분산 토픽맵의 다중 전략 매핑 기법
(A Multi-Strategic Mapping Approach for Distributed Topic Maps)

김 정 민 † 신 효 필 †† 김 형 주 †††
(Jung-Min Kim) (Hyo-phil Shin) (Hyoung-Joo Kim)

요 약 유사한 지식구조의 분산된 온톨로지들을 통합 및 연결하여 새로운 온톨로지를 생성하거나 확장 지식 검색을 효과적으로 제공하기 위해서는 온톨로지 모델 자체의 구조적 특성이나 제약조건을 고려한 온톨로지 매핑이 중요하다. 그러나 과거의 온톨로지 매핑은 범용성을 높이기 위해 대부분 그래프 모델을 기반으로 노드와 간선 중심의 매핑여부를 계산함으로써 온톨로지 모델의 특성과 제약조건을 매핑에 반영하지 못하는 문제점을 가진다. 본 논문에서는 RDF와 함께 온톨로지 모델로 사용되고 있는 토픽맵의 구문적 특성과 제약조건을 반영한 다중 매핑 전략의 토픽맵 매핑 기법을 제안한다. 다중 매핑 전략에는 토픽명 기반 매핑, 토픽 속성 기반 매핑, 계층 구조 기반 매핑, 연관관계 기반 매핑의 4가지 매핑 전략이 포함되어 있으며 개체들 사이의 매핑 여부를 결정하기 위해 각 매핑의 개별 유사도를 조합한 다음 단일 유사도를 결정하는 하이브리드 방식을 사용한다. 또한 토픽맵의 구문적 특성에 따라 매핑 계산 전에 매핑이 불가능한 개체들을 미리 제거함으로써 탐색 범위를 줄이고 있으며 토픽명 색인과 PSI 색인을 생성하여 매핑 계산의 효율을 높이고 있다. 제안하는 토픽맵 매핑 기법의 성능을 보이기 위해 동, 서양 철학 온톨로지들과 야후 철학 백과사전 및 독일 문학 백과사전을 토픽맵으로 구현하여 실험 데이타로 활용하였으며 그 결과 자동 생성된 매핑 집합이 전문가에 의해 생성된 매핑 집합을 대부분 포함함을 확인하였다.

키워드 : 온톨로지 매핑, 토픽맵 매핑, 다중 전략 매핑

Abstract Ontology mapping is the task of finding semantic correspondences between two ontologies. In order to improve the effectiveness of ontology mapping, we need to consider the characteristics and constraints of data models used for implementing ontologies. Earlier research on ontology mapping, however, has proven to be inefficient because the approach should transform input ontologies into graphs and take into account all the nodes and edges of the graphs, which ended up requiring a great amount of processing time. In this paper, we propose a multi-strategic mapping approach to find correspondences between ontologies based on the syntactic or semantic characteristics and constraints of the topic maps. Our multi-strategic mapping approach includes a topic name-based mapping, a topic property-based mapping, a hierarchy-based mapping, and an association-based mapping approach. And it also uses a hybrid method in which a combined similarity is derived from the results of individual mapping approaches. In addition, we don't need to generate a cross-pair of all topics from the ontologies because unmatched pairs of topics can be removed by characteristics and constraints of the topic maps. For our experiments, we used oriental philosophy ontologies, western philosophy ontologies, Yahoo western philosophy dictionary, and Yahoo german literature dictionary as input ontologies. Our experiments show that the automatically generated mapping results conform to the outputs generated manually by domain experts, which is very promising for further work.

Key words : Ontology mapping, Topic Maps mapping, Multi-strategic mapping

· 본 연구는 BK-21 정보기술사업단, 정보통신연구진흥원의 대학 IT연구센터육성지원사업(IITA-2005-C1090-0502-0016), 한국학술진흥재단 기초학문육성지원사업(2004-074-AL0036)의 지원을 받아 수행되었음
† 학생회원 : 서울대학교 컴퓨터공학 박사과정
jmkim@oopsla.snu.ac.kr
†† 정 회 원 : 서울대학교 언어학과 교수
hpshin@snu.ac.kr
††† 종신회원 : 서울대학교 컴퓨터공학 교수
hjk@oopsla.snu.ac.kr
논문접수 : 2005년 5월 4일
심사완료 : 2005년 10월 20일

1. 서 론

시맨틱 웹(Semantic Web)이나 지식 관리 시스템(Knowledge Management System)에서 지식 공유와 검색을 위한 핵심 요소는 온톨로지(ontology)이다. 온톨로지는 도메인의 지식을 개념화하고 의미적으로 연결시킴으로써 하이퍼링크 위주의 자료 연결로 인한 지식 처

리 및 검색의 여러 가지 문제를 해결하기 위해 사용된다. 인터넷 상의 웹페이지들이 여러 사이트에 분산되어 저장되어 있듯이 온톨로지도 여러 사이트에 독립적으로 구축되므로 분산된 개방형 구조를 가진다. 즉, 시맨틱 웹에서 여행, 증권, 쇼핑 등 각 지식 도메인의 온톨로지들은 서로 다른 사이트에 존재하며 비슷한 지식 도메인에 대해 여러 개의 온톨로지들이 서로 독립적으로 생성되어 유지된다. 지식 관리 시스템에서도 인트라넷의 모든 지식을 포함하는 하나의 단일 온톨로지를 생성하는 대신 분야별 세분화된 여러 온톨로지들을 단계적으로 생성하고 이들을 응용프로그램 수준에서 연결하는 구조를 가진다[1].

이와 같이 온톨로지들이 분산된 환경에서는 온톨로지들 사이의 상호 연결, 재사용 및 지식 공유의 필요성이 대두되며 이러한 문제를 해결하기 위한 연구들이 수행되어 왔다. 이들 온톨로지 통합에 관한 연구는 과거 관계형 데이타베이스나 객체지향 데이타베이스 및 XML 데이타베이스 등의 스키마 통합에 대한 연구에서 영향을 받았으며 많은 부분 이들 연구와 유사성을 가진다.

그러나 대부분 스키마 통합 및 온톨로지 통합에 관한 연구에서는 효과적인 통합을 위한 다양한 기법들의 제시에 초점이 맞추어져 있으며 현실적으로 적용이 가능한가에 대해서는 고려하지 않고 있다. 이들 연구에서 제시하는 기법들은 그래프 탐색, 자연어 처리, 기계 학습 등에 기반을 두기 때문에 복잡한 계산을 요구할 뿐만 아니라 실험적인 예제 수준에서 결과를 보이고 있다. 또 다른 문제는 통합을 하고자 하는 두 스키마 또는 온톨로지 A와 B가 있을 때 A의 한 개체에 대해 B의 모든 개체를 일대일로 비교함으로써 N × M의 비교 횟수를 가진다는 것이다. 이는 실시간으로 온톨로지들을 매핑하고 결합된 결과를 제시해야하는 경우에 복잡한 비교와 계산으로 인한 응답시간의 지연을 가져오게 된다.

또한, 이전까지의 연구들은 대부분 범용성을 지원하기 위해 특정 데이타 모델을 그래프 모델로 변환한 다음 두 그래프의 노드와 에지 사이의 매핑을 계산하는 알고리즘을 제시하고 있다. 그러나 현재 온톨로지를 표현하는 데이타 모델은 RDF(Resource Description Framework)[2]와 토픽맵(Topic Maps)[3] 및 OWL(Web Ontology Language)[4]로 표준화되어 있다. 이들 모델은 온톨로지 표현을 위한 구문(syntax), 의미(semantic), 제약조건(constraints)을 가지고 있으며 이러한 특성을 매핑 계산 시에 고려함으로써 그래프 모델의 노드와 에지들 사이에 노드명이나 경로 등의 단순 비교에서 찾을 수 없는 매핑 규칙들을 찾을 수 있다.

따라서 본 논문에서는 토픽맵들 사이의 매핑을 결정하기 위하여 토픽맵 모델의 구문적 및 의미적 특성과 여러 가지 제약조건들을 고려한 다중 전략 매핑 기법을 제시한다. 다중 전략 매핑 기법은 토픽명 기반 매핑, 토픽 속성 기반 매핑, 토픽 사이의 계층 구조 기반 매핑, 연관관계 기반 매핑의 4가지 구체적 매핑 기법들로 구성되어 있다. 또한 토픽맵 모델의 특성에 따라 매핑이 불가능한 토픽들을 매핑 계산에서 제외시킴으로써 탐색 범위를 줄인다. 그리고 PSI(Published Subject Indicator) 및 토픽단어(topic word) 색인을 생성하여 매핑 계산의 효율을 높인다.

제안된 매핑 기법의 성능을 평가하기 위해 실험 데이타를 3 그룹, 즉 동일 도메인의 동일 지식 구조의 온톨로지 집합 그룹, 동일 도메인의 다른 지식 구조의 온톨로지 집합 그룹, 그리고 다른 도메인의 다른 지식 구조의 온톨로지 집합 그룹으로 나누었으며 실험 온톨로지로는 현재 서울대학교 철학사상연구소에서 온톨로지 구축 프로젝트로 진행하고 있는 철학 온톨로지(Philosophy Ontologies)[5]와 야후 백과사전의 철학사전과 독일 문학사전을 토픽맵으로 구현하여 사용하였다. 실험 결과에 대한 평가는 전문가에 의해 수작업으로 생성된 매핑 집합과 시스템이 자동 생성한 매핑 집합 사이의 겹쳐지는 집합의 크기에 따른 정확율(precision), 재현율(recall), 종합율(overcome)을 산정하였으며 그 결과 시스템의 자동 생성 매핑 집합이 전문가의 매핑 집합을 대부분 포함하고 있음을 보이고 있다.

토픽맵과 RDF는 각각의 모델 요소들이 직접적으로 대응되기 때문에 상호 변환이 가능하므로 여기서 소개하는 기법들은 RDF 온톨로지의 매핑에도 쉽게 적용할 수 있으며 OWL 온톨로지의 매핑 연구에도 많은 부분 적용할 수 있다[7].

본 논문의 구성은 2장에서 관계형 스키마 및 온톨로지 매핑과 관련된 이전 연구들을 살펴보고 3장에서는 철학 온톨로지의 구조 및 구성 내용을 소개한다. 4장에서는 전체적인 매핑 프로세스와 각 단계별 자세한 처리 기법을 설명한다. 그리고 5장에서는 색인의 자료 구조와 매핑 알고리즘들을 자세히 소개하고 6장에서는 실험 데이타와 실험 결과를 보인다. 그리고 7장에서 논문의 결론과 향후 연구에 대해 기술한다.

2. 관련연구

본 연구와 관련된 연구로서 먼저 스키마 매핑은 두 스키마의 엘리먼트들 사이에 의미적 유사성을 찾는 것으로 전자상거래, 데이타 웨어하우스, 데이타 통합 등 여러 응용 분야에서 필요로 하고 있다[8]. 이 스키마 매핑을 반자동으로 처리하기 위한 연구들로서 COMA[9], Cupid[10], LSD[11], MOMIS[12], SemInt[13], Simila-

rity Flooding[14] 등이 있으며 이들 대부분은 관계형 데이타베이스, XML, ER, 그래프 등 특정 응용분야 및 데이타 모델을 대상으로 매핑문제를 해결하는 시스템 및 알고리즘을 제시하고 있다.

스키마 매핑 기법들은 비교 대상을 선정하는 방법에 따라 인스턴스 수준 접근법(instance-level approaches)과 스키마 수준 접근법(schema-level approaches) 및 엘리먼트 수준 접근법(element-level approaches)과 구조 수준 접근법(structure-level approaches)으로 나누어지고 비교 알고리즘에 따라 구문적 접근법(syntactic approaches), 구조적 접근법(structure approaches), 의미적 접근법(semantic approaches)으로 나누어진다[8]. SemInt의 경우 관계형 데이타베이스 스키마의 애트리뷰트들 사이의 유사성을 찾는 엘리먼트 수준의 접근법을 사용하고 있으며 매핑 기법도 데이타 형, 데이타 길이, 키 정보 등의 구조적 유사성에 기인하고 있다. 이와 달리 Cupid의 경우 구조 및 엘리먼트 수준에서 유사성을 찾으며 다양한 매핑 알고리즘을 포함하는 하이브리드 매핑 기법으로 구문적, 구조적, 의미적 매핑 탐색 알고리즘에 따라 두 스키마의 매핑 개체들을 구한다.

온톨로지 매핑은 스키마 매핑 연구에서 영향을 받았으며 많은 부분 위에서 언급한 연구들과 유사성을 가진다[15,16]. 온톨로지 매핑 및 통합과 관련된 연구로는 PROMPT[17], Anchor-PROMPT[18], Ctx-Match[19], Information flow[20], FCA-Merge[21], QOM[22] 등이 있다. PROMPT는 엘리먼트 수준의 구문적 접근법만을 지원하며 온톨로지의 개념명(concept name)이 완전히 일치하는 요소들 사이의 매핑을 처리한다. Anchor-PROMPT는 PROMPT에 의해 발견된 매핑들 사이의 경로를 확인하고 경로의 길이가 동일할 경우 그 중간에 존재하는 노드들을 매핑시키는 부분적인 구조적 접근법을 지원한다. QOM은 매핑을 위한 온톨로지 탐색 범위를 줄임으로써 매핑의 효율을 높이고 실제적인 응용프로그램에 적용할 수 있음을 보인다.

Topic Maps Reference Model[23]에서는 두 개의 서로 다른 토픽맵의 토픽들이 동일한 Subject Identity를 가지는 경우에만 통합됨을 설명하고 있다. 그러나 실제적으로 모든 토픽들이 주제 식별 정보를 가지지 않으며 토픽맵들 사이에 의미적으로 같은 토픽이지만 서로 다른 주제 식별 정보를 가질 수 있다. 이러한 문제를 해결하기 위해 SIM[24]에서는 구문적 접근법과 부분적인 구조적 접근법으로 토픽쌍의 유사값을 계산하고 매핑을 결정할 수 있음을 보인다. 그러나 SIM에서는 단순히 토픽명과 어커런스 데이타만을 비교하여 유사값을 계산하고 있으며 토픽맵 모델의 특성을 고려하지 않고 모든 토픽들을 비교 대상으로 하고 있다.

3. 철학 온톨로지 구조

철학 온톨로지는 크게 철학 분야의 포괄적 지식과 철학 텍스트들의 내용 지식으로 구성된다. 철학 분야의 포괄적 지식은 철학자, 철학문헌, 철학사, 철학분야, 철학이론, 철학학파, 철학용어의 7개 개념을 중심으로 각 개념의 상세 개념들로 구성되어 있으며 철학 텍스트의 내용 지식은 철학 텍스트의 내용을 분석하여 저자가 전달하고자 하는 핵심적인 철학 개념들을 추출하고 각 개념을 서술하는 텍스트 구조에서 하위 수준의 상세 개념들을 파악하여 계층 및 연관 관계에 따라 개념들을 연결한 지식 구조로 구성되어 있다.

다른 일반적인 온톨로지들과 비교해볼 때 철학 온톨로지가 가지는 의의는 저자, 출판사, 제목 등의 철학 텍스트의 메타데이타뿐만 아니라 텍스트 내용에 들어있는 지식을 텍스트 외부에 형상화함으로써 텍스트를 읽지 않더라도 주요 개념들을 검색하고 부분적으로 읽고 이해하는데 있다. 또한 여러 텍스트들이 내용적으로 어떤 연관성을 가지는지 보임으로써 철학 연구자나 일반 학습자들이 철학사적으로 주요 철학 사상이나 이론이 어떠한 방식으로 전개되어 왔으며 이론의 주장, 보완, 반론 등의 관계가 어떠한지 쉽게 파악할 수 있도록 한다.

철학 온톨로지에서 최상위 개념은 철학이다. 철학 개념 밑으로 'is-a' 또는 'part-of' 관계에 의하여 개념들의 계층 구조를 형성하는데 철학 개념은 그 아래에 직접적으로 'part-of' 관계를 가지는 철학자, 철학문헌, 철학분야, 철학사, 철학학파, 철학이론, 철학용어의 7가지 상세 개념들을 가진다. 이는 철학을 구성하는 일반적인 기준들로서 과거로부터 객관적으로 인정되는 분류 기준이다. 그리고 각 분류 항목은 그 하위에 시대적 및 지리적 분류에 따라 크게 동양철학과 서양철학으로 나누어지고 동양철학은 다시 한국철학, 중국철학, 인도철학으로 나누어진다. 그리고 서양철학은 시대적 기준에 따라 서양고대철학, 서양중세철학, 서양근대철학, 서양현대철학으로 나누어진다. 이러한 기준에 따라 철학자의 경우 한국철학자, 중국 철학자, 인도 철학자, 서양 고대 철학자, 서양 중세 철학자, 서양 근대 철학자, 서양 현대 철학자로 분류하였으며 철학자와 이들 하위 철학자들 사이에 'is-a' 관계를 설정하였다. 이외 철학문헌, 철학용어 등 다른 세부 개념들도 동일한 기준으로 분류하였다.

그림 1은 철학 온톨로지의 구조를 간략히 보이는 것으로 상위의 분류 계층과 하위의 내용기반 계층 사이의 연결을 보이고 있다. 칸트의 실천이성비판 문헌을 검색하고자 할 경우 최상위 철학에서부터 철학문헌, 서양 근대 철학문헌으로 탐색하면 그 하위의 실천이성비판 개념 노드를 찾을 수 있다. 실천이성비판 텍스트의 하위

그림 1 철학 온톨로지의 구조

개념들은 그 텍스트의 내용상에 존재하는 주요 개념들로서 이성, 자유, 도덕법칙, 의지 등이 있다. 한 텍스트에는 많은 개념들이 추출될 수 있으므로 이 개념들의 CID(Concept IDentifier)는 각각 c1, c2, c3, c4로 부여되며 이성 개념은 그 하위에 이성의 정의와 이성의 구분이라는 개념들로 세분화된다. 이때 세부 개념의 CID는 c1.1과 c1.2로 부여된다.

현재 철학 온톨로지는 철학의 일반적 지식과 함께 동, 서양 고전 텍스트 60여권을 분석한 텍스트 내용 지식으로 구성되어 있으며 지속적인 프로젝트 수행을 통해 향후 300여권까지 대상을 넓힐 예정이다.

4. 온톨로지 매핑 프로세스

두 온톨로지의 매핑 문제는 두 온톨로지의 엘리먼트(element)들 사이에 의미적 대응 관계를 찾는 것이다. 엘리먼트들 사이의 대응 관계를 찾기 위해서는 서로 다른 온톨로지에 속하는 두 엘리먼트들 사이에 어떤 유사성이 존재하는지 구문적, 구조적, 의미적 접근법에 따라 여러 가지 매핑 계산을 수행한 다음 산출된 유사값([0..1] 범위의 값)들을 조합하여 하나의 단일 유사값을 결정하고 이 유사값을 분석하여 매핑 여부를 결정하는 처리 과정을 따른다. 본 논문의 온톨로지 매핑 프로세스는 크게 준비 단계, 후보 선정 단계, 유사값 계산 단계, 유사값 조합 단계, 매핑 결정 단계로 나누어진다. 그림 2는 매핑 프로세스의 각 단계와 이들의 연결 구조를 보이고 있다.

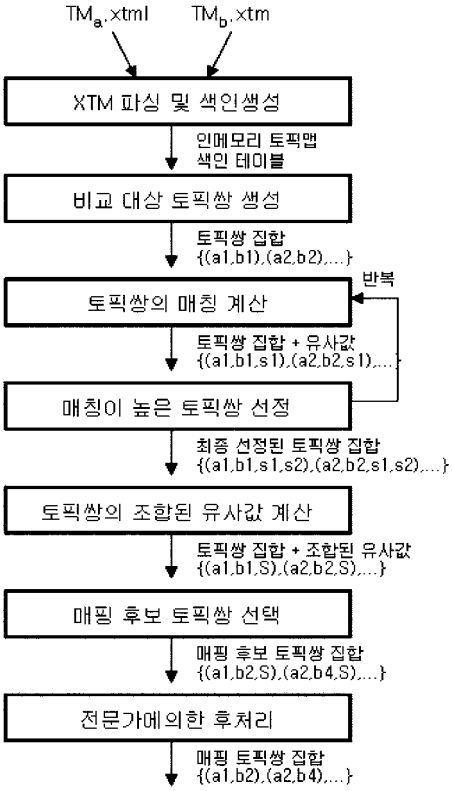


그림 2 온톨로지 매핑 프로세스

XTM 파싱 및 색인 생성. 토픽맵의 표준 기술 언어는 XTM이므로 두 개의 XTM 파일이 입력으로 주어지면 준비 단계에서는 각 파일의 구문을 파싱하여 메모리 내에 두 개의 토픽맵을 생성한다. 또한 파싱 과정에서 매핑 알고리즘이 필요로 하는 두 가지의 색인 즉, PSI 색인과 토픽단어 색인을 각 토픽맵마다 별도로 생성한다. PSI는 서로 다른 도메인의 여러 토픽맵들 사이에 공통의 개념 정의를 참조하도록 하기 위한 것으로 토픽맵내의 각 개념 토픽이 가질 수 있는 주제 식별자이다. 그러므로 두 토픽맵에서 서로 다른 이름의 개념 토픽이 동일한 PSI를 참조하고 있다면 토픽 이름이나 구조에 상관없이 두 토픽은 동일한 토픽으로 간주된다. PIS 색인은 토픽맵의 각 토픽들에 부여된 주제 식별값을 색인화 하는 것으로 매핑 계산을 적용할 비교 대상 토픽쌍을 생성하는데 필요하다.

토픽단어 색인은 토픽의 기본명(base name)과 별명(variant name) 및 자원데이타(resourceData) 형식의 어커런스에 대한 색인으로서 토픽맵 내의 모든 토픽들의 이름과 어커런스 문자열을 단어 단위로 분해한 다음 각 단어를 중심으로 그 단어가 존재하는 토픽들의 ID와 연결해 놓은 역색인(inverted index) 구조를 가진다. 이 토픽단어 색인은 PSI 색인과 함께 매핑 계산을 적용할 비교 대상 토픽쌍을 생성하는데 사용될 뿐만 아니라 토픽명 기반 및 토픽 속성 기반 매핑에서 동일한 단어를 내포하는 토픽들을 쉽게 파악하기위해 사용된다.

비교 대상 토픽쌍 생성. 토픽맵 및 색인 생성의 준비 단계가 끝나면 두 토픽맵의 토픽들을 일대일로 쌍을 지은 토픽쌍 집합({(a, b) | a∈TM_a, b∈TM_b})을 생성한다. 온톨로지 매핑과 관련된 이전 연구들에서는 한 온톨로지의 하나의 엘리먼트에 대해서 다른 온톨로지의 모든 엘리먼트와 일대일로 쌍을 이루도록 하였다. 그러나 본 연구에서는 온톨로지 매핑의 효율을 높이기 위해 매핑이 불가능한 토픽쌍이나 또는 구문적으로 매핑이 명백한 토픽쌍을 매핑 계산 전에 미리 파악하여 이들을 비교 대상 토픽쌍 집합 생성에서 제외시킴으로써 매핑 탐색 범위를 줄인다. 아래 수식은 두 토픽맵에서 PSI와 토픽 유형을 기준으로 매핑 계산이 불필요한 토픽쌍을 제거하는 것을 보이고 있다.

$$S = \{(a, b) \mid a \in TM_a, b \in TM_b\} - \{(a', b') \mid a'.psi = b'.psi, a' \in TM_a, b' \in TM_b\} - \{(a'', b'') \mid a''.type <> b''.type, a'' \in TM_a, b'' \in TM_b\} \quad (1)$$

탐색 범위를 줄이기 위해 비교 대상 토픽쌍 생성 단계에서는 먼저 PSI 색인을 이용하여 두 토픽맵에서 동일한 주제 식별 정보를 가지는 토픽들을 찾는다. 이들 토픽들은 매핑 계산을 하지 않아도 토픽맵 표준에 의해 동일한 토픽으로 간주되므로 비교 대상에서 제외시킨다. 토픽맵에서 주제 식별 정보는 토픽의 정체성을 보장하는 방법으로서 대체적으로 다른 토픽들을 카테고리화하는 상위 수준의 토픽들에 부여된다. 예를 들어, 철학 온톨로지에서 철학자 토픽은 아래 그림 3과 같이 정의된다. <SubjectIndicatorRef> 엘리먼트의 href 속성에 부여된 PSI 값에 의해 토픽 ID가 'philosopher'인 토픽은 토픽명에 상관없이 하나의 정체성을 가지게 되며 이는 동일한 PSI를 가지는 다른 토픽명의 토픽들과 본질적으로 동일한 개념임을 보장하는 수단이 된다.

동일한 도메인의 소그룹의 토픽맵들 사이에는 이들이 공유할 수 있는 상위 수준의 참조 토픽맵이 존재하며 이 참조 토픽맵의 토픽들은 주제 식별 정보를 가지도록 설계된다. 그러므로 이들 소그룹의 토픽맵들을 매핑할 경우 많은 수의 상위 토픽들이 토픽명에 상관없이 동일한 PSI를 가지고 있으므로 이들을 비교 대상 토픽쌍에서 제거하게 되면 매핑 탐색 범위를 상당히 줄일 수 있다.

토픽맵에서 토픽은 하위 토픽의 포함 여부에 따라 클래스 토픽(class topic)과 인스턴스 토픽(instance topic)으로 나눌 수 있는데 클래스 토픽은 토픽 계층 구조에서 'is-a' 또는 'part-of' 관계로 연결된 하위 토픽을 가지는 토픽이고 인스턴스 토픽은 하위 토픽을 가지지 않는 말단(leaf) 토픽을 말한다. 또한 참조 유형에 따라 토픽 타입(topic type), 어커런스 타입(occurrence type), 연관관계 타입(association type), 역할 타입(role type)의 4가지 유형으로 나누어진다. 이 중에서 토픽 타입은 토픽들 사이의 계층관계에서 하위 토픽이 참조하는 상위 토픽으로서 클래스 토픽과 동일하다. 위 그림 3에서 philosophy 토픽은 philosopher 토픽의 토픽 타입이 된다.

어커런스 타입은 토픽의 속성인 어커런스의 유형을 정의하는 토픽이다. 예를 들어, 철학 온톨로지에서는 철학자 토픽의 속성으로 생애요약, 생애연보, 출생도시, 대표저작 등이 있다. 토픽맵에서는 이 속성들도 토픽으로 정의되어야 하며 이러한 토픽을 어커런스 토픽이라고 한다. 이는 객체지향 언어에서 클래스의 멤버 타입이 또 다른 클래스로 정의될 수 있는 것과 같다. 연관관계 타입은 연관관계를 정의하기 위한 참조되는 토픽이다. 예를 들어, '칸트'와 '실천이성비판' 사이에는 'written by' 이라는 연관관계가 정의된다. 이때 'written by' 라는 토픽을 정의해야 하며 이 토픽은 철학자와 철학저서 사이에 연관관계를 설정하는 연관관계 타입으로 참조된다.

역할 타입은 연관관계를 갖는 토픽들이 각각의 관계상에서 어떤 역할을 하는지 설명하기 위해 정의된 토픽을 말한다. 예를 들어, 'written by' 연관관계에서 칸트는 저자 역할, 실천이성비판은 도서 역할을 하게 되는데, 이때 '저자'와 '도서' 토픽을 역할 타입으로 분류하게 된다. 4가지 타입으로 분류된 토픽들은 토픽맵에서 서로 다른 의미로 사용되며 이들 사이에는 매핑이 존재할 가

```
<topic id="philosopher">
 <instanceOf><topicRef xlink:href="#philosophy"/></instanceOf>
 <subjectIdentity>
   <subjectIndicatorRef
     xlink:href="http://plato.snu.ac.kr/psi/philosophy/philosophy.psi#philosopher"/>
 </subjectIdentity>
 <baseName>
   <baseNameString>철학자</baseNameString>
 </baseName>
</topic>
```

그림 3 XTM으로 기술된 철학자 토픽 정의 구문

능성이 낮기 때문에 비교 대상 토픽쌍을 생성할 때 서로 다른 타입의 토픽들 사이에는 토픽쌍을 생성하지 않음으로써 탐색 범위를 줄일 수 있다.

토픽쌍의 매핑 계산. 비교할 토픽쌍이 주어지면 두 토픽 사이의 매핑 정도를 알기 위해 유사값을 계산하는 매핑 기법들을 적용한다. 본 논문에서는 토픽명 기반 매핑, 토픽 속성 기반 매핑, 계층 구조 기반 매핑, 연관관계 기반 매핑의 4가지의 매핑 기법들을 조합하는 하이브리드 방식을 택한다. 즉 각 토픽쌍은 4가지 매핑 계산에 의한 유사값 4개를 가지게 되며 이 4개의 유사값을 조합하여 하나의 유사값을 산출함으로써 두 토픽 사이의 유사성을 결정하게 된다.

토픽명 기반 매핑은 가장 기본적인 것으로 두 토픽의 이름을 단어 수준에서 비교하여 유사값을 계산하는 기법이다. 토픽 속성 기반 매핑은 두 토픽의 어커런스들을 비교하는 것으로 어커런스 타입, 자원데이타(resourceData)나 자원참조(resourceRef)로 기술되는 어커런스 값, 어커런스 값의 영역(scope)으로 나누어 비교한 다음 각 항목별로 유사값을 계산하고 이를 조합하여 두 토픽의 어커런스 유사값을 결정한다.

계층 구조 기반 매핑은 두 토픽의 자식 토픽들 사이에 유사값을 계산한 다음 이 값에 따라 두 토픽의 계층 구조의 유사도를 측정하는 기법이다. 이는 두 토픽이 비록 토픽명이나 속성이 다르더라도 그 하위의 토픽들 사이에 유사도가 높을 경우 두 토픽의 매핑 가능성을 높이기 위해서 사용된다. 연관관계 기반 매핑은 연관관계 타입인 토픽쌍의 유사값을 계산하는 매핑 기법으로 각 토픽의 연관관계에서 멤버 토픽들과 역할 타입 토픽들을 서로 비교함으로써 유사값을 결정한다. 그림 4는 매핑 기법들의 적용 순서와 토픽쌍 행렬내에 각 셀마다 4가지 유사값을 가짐을 보이고 있다. 예를 들어 (a_1, b_1) 쌍의 유사값으로 S_n은 토픽명 기반 유사값, S_o는 토픽속성 기반 유사값, S_h는 계층 구조 기반 유사값, S_a는 연관관계 기반 유사값을 가진다.

토픽쌍의 조합된 유사값 계산. 모든 토픽쌍의 단일

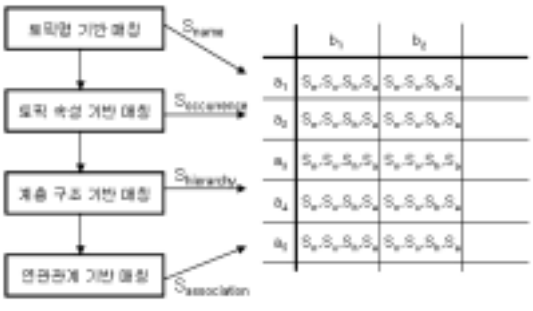


그림 4 매핑 기법의 유형 및 토픽쌍 행렬

화된 유사값을 결정하기 위해 매핑 기법들에 의해 계산된 각 유사값들을 조합한다. 유사값들을 하나로 조합하는 방식으로는 가장 큰 유사값을 선택하는 최대값 방식, 특정 매핑 기법에 가중치를 두어 유사값의 합계를 구하는 가중치 적용 방식, 모든 매핑 기법들에 동일한 가중치를 두는 평균값 방식, 그리고 가장 작은 유사값을 선택하는 최소값 방식이 있다[7]. 이 중에서 본 논문에서는 4가지 매핑 기법들에 동일한 가중치를 주는 평균값 방식을 사용한다. 예를 들어, a1과 b1 사이의 단일 유사값 SIM(a_1, b_1)은 (SIMname + SIMocc + SIM_H + SIMassoc) / 4로 계산된다.

매핑 후보 토픽쌍 선택. 매핑 후보를 결정하기 위해서 먼저 조합된 유사값의 내림차순으로 토픽쌍들을 정렬한다. 예를 들어, 토픽맵 A의 a_1 토픽과 쌍을 이루는 토픽맵 B의 b1, b2, b3, ..., b_m 중에서 매핑 후보를 결정하기 위해 각 쌍의 SIM 값에 따라 내림차순으로 정렬하는 것이다. 정렬 후 매핑 후보를 결정하는 방법은 상위의 몇 개를 선택하는 MaxN 방식, 최상위 유사값에서 d 만큼 내려온 값 사이에 존재하는 쌍들을 선택하는 MaxDelta 방식, 그리고 주어진 기준값(threshold)을 초과하는 쌍들을 선택하는 기준값 방식이 있다[9]. 본 논문에서는 기준값 0.5 이상의 유사값을 가지는 토픽쌍을 매핑 후보로 결정한다.

5. 색인 구조 및 매핑 알고리즘

5.1 색인 구조

PSI 색인은 XTM 구문에서 <SubjectIdentity> 엘리먼트를 하위 엘리먼트로 가지는 모든 토픽들을 대상으로 그 토픽의 ID와 <SubjectIdentity> 엘리먼트 값으로 기술된 URI(PSI로 기술된 경우)나 문자열(SubjectIdentifier로 기술된 경우)을 가진다. 먼저 PSI 색인은 아래 그림 5와 같은 구조를 가진다.

토픽단어 색인은 아래 그림 6과 같은 구조를 가진다. 토픽단어 색인은 토픽명과 토픽의 어커런스에 들어있는 단어를 역색인 구조로 색인한다. 이는 정보 검색에서의 역색인과 비슷한 구조이다.

기본명과 별명의 토픽명 및 어커런스 문자열은 단어 단위로 분해되어 Word 필드에 저장된다. 단어 단위로 분할하는 이유는 기호나 축약 문자로 컬럼명을 기술하는 데이타베이스 스키마와 달리 온톨로지에서는 명사형으로 개념명을 정의하기 때문에 문자 단위보다는 단어 단위로 비교하는 것이 더 적합하기 때문이다. 예를 들어, '임마누엘 칸트'의 경우 '임마누엘'과 '칸트'로 나누어서 두 개의 레코드로 색인에 저장된다. 이때 TopicID에는 동일한 토픽 ID가 저장된다. Scope는 토픽명이나 어커런스의 영역을 가리키는 것으로 동일한 토픽명이더라

PSI 색인 구조	
(TopicMap, TopicID, PSI, TreeLevel, Type/Instance)	
TopicMap	토픽맵의 ID.
TopicID	PSI를 가지는 토픽의 ID이다.
PSI	토픽의 Subject Identity로서 URI 혹은 문자열 값을 가진다.
TreeLevel	토픽이 위치하는 계층 구조의 레벨값을 가진다.
Type/Instance	클래스 토픽(type)인지 또는 인스턴스 토픽(instance)인지를 가리킨다.

그림 5 PSI 색인 구조 및 설명

토픽단어 색인 구조	
(TopicMap, Word, TopicID, TreeLevel, Type/Instance, Scope, Type)	
TopicMap	토픽맵의 ID.
Word	토픽명에 들어있는 단어.
Name/Occ	토픽명의 단어이면 Name, 어커런스의 단어이면 어커런스 타입명을 가진다.
TopicID	Word 필드의 단어를 토픽명에 포함하고 있는 토픽의 ID이다.
TreeLevel	토픽이 위치하는 계층 구조의 레벨값을 가진다.
Type/Instance	클래스 토픽(type)인지 또는 인스턴스 토픽(instance)인지를 가리킨다.
Scope	토픽명의 영역을 가리킨다.
Type	토픽의 타입 유형을 가리킨다. 토픽타입(TT), 어커런스 타입(OT), 연관관계 타입(AT), 역할 타입(RT)

그림 6 토픽단어 색인의 구조와 설명

도 영역이 다른 경우 서로 다른 의미의 토픽명이 된다. Type은 클래스 토픽인 경우 어떤 유형으로 참조되는지 구분하기 위해 사용된다.

XTM 파싱 과정에서 생성되는 PSI 색인과 토픽단어 색인은 토픽맵 매핑 계산중에는 메모리 내에 해쉬 테이블(hash table)로 저장되지만 매핑 후에는 토픽맵과 함께 관계형 데이타베이스에 별도의 테이블로 저장되어 새로운 토픽맵과의 매핑시 재사용된다.

5.2 매핑 알고리즘

이번 장에서는 4가지 매핑 기법들의 핵심 알고리즘과 매핑 처리 과정에 대해 기술한다.

토픽명 기반 매핑 기법. 토픽명 기반 매핑 기법은 두 개의 토픽단어 색인 테이블과 비교 대상 토픽쌍을 입력으로 받아들인 다음 먼저 각 토픽쌍에 대해 두 토픽이 토픽명에 동일한 단어를 가지는지 토픽단어 색인을 이용하여 검사한다. 이때 두 토픽의 동일한 단어가 하나 이상인 경우에만 토픽명에 대한 문자열 매핑 계산을 수행하며 동일한 단어를 가지지 않는 토픽쌍은 문자열 매핑 계산을 하지 않고 SIMname 값을 0으로 둔다. 동일한 단어가 존재할 경우에는 두 토픽 중에서 토픽명의 단어 수가 적은 토픽의 단어 수로 공통 단어수를 나누어 SIMname 값을 구한다. 관계형 데이타베이스 스키마 통합에서는 약자, 특수문자, 언더바 등의 기호 형태의 컬러명이 많으므로 이들 사이의 문자열 유사값 계산을 위해 문자 중심의 비교 기법을 사용하였으나 온톨로지의 경우에는 의미를 내포하는 명사형 단어(noun words)나 구(noun phrase)로 개념을 표현하기 때문에 문자 하나보다는 단어 자체의 비교가 중요하다. 토픽명 기반 매핑 기법의 알고리즘은 아래 그림 7과 같다.

토픽 속성 기반 매핑 기법. 토픽의 속성인 어커런스는 어커런스 타입과 어커런스 값으로 구성되며 어커런스 값으로는 문자열이나 외부 자원의 URI 주소를 가진다. 토픽 속성 기반 매핑 기법에서는 먼저 두 토픽의 어커런스 타입 집합을 각각 구한 다음 이 어컨런스 타입들 사이의 유사값을 계산한다. 어커런스 타입의 유사값은 토픽명 기반 매핑 기법에서 계산된 SIMname 값을 사용한다. 즉, 어커런스 타입도 하나의 토픽이므로 토픽명 기반 매핑 기법에서 어커런스 타입 토픽들 사이의 토픽명에 대한 유사값을 계산하기 때문이다. 그리고 토픽단어 색인을 이용하여 문자열 형태의 어커런스 값 사이의 유사값을 계산한다. 이후 어커런스 타입 유사값과 어커런스 값 유사값을 조합하여 두 토픽의 최종 어커런스 유사값을 결정한다. 아래 수식은 어커런스 타입의 유사값을 구하는 수식으로 T_1과 T_2는 각각 어커런스 타입 토픽들의 집합이며 t_1과 t_2는 어커런스 타입 토픽들이다. 그리고 C는 $|T_1| < |T_2|$인 경우 $|T_1|$의 값을 가진다. 이 수식은 어커런스 타입 토픽쌍의 SIMname 값의 최대값들을 구한 다음 이 최대값들의 평균을 구하는 식이다.

$$SIM_{occtype} = \frac{\sum_{i=m} \max_{j=n} SIM_{name}(t_i, t_j)}{c}, \quad t_i \in T_1, t_2 \in T_2, m = |T_1|, n = |T_2| \tag{2}$$

5.2.1 토픽명 기반 매핑 알고리즘

```
nameMatcher(nameIdx1:NameIndex, nameIdx2:NameIndex, pairSet:TopicMatrix)
    for each pairs of topics in pairSet
// ① 하나의 토픽쌍으로부터 비교할 토픽들의 ID를 가져온다.
        topic1 = topicPair.topic1;
        topic2 = topicPair.topic2;
// ② 토픽단어 색인으로부터 각 토픽의 단어수와 공통 단어수를 구한다.
        wordCount1 = countWord(topic1, nameIdx1);
        wordCount2 = countWord(topic2, nameIdx2);
        sameWordCount = countSameWord(topic1, topic2, nameIdx1, nameIdx2);
// ③ 구해진 공통 단어수에 따라 SIMname 값을 구하고 이를 토픽쌍에 저장한다.
        if sameWordCount = 0 then SIMname = 0
        else
            if wordCount1 < wordCount2 then
                SIMname = sameWordCount / wordCount1
            else
                SIMname = sameWordCount / wordCount2
            end if
        end if
        topicPair.SIMname = SIMname;
```

그림 7 토픽명 기반 매핑 알고리즘

5.2.2 토픽 속성 기반 매핑 알고리즘

```
occurrenceMatcher(nameIdx1:NameIndex, nameIdx2:NameIndex, pairSet:TopicMatrix)
    for each pairs of topics in pairSet
// ① 하나의 토픽쌍으로부터 비교할 토픽들의 ID를 가져온다.
        topic1 = topicPair.topic1;
        topic2 = topicPair.topic2;
// ② 두 토픽의 어커런스 타입에 대한 유사값을 구한다.
//    두 토픽의 어커런스 타입쌍의 SIMname 값을 읽어온 다음 최대값을 취한다.
//    최대값들의 평균을 구하여 SIMocctype 값으로 한다.
        SIMocctype = calcOccTypeSimilarity(topic1, topic2, nameIdx1, nameIdx2);
// ③ 두 토픽의 어커런스 데이터(문자열, URI)에 대한 유사값을 구한다.
//    문자열인 경우 토픽명 기반 매핑과 같이 토픽단어 색인에 공통 단어의 비율을 구한다.
//    URI인 경우 URI 전체를 비교하여 완전히 동일할 경우만 1을 그렇지 않으면 0으로 둔다.
//    유사값들의 평균을 구하여 SIMoccdata 값으로 둔다.
        SIMoccdata = calcOccDataSimilarity(topic1, topic2, nameIdx1, nameIdx2);
// ④ SIMocctype과 SIMoccdata의 평균을 구하여 SIMocc 값으로 저장한다.
        SIMocc = (SIMocctype + SIMoccdata) / 2
        topicPair.SIMocc = SIMocc;
```

그림 8 토픽 속성 기반 매핑 알고리즘

그림 8은 토픽 속성 기반 매핑 알고리즘을 기술하고 있다.

계층 구조 기반 매핑 기법. 계층 구조 기반 매핑 기법은 두 토픽의 하위 토픽 쌍들의 유사한 정도를 측정하여 두 토픽의 구조적 유사도를 판단하는 기법이다. 예를 들어, 그림 9에서 토픽맵 A의 철학자와 토픽맵 B의 철학자는 동일한 토픽명을 가지지만 구조적 유사도를 측정할 경우 토픽맵 A의 근대 철학자와 토픽맵 B의 철학자가 더 높은 유사도를 가지게 된다. 즉, 하위 토픽 쌍들 사이의 유사도를 비교해 볼 때 토픽맵 B의 철학자는 토픽맵 A의 철학자보다 근대 철학자와 매핑되어야 함을 알 수 있다.

(a) 토픽맵 A의 철학자 계층구조 (b)토픽맵 B의 철학자 계층구조

그림 9 계층 구조를 가지는 토픽들의 예

두 토픽 사이의 계층 구조 기반 유사값은 다음의 수식에 의해 계산된다. 여기서 T1, T2는 계층적 유사도를 측정할 두 토픽이고 각각 m 개의 하위 토픽들과 n 개의 하위 토픽들을 가지고 있다. t_i는 T_1의 i번째 하위 토픽이고 t_j는 T_2의 j번째 하위 토픽으로 이 두 토픽쌍의 SIMname, SIMocc, SIM_H 값을 토픽쌍 행렬에서 읽어들인 다음 이들 값의 평균을 구하여 두 토픽의 단일 유사값을 구한다. 모든 하위 토픽 쌍들의 단일 유사값이 계산되면 이 값들의 합계를 구하고 다시 하위 토픽 쌍의 수로 나누어 T_1과 T_2의 계층적 유사값을 결정하게 된다.

$$SIM_H(T_1, T_2) = \sum_{\substack{i=m \\ j=n}} \{AVG(SIM_{name}(t_i, t_j) + SIM_{occ}(t_i, t_j) + SIM_H(t_i, t_j))\} / (m \times n) \quad (3)$$

이때 T_1과 T_2 둘 다 하위 토픽을 가지지 않는 인스턴스 토픽인 경우와 둘 중 하나가 인스턴스 토픽인 경우 $SIM_H(T_1, T_2)$는 0값을 가진다. 그러므로 두 토픽맵에서 클래스 토픽인 토픽 타입 쌍에 대해서만 계층 구조 유사값을 계산하고 나머지는 0으로 둔다. 위 수식에서 보면 T_1과 T_2의 SIM_H값을 산출하기 위해 재귀적으로 하위 토픽들의 SIM_H값을 필요로 함을 알 수 있다. 이러한 재귀적 호출의 부담을 줄이기 위해 계층 구조의 하위에서부터 상위로 SIM_H값을 계산해서 올라가는 방식을 취한다. 즉, 하위 계층의 SIM_H값을 먼저 계산한 다음 이 값을 토픽쌍 행렬에 저장해 둠으로써 상위 계층의 SIM_H값 계산 시에 하위 계층의 SIM_H값을 토픽쌍 행렬로부터 직접 얻을 수 있다.

그림 10은 계층적 유사값을 계산하기 위한 또 다른 예로서 철학자들을 서로 다른 기준에 따라 분류한 철학자 계층 구조를 보이고 있다. 직관적으로 토픽맵 B의 한국 철학자와 중국 철학자는 토픽맵 A의 동양 철학자에 매핑되고 토픽맵 B의 서양근대 철학자는 토픽맵 A의 서양 철학자에 매핑됨을 알 수 있다. 그러나 토픽명 기반 유사도를 볼 때 SIMname(동양 철학자, 한국 철학자)와 SIMname(동양 철학자, 서양근대 철학자)는 유사한 값을 가지게 된다. 그러나 계층 구조 기반 유사도를 보면 SIM_H(동양 철학자, 한국 철학자)는 SIM_H(동양 철학자, 서양근대 철학자)보다 더 큰 값을 가진다. 위 예에서 볼 때 SIM_H(동양 철학자, 한국 철학자)는 하위 토픽 쌍에서 율곡, 이이가 정확히 매핑되기 때문에 0.18값을 가지지만 SIM_H(동양 철학자, 서양근대 철학자)는 매핑되는 하위 토픽들이 없으므로 0값을 가짐을 알 수 있다. 두 토픽맵의 SIM_H(철학자, 철학자)는 그 하위 토픽 쌍들의 SIM_H 값을 모두 계산한 다음 이 값을 이용하여 계산되며 위 예에서 볼 때 하위 토픽들이 구조적으로 유사하기 때문에 0보다 큰 값을 가진다. 그림 11은 계층 구조 기반 매핑 기법의 알고리즘을 보이고 있다.

연관관계 기반 매핑 기법. 연관관계 기반 매핑은 연관관계 타입들 사이의 매핑을 의미한다. 예를 들어, 그림 12의 토픽맵 A에서는 'author' 역할의 칸트와 'book' 역할의 실천이성비판 토픽 사이에 'author of'의 연관관계가 있고 토픽맵 B에서는 'philosophical text' 역할의 순수이성비판과 'write' 역할의 임마누엘 칸트 토픽 사이에 'written by'의 연관관계가 있을 경우, 연관관계 기반 매핑에서는 연관관계 타입 토픽인 두 토픽 'author of' 와 'written by' 사이에 연관관계 정의 측면에서 유사성이 존재하는지 결정하는 것이다.

토픽맵의 연관관계는 대부분 이진관계로서 두 토픽사이의 의미적 연관성을 설정한다. 따라서 두 연관관계 타입이 매핑되기 위해서는 각각의 연관관계 타입으로 연결된 멤버 토픽들 사이의 유사도를 측정하고 그 결과에 따라 연관관계 타입 사이의 유사값을 산출하게 된다.

즉, 'author of'와 'written by' 사이의 연관관계 유사값을 구하기 위해서는 두 연관관계 타입의 멤버 쌍인 (칸트, 순수이성비판), (칸트, 임마누엘 칸트), (실천이성비판, 순수이성비판), (실천이성비판, 임마누엘 칸트) 쌍들 사이의 유사값을 계산한 다음 이를 평균하여 구하게 된다. 이는 멤버들 사이의 유사성이 높을 경우 그 멤버의 연관관계 또한 유사성이 높다는 것을 말한다.

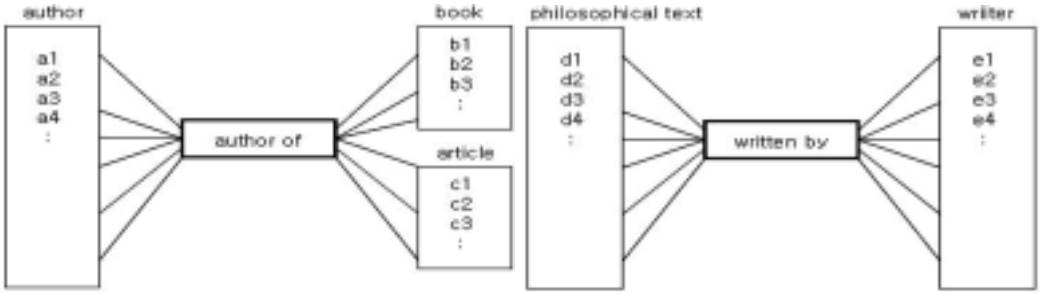
(a) 토픽맵 A의 철학자 계층구조 (b)토픽맵 B의 철학자 계층구조

그림 10 자식토픽들이 전부 클래스토픽인 경우

5.2.3 계층구조 기반 매핑 알고리즘

```
calcSIMHierarchy(TList1:TopicSet, TList2:topicSet, pairSet:TopicMatrix)
// ① 두 토픽집합의 원소 쌍의 유사값을 구한다.
   for(i = 0; i <= TList1.count(); i++) {
        childTopic1 = TList1.getTopic(i);
        for(j = 0; j <= TList2.count(); j++) {
            childTopic2 = TList2.getTopic(j);
// ② 토픽쌍 행렬로부터 두 토픽의 쌍에 대한 주소를 구한다.
            topicPair = pairSet.getPair(childTopic1, childTopic2);
            SIMhierarchy = topicPair.getSIMh();
// ③ 토픽 쌍의 SIMH 값이 계산되어 있지 않은 경우 그 값을 산출한다.
            if NOT_FOUND then
                if isLeaf(childTopic1) and isLeaf(childTopic2) then
                   SIMhierarchy  = 0;
                else if isLeaf(childTopic1) and NOT isLeaf(childTopic2) then
                   topicPair.SIMh = 0;
                else if NOT isLeaf(childTopic1) and isLeaf(childTopic2) then
                   topicPair.SIMh = 0;
                else SIMhierarchy = calcSIMHierarchy(childTopic1, childTopic2, pairSet);
            end if
// ④ 토픽 쌍의 단일 유사값을 계산하고 집합 내의 모든 토픽 쌍의 유사값을 누적한다.
            SIMofPair = (topicPair.getSIMname() + topicPair.getSIMocc + SIMHierarchy)/3;
            accSIMofPairs = accSIMofPairs + SIMofPair;
        }
   }
// ⑤ 최종 계층 구조 유사값을 산출한다.
   SIMHierarchy = accSIMofPairs / (TList1.count() * TList2.count());
```

그림 11 계층구조 기반 기반 매핑 알고리즘



(a) 토픽맵 A의 author of 연관관계 (b)토픽맵 B의 written by 연관관계

그림 12 연관관계를 가지는 두 토픽의 연관관계 구조

그림 13은 연관관계 기반 매핑 기법의 알고리즘을 보이고 있다.

6. 구현 및 실험

본 연구의 이전 연구에서는 토픽맵 기반의 온톨로지 관리 시스템 K-Box를 구현하였다[25]. K-Box 시스템은 온톨로지의 데이타모델로 토픽맵을 사용하기 때문에 토픽맵을 생성하고 변경, 저장하며 키워드 검색 및 주제 검색등을 지원하기 위한 여러 컴포넌트들로 구성된다. K-Box의 컴포넌트들은 토픽맵 객체들의 단일 인터페이스 제공을 위한 토픽맵 오브젝트 래퍼(Topic Map Object Wrappers), 토픽맵 객체들을 저장하기 위한 스토리지 래퍼(Storage Wrappers), 사용자가 토픽맵에 접근할 수 있게 하는 토픽맵 제공자(Topic Map Provider), 대용량의 토픽맵의 효율적인 검색을 위한 토픽맵 캐쉬 관리자(Topic Map Cache Manager), 토픽맵 객체들을 생성하는 토픽맵 생성자(Topic Map Factory), 토픽맵 관리를 위한 토픽맵 관리자(Topic Map Manager), 토픽맵의 가져오기 및 내보내기등 여러 유용한 기능들을 제공하는 토픽맵 도구(Topic Map Utilities)등이 있다.

K-Box 시스템은 클라이언트-서버 구조로 사용자의 토픽맵 접근을 지원하는 프론트-엔드(front-end)와 토

5.2.4 연관관계 기반 매핑 알고리즘

```
assocationTypeMatcher(nameIdx1:NameIndex, nameIdx2:NameIndex, pairSet:TopicMatrix)
    for each pairs of association type topics in pairSet
        SIMmem1 = SIMmem2 = SIMmem3 = SIMmem4 = 0;
// ① 각 연관관계 타입의 멤버 토픽들의 집합을 구한다.
        firstMemberSet1 = getMember(topicPair.topic1, FIRST);
        secondMemberSet1 = getMember(topicPair.topic1, SECOND);
        firstMemberSet2 = getMember(topicPair.topic2, FIRST);
        secondMemberSet2 = getMember(topicPair.topic2, SECOND);
// ② 두 연관관계 타입의 멤버들 사이에 쌍을 생성하고 멤버들 사이의 SIMmem 값을 산출한다.
//     SIMmem1은 firstMemberSet1 집합의 토픽들과 firstMemberSet2 집합의 토픽들 사이에
//     쌍을 생성한 다음 pairSet에서 각 토픽쌍의 SIMname+occ 값을 읽어온다. 여기서 가장 큰
//     SIMname+occ 값을 SIMmem1 값으로 한다.
//     SIMmem2, SIMmem3, SIMmem4도 동일하게 구한다.
        SIMmem1 = calcMemberSimilarity(firstMemberSet1, firstMemberSet2, pairSet);
        SIMmem2 = calcMemberSimilarity(firstMemberSet1, secondMemberSet2, pairSet);
        SIMmem3 = calcMemberSimilarity(secondMemberSet1, firstMemberSet2, pairSet);
        SIMmem4 = calcMemberSimilarity(secondMemberSet1, secondMemberSet2, pairSet);
// ③ 4개의 SIMmem 값의 평균을 구하여 SIMassoc 값으로 한다.
        SIMassoc = (SIMmem1 + SIMmem2 + SIMmem3 + SIMmem4) / 4
        topicPair.SIMassoc = SIMassoc;
```

그림 13 연관관계 기반 매핑 알고리즘

픽맵 엔진 역할의 백-엔드(back-end)로 나누어지며 K-Box API를 제공함으로써 지식 관리 시스템, 콘텐츠 관리 시스템 등의 내부 온톨로지 관리 모듈로 활용될 수 있다. 본 연구에서는 K-Box 시스템의 토픽맵 매핑 관리자(Topic Map Mapping Manager) 모듈을 구현하였다. 매핑 관리자는 PSI 색인과 토픽단어 색인의 생성 및 관리 기능, 토픽쌍의 유사값 계산 기능, 매핑 후보쌍 선택 기능 등을 수행한다.

6.1 실험 데이타

실험을 위한 데이타는 네 그룹으로 A 그룹은 철학 분야에 대하여 동일한 전문가 그룹에 의해 생성된 온톨로지들의 그룹으로서 동양철학 전문가와 서양철학 전문가에 의해 생성된 동양철학, 서양근대철학, 서양현대철학 온톨로지가 있다. 여기서 동일한 전문가 그룹은 동일한 프로젝트에 소속된 서로 다른 전문분야의 연구자들을 의미한다. A 그룹의 온톨로지들은 상위 수준의 철학 지식 분류체계와 참조 토픽들을 공유하고 있으며 각각 인스턴스에 적합한 어커런스와 연관관계를 정의하고 있다.

B 그룹은 철학 분야에 대하여 서로 다른 전문가 그룹에 의해 생성된 온톨로지들의 그룹으로서 야후 코리아 포털에서 제공하는 백과사전에서 서양 근대철학과 현대철학 부분을 토픽맵으로 작성한 온톨로지들을 가진다. C 그룹은 철학외의 다른 지식 분야의 온톨로지로서 야후 코리아 백과사전과 네이버 백과사전에서 독일문학 부분을 토픽맵으로 작성한 온톨로지들을 가진다. D 그룹은 제안된 매핑 알고리즘의 범용성을 보이기 위한 실험 데이타로서 모두 웹상에서 구할 수 있는 토픽맵들이다. 그리고 A, B, C 그룹의 온톨로지는 한글로 작성되었으며 D 그룹의 온톨로지는 영어로 작성되었다. 표 1은 A, B, C 그룹의 실험 데이타의 종류와 특성을 보이고 있으며 표 2는 D 그룹의 실험 데이타의 종류와 특성을 보이고 있다.

표 1에서 동양철학 온톨로지는 한국, 중국, 인도 철학의 지식을 포함하고 있으며 서양근대철학 온톨로지는 칸트, 헤겔, 데카르트 등 근대시대의 철학 지식을 포함한다. 그리고 서양현대철학 온톨로지는 마크르스, 흄, 러셀 등 현대시대의 철학 지식을 포함한다. 온톨로지에 구성된 철학 지식 내용은 철학자, 철학문헌, 철학이론, 철학용어 및 철학 텍스트의 내용 지식으로 구성되어 있다. 그림 14는 A 그룹의 서양근대철학 온톨로지(T_2)와 B 그룹의 야후 근대철학 온톨로지(T_4)의 계층 구조를 보이고 있다.

계층 구조에서 두 온톨로지의 가장 큰 차이점은 서양근대철학 온톨로지의 경우 철학자, 철학문헌 하위에 시대적으로 서양근대철학이 존재하는 것에 반해 야후 근대철학 온톨로지는 근대철학 하위에 철학자와 철학저서가 존재한다는 것이다. 또한 야후 근대철학 온톨로지는 백과사전으로부터 생성된 온톨로지이기 때문에 사전식

표 1 A, B, C 그룹의 온톨로지의 구조적 특성

온톨로지	A 그룹			B 그룹		C 그룹	
	동양철학 온톨로지 (T_1)	서양근대철학 온톨로지 (T_2)	서양현대철학 온톨로지 (T_3)	Yahoo 근대철학 (T_4)	Yahoo 현대철학 (T_5)	Yahoo 독일문학 (T_6)	네이버 독일문학 (T_7)
최대 깊이	11	10	9	5	5	4	4
토픽 수	1748	740	937	95	231	121	210
토픽타입 수	1332	378	582	5	5	8	8
어커런스타입 수	86	56	62	2	2	2	9
연관관계타입 수	47	40	43	2	2	2	2
역할타입 수	22	15	18	2	2	2	2
PSI 수	653	328	345	5	5	8	8

표 2 D 그룹의 온톨로지의 구조적 특성

온톨로지	D 그룹					
	Geography (T_8)	Country (T_9)	UNSPSC71 (T_{10})	UNSPSC11 (T_{11})	Tm-world (T_{12})	XML-tools (T_{13})
최대 깊이	4	3	5	5	3	4
토픽 수	308	393	126	21564	226	603
토픽타입 수	21	3	17	5	23	8
어커런스타입 수	6	0	1	0	21	11
연관관계타입 수	2	0	3	4	13	6
역할타입 수	4	0	2	6	20	7
PSI 수	126	1	114	21499	41	8

(a) 서양근대철학 온톨로지의 계층구조 (b) 서양근대철학 온톨로지의 계층구조

그림 14 실험데이타의 근대철학 온톨로지 구조

나열로 인하여 계층 구조가 단순하다는 점이다.

B 그룹의 야후 철학 온톨로지는 토픽타입, 어커런스 타입, 연관관계 타입 토픽의 수가 적다. 토픽타입의 경우 철학, 서양철학, 근대철학, 철학자, 철학저서의 5개 토픽이 하위 토픽을 가지는 토픽타입이고 나머지 토픽은 하위 토픽을 가지지 않는다. 그리고 토픽의 어커런스 타입은 요약, 설명의 2개 토픽이고 연관관계 타입은 관련항목, 참조항목의 2개 토픽이다. 역할 타입은 관련대상, 참조대상의 2개 토픽이다.

C 그룹의 독일문학 온톨로지들은 문학, 독일문학, 독일문학일반, 시, 희곡, 소설 등 동일한 계층구조를 가지면서 토픽 수에 있어서는 차이를 보인다. 즉, 야후 독일문학에 비해 네이버 독일문학이 더 많은 정보를 제공하고 있으며 토픽의 속성에 있어서도 야후보다 많은 속성들을 가지고 있다. 그러나 독일문학이라는 동일한 지식을 표현하기 때문에 지식 구조, 토픽명, 요약 및 해설 등에 있어서 많은 유사성을 가지고 있다.

표 2에서 Geography.xtm은 주요 국가명과 각각의 주요 도시들의 정보를 가지는 토픽맵이고 country.xtm은 ISO 국가 코드와 표준 국가명을 표현하고 있는 토픽맵이다. UNSPSC71.xtm과 UNSPSC11.xtm은 상품 및 서비스 분류 체계인 UNSPSC(Universal Standard Products and Services Classification) 정보를 가지는 토픽맵으로서 UNSPSC71.xtm은 세그먼트 71 영역의 상품 및 서비스 코드만 가지는데 반하여 UNSPSC11.xtm은 전체 상품 및 서비스 분류코드를 가지고 있다. Tm-world.xtm은 토픽맵 관련 소프트웨어들과 표준 문서, 회사 등의 정보를 가지는 토픽맵이고 이와 유사하게 XML-tools.xtm은 XML 관련 도구, 도서 및 회사 정보를 가지는 토픽맵이다. 여기서 Country. xtm, UNSPSC71.

xtm, UNSPSC11.xtm은 코드 데이타를 토픽맵으로 변환한 것이기 때문에 토픽의 어커런스를 정의하지 않고 있다.

6.2 실험 결과

본 논문에서는 매핑 성능을 평가하기 위한 척도로서 정보 검색에서 사용되는 척도인 정답율(precision), 재현율(recall), 종합율(overall)이 사용된다[9]. 매핑 성능 측정을 위해 먼저 철학 전문가들에 의해 수작업으로 생성된 매핑 집합(R)과 본 논문의 매핑 알고리즘에 의해 생성된 매핑 집합(P)을 비교하여 두 집합의 교집합(I)인 정답 집합(true-positive)와 P - I인 오답 집합(true-negative), 그리고 R - I인 미발견 집합(false negative)을 구한다. 그리고 아래 수식 (4)와 같이 정답율, 재현율, 종합율 값을 구한다. 정답율의 의미는 자동으로 생성된 매핑 집합에서 전문가에 의해 선택될 수 있는 매핑의 정답율이고 재현율은 전문가에 의해 수작업으로 생성된 매핑 집합에서 자동 생성이 정답으로 찾은 매핑 집합의 비율이다. 종합율의 의미는 자동으로 생성된 매핑 집합에서 오답 집합은 제거하고 미발견 집합은 추가하기 위한 보정치를 가리키는 것으로 값이 작을수록 더 많은 보정 작업을 해야 함을 의미한다. 가장 이상적인 경우는 정답율, 재현율, 종합율 모두 1인 경우이다. 표 3과 4는 이들 척도에 의해 산출된 매핑 성능을 보이고 있다.

$$precision = \frac{I}{P} \quad recall = \frac{I}{R}$$

$$overall = recall * \left(2 - \frac{1}{precision}\right) \qquad (4)$$

실험 결과를 볼 때 모든 온톨로지 쌍의 재현율이 80% 이상임을 알 수 있는데 이는 시스템에 의해 자동으로 생성된 매핑 집합이 전문가들이 수작업으로 결정한 매핑 집합을 대부분 포함하고 있음을 가리킨다.

A 그룹의 온톨로지 쌍은 대부분 온톨로지 스키마 차원에서 매핑이 이루어진다. 즉, 이들 온톨로지는 동일한 도메인과 동일한 작업 그룹에 의해 생성된 온톨로지이므로 각 온톨로지들이 동일한 스키마를 참조하고 있기 때문이다. 그러므로 대부분 토픽타입, 어커런스 타입, 연관관계 타입에서 매핑이 이루어지고 인스턴스 토픽들도 철학자, 철학문헌 및 텍스트 내용 토픽들에서 유사한 토픽명과 계층 구조에 의해 매핑이 이루어진다. 동양철학에 비해 서양근대철학 온톨로지(T_2)와 서양현대철학 온톨로지(T_3) 사이의 매핑이 더 높게 이루어지는 이유는 근대와 현대 철학 사이의 철학자 참조 및 텍스트 내용상의 관련성등이 더 높기 때문이다.

B 그룹의 (T_2, T_4)와 (T_3, T_5) 온톨로지 쌍에서는 주로 '칸트', '흄', '마르크스' 등의 철학자나 '법철학' 등의 철학 텍스트, '도덕법칙', '자유주의' 등의 주요 용어 등에서 토픽명이 일치하는 토픽 쌍들 사이에 매핑이 이루어진다. 그러나 그림 16에서 보듯이 T_2, T_4 두 온톨로지의 철학자 토픽은 동일한 토픽명을 가지지만 의미적으로 볼 때 T_4의 철학자 토픽은 T_2의 서양근대철학자와 매핑되어야 한다. 자동 생성에서는 계층 구조 기반 매핑에 의하여 T_2의 서양근대철학자와 T_4의 철학자가 유사한 하위 토픽들을 가지고 있음을 알게 되고 따라서 이 둘 사이에 매핑이 존재함을 보인다. T_2의 서양근대철학 문헌과 T_4의 철학저서도 계층 구조 기반 매핑에 의해 토픽명이 일치하지 않더라도 매핑 결과에 포함된다.

C 그룹의 (T_2, T_6) 온톨로지 쌍의 경우 재현율이 1이 되는 이유는 서로 다른 내용의 온톨로지 사이에 매핑되는 요소의 수가 극히 적고 백과사전이라는 실험 데이타의 특성상 '니체', '법철학' 등 양쪽 온톨로지에서 정확히 명사형으로 일치되기 때문이다. 시스템 매핑 집합(P)이 전문가 매핑 집합(R)보다 많은 7개의 매핑을 가지는 이유는 '독일 종교와 철학의 역사'에서 '독일', '종교', '철학'과 같이 일부 문자열을 공통적으로 가지므로 SIMname 값이 기준 이상 되는 쌍들을 매핑으로 산출하기 때문이다. 종합율이 -0.38인 것은 재현율에 비해 정답율이 너무 낮기 때문이며 이것은 자동으로 생성된

표 4 실험 데이타 D 그룹의 매핑 결과

온톨로지 쌍	(T_8,T_9)	(T_{10},T_{11})	(T_{12},T_{13})
전문가매핑집합(R)	28	121	27
시스템매핑집합(P)	34	147	38
일치매핑집합(I)	28	115	24
정답율	0.82	0.78	0.63
재현율	1	0.95	0.89
종합율	0.78	0.68	0.37

표 3 실험 데이타 A, B, C 그룹의 매핑 결과

온톨로지 쌍	(T_1,T_2)	(T_1,T_3)	(T_2,T_3)	(T_2,T_4)	(T_3,T_5)	(T_2,T_6)	(T_6,T_7)
전문가매핑집합(R)	172	193	211	57	93	3	83
시스템매핑집합(P)	244	277	292	69	116	7	97
일치매핑집합(I)	144	169	196	51	85	3	76
정답율	0.59	0.61	0.67	0.74	0.73	0.42	0.78
재현율	0.84	0.88	0.93	0.89	0.91	1	0.92
종합율	0.26	0.32	0.47	0.58	0.57	-0.38	0.66

매핑 결과를 보정하는데 더 많은 노력이 소모된다는 것을 의미한다. 즉, 서로 다른 도메인의 온톨로지 사이에서의 자동 매핑은 무의미함을 보여주는 것이다.

(T_6, T_7) 온톨로지들은 백과사전으로부터 생성된 온톨로지이므로 기본적으로 토픽명에 있어서 동일성을 보이고 있다. 따라서 시스템에 의해 생성된 매핑들도 대부분 토픽이름 기반 유사성에 의해 결정되었다. 시스템에서 찾지 못한 매핑을 살펴보면 토픽명이 상이하면서 토픽의 속성 또한 많은 차이를 보이는 경우이다. 예를 들어, 야후 독일문학에는 '독일문학일반' 토픽 하위에 '직공(Die Weber)' 토픽이 있고 네이버 독일문학에는 '희곡' 토픽 하위에 '직조공들(Die Weber)' 토픽이 있다. 이 두 토픽은 동일한 대상을 가리키는 토픽이지만 토픽명이 상이하고 속성에 있어서도 야후에서는 요약, 해설의 두 가지 속성뿐이지만 네이버에서는 대본작가, 국적, 구성, 초연연출가, 초연연월 등 9가지의 속성을 가진다. 이 경우 시스템의 매핑 알고리즘의 실행결과 매핑 선택 기준값보다 낮은 속성기반 유사값을 산출하므로 최종적으로 매핑 결과에서 누락시키게 된다.

D 그룹의 데이타도 A, B, C 그룹의 데이타와 비슷한 결과를 보인다. 이는 유사값을 산출하기 위한 알고리즘이 특수한 데이타 형식이나 값에 의존하지 않음을 보인다. 예를 들어, T_{12} 온톨로지에서는 'Standard' 토픽 아래에 'HTML' 토픽이 정의되어 있고 T_{13} 온톨로지에서는 토픽의 속성을 기술하기 위해 어커런스 타입으로만 사용되는 'HTML representation' 토픽이 정의되어 있다. 이 두 토픽은 토픽명만 비교할 경우 유사값이 기준값을 초과하기 때문에 시스템매핑집합(P)에 포함되어야 하지만 본 논문의 제안 알고리즘에서는 토픽 타입과 어커런스 타입 사이에는 매핑 쌍을 생성하지 않기 때문에 이 두 토픽은 매핑되지 않는 것으로 결정된다.

시스템에서 매핑을 결정하지 못하는 미발견 매핑집합(R – I)에는 대부분 도메인 전용의 의미사전이 필요한 경우이다. 일반적인 범용 용어의 시소러스(thesaurus)로서 한글 워드넷(WordNet)과 영어 워드넷 등이 있지만 철학 용어 '이성'과 '오성', '인식'과 '인지', '업'과 '행위' 등의 매핑을 자동적으로 찾기 위해서는 표준 철학 용어 시소러스가 정의되어 있어야 한다. 이러한 시소러스가 구축되고 접근 가능할 경우 본 논문의 토픽명 기반 매핑 기법에서는 단순한 단어 매핑뿐만 아니라 시소러스를 통한 동의어, 확장어 등의 의미 관계를 활용한 매핑도 수행할 수 있다.

그림 15는 제안된 매핑 알고리즘들을 개별적으로 적용할 경우 각각의 매핑 결과에 대한 정답율, 재현율, 종합율을 보이는 그래프이다. 이 그래프에서는 4가지 알고리즘의 조합을 보이고 있다. Name은 토픽이름 기반 매

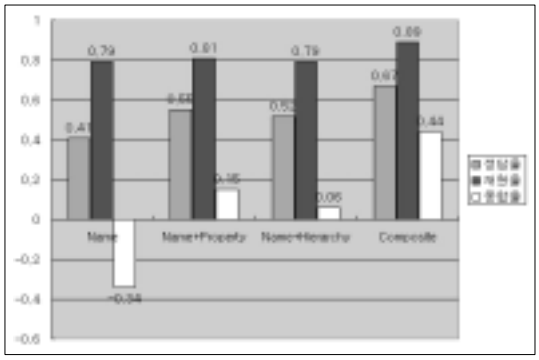


그림 15 개별 매핑 기법의 평균 정답율, 재현율, 종합율 그래프

핑 알고리즘만 수행한 경우이고 Name+Property는 토픽이름 기반 및 속성 기반 매핑 알고리즘을 조합하여 수행한 경우이다. Name+Hierarchy는 토픽이름 기반 및 계층 구조기반 매핑 알고리즘을 조합하여 수행한 경우이고 Composite은 모든 매핑 알고리즘을 조합하여 적용한 경우이다. 연관관계 기반 매핑의 경우 오직 연관관계 타입의 토픽들 사이에 매핑을 결정하기 위한 알고리즘이기 때문에 성능 비교에 적합하지 않으므로 생략되었다.

그래프의 성능 측정값들을 보면 대체적으로 비슷한 재현율을 보이는 대신 정답율과 종합율에 있어서 차이를 보임을 알 수 있다. 즉, 토픽이름 기반 매핑 알고리즘만 적용한 경우 시스템에 의해 자동 생성된 매핑 집합에서 오류 매핑이 많은 부분을 차지하기 때문에 정답율과 종합율이 낮은 값을 가지게 된다. 예를 들어, (철학자, 철학자), (철학자, 근대 철학자), (철학자, 동양 철학자), (철학자, 현대 철학자) 등이 전부 매핑 집합에 포함된다.

Name+Property와 Name+Hierarchy는 Name에 비해 높은 정답율과 종합율을 가지는데 이는 토픽이름으로만 유사값을 산출하여 매핑을 결정하는 것에 비해 보다 정확하게 매핑을 결정하도록 매핑 선택을 제한하는 역할을 수행하기 때문이다. 예를 들어, (칸트 이성, 헤겔 이성)의 경우 Name에서는 매핑되는 것으로 결정하는데 반하여 Name+Property에서는 이 두 토픽의 속성을 비교함으로써 낮은 속성 유사값에 따라 매핑이 안되는 것으로 결정한다. Name+Hierarchy도 마찬가지로 Name과 달리 (철학자, 근대 철학자) 또는 (철학자, 현대 철학자) 사이에서만 매핑이 있는 것으로 결정한다.

이러한 실험 결과로 볼 때 단일 특성만을 고려한 매핑 알고리즘은 불필요한 매핑 결과를 산출하는 문제점을 보이므로 보다 정확한 매핑 결과를 얻기 위해서는

온톨로지 모델의 여러 가지 특성을 조합한 다중 기법 매핑 알고리즘의 적용이 필요함을 알 수 있다.

7. 결론 및 향후 연구

본 논문에서는 토픽맵 기반의 두 온톨로지 사이에 매핑 토픽쌍을 결정하기 위해서 적용 가능한 4가지의 매핑 기법들을 제시하였고 토픽맵의 구문적 특성과 제약조건에 의하여 매핑 탐색 범위를 줄일 수 있음을 보였다. 또한 이전 연구에서 구현한 토픽맵 관리 시스템인 K-Box 시스템의 추가 모듈로 토픽맵 매핑 관리자 모듈을 구현하였으며 현재 철학 전문가들에 의해 구축중인 철학 온톨로지 및 야후 백과사전의 철학 사전을 대상으로 실험을 하였다. 실험의 결과 매핑 모듈에 의해 자동적으로 계산된 매핑 후보들의 수가 전문가에 의해 계산된 매핑의 수보다 많게 산출됨으로써 정답율은 50%~70% 정도로 낮지만 전문가의 수작업 매핑과 일치하는 자동 생성 매핑 비율인 재현율은 80% 이상임을 알 수 있다.

정답율을 높이기 위해서는 매핑 후보를 선택하는 과정에서 기준값을 초과하는 유사값을 가지는 모든 쌍을 선택하는 대신 재현율에 영향을 주지 않으면서 정답율을 높일 수 있도록 불필요한 매핑을 제거할 수 있는 필터링 기법이 필요하다. 기본적으로 1:1 매핑의 경우 각 토픽에 대해 최대 유사값을 가지는 하나의 토픽쌍만을 선택하게 되는데, 이 경우 정답율은 높은 반면 재현율이 낮아지게 된다. 정답율을 높이기 위해 향후연구에서는 다의어 매핑을 위한 철학 의미사전의 구축과 함께 가변적 가중치를 적용한 필터링, 유사값의 가변 범위 필터링 기법 등을 다루고자 한다. 가중치를 적용한 필터링은 개별 매핑 기법에 가변적인 가중치를 적용하여 높은 유사값을 가지는 매핑 기법의 매핑 후보를 선택하는 것이고 가변 범위 필터링은 비슷한 유사값을 가지는 토픽쌍들에서 일정 범위 내에 속하는 매핑 후보를 선택하는 기법이다.

계산된 매핑 후보들은 두 온톨로지의 통합이나 링크에 사용된다. 즉, 매핑 후보의 두 토픽을 통합하여 하나의 토픽으로 만들거나 아니면 두 토픽을 연결하는 연관관계를 정의하는 것이다. 또한 온톨로지 기반의 검색 에이전트가 여러 온톨로지에서 서로 관련된 지식을 검색하는 데에도 사용된다. 후속 연구에서는 매핑 후보들의 통합이나 연결에 있어서 토픽들 사이의 충돌 문제와 통합 온톨로지의 갱신 문제를 해결하고자 한다. 또한 여러 버전의 온톨로지들 사이의 통합 문제 또한 다루고자 한다.

본 논문에서 다루고 있는 토픽맵 매핑 기법들은 RDF와 OWL에도 적용이 가능하다. 이는 토픽맵과 RDF가 상호 호환이 가능하기 때문이다. OWL은 RDF나 토픽맵에 비해 더 많은 제약조건을 정의할 수 있으므로 OWL 온톨로지에 적합한 매핑 기법의 구현도 의미 있는 연구 주제이다.

참 고 문 헌

[1] T. Berners-Lee, J. Hendler, and O. Lassila. "The Semantic Web," Scientific American, 279, 2001.

[2] Ora Lassila and Ralph R. Swick. "Resource Description Framework(RDF) Model and Syntax Specification," W3C Recommendation 22 February 1999, URL:http://www.w3.org/TR/REV-rdf-syntax.

[3] Michel Biezunski, Martin Bryan and Steve Newcomb. ISO/IEC 13250 TopicMaps.

[4] D. L. McGuinness and F. Harmelen. "OWL Web Ontology Language Overview," W3C Recommendation, 10 February 2003, http://www.w3.org/TR/owl-features/.

[5] 김정민, 최병일, 김형주. "텍스트 내용지식 기반의 철학 온톨로지 구축", 정보과학회논문지(컴퓨팅의 실제), 게재예정.

[6] Steve Pepper and Graham Moore. "XML Topic Maps(XTM) 1.0," TopicMaps.Org.

[7] L. M. Garshol, "Living with Topic Maps and RDF," In Proceedings of the XML Europe 2003 Conference, 2003.

[8] E. Rahm and P. Bernstein. "On Matching Schemas Automatically," VLDB Journal, 10(4), 2001.

[9] H. H. Do and E. Rahm. "COMA - a system for flexible combination of schema matching approaches," In Proceedings of VLDB, 2001.

[10] J. Madhavan, P. Bernstein, and E. Rahm. "Generic Schema Matching with Cupid," In Proceedings of VLDB, 2001.

[11] AH. Doan, P. Domingos, and A. Halevy. "Reconciling schemas of disparate data sources: a machine-learning approach," In Proceedings ACM SIGMOD Conference, 2001.

[12] D. Beneventano, S. Bergamaschi, F. Guerra, and M. Vincini. "The MOMIS approach to Information Integration," IEEE and AAAI International Conference on Enterprise Information Systems, 2001.

[13] W. Li, C. Clifton, and S. Liu. "Database Integration using neural network: implementation and experiens," Knowledge Information Systems, 2(1), 2000.

[14] S. Melnik, H. Garcia-Molina, and E. Rahm. "Similarity flooding: A versatile graph matching algorithm," In Proceedings of ICDE, 2002.

[15] F. Giunchiglia and P. Shvaiko. "Semantic matching," In The Knowledge Engineering Review Journal, 18(3), 2004.

[16] P. Shvaiko and J. Euzenat. "A survey of schema-based matching approaches," University of Trento,

Technical Report #DIT-04-087, 2004.
[17] N. Noy and M. Musen. "PROMPT: Algorithm and Tool for Automated Ontology Merging and Alignment," In Proceedings of the National Conference on Artificial Intelligence(AAAI), 2000.
[18] N. Noy and M. Musen. "Anchor-PROMPT: Using Non-Local Context for Semantic Matching," In Proceedings of the Workshop on Ontologies and Information Sharing at the International Joint Conference on Artificial Intelligence(IJCAI), 2001.
[19] P. Bouquet, L. Serafini, and S. Zanobini. "Semantic coordination: A new approach and an application," In Proceedings of ISWC, 2003.
[20] Y. Kalfoglou and M. Schorlemmer. "Information-Flow-based Ontology Mapping," In Proceedings of the 1st International Conference on Ontologies, Database and Application of Semantics, 2002.
[21] G. Stumme and A. Madche. "FCA-Merge: Bottom-up Merging of Ontologies," In Proceedings of 17th Internaltional Joint Conference on Artificial Intelligence(IJCAI), 2001.
[22] M. Ehrig and S. Staab. "QOM: Quick ontology mapping," In Proceedings of ISWC, 2004.
[23] ISO/IEC JTC1/SC34. "Topic Maps - Reference Model," Available at: http://www.isotopicmaps.org/TMRM/TMRM-latest-clean.html, 2003.
[24] L. Maicher and H. F. Witschel. "Merging of Distributed Topic Maps based on the Subject Identity Measure(SIM) Approach," In Proceedings of LIT, 2004.
[25] 김정민, 박철만, 정준원, 이한준, 민경섭, 김형주, "K-Box: 토픽맵 기반의 온톨로지 관리 시스템", 정보과학회논문지(컴퓨팅의 실제), 10(1), February 2004.

김 정 민
1992년 홍익대학교 전자계산학과 졸업(학사). 1994년 홍익대학교 전자계산학과 졸업(석사). 2002년 서울대학교 전기, 컴퓨터공학부 박사과정 수료. 관심분야는 Semantic Web, Ontology, IR, Database

신 효 필
1988년 서울대학교 언어학과 학사. 1990년 서울대학교 언어학과 석사. 1994년 서울대학교 언어학과 박사. 1997년 University of Missouri-Kansas City, Computer Science 석사. 1998년 1월~2001년 1월 Researcher, CRL (Computing Research Laboratory), New Mexico State University, USA. 2001년 1월~2001년 12월 Senior Researcher, YY Technologies in Silicon Valley, USA. 2001년 9월~2003년 2월 서울대학교 공과대학 전기공학부 계약조교수. 2003년 3월~현재 서울대학교 인문대학 언어학과 조교수. 관심분야는 자연언어처리, 온톨로지, 정보검색

김 형 주
1982년 서울대학교 전산학과(학사). 1985년 미국 텍사스 대학교 대학원 전산학(석사). 1988년 미국 텍사스 대학교 대학원 전산학(박사). 1988년 5월~1988년 9월 미국 텍사스 대학교 POST-DOC 1988년 9월~1990년 12월 미국 조지아 공과대학 조교수. 1991년~현재 서울대학교 컴퓨터공학부 교수