," in *Proc. 22nd Int. Conf. Mach. Learning, ACM*, pp. 273-280, Bonn, Germany, Aug. 2005.

[6] J. Kim, J. Kong, D. Park, and M. Sohn, "Predicate clustering-based entity-centered graph pattern recognition for query extension on the LOD," in *Int. Conf. Innovative Mob. and Internet Serv. in Ubiquitous Computing*, Springer, pp. 159-170, Matsue, Japan, Jul. 2018.

[7] H.-H. Do and E. Rahm, "COMA: a system for flexible combination of schema matching approaches," in *Proc. 28th Int. Conf. Very Large Data Bases*, pp. 610-621, Hong Kong

SAR, China, Aug. 2002.

[8] N. Zong, S. Nam, J-H. Eom, J. Ahn, H. Joe, and H-G. Kim, "Aligning ontologies with subsumption and equivalence relations in linked data," *Knowledge-Based Syst.*, vol. 76, pp. 30-41, Nov. 2015.

[9] S. Hertling and H. Paulheim, "WikiMatch: Using wikipedia for ontology matching," *Ontology Matching*, vol. 946, 2012.

[10] D. Faria, C. Pesquita, E. Santos, M. Palmonari, I. F. Cruz, and F. M. Couto, "The agreementmakerlight ontology matching system," in *OTM Confederated Int. Conf. "On the Move to Meaningful Internet Systems,"* Springer, pp. 527-541, Graz, Austria, Sep. 2013.

[11] J. Li and H. Zha, "Two-way poisson mixture models for simultaneous document classification and word clustering," *Computational Statistics & Data Analysis*, vol. 50, no. 1, pp. 163-180, 2006.

[12] T. Hofmann, "Probabilistic latent semantic analysis," in *Proc. Fifteenth Conf. Uncertainty in Artificial Intelligence*, pp. 289-296, Stockholm, Sweden, Jul. 1999.

[13] I. Nkisi-Orji, N. Wiratunga, S. Massie, K-Y. Hui, and R. Heaven, "Ontology alignment based on word embedding and random forest classification," *Joint Eur. Conf. Mach. Learning and Knowledge Discovery in Databases*, Springer, pp. 557-572, Dublin, Ireland, Sep. 2018.

김 민 환 (Minhwan Kim)

2019년 2월 : 성균관대학교 시스템경영공학과 졸업

2019년 3월~현재 : 성균관대학교 산업공학과 석박통합과정

<관심분야> 시맨틱웹, 온톨로지, IOT, 기계학습

김 종 모 (Jongmo Kim)

2014년 2월 : 성균관대학교 시스템경영공학과 졸업

2014년 9월~현재 : 성균관대학교 산업공학과 석박통합과정

<관심분야> 시맨틱웹, 온톨로지, LOD, Web-of-Things

손 미 애 (Mye Sohn)

1985년 2월 : 성균관대학교 산업공학과 졸업

1988년 2월 : 한국과학기술원 산업공학과 석사

2002년 8월 : 한국과학기술원 경영공학과 졸업

1998년 3월~2004년 2월 : 한국국방연구원 선임연구원

2004년 3월~현재: 성균관대학교 산업공학과 교수

<관심분야> 시맨틱웹, 온톨로지, IOT, 기계학습, SNS 추천

[ORCID:0000-0002-1951-3493]

박 규 동 (Gyudong Park)

1994년 2월 : 홍익대학교 컴퓨터공학과 졸업

1996년 2월 : 홍익대학교 컴퓨터공학과 석사

2014년 2월 : 홍익대학교 컴퓨터공학과 박사

1996년 3월~현재 : 국방과학연구소 연구원

<관심분야> C4I, 가상화, 클라우드, 네트워크