논문 2004-41CI-4-3

지역 및 전역 의미집합을 이용한 온톨로지 병합 및 정렬 알고리즘

(An Algorithm for Ontology Merging and Alignment using Local and Global Semantic Set)

김 재 홍*, 이 상 조**

(Jaehong Kim and Sang Jo Lee)

요 약

기존 웹의 단점을 보완하기 위해 시맨틱 웹 개념이 제안되었고, 시맨틱 웹에서 중요한 역할을 하는 온톨로지는 분산 독립된 형태로 개발되는 특성으로 인해 동일한 도메인에 대해 중복 저작될 수 있는 문제점을 가지고 있다. 따라서 온톨로지의 공유와 재사용이 중요한 문제로 부각되고 있으며, 온톨로지 병합 및 정렬이 한 해결책이 될 수 있다. 현재까지 제안된 반자동 방식의 온톨로지 병합 및 정렬 알고리즘은 온톨로지 전체에서 가지는 의미정보가 아닌 지역적 구문정보만을 이용하고, 반자동 작업 특징으로 인해 온톨로지 엔지니어에게는 지루한 작업이 되어 결과의 품질이 낮아질 수 있다는 단점이 있다. 본 논문에서는 지역 및 전역 의미집합 개념을 이용하여 이러한 단점을 개선한 온톨로지 병합 및 정렬 알고리즘을 제안하였다. 제안된 알고리즘을 구현하여 OWL 언어로 작성된 온톨로지에 대해 실험한 결과 91%의 정확도를 보였다. 본 논문에서 제안하는 알고리즘을 이용하여 온톨로지 병합 및 정렬 작업을 수행하면 온톨로지 공유 및 재활용률을 높이고, 기존 온톨로지를 활용한 새로운 온톨로지의 저작시간도 단축시킬 수 있을 것으로 기대된다. 또한, 온톨로지 매핑등 온톨로지들 간의 의미 정보 교환이 필요한 다른 어떤 분야에도 쉽게 적용이 가능할 것으로 기대된다.

Abstract

Ontologies play an important role in the Semantic Web by providing well-defined meaning to ontology consumers. But as the ontologies are authored in a bottom-up distributed manner, a large number of overlapping ontologies are created and used for the similar domains. Ontology sharing and reuse have become a distinguished topic, and ontology merging and alignment are the solutions for the problem. Ontology merging and alignment algorithms previously proposed detect conflicts between concepts by making use of only local syntactic information of concept names. And they depend only on a semi-automatic approach, which makes ontology engineers tedious. Consequently, the quality of merging and alignment tends to be unsatisfying. To remedy the defects of the previous algorithms, we propose a new algorithm for ontology merging and alignment which uses local and global semantic set of a concept. We evaluated our algorithm with several pairs of ontologies written in OWL, and achieved around 91% of precision in merging and alignment. We expect that, with the widespread use of web ontology, the need for ontology sharing and reuse will become higher, and our proposed algorithm can significantly reduce the time required for ontology development. And also, our algorithm can easily be applied to various fields such as ontology mapping where semantic information exchange is a requirement.

Keywords: Ontology, Merging, Alignment, Local Semantic Set, Global Semantic Set

I. 서 론

1. 시맨틱 웹과 온톨로지

팀 버너스-리가 제안한 웹은 지난 10여 년간 웹 공간상에 존재하는 사이트의 수와 이를 이용하는 사용자의

* 정회원, 한국전자통신연구원 지능형로봇연구단
(Intelligent Robot Research Division, ETRI)
** 정회원, 경북대학교 컴퓨터공학과
(Department of Computer Engineering, Kyungpook National University)
접수일자: 2004년2월11일, 수정완료일: 2004년6월30일

수 측면에서 급속히 성장하였다. 그러나 정보의 표현에 중점을 둔 기존의 웹은 사람이 이해하는 데는 별 문제가 없으나 표현의 비정형성으로 인해 컴퓨터가 문서의 의미를 이해하고 처리하는 데에는 한계가 있다. 이에 대한 해결책으로 팀 버너스-리는 웹상의 정보에 "잘 정의된 의미"를 부여함으로써 사람뿐만 아니라 컴퓨터도 쉽게 문서의 의미를 해석할 수 있도록 하여 컴퓨터를 통한 정보의 검색 및 해석, 통합 등의 업무를 자동화하기 위한 목적으로 시맨틱 웹을 제안하였다. 여기에서 "잘 정의된 의미"를 다루고자 하는 것이 바로 시맨틱 웹상의 온톨로지 언어의 역할이다[14]. 시맨틱 웹에서의 온톨로지의 정의는 "개념의 공유화를 위해 명시적으로 형식화한 명세(A formal explicit specification of a shar-ed conceptualization)"이며[13], 풀어쓰면 광범위한 구성원 간에 합의되어 애매모호하지 않게 통용되며 컴퓨터 프로그램에 의해 처리하기 용이한 지식체계를 온톨로지라고 할 수 있다[14].

온톨로지를 구체적으로 표현하기 위해서는 스키마와 구문구조 등을 정의한 언어가 필요하며, 이것이 온톨로지 언어이다. 시맨틱 웹상의 온톨로지 언어로 DAML (DARPA Agent Markup Language)[3], OIL(Ontology Inference Layer)[5] 및 이들의 결합을 통해 만들어진 DAML+OIL[4] 등이 있으며, 이것은 현재 OWL[11]로 계승 발전되고 있다. 본 논문에서는 제안한 알고리즘의 실용성을 증명하기 위해 OWL로 작성된 온톨로지를 대상으로 실험을 하였으나, 지역 및 전역 의미집합을 이용한 온톨로지 병합 및 정렬 개념의 적용은 OWL 뿐 아니라, 이전의 시맨틱 웹 언어 또는 시맨틱 웹 등장 이전에 존재하던 언어로 작성된 온톨로지에도 적용가능하다.

2. 온톨로지 병합 및 정렬의 필요성

I장 1절에서 설명한 온톨로지는 시맨틱 웹상에서 다음과 같은 특징을 가진다[14].

- 온톨로지의 작성 주체로 국가나 대규모 단체와 같은 집단뿐만 아니라 개인이나 그룹 같은 소규모의 다수 집단을 지향한다.
- 온톨로지의 확장과 수정을 전제로 작성된다. 시맨틱 웹에서는 이미 정의된 다른 온톨로지를 검색하여 이를 수정하고 확장하여 개별적 목적에 맞게 진화시키는 기능을 추구한다. 또한 하나 이상의 온톨로지를 결합하여 새로운 온톨로지를 만들어 낼 수도 있다.
- 온톨로지의 작성이 자발적 무의식적으로 이루어진다. 사용자는 현재의 HTML 작성과 같이 온톨로지에 대한 이해없이 또는 최소한의 이해만으로 일반 문서 작업과 유사한 작업을 하면서 자연스럽게 이미 만들어진 온톨로지를 이용하고 수정하고 확장하는 작업을 하게 될 것이다.

이상과 같은 온톨로지의 특징은 온톨로지 병합 및 정렬 기능의 필요성을 내포하고 있다. 온톨로지가 다수에 의해 자발적으로 작성된다는 것은 작성되는 온톨로지 상에 다양한 형태의 개념 중복이 발생할 수 있음을 의미하고, 확장과 수정을 전제로 작성된다는 것은 원하는 도메인과 연관된 기존 온톨로지를 검색하여 이를 하나의 온톨로지로 병합하거나 정렬하여 재활용할 수 있다는 것을 의미한다. 또, 온톨로지 작성자가 최소한의 이해만으로 일반작업과 유사하게 이미 만들어진 온톨로지를 이용하고 수정, 확장하기 위해서는 개념 중복과 개념간의 관계들을 쉽게 검출하고 처리하는 도구가 제공되어야 한다는 것을 의미한다. 이와 같이 기존의 온톨로지들을 공유하고 재활용하는 과정에 있어서 온톨로지의 병합 및 정렬은 반드시 필요한 기능이다.

3. 본 논문에서의 온톨로지 병합 및 정렬

가. 표기

이하의 설명에서 사용하는 온톨로지 및 클래스의 표기를 위해 다음과 같은 기호를 사용한다.

O_{id}: ID(rdf:ID, Namespace 제외)가 id인 온톨로지

C_{id}: ID가 id인 클래스

$O_{id}(C_{id})$: 온톨로지 O_{id}의 클래스 C_{id}

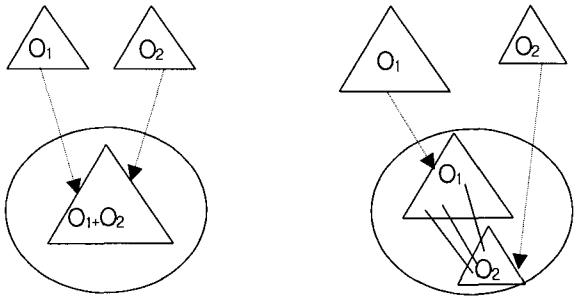
$O_{id1}+O_{id2}$: O_{id1}과 O_{id2}가 병합된 온톨로지

$C_{id1}+C_{id2}$: C_{id1}과 C_{id2}가 병합된 클래스

여기서, id는 의미를 가지는 유한개의 단위 문자열(이하에서 wi-i는 양의 정수-로 표기, 주로 사전에 등록된 단어)들로 구성된다.

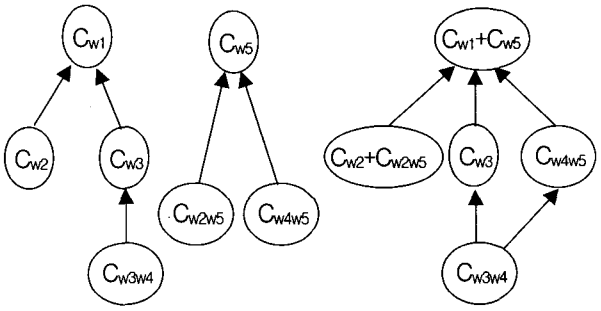
나. 본 논문에서의 온톨로지 병합 및 정렬

온톨로지의 병합(Merging)과 정렬(Alignment)의 차이를 스탠포드 대학의 Noy는 다음과 같이 설명하고 있다[12]. 온톨로지 병합(<그림 1-a>)은 두개의 원본 온톨로지(O_1, O_2)를 합하여 하나의 결과 온톨로지(O_1+O_2)를 생성하는 것으로 유사하거나 중복되는 도메인을 정의한 온톨로지들을 통합(실제로는 온톨로지 내의 중복되는 개념을 통합)한 하나의 온톨로지를 생성할 때 주로 사용될 수 있다. 온톨로지 정렬(<그림 1-b>)은 독립적인



(a) (b)

그림 1. 온톨로지 병합(a)과 정렬(b)
Fig 1. Ontology merging(a) and alignment(b).



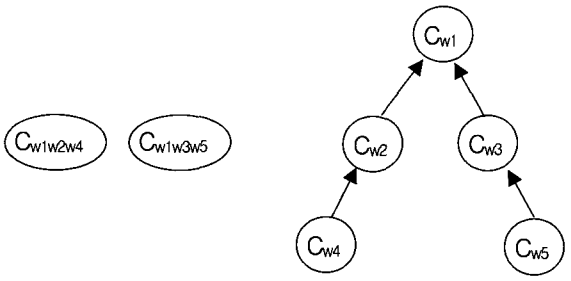
(a) 온톨로지 O_1 및 O_2 (b) 병합/정렬된 온톨로지 O_1+O_2

그림 2. 온톨로지 병합/정렬의 예
Fig 2. Example of ontology merging and alignment.

원본 온톨로지들 간의 관계(온톨로지 내의 개념들 간의 관계)를 만들어 주는 개념으로 상보적인(complemen-tary) 도메인을 다룬 온톨로지들 간의 관계 정립을 위해 주로 사용될 수 있다. 온톨로지 내 개념들 간의 관계 정보로는 어떠한 것들도 가능하나 본 논문에서는 모든 온톨로지 언어들에서 공통적으로 존재하는 관계인 상하위 관계(OWL의 rdfs:subClassOf 및 rdfs:subProperty Of)를 대상으로 하였다.

<그림 2-a>와 같이 원본 온톨로지 O_1, O_2가 있고, 클래스 C_{w1}과 C_{w5}, C_{w2}와 C_{w2w5}가 동일한 개념이고, C_{w3w4}는 C_{w4w5}의 하위 개념이라는 것이 각각 온톨로지 병합 및 정렬을 위한 알고리즘의 처리 결과로 생성되었을 때 병합 및 정렬을 동시에 처리한 결과로 <그림 2-b>를 얻을 수 있다.

본 논문에서의 온톨로지 병합 및 정렬 알고리즘은 위와 같은 경우에서 서로 다른 두 온톨로지(O_1, O_2) 간의 개념 쌍이 동일한지(<그림 2>에서 (C_{w1}, C_{w5}) 및 (C_{w2}, C_{w2w5})), 또는 상하위 관계(<그림 2>에서 (C_{w3w4}, C_{w4w5}))가 존재하는 지를 찾아내어 찾아낸 동일 개념 쌍들을 병합(C_{w1}+C_{w5}, C_{w2}+C_{w2w5})하고, 상하위 관계가 존재하는 쌍에 대해서는 온톨로지에 이 관계(C_{w3w4} rdfs:subClassOf C_{w4w5})를 추가해 주는 것이다.



(a) (b)

그림 3. 수평적(a) vs 수직적(b)으로 설계된 온톨로지
Fig 3. Ontology designed (a)horizontally (b)vertically.

Ⅱ. 관련 연구

현재까지의 온톨로지 병합 또는 정렬에 관한 연구 중 대표적인 것이 PROMPT[10]와 Chimaera[8]로서 기존에 존재하는 유사한 온톨로지를 병합하여 중복되는 요소들을 제거하여 새로운 온톨로지를 생성해내는 개념이다. PROMPT 및 Chimaera는 프레임(클래스, 슬롯 및 인스턴스)에 기초한 OKBC 모델[1]에 기반하고 있으며 본 논문에서 제안하는 알고리즘과 비교될 수 있는 두 프레임 간의 유사성 비교는 프레임 이름의 구문 정보(문자열 자체의 접미사, 접두사, 문자열 부분 매칭, 약어, 이름의 확장된 형태 등)를 이용하고 있다. 프레임 이름의 구문 정보를 이용하여 유사한 프레임 쌍을 찾은 후 사용자로 하여금 그 쌍에 대해 병합 작업을 수행할 것인지 아닌지에 대해 사용자로 하여금 선택하도록 하는 반자동방식이다.

그러나 프레임 이름의 구문 정보만 사용하고 의미 정보는 사용하지 않기 때문에, 이름이 구문적으로는 전혀 다르지만 의미는 유사한 프레임 쌍을 찾아낼 수는 없다. 또한, 의미 정보를 사용한다고 하더라도 상위 계층의 구문 또는 의미 정보가 하위 계층 프레임의 이름에 반영되어 있지 않은 경우 수평적으로 설계된 온톨로지와 수직적으로 설계된 온톨로지의 유사성을 발견하기가 쉽지 않다. 즉, <그림 3>과 같이 실제로는 유사한 온톨로지이지만, 설계자의 특성에 따라 하나는 수평적으로 설계되고, 다른 하나는 수직적으로 설계된 온톨로지가 있을 경우 C_{w1w2w4}와 C_{w4}가 동일하다 것과 C_{w1w3w5}가 C_{w5}와 동일하다는 것을 발견할 수 없다.

상기 두가지 도구에서 <그림 2>에서 C_{w1}과 C_{w5}의 병합에 따라 C_{w3}가 C_{w1}+C_{w5}의 하위 클래스가 되는 것과 같이 온톨로지 병합 결과에 따라 부수적으로 발생하는 정렬 기능은 지원되지만, C_{w3w4}가 C_{w4w5}의 하위 클래

스가 되는 것과 같은 병합에 부수적으로 따르는 결과가 아닌 것은 찾아내지 않는다. 실제로 상기와 같은 병합 도구들에서 병합할 대상이라고 추천하는 프레임 쌍의 상당 부분이 병합이 아닌 정렬이 되어야 하는 쌍으로 도구 사용자에게 혼란을 주고 병합 시간을 증가시키는 단점이 있다. 이런 점을 고려하여 본 논문에서는 단순히 유사한 개념의 쌍을 다수개 추천하는 것이 아니라 실제 병합이 되어야 하는 것인지, 정렬이 되어야 하는 것인지를 명확히 구분하는 방식을 사용하였다.

본 논문의 알고리즘은 주로 위에서 나열한 단점들을 개선한 것이다.

Ⅲ. 지역 및 전역 의미집합을 이용한 온톨로지의 병합과 정렬

1. 지역 및 전역 의미집합의 정의

본 논문의 주요 개념인 지역 및 전역 의미집합을 OWL 모델의 용어를 이용하여 설명한다. OWL의 주요 구성요소는 클래스(Class), 속성(Property), 인스턴스(Individual)로 이것은 각각 OKBC 모델의 클래스(Class), 슬롯(Slot), 인스턴스(Instance)에 해당한다. 온톨로지 설계 및 이용에서 가장 중심이 되는 부분이 클래스이므로 본 장에서는 클래스 위주로 설명한다.

각 정의의 예를 쉽게 이해할 수 있도록 다음과 같은 두 개의 예제 온톨로지를 사용한다.

```
<owl:Class rdf:ID="Faculty"/>
<owl:Class rdf:Id="Administrative">
  <rdfs:subClassOf rdf:resource="#Faculty"/>
</owl:Class>
<owl:Class rdf:ID="Research">
  <rdfs:subClassOf rdf:resource="#Faculty"/>
</owl:Class>
<owl:Class rdf:ID="ResearchAssistant">
  <rdfs:subClassOf rdf:resource="#Research"/>
</owl:Class>
```

(a) 온톨로지 O_1

```
<owl:Class rdf:ID="Staff"/>
<owl:Class rdf:Id="AdministrativeStaff">
  <rdfs:subClassOf rdf:resource="#Staff"/>
</owl:Class>
<owl:Class rdf:ID="AssistantStaff">
  <rdfs:subClassOf rdf:resource="#Staff"/>
</owl:Class>
```

(b) 온톨로지 O_2

그림 4. 예제 온톨로지
Fig 4. Example ontology.

(정의 1) 클래스의 지역 이름 집합(Local ID Set, L_{ids}): 클래스의 ID를 구성하는 단어(wi)들의 집합.

(L_{ids}의 예) $L_{ids}(O_1(C_{Faculty}))$ = {Faculty}, $L_{ids}(O_1(C_{ResearchAssistant}))$ = {Research, Assistant}

(정의 2) 단어의 의미 그룹 번호 집합(Semantic Gro-up ID Set, S_{gid}): 단어가 동일한 의미를 가지는 의미 그룹단위로 분류되는 상황에서 그룹을 지칭할 수 있는 식별번호들의 집합.

본 논문에서는 WordNet[2]에 정의된 Synset의 offset 값을 의미 그룹 번호로 사용한다. 만약, 단어가 Word-Net에 등록된 것이 아닐 경우에는 그 자체를 하나의 의미 그룹으로 분류하며, 이때 단어의 해쉬값을 의미 그룹 번호로 할당하여 사용한다.

(S_{gid}의 예) S_{gid}(Faculty) = {6859293, 4851828}, S_{gid}(Staff) = {3744397, 6859293, 6056474, 12729260, 5730457, 6964447, 858527}

(정의 3) 클래스의 지역 의미 집합(Local Semantic Set, LSS): 클래스의 지역 이름 집합(정의 1)을 구성하는 엘리먼트의 의미 그룹 번호 집합(정의 2)의 집합.

(LSS의 예) $LSS(O_1(C_{Faculty}))$ = $\{S_{gid}(Faculty)\}$ = {{6859293, 4851828}}, $LSS(O_2(C_{Staff}))$ = {{3744397, 6859293, 6056474, 12729260, 5730457,6964447, 858527}}

(정의 4) 클래스의 전역 이름 집합(Global ID Set, G_{ids}): 클래스 및 이 클래스의 모든 상위 클래스들(super-classes)의 지역 이름 집합의 합집합.

(G_{ids}의 예) $G_{ids}(O_1(C_{Administrative}))$ = $L_{ids}(O_1(C_{Administra-tive}))$ ∪ $L_{ids}(O_1(C_{Faculty}))$ = {Administrative, Faculty}, $G_{ids}(O_2(C_{AdministrativeStaff}))$ = $L_{ids}(O_2(C_{AdministrativeStaff}))$ ∪ $L_{ids}(O_2(C_{Staff}))$ ={Administrative, Staff} ∪ {Staff} = {Administrative, Staff}

(정의 5) 클래스의 전역 의미 집합(Global Semantic Set, GSS): 클래스의 전역 이름 집합(정의 4)을 구성하는 엘리먼트의 의미 그룹 번호 집합(정의 2)의 집합.

(GSS의 예) $GSS(O_1(C_{Faculty}))$ = $\{S_{gid}(Faculty)\}$ = {{6859293, 4851828}}, $GSS(O_1(C_{Administrative}))$ = $\{S_{gid}(Administrative), S_{gid}(Faculty)\}$ = {{2761145}, {6859293, 4851828}}, $GSS(O_1(C_{Research}))$ = {{694771, 4968857, 475490, 516559}, {6859293, 4851828}}, $GSS(O_1(C_{ResearchAssistant}))$ = $\{S_{gid}(Research), S_{gid}(Assistant), S_{gid}(Faculty)\}$ = {{694771, 4968857, 475490, 516559}, {8062152, 763988}, {6859293, 4851828}}, $GSS(O_2(C_{Staff}))$ = {{3744397, 6859293, 6056474, 12729260, 5730457, 6964447, 858527}}, $GSS(O_2(C_{AdministrativeStaff}))$ = {{2761145}, {3744397, 6859293, 6056474, 12729260, 5730457, 6964447, 858527}}, $GSS(O_2(C_{AssistantStaff}))$ = {{8062152, 763988}, {3744397, 6859293, 6056474, 12729260, 5730457, 6964447, 858527}}

본 논문에서 최종적으로 사용하게 되는 정보는 정의 3(지역 의미 집합)과 정의 5(전역 의미 집합)이다.

2. 온톨로지 병합 및 정렬 알고리즘

두 클래스의 의미집합 간의 관계를 <그림 5>와 같은 4가지 경우로 나누어 생각해 볼 수 있다. 그림에서 점선으로 표시된 것이 C_1의 의미집합이고, 실선으로 표시된 것이 C_2의 의미집합이다.

<그림 5-a>는 두 클래스의 의미집합이 동일한 것을 표현한 것이다. 본 논문에서는 다음과 같은 경우 두 개의 클래스를 병합 대상이 되는 동일한 클래스로 간주한다.

(정의 6) 두 클래스의 의미적 동치: $LSS(O_X(C_A))$ = $LSS(O_Y(C_B))$ 또는 $GSS(O_X(C_A))$ = $GSS(O_Y(C_B))$인 경우 두 클래스 $O_X(C_A)$와 $O_Y(C_B)$는 의미적으로 동일하다. 이때 지역 또는 전역 의미 집합의 엘리먼트인 집합 S_{gid} 간의 관계는 S_{gid}들 간에 공통된 엘리먼트가 존재하면 동일(=)한 것으로 간주한다.

예를 들면, "Faculty"의 지역 의미 집합 {{**6859293**, 4851828}}와 "Staff"의 지역 의미 집합 {{3744397, **6859293**, 6056474, 12729260, 5730457, 6964447, 858527}}는 동일한 의미 집합이다. 이런 조건을 부가함으로써, "Faculty"라는 개념과 "Staff"라는 개념은 비록 의미 집합 전체는 다르지만, 특정한 문맥(의미그룹 번호 6859293에 해당) 내에서는 "학교에서 교사와 관리자들의 조직"이라는 동일한 의미를 가지므로 같은 개념으로 간주될 수 있다. 아래 설명에서도 S_{gid}들 간의 동일성 여부는 이와 같은 방식으로 정의한다.

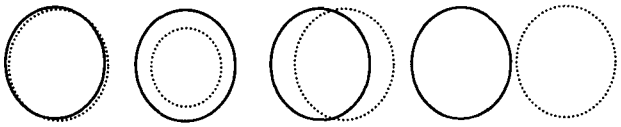


그림 5. 클래스의 의미집합간의 관계
Fig 5. Semantic relationship between two classes.

위 동치 조건에 의해 $O_1(C_{Faculty})$와 $O_2(C_{Staff})$ 및 $O_1(C_{Administrative})$와 $O_2(C_{AdministrativeStaff})$는 동일한 관계라는 것을 도출할 수 있다.

<그림 5-b>는 한 클래스의 의미 집합이 다른 의미 집합에 포함되는 관계를 나타내고 있으며, 다음과 같은 경우 하나의 클래스가 다른 클래스의 하위 개념이라고 간주한다.

(정의 7) 두 클래스간의 의미적 상하위 관계: $LSS(O_X(C_A)) \subset LSS(O_Y(C_B))$ 또는 $GSS(O_X(C_A)) \subset GSS(O_Y(C_B))$인 경우, 클래스 $O_Y(C_B)$는 클래스 $O_X(C_A)$의 하위 클래스이다.

이 정의에 의해 $O_1(C_{Administrative})$, $O_1(C_{Research})$ 및 $O_1(C_{ResearchAssistant})$가 $O_2(C_{Staff})$의 하위 개념이라는 것, $O_2(C_{AdministrativeStaff})$ 및 $O_2(C_{AssistantStaff})$가 $O_1(C_{Faculty})$의 하위 개념이라는 것 및 $O_1(C_{ResearchAssistant})$가 $O_2(C_{Assistant-Staff})$의 하위 개념이라는 관계 도출이 가능하다.

<그림 5-c>는 의미 집합에 공통 요소가 존재하는 것으로 두 클래스들 간에 공유되는 의미가 있다는 것을 의미하고, <그림 5-d>는 두 클래스 간에 공통되는 의미가 전혀 없는 경우이다. 이들 두 가지 관계 유형은 본 논문에서는 직접적으로 이용하지 않는다.

지금까지 클래스에 대해서만 설명을 하였으나, 속성과 인스턴스에 대해서도 동일한 방식으로 적용할 수 있다. 단, 인스턴스는 상하위관계가 존재하지 않으므로 지역 의미 집합과 전역 의미 집합이 동일하며, 두 인스턴스 간의 동치 관계만 도출하고 상하위 관계는 도출하지 않는다.

3. 제안된 알고리즘의 장점

본 논문에서 도입한 개념인 지역 및 전역 의미 집합을 이용한 온톨로지 병합 및 정렬을 이용하면 기존 알고리즘의 문제점을 다음과 같은 방법으로 해결할 수 있다.

첫째, 구문적 형태는 다르지만 의미적으로 유사한 개념 쌍을 찾을 수 없는 문제점은 본 논문에서 의미 그룹 개념의 도입으로 해결할 수 있다. 의미 그룹은 동일한 의미를 가지는 개념들의 집합으로 이것을 활용함으로써, 구문적 형태는 전혀 다르지만 의미는 동일한 개념들을 찾아낼 수 있다. 둘째, 수평적으로 설계된 온톨로지와 수직적으로 설계된 온톨로지의 유사성 발견이 어려운 문제는 전역 의미 집합 개념을 통해 지역적 의미 정보뿐만 아니라 전역적인 의미 정보를 활용함으로써 해결할 수 있다.

마지막으로, 정렬해야 할 개념 쌍을 병합의 대상으로 판단하는 문제점은 Ⅲ장 2절에서와 같이 동치 조건과 상하위 관계 조건을 달리함으로써 해결할 수 있다. 기존의 알고리즘은 동치 조건과 상하위 관계 조건을 별도로 구분하지 않고, 공통되는 구문 정보가 있으면 병합의 대상으로 판단하였으나 본 논문에서는 두 가지 형태를 명시적으로 구분함으로써, 사용자의 혼란을 방지하고 병합 및 정렬에 소요되는 시간을 감소시킬 수 있다.

Ⅳ. 구현 및 실험

1. 구현

제안된 개념을 실험하기 위해 Ⅲ장에서 설명한 알고리즘과 알고리즘의 결과에 따라 OWL로 작성된 온톨로지를 실제로 병합 및 정렬하는 기능을 Java 언어로 구현하였다. 온톨로지를 병합 및 정렬하는 과정은 <그림 6>과 같다.

두 개의 온톨로지를 로드한 후 클래스 정렬과 병합, 속성의 정렬과 병합 마지막으로 인스턴스의 병합 과정을 수행한다. 클래스, 속성 또는 인스턴스의 정렬 및 병합을 하기 위한 알고리즘은 <그림 6>의 오른쪽 부분과 같이 거의 동일하다. 우선 ID를 가져온 후 사전 검색이 가능하도록 토큰(단어) 단위로 나눈 후 형태소 분석을 한다. 형태소 분석된 모든 단어들에 대해 Ⅲ장 1절의 정의에 따라 LSS, GSS를 구해서, 이것들을 Ⅲ장 2절에서 설명한 것과 같은 방식으로 비교하여 최종적으로 병합의 대상과 정렬의 대상을 구해 실제 온톨로지에 대해 이 작업을 수행한다.

속성의 병합 기능 수행시 부가적으로 한가지 조건을 추가하였다. 즉, 의미적으로는 동일하더라도 정의구역(domain)과 치역(range)이 동일할 경우에만 속성의 병합을 하도록 하였다. 인스턴스의 경우에는 병합 후보가 된 두 인스턴스의 타입이 동일한 클래스일 경우에만 병합을 하도록 하였다.

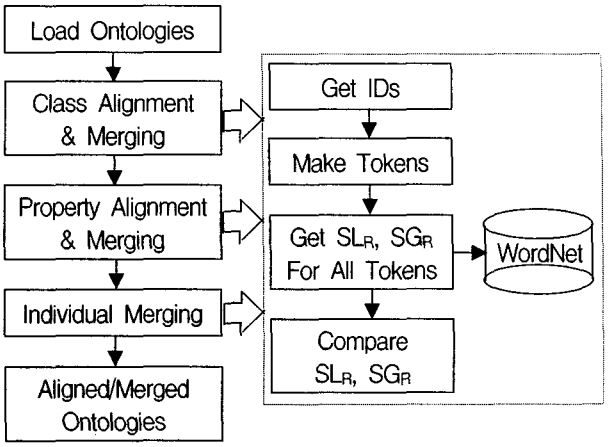


그림 6. 온톨로지 병합 및 정렬 과정
Fig 6. Ontology merging and alignment process.

구현에서 OWL 온톨로지를 다루기 위해 Jena 2.0[7]을 사용하였고, 개념의 의미 그룹 번호를 얻기 위해 WordNet을 사용하였다. WordNet을 Java 언어에서 직접 이용할 수 있도록 JWNL(Java WordNet Library)[6]을 이용하였다. 개념의 이름으로 사용된 단어 또는 구의 형태소 분석은 JWNL에서 제공하는 형태소 분석기를 사용하였다.

2. 실험

본 논문에서 제안한 알고리즘의 성능을 평가하기 위해 <표 1>과 같은 OWL로 구현된 4개의 온톨로지 쌍에 대해 실험을 하였다.

온톨로지 쌍 A는 단순한 형태의 항공 예약 및 자동차 대여 온톨로지로 PROMPT의 실험에 사용된 온톨로지이고, 온톨로지 쌍 B는 자체적으로 작성한(두명에 의해 독립적으로 작성) 현대자동차 및 삼성 자동차 온톨로지이다. 온톨로지 쌍 C는 Anchor-PROMPT[9]의 실험에서 사용된 UMD 및 CMU 온톨로지로 각각 전혀 다른 대학에서 대학 내의 조직 구조를 표현하기 위해 작성된 것이다. 온톨로지 쌍 D(Transportation 온톨로지 및 Cyc-Transportation 온톨로지)는 교통수단을 표현하기 위해 각기 다른 프로젝트에서 보다 전문적인 방식으로 체계적으로 작성된 온톨로지이다. 실험에 사용된 온톨로지 중 일부는 원본이 DAML로 작성되어 있어 이를 OWL로 변환한 후에 실험에 사용하였다.

3. 실험 결과

실험 결과의 평가는 다음과 같이 정확도(Precision)를 기준으로 하였다.

표 1. 실험에 사용된 온톨로지
Table 1. Ontologies used in experiment

	온톨로지 쌍 A	온톨로지 쌍 B	온톨로지 쌍 C	온톨로지 쌍 D	합계
클래스의 수	23	54	170	726	973
속성의 수	42	18	107	84	251
인스턴스의 수	0	96	12	40	148
합계	65	168	289	850	1372

A: 검색된 적합 병렬 및 정렬 정보
B: 검색된 부적합 병합 및 정렬 정보
정확도 = A/(A+B) (1)

검색된 적합 병합 및 정렬 정보(A)는 알고리즘에 의해 생성된 병합 및 정렬 정보 중 적합하다고 판단되는 쌍들이고, 검색된 부적합 병합 및 정렬 정보(B)는 적합하지 않다고 판단되는 쌍들이다. <표 2>는 제안된 알고리즘에 의한 온톨로지 병합 및 정렬의 정확도를 나타낸 것이다.

실험에서 정확도의 측정은 온톨로지 관련 전문적인 지식을 가진 3명에 의해 이루어졌으며, 적합 및 부적합의 판정은 3명 중 2명 이상이 적합 또는 부적합하다고 판단한 내용을 바탕으로 하였으며, 적합하다 아니다를 판단하기 곤란하다고 결론이 난 것들도 모두 부적합한 것이라 간주하였다.

4. 개선할 점

실험 결과로 얻은 평균 91%의 정확도는 본 논문에서 제안하는 알고리즘이 비교적 단순한 것이라는 것을 고려하면 만족할 만한 결과이다. 아래에서 나열한 부적합하다고 판단된 유형들에 대한 처리를 보완하면 보다 향상된 정확도를 보일 수 있을 것으로 기대된다.

가. 부적합하다고 판단된 병합의 유형

· 동일한 단어로 다른 개념을 나타내는 경우: CMU 온톨로지의 Research 클래스는 Research Faculty를, UMD의 Research 클래스는 Research Activity를 의미한다. 이런 경우 본 논문의 알고리즘은 이들 두 클래스가 동일한 것이라 판단하지만(LSS가 같기 때문에) 실제는 그렇지 않다.

표 2. 제안된 알고리즘의 정확도
Table 2. Precision of proposed algorithm.

	온톨로지 쌍 A	온톨로지 쌍 B	온톨로지 쌍 C	온톨로지 쌍 D	평균
정확도	86%	100%	95%	90%	91%

나. 부적합하다고 판단된 정렬의 유형

· 온톨로지 내에서 한 단어가 여러 의미를 가지는 경우: CMU 온톨로지의 Paper_Writing 클래스는 원래 논문 작성이라는 의미로 사용되었지만, 본 알고리즘은 이것이 논문을 뜻하는지 신문을 뜻하는지 명확히 구분하지 못하기 때문에 Paper_Writing 클래스가 UMD 온톨로지의 Newspaper 클래스의 하위 클래스라는 결과를 제시한다.

· 중심 단어를 찾아야 하는 경우: "HelicopterCruiser 클래스는 Helicopter 클래스의 하위 클래스이다"라는 결과가 본 논문의 알고리즘의 결과로 제시된다. 그러나 앞의 클래스는 Cruiser 종류로 보는 것이 더 타당하다.

V. 결 론

본 논문에서는 기존의 알고리즘들이 가지는 단점을 개선한 지역 및 전역 의미집합 개념을 이용한 온톨로지 병합 및 정렬 알고리즘을 제안하였다. 제안된 알고리즘을 구현하여 OWL 언어로 작성된 온톨로지에 대해 실험한 결과 91%의 정확도를 보였다. 이것은 기존 알고리즘보다 정확도가 개선되었을 뿐 아니라 기존의 방식이 반자동으로 이루어진 것에 대한 결과이고, 본 논문에서는 자동 방식으로 처리해서 얻은 결과이기 때문에 본 논문에서 제안한 알고리즘이 보다 편리하고, 좋은 성능을 가진다는 것을 알 수 있다. 본 논문에서 제안하는 알고리즘을 이용하여 온톨로지 병합 및 정렬 작업을 수행하면 온톨로지 공유 및 재활용률을 높이고, 기존 온톨로지를 활용한 새로운 온톨로지의 저작시간도 단축시킬 수 있을 것으로 기대된다. 또한, 온톨로지 매핑등 온톨로지들 간의 의미 정보 교환이 필요한 다른 어떤 분야에도 쉽게 적용이 가능할 것으로 기대된다.

본 논문에서는 OWL 온톨로지에 대해서만 실험을 하였으나, 제안된 알고리즘은 기존의 어떤 온톨로지 언어에 대해서도 적용가능하며, ID가 아닌 다른 정보를 알고리즘 적용 대상으로 활용하는 방식도 적용해 볼 수 있을 것으로 예상된다. 또, 본 논문에서는 온톨로지 병합 및 정렬시 유사한 의미관계 정보만을 이용하였으나

상위 개념(Hypernyms), 하위 개념(Hyponyms), 부분 정보(Part of) 및 기타 자연어 처리에서 사용되는 다른 기술을 활용할 경우 보다 정확하고 다양한 결과를 얻을 수 있을 것으로 기대된다.

참 고 문 헌

[1] Vinary K. Chaudhri, Adam Farquhar, Richard Fikes, Peter D. Karp, James P. Rice, "OKBC: A Programmatic Foundation for Knowledge Base Interoperability," AAAI'98 Conference, pp. 600-607, Madison, WI, July 1998.

[2] Cognitive Science Laboratory at Princeton Uni -versity, "WordNet: a lexical database for the English language," http://www.cogsci.princeton.edu/~wn/.

[3] DARPA, "DARPA Agent Markup Language (DAML)," http://www.daml.org.

[4] Debora L. McGuinness, Richard Fikes, James He -ndler and Lynn Andrea Stein, "DAML+OIL: An Ontology Language for the Semantic Web," IEEE Intelligent Systems, vol.17, no.5, pp. 72-80, September/October, 2002.

[5] Dieter Fensel, Frank van Harmelen, Ian Horr -ocks, Debora L. McGuinness, Peter F. Patel -Schneider, "OIL: An Ontology Infrastructure for the Semantic Web," IEEE Intelligent Systems, vol.16, no.2, pp. 38-45, March/April, 2001.

[6] Greg Barton, John Didion, "JWNL(Java WordNet Library) Project," http://sourceforge.net/projects/jwordnet.

[7] HP Labs Semantic Web research group, "Jena 2.0," http://www.hpl.hp.com/semweb/jena2.htm.

[8] McGuinness, Deborah L., Fikes Richard, Rice James and Wilder Steve, "An Environment for Merging and Testing Large Ontologies. Princi -ples of Knowledge Representation and Reason -ing," Proceedings of the Seventh International Conference, pp. 483-493, San Francisco, CA, April 2000.

[9] N. F. Noy and M. A. Musen, "Anchor -PROMPT: Using non-local context for seman -tic matching," In Workshop on Ontologies and Information Sharing at the Seventeenth Inter -national Joint Conference on Artificial Intelli -gence, pp. 63-70, Seattle, WA, August 2001.

[10] N. Fridman Noy, M.A. Musen, "PROMPT: Algo -rithm and Tool for Automated Ontology Merg -ing and Alignment," Proc. 17th Natl. Conf. on Artificial Intelligence(AAAI'2000), pp. 450-455, Austin, TX, July/August 2000.

[11] W3C, "OWL Web Ontology Language Guide," http://www.w3.org/TR/owl-guide/.

[12] Fridman Noy N., Musen M. A, "An Algorithm for Merging and Aligning Ontologies: Automa -tion and Tool Support," Proc. 16th Natl. Conf. on Artificial Intelligence(AAAI'99), pp. 17-27, Orlando, FL, July 1999.

[13] Gruber, T., "A Translation Approach to Protable Ontologies," Knowledge Acquisition, vol. 5, no. 2, pp. 199-220, 1993.

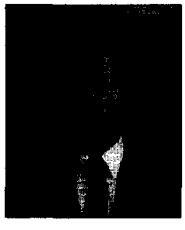
[14] 이재호, "시맨틱 웹의 온톨로지 언어, " 정보과학회지 제21권, 제3호, 18-27쪽, 2003년 3월

저 자 소 개

김 재 홍(정회원)

1994년 경북대학교 컴퓨터공학과 학사 졸업.
1996년 경북대학교 컴퓨터공학과 석사 졸업.
1998년 경북대학교 컴퓨터공학과 박사 수료.

1998년~2001년 3월 (주)필컴 팀장.
2001년 4월~현재 한국전자통신연구원 선임연구원.
<주관심분야: 언어처리, 시맨틱 웹, 지식처리, 온톨로지, 지능형로봇 인터페이스>



이 상 조(정회원)

1974년 경북대학교 사범대학 수학교육과 학사 졸업.
1976년 한국과학원 전산학과 석사 졸업.
1993년 서울대학교 컴퓨터공학과 박사 졸업.

1991년~현재 경북대학교 컴퓨터공학과 정교수.
<주관심분야: 언어처리, 지식처리, 정보검색, 기계학습, 시맨틱 웹, 온톨로지>