토픽 분할에 의한 토픽맵 매칭 및 통합 기법

김 정 민[†] · 정 현 숙[††]

요 약

본 논문에서는 토픽맵의 모델 특성을 고려한 토픽맵 매칭 및 통합 기법을 제안한다. 이전까지의 대부분의 스키마 매칭 연구들은 계산 시간의 효율성을 고려하지 않고 매칭 기법의 범용성 및 정확성을 높이기 위한 목적으로 개발되어 왔다. 그러나 현재 표준적인 온톨로지 언어로 RDF/OWL과 토픽맵이 사용되고 있으며 앞으로 많은 온톨로지들이 이들 언어로 구현될 것이다. 따라서 본 논문에서는 토픽맵 데이터 모델의 구조적 특성 및 제약조건을 고려하여 토픽 분할, 토픽명기반 매칭연산, 속성기반 매칭연산, 계층구조기반 매칭연산, 연관관계기반 매칭연산 및 통합 알고리즘을 개발함으로써 효과적이면서 효율적인 토픽맵 매칭 및 통합이 가능함을 보인다.

키워드 : 토픽맵 매칭, 토픽맵 통합, 온톨로지 통합, 토픽 분할

Topic maps Matching and Merging Techniques based on Partitioning of Topics

Jung-Min Kim[†] · Hyun-Sook Chung[††]

ABSTRACT

In this paper, we propose a topic maps matching and merging approach based on the syntactic or semantic characteristics and constraints of the topic maps. Previous schema matching approaches have been developed to enhance effectiveness and generality of matching techniques. However they are inefficient because the approaches 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. Now, standard languages for developing ontologies are RDF/OWL and Topic Maps. In this paper, we propose an enhanced version of matching and merging technique based on topic partitioning, several matching operations and merging conflict detection.

Key Words : Topic Map Matching, Topic Maps Merging, Ontology Merging, Topic Partitioning

1. 서 론

온톨로지를 포함한 스키마 매칭은 데이터 통합, 데이터 웨어하우스, 시맨틱 웹, 전자상거래 등 다양한 분야에서 두 스키마의 요소들 사이에 의미적 대응 관계를 찾기 위해 요구되는 중요한 프로세스이지만 효과적이고 효율적인 매칭 기법을 개발하는 것은 상당히 어려운 문제로서 실제적으로 대부분의 응용 분야에서는 스키마 전문가에 의한 수작업으로 매칭이 이루어지고 있다[1, 2].

스키마 매칭이 어려운 이유는 두 스키마 사이의 구문적 차이, 의미적 차이, 계산의 복잡성 등에서 찾을 수 있다[3]. 구문적 차이는 두 스키마가 서로 다른 관점에서 설계됨으로써 발생하는 문제로서 데이터 모델, 요소명, 구조 등이 상이하기 때문에 발생한다. 의미적 차이는 두 스키마의 요소들에 내재되어 있는 의미를 유추하기에는 요소명, 데이터형, 요소값, 데이터 구조 등의 실마리만으로 해결이 어렵기 때문에 발생한다.

계산의 복잡성은 두 스키마의 모든 요소들 사이에 최소 한번의 매칭 연산이 수행됨으로 인해서 발생하는 문제로서 스키마 S_1의 요소 s_i가 스키마 S_2의 s_j와 가장 잘 대응된다는 것을 알기 위해서는 s_i와 스키마 S_j의 모든 요소들 사이에 매칭 연산을 수행한 다음 가장 높은 유사값을 가지는 것이 s_j라는 결과를 얻어야 하는 것이다. 최근의 스키마 매칭 기법들은 상당히 효과적이지만 이러한 어려움들을 극복하고 효율성과 정확성을 유지하도록 스키마 매칭의 품질을 높이는 것은 여전히 연구가 필요한 분야이다[4].

온톨로지 매칭은 두 온톨로지에서 의미적으로 대응되는 요소들을 찾는 프로세스로서 스키마 매칭 기법들을 이용할 수 있으나 몇 가지 차이점을 가진다. 온톨로지의 요소명은 대부분 개념 용어이므로 축약어, 기호, 숫자 대신 명사어, 명사구 등으로 정의되어 있다. 또한 온톨로지 언어로 RDF/OWL[5]과 토픽맵(Topic Maps)[6]이 표준으로 사용되고 있으며 이들 언어는 온톨로지 표현 및 생성을 위한 구문, 의미, 제약조건 등을 정의하고 있다. 따라서 온톨로지 매칭에서는 효과적인

※이 논문은 2006년도 조선대학교 학술연구비의 지원을 받아 연구되었음.
† 정 회 원 : 서울대학교 언론정보연구소 박사후연구원
†† 정 회 원 : 조선대학교 컴퓨터공학부 전임강사 (교신저자)
논문접수 : 2007년 9월 7일, 심사완료 : 2007년 10월 15일

매칭 결과를 얻기 위해 이들 언어의 구조적 특징 및 제약조건 등을 고려해야 한다[7].

본 논문에서는 토픽맵 구문으로 생성된 온톨로지들 사이의 효과적이고 효율적인 매칭 및 통합 기법을 제안한다. 본 논문의 토픽맵 매칭 및 통합 기법의 특징은 다음과 같이 요약할 수 있다. 첫째, 계산 시간의 효율성을 높이기 위해 이전의 스키마 매칭 연구들과는 달리 스키마를 그래프로 변환하는 과정을 생략한다. 둘째, 토픽분할을 통해 대응 가능성이 낮은 요소들을 미리 제외시킴으로써 매칭 계산의 복잡성을 줄인다. 토픽분할은 모델기반 토픽분할과 개념기반 토픽분할로 나누어진다. 셋째, 개념 용어의 효과적인 매칭 처리를 위해 토큰 또는 형태소 분석 기반의 토픽명 유사값을 산출한다. 또한 토픽 속성, 계층 구조 및 연관관계 매칭 기법을 복합적으로 적용한다. 넷째, 토픽맵 통합시에 발생 가능한 충돌의 유형을 분류하고 탐지 및 해결하는 기법을 제안한다.

본 논문의 구성은 다음과 같다. 2장에서는 스키마 매칭과 온톨로지 매칭의 관련 연구들에 대해 살펴보고 3장에서는 토픽맵 매칭 및 통합 문제와 프로세스를 정의한다. 4장에서는 토픽맵 매칭 및 통합의 단위 연산들을 정의하고 5장에서는 토픽맵 통합 충돌 유형을 분류하고 탐지 기법을 정의한다. 6장에서는 실험을 통해 제안된 방법의 성능을 보임과 동시에 계산 복잡도를 분석한다. 7장에서는 결론 및 향후 연구 방향을 제시한다.

2. 관련연구

본 연구와 관련된 연구로서 먼저 스키마 매핑과 온톨로지 매핑이 있다. 스키마 매핑은 두 스키마의 엘리먼트들 사이에 의미적 유사성을 찾는 것으로 전자상거래, 데이터 웨어하우스, 데이터 통합 등 여러 응용 분야에서 필요로 하고 있다[8]. 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]. SemInt[9]의 경우 관계형 데이터베이스 스키마의 애트리뷰트들 사이의 유사성을 찾는 엘리먼트 수준의 접근법을 사용하고 있으며 매핑 기법도 데이터 형, 데이터 길이, 키 정보 등의 구조적 유사성에 기인하고 있다. 이와 달리 Cupid[8]의 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[10]. 온톨로지 매핑 및 통합과 관련된 연구로는 PROMPT[11], Anchor-PROMPT[12], Ctx-Match[13], Information flow[14], FCA-Merge[15], QOM[16] 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.

본 연구와 직접적으로 관련을 가지는 토픽맵 매칭 기법인 SIM[17]에서는 구문적 접근법과 부분적인 구조적 접근법으로 토픽쌍의 유사값을 계산하고 매핑을 결정할 수 있음을 보인다. 그러나 SIM에서는 단순히 토픽명과 어커런스 데이터만을 비교하여 유사값을 계산하고 있으며 토픽맵 모델의 특성을 고려하지 않고 모든 토픽들을 비교 대상으로 하고 있다.

3. 토픽맵 매칭 및 통합 문제 정의

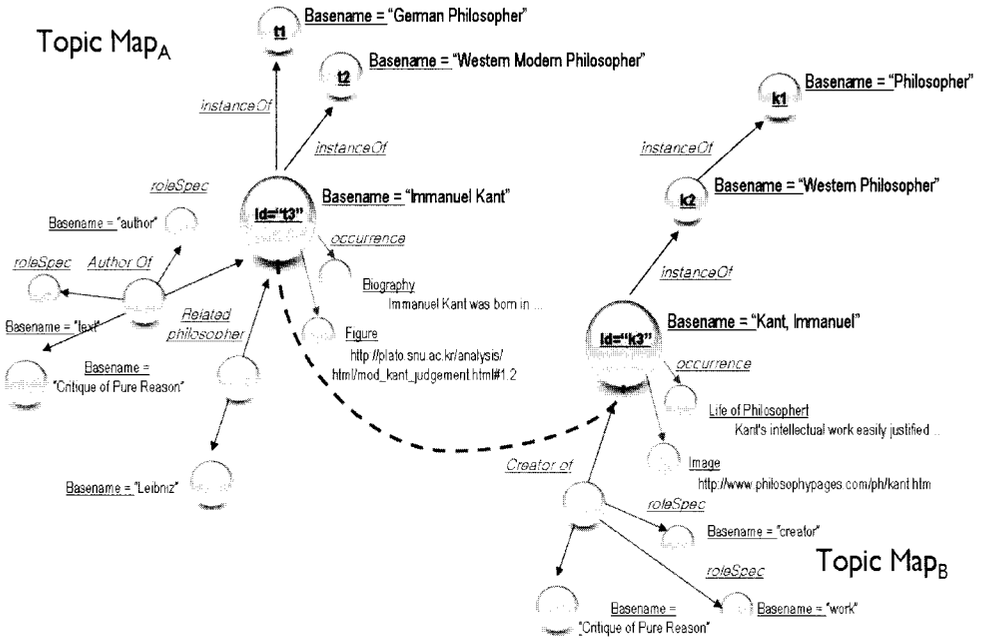
3.1 토픽맵 매칭 사례

효과적이고 효율적인 토픽맵 매칭 및 통합 기법을 정의하기 위해서는 먼저 토픽맵 매칭 및 통합 문제에 대한 이해와 정형화된 정의가 필요하다. (그림 1)은 토픽맵 매칭의 사례로서 철학 관련 토픽맵의 일부분을 보이고 있다. 토픽맵 A에서 ID값이 t3인 노드는 인스턴스 토픽으로서 토픽명이 Immanuel Kant이고 Biography, Figure 등의 속성을 가진다. 또한 부모 클래스 토픽으로 각각 ID값이 t1, t2이면서 토픽명이 German Philosopher, Western Modern Philosopher인 노드를 참조하고 있다. AuthorOf와 RelatedPhilosopher는 연관관계 토픽으로서 t3 토픽과 다른 토픽들 사이에 의미적 연결망을 형성하고 있다.

토픽맵 B에서는 ID값이 k3이면서 토픽명이 Kant, Immanuel인 토픽이 인스턴스 토픽으로서 Life of Philosopher, Image 등의 속성을 가지고 있으며 부모 클래스로 Western Philosopher를 참조하고 있다. 이 두 토픽맵의 요소들 사이에 매칭 여부를 판단하기 위해서는 (그림 2)와 같이 매칭 가능한 요소들 사이에 유사값을 산출하여 최대값을 가지는 요소쌍을 선택한다.

3.2 토픽맵 매칭 및 통합 문제 정의

토픽맵은 지식 도메인의 개념을 정의하고 개념들 사이의 지식 구조를 기술하기 위해 토픽(Topic), 어커런스(Occurrence), 연관관계(Association) 등을 중심으로 여러 구문 요소들을 가지고 있다. 따라서, 토픽맵 모델은 <C, I, O, A, R, R_H, R_A>의 7-튜플로 정의할 수 있다. 토픽맵에서 토픽은 정의하는 대상 개념의 역할에 따라 클래스 토픽(Class Topic), 인스턴스 토픽(Instance Topic), 어커런스 토픽(Occurrence Topic), 연관관계 토픽(Association Topic), 역할 토픽(Role Topic)로 분류할 수 있다. 이들 각각은 C, I, O, A, R로 표시된다. 클래스 토픽 및 인스턴스 토픽들 사이의 관계는 부모-자식 클래스(superclass-subclass) 및 클래스-인스턴스



(그림 1) 매칭이 가능한 철학 관련 지식 토픽맵의 예제

Line#	EntityType	Entity in TM_A	Entity in TM_B	Similarity Expression
1	Occurrence	Biography	Life of Philosopher	$SIM_{name} + SIM_{occ}$
2	Occurrence	Biography	Image	$SIM_{name} + SIM_{occ}$
3	Occurrence	Figure	Life of Philosopher	$SIM_{name} + SIM_{occ}$
4	Occurrence	Figure	Image	$SIM_{name} + SIM_{occ}$
5	Role	author	creator	$SIM_{name} + SIM_{occ}$
6	Role	author	work	$SIM_{name} + SIM_{occ}$
7	Role	text	creator	$SIM_{name} + SIM_{occ}$
8	Role	text	work	$SIM_{name} + SIM_{occ}$
9	Association	Author of	Creator of	$SIM_{name} + SIM_{occ} + SIM_{ass}$
10	Topic	Immanuel Kant	Kant, Immanuel	$SIM_{name} + SIM_{occ}$
11	Topic	Immanuel Kant	Western Philosopher	$SIM_{name} + SIM_{occ}$
12	Topic	Immanuel Kant	Philosopher	$SIM_{name} + SIM_{occ}$
13	Topic	WesternModern Philosopher	Kant, Immanuel	$SIM_{name} + SIM_{occ}$
14	Topic	WesternModern Philosopher	Western Philosopher	$SIM_{name} + SIM_{occ} + SIM_{H}$

(그림 2) 매칭 가능한 요소쌍과 유사값 산출을 위한 연산식

(class-instance)의 계층적 관계와 연관관계 토픽에 의해 정의되는 의미 관계(semantic relation)로 분류된다. 이들 각각은 R_H, R_A로 표시된다.

이전까지의 스키마 매칭과 달리 계산의 복잡도를 줄이기 위해 토픽맵 매칭에서는 구조기반 및 개념기반 토픽 분할에 의해 각 유형별 토픽들 사이에서만 매칭 함수를 적용한다.

예를 들어,description, biography, korean name 등과 같이 클래스 토픽의 속성 정의를 위해 사용되는 토픽들과 authorOf, createBy, affectedFrom 등과 같은 연관관계 정의를 위해 사용되는 토픽들 사이에는 서로 상이한 역할 및 용도로 인하여 매칭 가능성이 낮다. 정의 1은 모델기반 토픽 분할에 의한 토픽맵 매칭 문제를 정형화된 형식으로 정의하고 있다.

정의 1(토픽맵 매칭). 두 토픽맵의 요소 집합 A, B와 도메인 용어 사전 D가 주어졌을 때 토픽맵 매칭 함수 f는 다음과 같이 정의된다. 여기서 토픽맵 요소 집합 A와 B는 토픽맵 모델 정의에 따라 다음과 같이 부분집합들을 원소로 가진다.

$$A = \{C_1, I_1, O_1, A_1, R_1\},\ B = \{C_2, I_2, O_2, A_2, R_2\}$$
$$f(A, B, E) = (f_1(C1, C2, E), f_2(C1, I2, E), f_3(I1, C2, E), f_4(I1, I2, E), f_5(O1, O2, E), f_6(A1, A2, E), f_7(R1, R2, E))$$
$$e_1 \in A,\ e_2 \in B \text{ 이라고 한다면, } f(e_1, B, E) = e_2 \quad (1)$$

매칭연산의 결과는 <ID, e_i, e_j, s_1, s_2, s_3, s_4, S>의 8-튜플로 정의된다. ID는 매칭 식별자이고 e_i는 토픽맵 A의 요소이며 e_j는 토픽맵 B의 요소를 가리킨다. s_1, s_2, s_3, s_4는 4가지 매칭 연산(토픽명기반 매칭연산, 속성기반 매칭연산, 계층구조기반 매칭연산, 연관관계기반 매칭연산)에 의해 산출된 두 요소간의 유사값으로서 0과 1사이의 값을 가진다. 0인 경우 두 요소 사이에 유사성이 전혀 없다는 것이고 1인 경우 두 요소는 완전히 일치한다는 것을 의미한다. S는 4가지 유사값을 조합한 단일값으로서 두 요소의 실제적인 유사정도를 나타낸다.

개념기반 토픽분할은 동일한 개념을 가리키는 토픽들을 하나의 그룹으로 묶거나 완전히 일치하는 토픽들을 매칭 연산 과정에서 제외시킴으로써 계산의 복잡도를 줄이는 것을 의미한다. 토픽맵 모델에서 *Subject Identity*는 특정 토픽이 정의하고 있는 개념에 대한 식별자를 기술하기 위해 사용된다. 만일 두 토픽이 동일한 *Subject Identity*를 가지고 있다면 두 토픽은 토픽명이나 토픽 구조가 서로 다르더라도 동일한 개념에 대해 정의하고 있는 것이므로 매칭 된다고 볼 수 있다.

토픽맵 통합 프로세스는 두 토픽맵의 요소들 사이에 합집합을 구하는 연산과 같다. 따라서 정의 2에서는 합집합에 기반한 두 토픽맵 통합 문제를 정의하고 있다.

정의 2(토픽맵 통합). 여러 토픽맵들의 집합 S가 주어졌을 때 통합 함수 g는 다음 수식 2와 같이 정의된다.

$$g:(S \times S) \rightarrow S \tag{2}$$

만일 토픽맵 A와 B가 집합 S의 원소이고 이 두 토픽맵의 매칭 연산 결과 집합 M이 주어졌을 경우 통합 함수 g는 두 토픽맵 A와 B 요소들 사이의 합집합 연산으로 정의된다.

$$g(A, B, M) = \{e_i \mid e_i \in A\} \cup \{e_j \mid e_j \in B\} \tag{3}$$

3.3 토픽맵 매칭 및 통합 프로세스 구조

토픽맵 매칭 프로세스의 흐름은 기본적으로 스키마 매칭 프로세스의 흐름을 따른다. 일반적인 스키마 매칭 프로세스는 모델 변환 단계, 매칭 연산 단계, 매칭 결정 단계로 세분화된다. 모델 변환 단계에서는 입력된 두 스키마를 범용 구조인 그래프 모델로 변환하는 단계이고 매칭 연산 단계에서는 두 그래프의 요소인 노드와 간선들 사이에 유사값을 산출하는 단계이다. 매칭 결정 단계에서는 유사값이 산출된 요소쌍들 중에서 일정 기준 이상의 높은 유사값을 가지는 요소쌍들을 추출하여 매칭 결과를 생성하는 단계이다. 본 논문의 토픽맵 매칭에서는 모델 변환 단계를 거치지 않고 (그림 3)에서 보듯이 매칭 연산 단계와 매칭 결정 단계의 간략화된 프로세스로 처리된다.

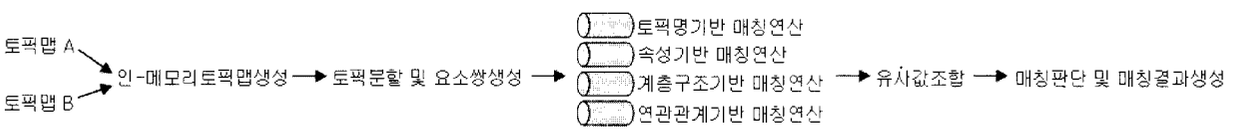
XTM 구문 형식으로 입력된 두 토픽맵을 파싱(parsing)하는 과정에서 메모리내에 토픽맵을 생성하고 매칭 요소들을 분류한다. 다음으로 매칭 대상 토픽쌍들을 생성하고 각 쌍에 대해 4가지 매칭연산을 적용하여 유사값들을 산출한다. 이들 유사값들은 아래 연산식에 의해 하나의 단일 유사값으로 조합된다. 매칭 판단 및 매칭 결과 생성 단계에서는 각 쌍들의 단일 유사값을 큰 값에서 작은 값 순으로 정렬한 다음 일정 경계값(threshold) 이상인 값들 중에서 최대값으로부터 일정 거리(displacement)내에 속하는 쌍들을 매칭 결과로 생성한다. 경계값과 일정 거리를 조합한 결정 방법의 장점은 하나의 토픽과 대응되는 토픽들이 하나 이상인 경우 이들을 매칭 결과에 포함할 수 있다는 것이다.

매칭 결과에 기반한 토픽맵 통합 프로세스의 흐름은 (그림 4)와 같이 정의된다. 토픽맵 통합에서는 매칭되는 두 요소를 하나의 단일 요소로 생성하는 과정에서 상이한 구조로 인해 발생할 수 있는 통합 충돌의 유형을 정의하고 탐지 및 해결하는 단계를 포함하고 있다. 매칭요소쌍 선택, 통합충돌 탐지, 충돌해결 및 통합연산은 매칭 요소쌍이 존재하는 동안 반복적으로 수행된다.

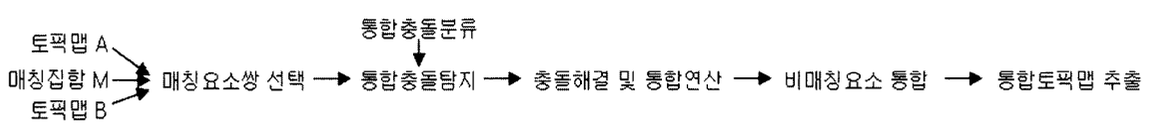
4. 토픽맵 매칭 및 통합 연산

4.1 토픽맵 매칭 연산

매칭 알고리즘은 요소들 사이의 유사값을 계산하기 위해 사용되는 데이터와 계산 방법에 따라 개별 매칭 연산자(individual matcher), 복합 매칭 연산자(composite matcher), 규칙기반 매칭(rule-based matching), 학습기반 매칭(learner-based matching) 등으로 분류할 수 있다.[2] 본 논문의 토픽



(그림 3) 토픽맵 매칭 프로세스



(그림 4) 토픽맵 통합 프로세스

〈표 1〉 토픽맵 매칭 연산자와 연산식

매칭 연산자	연산식
토픽명기반 매칭 연산자(SIMname)	$SIM_{substring}(x, y) = 2\|c\| / \|x\| + \|y\|$ $SIM_{string}(a, b) = \sum SIM_{substring}(x_i, y_j) / \| a \cup b \|$ $SIM_{name}(t_1, t_2) = (SIM_{dict}(t_1.names, t_2.names)+SIM_{string}(t_1.names, t_2.names)) / 2$
속성기반 매칭 연산자(SIMocc)	$SIM_{occ}(t_1, t_2) =$ $\sum(SIM_{occtype}(t_1.occurrence_i, t_2.occurrence_j) \times SIM_{occvalue}(t_1.occurrence_i, t_2.occurrence_j) / \|m\| \times \|n\|,$ $for\ 1 \le i \le m\ and\ 1 \le j \le n$
계층구조기반 매칭 연산자(SIMH)	$SIM_H(t_1, t_2) =$ $(1-w)(\sum(SIM_{name+occ}(t_1.parent_i, t_2.parent_j))/\|x\| \times \|y\|) + w(\sum(SIM(t_1.child_k, t_2.child_l))/\|x'\| \times \|y'\|)$
연관관계기반 매칭 연산자(SIMassoc)	$SIM_{assoc}(t_1,t_2)=\sum SIM(m_i,m_j) \cdot SIM(r_i,r_j)/\|M\| \times \|N\|,\ for\ 1=i=M,\ 1=j=N$

〈표 2〉 토픽 유형별 적용 가능한 매칭연산

토픽 유형	적용 가능한 매칭연산
클래스 토픽, 인스턴스 토픽	$SIM_{name} + SIM_{occ} + SIM_H$
어커런스 토픽	$SIM_{name} + SIM_{occ} + SIM_H$
	$SIM_{name} + SIM_{occ} + SIM_{assoc}$
연관관계 토픽	$SIM_{name} + SIM_{occ}$
역할 토픽	

맵 매칭 알고리즘은 4가지 개별 매칭 연산자들을 복합적으로 적용하는 복합 매칭 연산자이면서 계산 방법에 있어서는 토픽맵 데이터 모델 특성을 고려하는 규칙기반 매칭 기법으로 분류할 수 있다. 다음 <표 1>은 토픽맵 매칭 연산자의 연산식을 보여주고 있다.

온톨로지에서는 서로 다른 용어가 동일한 개념을 가리킬 수도 있고 동일한 용어이지만 서로 다른 개념을 가리킬 수도 있다. 또한 용어는 개념어이기 때문에 기호, 약어, 특수어 등이 들어 있지 않은 명사어 또는 명사구 형태를 가진다. 토픽명기반 매칭 연산은 이러한 특징에 기반하여 문자열 비교 연산을 통해 두 토픽명의 유사값을 산출하는 기법이다. 문자열 비교 연산은 기본적으로 Jaccard 기법을 적용하지만 토큰의 비교에 있어서 완전한 일치 여부 대신 두 토큰의 공통 부분문자열이 차지하는 비중으로 유사값을 산출한다.

<표 1>에서 $SIM_{substring}$은 두 토큰 x, y의 유사값을 최대 공통 부분문자열 c 값을 이용하여 산출한다. 토픽명은 하나 이상의 토큰들로 분리될 수 있으며 a, b는 각각 두 토픽명의 토큰 집합을 가리킨다. 예를 들어, 토픽명이 "독일 철학자"인 경우 "독일"과 "철학자"의 토큰들로 분리된다. 토큰 분리 방법은 토픽명이 명사어인 경우 빈칸없는 복합명사의 토큰 분리를 위해 형태소 분석기에 의한 방법을 사용하고 명사구인 경우 빈칸에 의한 방법을 사용한다. SIM_{dict}는 도메인 용어 사전에 의해 두 토픽명의 유사값을 산출하는 것으로 도메인 용어 사전은 <$term_1$, $term_2$, sim> 구조를 가지는 용어 집합이다.

속성기반 매칭 연산은 두 토픽의 속성들 사이의 유사값을 산출하는 것으로 속성 타입과 속성 값을 기반으로 한다. 표 1에서 m, n은 각각 두 토픽의 속성 집합을 가리키며 $SIM_{occtype}$, $SIM_{occvalue}$는 각각 두 토픽의 속성 타입 기반의 유사값과 속성값 기반의 유사값을 가리킨다. 속성 타입은 어커런스 토픽이므로 두 속성 타입의 유사값은 결과적으로 두 어커런스 토픽의 유사값을 가리킨다.

두 토픽의 유사값은 토픽명과 속성의 토픽 내부적인 구조의 유사성외에 토픽들 사이의 계층관계에 있어서 직전 부모 또는 직전 자식 토픽들 사이의 유사성에 의해서도 영향을 받는다. 예를 들어, 두 토픽의 토픽명이 각각 Philosopher와 Oriental Philosopher이면서 자식 토픽들로 각각 {Kant, Mencius, Nagarjuna, Hegel, Confucius}와 {Mencius, Confucius, Zhuxi}를 가지는 경우 이 두 토픽들은 표 1에서 x, y는 부모 토픽들의 개수이고 x', y'은 자식 토픽들의 개수이며 w는 가중치이다.

연관관계기반 매칭 연산은 연관관계 토픽들 사이의 유사값을 산출하는 것으로 예를 들어, authorOf와 createdBy 연관관계 토픽이 있을 때 이 두 토픽의 유사값은 토픽명기반 유사값 보다 각각의 멤버들 사이의 유사도에 의해 결정된다. <표 1>에서 m_i, m_j는 두 연관관계 토픽의 멤버들이고 r_i, r_j는 멤버들의 역할 토픽을 가리킨다. M과 N은 멤버들의 수이다.

모든 토픽들에 대해 4가지 매칭 연산을 적용하는 것은 아니며 아래 <표 2>와 같이 토픽 유형별로 적용 가능한 연산을 다르게 적용함으로써 계산의 복잡도를 줄일 수 있다.

4.2 토픽맵 통합 연산

매칭연산의 결과는 의미적으로 대응된다고 판단되는 토픽 쌍들의 집합이다. 토픽맵 통합에서는 두 토픽맵을 통합하여 하나의 새로운 토픽맵으로 생성하기 위해 두 토픽이 가지는 의미 정보의 합집합을 구하는 것이다. 기본적인 통합 과정은 두 토픽을 통합할 새로운 토픽을 생성한 다음 각 토픽이 가지는 의미 정보 유형에 따라 중복 값을 제거하면서 새로운 토픽으로 의미 정보를 복사한 다음 두 토픽을 제거하는 것으로 완료된다.

(그림 1)에서 볼 수 있듯이 단일 토픽은 내적 속성들과 다른 토픽들과의 외적연결로 구성된다. 토픽 내적 속성은 토픽 자체를 설명하기 위한 의미 정보로서 속성타입과 속성값으로 이루어진다. 예를 들어, 철학자 토픽은 생애해설, 대표저작, 주요사상 등의 내적 속성을 가지도록 정의할 수 있다. 여기서 생애해설, 대표저작, 주요사상 등은 속성타입으로 철학자 토픽과 별개의 토픽으로 정의되어 있으며 철학자 토픽에서는 속성값을 기술하기 위해 해당 속성타입 토픽으로의 참조를 가진다.

토픽의 외적 연결은 계층적 관계를 표현하는 상위토픽 및 하위토픽과의 연결이 있고 의미적 연관성을 표현하기 위한 연관관계가 있다. 또한 토픽 자체의 정체성(identity)을 부여하기 위한 주제식별참조(SubjectIndicatorRef) 연결이 있다. 매핑 토픽의 통합 알고리즘은 이러한 토픽 구조의 각 요소를 차례로 통합함으로써 두 매핑 토픽들을 하나로 합친다.

5. 통합 충돌 정의 및 탐지

통합충돌은 의미적으로는 대응되지만 구조적으로 상이한 토픽들을 통합하여 하나의 단일 토픽으로 생성하고자 하는 경우 발생한다. 따라서 성공적인 통합을 위해서는 통합 과정에서 발생 가능한 충돌의 유형을 정의하고 각각의 충돌을 탐지할 수 있는 기법이 필요하다. <표 3>은 통합 충돌의 유형과 각각의 탐지 방법을 정의하고 있다.

토픽명 충돌, 속성타입 충돌 및 속성값 충돌은 통합하고자 하는 두 토픽의 구조가 상이함으로 인하여 발생하는 충돌로서 두 토픽의 매칭 연산별 유사값을 확인함으로써 충돌 발생 여부를 판단할 수 있다. 참조 충돌과 미정의 충돌은 토픽맵 통합 과정에서 부분적으로 발생하는 임시적 충돌로서 두 토픽의 통합 후 정당성 검사를 통해 오류를 찾을 수 있다.

임시적 충돌은 탐지와 함께 토픽맵 데이터 모델에 기반한 정당성 검사를 통해 기계적으로 해결이 가능하다. 예를 들어, 두 토픽의 통합 후 발생 가능한 참조 손실은 통합된 토픽이 참조하는 대상이 없는 경우인 외부로의 참조(outgoing reference) 손실과 통합되기 전 두 토픽으로의 참조가 통합으로 인해 손실되는 내부로의 참조(ingoing reference) 손실의 두 가지 유형이다. 이 경우 각각에 대해 참조 또는 참조 대상을 재생성함으로써 해결이 가능하다.

이와 달리 토픽명 충돌, 속성타입 및 속성값 충돌의 해결은 여러 가지 모호성으로 인해 전문가의 판단에 의존적이다. 예를 들어, 의미적으로 대응되는 두 어커런스 토픽 books와 texts를 통합하여 books_and_texts 어커런스 토픽을 생성해 놓았을 경우 두 토픽 T_a와 T_b의 속성타입 books와 texts가 충돌했을 때 T_a와 T_b의 통합 토픽 T_c에서는 books_and_texts 속성타입을 가지며 books와 texts의 중복된 속성 값은 하나만 가지도록 함으로써 기계적으로 해결이 가능하다.

그러나 두 토픽 사이에 속성 값 충돌이 존재하는 경우는 각 토픽이 동일한 속성타입에 대해 속성 값을 서로 다른 내용으로 기술한 것이므로 시스템에서는 중복이 존재하는지 판단하기 어려우므로 통합 토픽의 속성타입에는 두 토픽의 속성 값 모두를 기술한다. 예를 들어, 두 토픽의 속성타입이 biology로 동일하지만 서로 다른 관점에서 생애해설을 한 경우 속성 값이 서로 상이한 속성 값 충돌을 가진다. 이 경우 통합 토픽에서는 biology 속성타입에 두 토픽의 생애해설 모두를 가져야 하며 어느 생애해설을 선택할지는 전문가가 후처리 단계에서 판단하여야 한다.

6. 실험 및 결과

6.1 토픽맵 매칭 및 통합의 실험결과

토픽맵 매칭 및 통합 모듈의 구현을 위해 본 논문에서는

〈표 3〉 통합충돌 유형 및 탐지 방법

통합충돌 유형	충돌 탐지 방법
토픽명 충돌	$SIM_{name}(t_a, t_b) < 1$, 토픽명 상이(충돌 발생) 1) 토픽명의 포함관계 존재. t_a.Name $\subset$ t_b.Name 또는 t_a.Name $\supset$ t_b.Name(부분 충돌) 2) 토픽명의 포함관계 없음(완전 충돌)
속성타입 충돌	$SIM_{occ}(t_a, t_b) < 1$, 속성 상이(충돌 발생) t_a.OccType <> t_b.OccType and t_a.OccVal = t_b.OccVal(속성타입 충돌)
속성값 충돌	$SIM_{occ}(t_a, t_b) < 1$, 속성 상이(충돌 발생) t_a.OccType = t_b.OccType and t_a.OccVal <> t_b.OccVal(속성 값 충돌)
참조손실 충돌	t_c.TopicType$_k$ $\notin$ T_c:{TopicType$_i$ | $1 \le i \le p$}
속성타입 미정의 충돌	t_c.OccType$_k$ $\notin$ T_a:{OccType$_i$ | $1 \le i \le n$}
연관관계타입 미정의 충돌	t_c.AssocType$_k$ $\notin$ R_a:{AssocType$_i$ | $1 \le i \le m$}
역할타입 미정의 충돌	t_c.RoleType$_k$ $\notin$ T_r:{RoleType$_i$ | $1 \le i \le l$}

오픈소스로 제공되는 TM4J(http://www.tm4j.org)를 기반으로 TM4J의 토픽맵 파싱(parsing), 토픽맵 팩토리(factory) 클래스를 활용하여 토픽맵 매칭 및 통합을 위한 색인 관리자(index manager), 토픽맵 매칭 관리자(matching manager), 통합 관리자(merging manager) 클래스를 구현하였다. 구현은 자바 언어를 사용하였으며 실험은 단일 제온(xeon) CPU, 2 기가 메모리, 160 기가 하드디스크와 윈도우 2003 서버 운영체제 하에서 수행하였다.

실험 데이터는 토픽맵으로 작성된 철학 분야의 온톨로지들로서 먼저, 철학 고전 텍스트 내에 존재하는 주요 개념들을 지식 구조화하는 연구과제의 결과물인 철학온톨로지[18]로부터 발췌한 서양근대철학 온톨로지와 서양현대철학 온톨로지가 있으며 위키피디아(wikipedia)에서 철학 용어들을 발췌하여 생성한 위키철학 온톨로지가 있다. 그리고 야후 및 네이버 백과사전의 철학 관련 용어들을 발췌하여 생성한 백과사전 철학 온톨로지가 있다.

토픽맵 매칭의 유효성(effectiveness)은 매칭 결과의 정확성에 의존하며 정확성에 대한 판단은 전문가에 의해 결정된다. 먼저 전문가에 의해 수작업으로 매칭 결과를 생성하고 이를 정답 집합 A라고 한다. 다음으로 시스템에 의해 매칭 연산의 결과로 생성된 매칭 집합을 B라고 하면 토픽맵 매칭의 유효성은 이 두 집합의 일치 정도를 평가함으로써 측정이 가능하다. 이때 두 집합의 교집합을 C라고 할 때 (그림 5)에서와 같이 두 집합의 포함 범위에 따라 4가지의 집합 유형을 분류할 수 있다.[2]

전문가에 의해 산출된 정답집합 A에서 교집합 C를 제외한 A-C 집합은 시스템에서 산출하지 못한 정답집합이므로 *false negatives* 라고 하고 교집합 C는 *true positives* 라고 한다. 시스템에 의해 산출되었으나 정답집합에 포함되지 못하는 B-C 집합은 *false positives* 라고 하고 시스템에 의해 매칭 결과에서 정상적으로 제외된 집합 D는 *true negatives* 라고 한다.

이들 집합에 근거하여 토픽맵 매칭의 유효성을 판단하는 척도는 정확율(precision)과 재현율(recall), F-측정(F-measure), 종합율(overall)이 있다. F-측정과 종합율은 정확율과 재현율을 조합한 측정값으로 F-측정값은 정확율과 재현율의 평균치를 구하기 위해 두 수치에 대한 가중치가 적용된 조화평균이며 종합율은 시스템의 의해 산출된 매칭 집합에 *false negatives*를 추가하고 *false positives*를 제거하는 데 요구되는 노력을 수치화한 것이다. 아래 수식은 각각 정확율(P), 재현율(R), F-측정(F), 종합율(O)을 가리킨다.

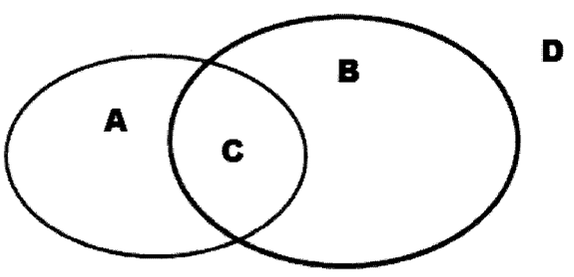
$$P=\frac{|C|}{|B|} \quad R=\frac{|C|}{|A|} \quad F(\alpha)=(1+\alpha)*\frac{P*R}{\alpha*P+R} \quad O=R*(2-\frac{1}{P}) \tag{4}$$

실험 데이터인 서양근대철학 온톨로지를 T1, 서양현대철학 온톨로지를 T2, 위키철학 온톨로지를 T3, 백과사전 철학 온톨로지를 T4라고 했을 때 실험 결과는 (그림 6)과 같다. 실험 결과로 볼 때 시스템에 의한 재현율은 모든 토픽쌍에 대해 85% 이상임을 알 수 있는데 이는 시스템에 의해 자동으로 생성된 매칭 집합이 전문가들의 수작업 매칭 결과를 대부분 포함하고 있음을 보여준다.

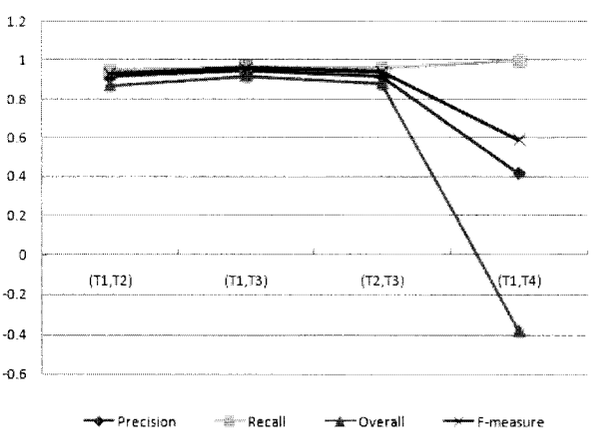
T1과 T2의 경우 동일한 전문가 집단에 의해 설계된 스키마에 따라 각자 근대철학 및 현대철학 분야별 전문가들이 토픽맵을 구축한 것이므로 대부분 스키마 계층에서의 토픽 타입, 어커런스 타입, 역할 타입, 연관관계 타입 등에서 매칭이 이루어지고 인스턴스 토픽들 사이에서도 철학자, 철학문헌 및 텍스트 내용 토픽들에서 유사한 토픽명과 계층 구조에 의해 매칭이 이루어진다.

T1과 T3 또는 T2와 T3 사이의 토픽맵 쌍에서는 주로 '칸트', '흄', '마르크스' 등의 철학자나 '법철학' 등의 철학 텍스트, '도덕법칙', '자유주의' 등의 주요 용어 등에서 토픽명이 일치하는 토픽 쌍들 사이에 매핑이 이루어진다. 이 경우 동일한 스키마가 아닌 서로 다른 구조의 토픽맵이므로 이들 사이에는 동일한 토픽명을 가지지만 그 하위의 자식 토픽들이 완전히 달라서 유사값이 낮게 나오는 경우가 존재한다.

T1과 T4의 경우 특이한 사항은 재현율은 1이면서 종합율은 -4에 가까운 값이 나온다는 것인데 이는 백과사전 철학 토픽맵의 경우 서양근대철학 온톨로지보다 토픽 수가 현저히 적으며 대부분 철학자 이름 또는 철학 텍스트 제목, 철학 사상 이름 등의 명사형의 토픽들 사이에서 매칭이 정확히 일어나기 때문이다. 종합율이 마이너스 값인 이유는 재현율에 비해 정답율이 너무 낮기 때문이며 이것은 자동으로 생성된 매핑 결과를 보정하는데 더 많은 노력이 소모된



(그림 5) 매칭 집합 유형



(그림 7) 토픽맵 매칭 실험 결과

〈표 4〉 토픽맵 통합 실험을 하기위한 tolog 질의 유형

질의유형	설명	질의어 조건문
토픽	특정 이름을 가지는 토픽들 검색	Topic-name($TOPIC,$NAME)
어커런스	특정 토픽의 속성을 가지는 토픽들 검색	Occurrence(Philosopher, $OCCURRENCE)
계층관계	토픽 타입 또는 특정 토픽 타입의 인스턴스 토픽들을 검색	Instance-of($INSTANCE,Philosopher) Instance-of(Kant, $TYPE) Direct-instance-of($INSTANCE, Philosopher)
연관관계	특정 연관관계를 가지는 멤버 토픽들을 검색	Association($ASSOC) Contributed-branch($PHILOSOPHER, $BRANCH) Association-role($ASSOC, $ROLE)

다는 것을 의미한다. 즉, 서로 다른 도메인의 온톨로지 사이에서의 자동 매핑은 무의미함을 보여주는 것이다.

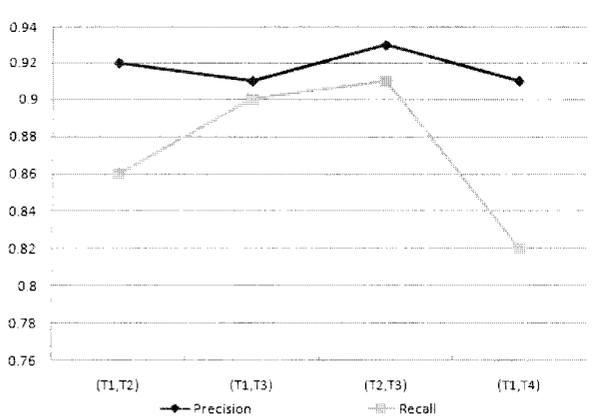
토픽맵 매칭 결과에 따라 두 토픽맵을 하나로 통합하는 토픽맵 통합에서는 실험을 위해 토픽맵 질의어 중의 하나인 tolog를 활용한다. 즉 통합되기 전의 두 토픽맵에 tolog 질의를 실행하여 얻은 검색 결과 집합과 통합 후 단일화된 토픽맵에 동일한 질의를 실행하여 얻은 검색 결과 집합을 비교함으로써 정확율과 재현율을 산출할 수 있다. 아래 <표 4>는 실험에 사용된 tolog 질의어의 유형을 보여주고 있다.

토픽맵 통합의 실험 결과는 (그림 7)과 같다. 전체적으로 90% 이상의 정답율과 재현율을 보이고 있으며 통합 토픽맵이 소스 토픽맵을 손실없이 통합함을 보이고 있다. 두 소스 토픽맵 사이에 매핑되는 토픽이 많을 경우 재현율이 낮게 나오고 있다. 그 이유는 두 소스 토픽맵의 매핑 토픽들이 통합 토픽맵에서는 하나의 토픽으로 통합되기 때문에 동일한 질의어에 대해 통합 토픽맵의 질의 결과 수가 더 적게 나오기 때문이다. T1과 T2, T1과 T4의 경우 전체 토픽 수에 비해 매핑 토픽 수의 비중이 높기 때문에 재현율이 낮게 나왔다.

6.2 토픽맵 매칭의 복잡도 분석

토픽맵 매칭의 계산 복잡도를 분석하기 위해서 매칭 과정의 각 단계에 대해 정리를 하면 다음과 같다.

S1 - 온톨로지를 그래프 모델로 변환하는 단계
S2 - 매칭 연산을 적용할 온톨로지 요소쌍을 생성하는 단계



(그림 7) 토픽맵 통합의 실험 결과

S3 - 각 요소쌍에 대해 k 개의 서로 다른 매칭 연산을 적용하는 단계
S4 - k 개의 서로 다른 유사값들을 조합하는 단계
S5 - 매칭 연산 결과를 생성하기 위해 단일화된 유사값을 평가하는 단계

이 매칭 연산 단계를 기반으로 계산 복잡도 C를 다름 수식 5와 같이 정의할 수 있다. 여기서 N은 매칭 연산을 수행할 요소쌍들의 수이고 k는 개별 매칭 연산의 수이다.

$$C = (S1 + S2 + N * (\Sigma_k SIM_k + S4) + S5) \quad (5)$$

전체 계산 복잡도에 영향을 끼치는 것은 요소쌍들의 수와 각 요소쌍에 적용되는 개별 매칭 연산의 복잡도이다. 이들 개별 매칭 연산의 복잡도를 살펴보면 먼저, 토픽의 *subject identity*를 비교하는 연산은 $O(1)$이고 토픽명, 속성값 등의 문자열을 비교하는 연산은 문자열의 길이 $O(length)$에 의존적이다. 그러나 문자열의 길이가 일정 크기로 제한되어 있다고 한다면 문자열 비교 연산의 복잡도는 $O(1)$로 둘 수 있다.

계층구조기반 매칭 연산이나 연관관계기반 매칭 연산의 경우 토픽 집합들 사이에 유사성이 높은 것들을 찾는 집합 연산이다. 즉 계층구조기반 매칭에서 두 토픽의 유사값은 그들의 직전 하위 토픽들 사이의 유사성에 의존하고 있다. 이때 두 집합의 크기를 각각 K, L이라고 한다면 집합 연산을 위한 복잡도는 $O(|K+L|^2)$으로 둘 수 있다. 이러한 기준하에 복잡도를 산출하면 다음 수식 6과 같다.

$$C = (0 + O(n \bullet log(n)) + O(n) \bullet k \bullet O(1) + O(n) + O(n)) = O(n \bullet log(n)) \quad (6)$$

본 논문의 토픽맵 매칭 연산에서는 그래프 모델로의 변환 과정이 필요없으므로 $S1$은 0이다. 그리고 토픽 분할에 따라 매칭 연산의 대상이 되는 토픽쌍들을 줄임으로써 $S2$는 $O(n \bullet log(n))$이다. 개별 매칭 연산에서 집합 연산의 경우 두 토픽의 모든 서브트리를 대상으로 하지 않고 단지 바로 아래 하위 토픽들만을 대상으로 함으로 제한된 수 l 만큼의 토픽들과 연결되어 있다고 할 때 $O(1)$의 복잡도를 가진다. $S4$ 단계는 모든 토픽쌍에 대해 한번만 실행되므로 복잡도는 $O(n)$이다.

선행 연구와의 직접적인 성능 비교는 매칭 적용 모델 및

〈표 5〉 선행연구와의 간접적인 성능비교

	Anchor-PROMPT	Ctx-Match	IF-MAP	FCA-Merge	QOM	Our Approach
매칭연산	T/ES	T/ES	T/I	T/I	T/IS/ES/E	T/IS/ES/E
적용모델	Graph	Graph	Graph	Graph	RDF/OWL	TopicMaps
연산결과	Merging	Matching	Matching	Matching	Matching	Merging
공간복잡도	$O(n^2)$	$O(n^2)$	$O(n^2)$	$O(n^2)$	$O(n \cdot \log(n))$	$O(n \cdot \log(n))$
전체복잡도	$O(n^2)$	$O(n^2)$	$O(n^2)$	$O(n^2)$	$O(n \cdot \log(n))$	$O(n \cdot \log(n))$

실험 데이터가 서로 상이하고 매칭 알고리즘을 구현한 프로그램을 구하기가 어려운 문제 등으로 인하여 불가능함으로 성능 비교를 위해 표 5와 같이 매칭 적용 모델, 매칭 연산, 공간 및 전체 복잡도 등을 분석함으로써 간접적인 성능 비교를 보인다. 매칭 연산의 유형은 크게 5가지로 분류할 수 있는데 T(Terminological)는 용어들 사이의 유사성을 계산하기 위한 문자열 비교 기법이고 IS(Internal Structure)는 속성과 같은 내적 구조 사이의 비교 기법이다. ES(External Structure)는 용어들의 계층구조 사이의 비교 기법이고 E(Extensional)는 용어들 사이의 연관관계에 대한 비교 기법이다. 그리고 I(Instance)는 인스턴스 수준에서의 비교 기법이다.

복잡도의 경우 전체 성능에 절대적으로 영향을 주는 공간 복잡도를 별도로 보이고 있는데 대부분의 선행연구에서는 두 그래프의 전체 노드들에 대해 비교 연산을 수행함으로써 $O(n^2)$의 복잡도를 가지고 있다. 선행 연구 중 QOM은 본 논문의 매칭 기법과 아주 유사한 접근법을 가지고 있으나 각각 분석 대상 모델이 RDF/OWL과 토픽맵으로 상이하고 연산의 결과 또한 각각 상이한 결과를 보이고 있다.

7. 결론 및 향후연구

본 논문에서는 토픽맵 구문으로 생성된 온톨로지들 사이의 효과적이고 효율적인 매칭 및 통합 기법을 제안하였다. 효과적인 매칭을 위해 토픽맵 데이터 모델의 구조의 특징 및 제약조건 등을 매칭 연산에 반영하고 있으며 토픽명기반, 속성기반, 계층구조기반 및 연관관계 기반의 4가지 개별 매칭 연산을 적용하고 있다.

또한 계산 시간의 효율성을 높이기 위해 이전의 스키마 매칭 연구들과는 달리 스키마를 그래프로 변환하는 과정을 생략하고 토픽분할을 통해 매칭 가능성이 낮은 요소들을 미리 제외시킴으로써 매칭 계산의 복잡성을 줄인다. 토픽분할은 모델기반 토픽분할과 개념기반 토픽분할로 나누어진다.

토픽맵 통합은 매칭 연산 결과를 토대로 두 토픽맵을 하나로 통합하는 것으로 본 논문에서는 토픽맵 통합시에 발생 가능한 충돌의 유형을 분류하고 탐지 및 해결하는 기법을 제안하였다.

토픽맵 매칭 및 통합의 실험은 서양근대철학 온톨로지, 서양현대철학 온톨로지, 위키피이아 철학 온톨로지 및 백과사전 온톨로지를 대상으로 하였으며 실험의 결과 80% 이상의 매칭 재현율을 보였고 통합에 있어서도 높은 정확율을 보이고 있다.

본 논문의 토픽맵 매칭 및 통합 기법은 토픽맵을 기반으로 온톨로지를 구축하고 이를 토대로 지식 서비스를 제공하는 시맨틱 포털이나 콘텐츠 관리 시스템 등에서 여러 토픽맵들을 연계하여 지식 검색이 가능하도록 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] X. Sun and E.Rose. "Automated Schema Matching Techniques: An Exploratory Study," Res. Lett. Inf. Math. Sci., pp.113-136, 2004.

[2] E. Rahm and P. Bernstein. "On Matching Schemas Automatically," VLDB Journal, Vol.10, No.4, 2001.

[3] P. Shvaiko and J. Euzenat. "A Survey of Schema-based Matching Approaches," Technical Report DIT-04-087, University of Trento, Italy, 2004.

[4] H. Do, S. Melnik, and E. Rahm. "Comparison of schema matching evaluations," In Proceedings of the 2nd Int. Workshop on Web Databases, 2002.

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

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

[7] J.M.Kim, H.P.Shin, and H.J.Kim, "Schema and constraints-based matching and merging of topic maps," Information Processing and Management, Vol.43, No.4, pp.930-945, 2007.

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

[9] W. Li, C. Clifton, and S. Liu. "Database Integration using neural network: implementation and experiens," Knowledge Information Systems, Vol.2, No.1, 2000.

[10] F. Giunchiglia and P. Shvaiko. "Semantic matching," In The Knowledge Engineering Review Journal, Vol.18, No.3, 2004.

[11] 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.

[12] 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.

[13] P. Bouquet, L. Serafini, and S. Zanobini. "Semantic coordination: A new approach and an application," In Proceedings of ISWC, 2003.

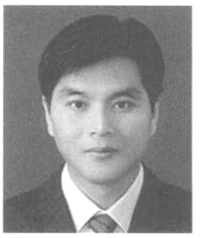
[14] Y. Kalfoglou and M. Schorlemmer. "Information-Flow-based Ontology Mapping," In Proceedings of the 1st International Conference on Ontologies, Database and Application of Semantics, 2002.

[15] G. Stumme and A. Madche. "FCA-Merge: Bottom-up Merging of Ontologies," In Proceedings of 17th Internaltional Joint Conference on Artificial Intelligence(IJCAI), 2001.

[16] M. Ehrig and S. Staab. "QOM: Quick ontology mapping," In Proceedings of ISWC, 2004.

[17] L. Maicher and H. F. Witschel. "Merging of Distributed Topic Maps based on the Subject Identity Measure(SIM) Approach," In Proceedings of LIT, 2004.

[18] J.M.Kim, B.I.Choi, H.P.Shin, and H.J.Kim "A Methodology for Constructing of Philosophy Ontology based on Philosophical Texts," Computer Standards & Interfaces, Vol.29, No.3, pp.302-315, 2007.



김 정 민

e-mail : jmkim12@snu.ac.kr

1992년 홍익대학교 전자계산학과(학사)

1994년 홍익대학교 대학원 전자계산학과 (이학석사)

2007년 서울대학교 대학원 전기컴퓨터공학부 (공학박사)

2007년~현 재 서울대학교 언론정보연구소 박사후연구원

관심분야 : 시맨틱웹, 온톨로지, 데이터베이스 등

정 현 숙

e-mail : hsch@chosun.ac.kr

1993년 대구가톨릭대학교 물리학과 (학사)

1995년 대구가톨릭대학교 전산학과 (이학석사)

1995년 연세대학교 대학원 컴퓨터과학과 (공학박사)

2006년~현 재 조선대학교 컴퓨터공학부 전임강사

관심분야 : 캐릭터 모델링, 모션그래픽스, 이러닝, 시맨틱웹 등