한국지능정보시스템학회논문지 제13권 제4호 2007년 12월(pp.91~104)

응용환경 적응을 위한 온톨로지 매칭 방법론에 관한 연구*

김우주
연세대학교 공과대학
산업정보시스템공학과
(E-mail: wkim@yonsei.ac.kr)

안성준
연세대학교 공과대학
산업정보시스템공학과
(E-mail: sungjun@yonsei.ac.kr)

강주영
아주대학교 e비즈니스학부
(E-mail:jykang@ajou.ac.kr)

박상언
경기대학교 경상대학
(E-mail:supark@kgu.ac.kr)

온톨로지 매칭 기술은 시맨틱 웹을 비롯한 여러 분야에서 중요한 기술 중 하나이다. 온톨로지 매칭은 두 개의 온톨로지를 입력으로 받고, 이를 몇 개의 매개변수로 구성된 특정 알고리즘을 이용하여 두 온톨로지 간의 매칭 관계를 알아내는 절차를 말한다. 온톨로지 매칭은 대용량 온톨로지의 통합이나, 지능화된 통합 검색의 구현 및 여러 응용프로그램에 의한 도메인의 공유 등 여러 분야에서 유용하게 활용될 수 있다. 일반적으로 온톨로지 매칭의 성능은 온톨로지 매칭이 사용되는 환경과 관계없이 매칭 결과에 대한 측정만으로 평가되어 왔다. 따라서 대부분의 연구는 매칭 결과를 최적화하기 위해 매개변수를 조절하는 것에 집중하였다. 본 연구에서는 기존의 측정방법에 따른 높은 측정결과만을 목표로 하지 않고 온톨로지의 성격과 매칭 결과의 사용 목적에 따라 매개변수를 적절히 변화시켜야 한다는 점에 주목하고, 주어진 환경에 맞게 매개변수를 조정하는 방법론을 제안하고자 한다.

논문접수일 : 2007년 09월 게재확정일 : 2007년 12월 교신저자 : 강주영

1. 서론

매칭 기술은 시맨틱 웹을 비롯한 여러 분야에서 중요한 기술 중 하나이다. 온톨로지란 특정 도메인에 대한 내용을 형식적 혹은 공식적인 언어로 표현한 것을 말하며, 일반적으로 온톨로지를 활용하는 응용프로그램은 온톨로지를 활용함으로써 도메인에 대한 정보를 얻거나, 이를 바탕으로 문제해결에 필요한 추론을 한다. 온톨로지 매칭은 둘 이상의 온톨로지에 기술되어있는 도메인 정보들을 비교하여 온톨로지 간의 유사성을 측정하고 그 결과를 표현하는 것을 말한다. 온톨로지 매칭을 통하여, 동일한 도메인을 다룸에도 불구하고 서로 이질적인 구조를 갖는 온톨로지들을 통합함으로써, 자료의 통합관리를 이루거나, 에이전트가 매칭 결과를 이용하여 지능적인 통합검색을 수행하거나, 서로 다른 시스템간의 상호운용성을 보장하는 등 여러 가지 방안으로 사용할 수 있다. 시맨틱 웹의 기본적인 연구 방향이 시스템간에 주고 받는 정보의 의미를 파악하여 이를 자동으로 처리하고자 하는 것임을 감안할 때, 앞서 언급된 바와 같이 온톨로지 매칭은 시맨틱 웹에서 매우 중요한 기술이라

* 이 논문은 2005-2학기 아주대학교 정착연구비 지원에 의하여 연구되었음.

할 수 있다(Noy, 2004).

온톨로지 매칭의 결과를 평가하기 위한 방법으로 여러 방법이 고안되어 왔다. 매칭 결과를 평가하기 위한 대표적인 척도로는 정확도(Precision)과 재현율(Recall)이 있다(Yatskevich et al., 2006). 정확도는 알고리즘에 의해 제안된 매칭 결과 중에서 실제로 맞은 결과의 비율를 나타내고, 재현율은 매칭이 되어야 하는 대상 중에서 알고리즘이 매칭에 성공한 대상의 비율을 나타낸다. 매칭 결과 평가 방법론에서 사용하는 성능측정 방법 중 대표적으로 F-Measure가 있다(Giunchiglia et al, 2005). F- Measure 는 정확도와 재현율을 통합하여 하나의 척도로 온톨로지 매칭의 결과를 평가한다. F-Measure 는 다음의 식으로 정의된다.

$$F = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Reall}} \quad (1)$$

지금까지의 연구에서는 온톨로지 매칭의 결과를 평가할 때 정확도와 재현율 그리고 F-Measure의 값을 높이는 것에 집중해왔다. 특히 정확도가 우선시되는 경우가 많았다. 하지만 이러한 접근 방법은 상황에 따라서 의도된 바와 다른 결과를 가져올 수 있다. 예를 들어 전자상거래 환경에서 상이한 쇼핑 사이트의 상품들을 통합하여 관리 또는 검색하기 위한 목적으로 상품 카테고리에 대해 온톨로지 매칭을 실시할 경우, 매칭의 정확성을 지나치게 강조하다 보면 매칭의 재현율이 떨어져 실제로 사용자가 얻을 수 있는 정보의 양이 급격하게 줄어들 수 있다. 보스톤 컨설팅 그룹에 의하면 전자상거래 환경에서 사용자들이 시도한 전체 검색 시도의 28%는 원하는 상품을 검색하지 못했으며, 사용자들의 48%가 원하는 상품에 대해 만족스럽지 못한 검색결과를 경험했다(Pecaut et al., 2005). 이와 같은 결과는, 전자상거래 환경에서 정확도보다 재현률이 오히려 더 중요한 척도임을 보여 준다. 따라서 본 연구에서는 온톨로지 매칭시 도메인과 관계 없이 높은 척도값 만을 목표로 매칭을 하는 것이 아니라 온톨로지 매칭을 실행하는 환경에 따라 척도값들의 적절한 조합을 얻기 위한 매칭을 실행해야 한다는 점에 초점을 맞추고자 한다. 이를 위하여 온톨로지 매칭에 사용되는 매개변수들을 조절함으로써 정확도와 재현율이 어떻게 변화하는지 살펴보았다.

일반적으로 정확도와 재현율은 반비례 관계를 갖게 된다. 즉 정확한 매칭 결과를 위하여 정확도를 높일 경우 재현율은 떨어지게 된다. 반대로 보다 많은 결과를 얻기 위해 재현율을 높이면 정확도는 떨어지게 된다. 앞서 예를 든 전자상거래 환경에서는 정확도를 희생하는 대신 재현율을 높이는 방향으로 매칭을 적용하는 것이 좋다고 볼 수 있다. 그러나 이 경우에 정확도의 희생과 이로 인한 재현율의 증가를 서로 비교하여 정확도가 지나치게 떨어지지 않는 적절한 수준을 찾아야 할 것이다. 반대로 온톨로지 매칭을 통해 정확한 자료 통합을 하고자 하는 경우는 재현율보다 정확도를 더욱 고려하여 매칭을 수행하는 것이 좋다고 볼 수 있다.

본 연구에서는 정확도와 재현율 중 하나의 척도를 우선으로 하여 온톨로지 매칭을 수행할 때 다른 척도의 손실률을 최소화하는 것을 지원하기 위해, 실험을 통해 정확도와 재현율 사이의 관계에 대해 알아보고자 한다. 본 논문의 구성은 다음과 같다. 제 2장에서 온톨로지 매칭에 대한 관련 연구를 설명하고, 제 3장에서 실험에 사용된 온톨로지 매칭 방법론과 온톨로지 매칭 알고리즘에 대하여 설명할 것이다. 제 4장에서는 실험의 결과를 분석하고, 마지막으로 제 5장에서 결론과 향후 연구에 대하여 다루고자 한다.

2. 관련연구

온톨로지 매칭과 관련된 연구분야는 다음과 같이 온톨로지 매핑 발견(Mapping Discovery), 온톨로지 매핑을 위한 유사성 표현(Declarative formal representations of mappings), 매핑을 이용한 추론(Reasoning with mappings) 연구로 나눌 수 있다 [Giunchiglia, 2005]. Mapping Discovery는 주어진 두 온톨로지에 대해 서로 유사성을 보이는 클래스나 속성을 찾는 방법을 연구하는 분야이고, Declarative formal representation of mappings는 추론(Reasoning)을 위해 두 온톨로지 간의 유사성을 어떤 식으로 표현(Representation)할 것인가 에 대한 연구이다. Reasoning with mappings은 매핑이 이루어진 후 그것을 이용하는 것을 뜻한다. 이 본 연구에서 주로 다룰 내용은 온톨로지 간의 유사성을 알아보는 Mapping Discovery에 속한다고 볼 수 있다.

온톨로지 매칭과 관련하여 지금까지 다양한 방법론들이 연구되고 개발되어 왔다(Benetti et al., 2002; Decker et al., 1999; Ehrig and Staab, 2004; Guarino et al, 1999; Noy and Musen, 2003; Veltman, 2001). 온톨로지를 매칭하는 방법에 따라 크게 공유 온톨로지(shared ontology)를 사용한 후 각 사용처에 따라 온톨로지를 확장하여 추후 매칭을 실시할 때 보다 효율적인 매칭을 추구하는 방법과, 공유 온톨로지를 사용할 수 없는 상황의 경우 경험적, 혹은 학습을 통한 정보를 이용해 매칭을 하는 휴리스틱적, 기계학습적 방법으로 나눌 수 있다(Doan et al., 2003; Modica et al, 2001; Noy, 2003).

온톨로지 매칭은 대상이 되는 두 온톨로지에 존재하는 여러 클래스들에 대해서 이루어지는데, 매칭이 이루어지는 범위에 따라서 온톨로지의 클래스와 서브클래스들만 매칭을 수행하는 것과 해당 클래스들의 인스턴스들까지 매칭의 대상으로 삼는 방법이 존재한다. 온톨로지 매칭의 정확성 관점에서는, 인스턴스까지 고려하는 것이 더 정확하다고 볼 수 있다. 그러나 이 방법의 경우 매칭 시간이 온톨로지에 포함되어 있는 인스턴스의 양에 따라서 기하급수적으로 늘어날 수 있다는 단점 역시 존재한다. 이상과 같은 이유로 본 연구에서는 온톨로지 매칭에 인스턴스를 사용하지 않고, 온톨로지 스키마만으로 매칭을 수행하는 스키마 기반 매칭(Schema-Based Matching)(Magnini et al., 2004)을 이용한 매칭 방법론에 초점을 맞추고자 한다.

스키마 기반 매칭에선 입력 받은 온톨로지를 매칭하는 대상에 따라 요소 수준(Element-level)과 구조 수준(Structure-level)으로 나눌 수 있다. 요소 수준(Element-level) 매칭은 온톨로지내의 클래스의 이름만을 분석하는 것이고, 구조 수준(Structure-level) 매칭은 온톨로지내의 클래스들의 서브, 슈퍼클래스 관계를 분석하여 매칭을 수행하는 것이다. 요소 수준(Element-level) 매칭과 구조 수준(Structure- level) 매칭은 매칭을 하기 위해 입력된 정보를 해석하는 방식에 따라 형식적(Syntactic) 방식, 외부(External) 방식, 의미적(Semantic) 방식으로 다시 나눌 수 있는데, 형식적(Syntactic) 방식은 단순한 문자열(String) 비교나, 단어 비교 등으로 클래스간의 유사성을 판별하는 것이고 외부(External) 방식은 클래스간의 유사성을 판단할 때 WordNet과 같은 외부 자원을 참고하여 클래스간의 유사성을 판단하는 것이고, 의미적(Semantic) 방식은 매칭을 하기 위해 입력된 온톨로지를 해석할 때 공식적인 의미(formal semantic)를 이용하는 것이다.

앞서 언급한 요소 수준(Element-level)의 매칭 방법, 구조 수준(Structure-level)의 매칭 방법과

형식적(Syntactic) 방식, 외부(External) 방식, 의미적(Semantic) 방식과 같은 정보 해석방법을 사용함에 따라 각기 문자열 기반(String-based), 언어학 기반(Linguistic based), 매칭 결과 재사용(Alignment reuse), 그래프 기반(Graph based), 분류체계 기반(Taxonomy based) 등으로 나눌 수 있다. 최근의 매칭 방법론은 위에서 언급한 문자열 기반(String-based), 언어학 기반(Linguistic based), 매칭결과 재사용(Alignment reuse) 등 여러 가지 방법론을 혼합하여 사용하고 있다(Magnini et al., 2004; Ehrig and Sure, 2004).

본 연구의 매칭 방법은 소스 온톨로지와 목표 온톨로지에 있는 클래스들의 하위클래스와 상위클래스구조를 이루는 경로를 매칭하는 방법으로 기본적으로 클래스들의 이름을 비교하여 매칭을 수행한다. 매칭 수행 시 동의어, 유사어로 기술된 클래스 이름에 대비하여 WordNet(Miller, 1995)을 사용하여 클래스 이름의 의미를 확장하여 이를 가지고 매칭을 수행한다(Guarino, 1999). 이러한 방법은 앞서 언급한 매칭 방법론 분류 절차에 의하면 구조 수준(Structure-level)의 분류 체계 기반(taxonomy-based) 방식이나, 구조 저장소(repository of structures) 방식과 유사한 방법이라고 할 수 있다.

3. 온톨로지 매칭 방법론

3.1 온톨로지 매칭 개요

일반적인 온톨로지 매칭 프로세스는 다음과 같다. 온톨로지 매칭은 두 개 이상의 온톨로지를 입력으로 읽어 들인 후, 이를 특정 알고리즘을 이용하여 각 온톨로지에 있는 클래스들의 유사성을 평가한다. 매칭 방법론에 따라서 매개변수가 유사성을 판단하는데 이용될 수도 있다. 매칭을 하려는 두 개의 온톨로지 중 매칭을 할 온톨로지를 소스 온톨로지라고 하고 매칭의 대상이 되는 온톨로지를 목표 온톨로지라고 한다. 유사성을 판단하는 매개변수에는 여러 가지가 있는데 대표적으로는 가중치(weight)와, 쓰레숄드(threshold), 개념별 분류어휘집(Thesaurus)등이 존재한다. 예를 들어 알고리즘을 통해 각 온톨로지에 있는 용어인 a와 a'의 유사정도가 0.65이고, 이 방법론이 사용하고 있는 매개변수 중 쓰레숄드 역할을 하는 매개변수의 값이 0.63이며, 유사도가 쓰레숄드보다 높아야 유사성이 있다고 판단하는 것이 유사성에 대한 알고리즘의 판단기준이라면, 두 클래스는 같은 것을 지칭하는 것으로 결정된다(Magnini et al., 2004).

위와 같이 특정 프로세스에 의해서 수행된 매칭된 결과는 매칭 평가방법론에 의해서 평가될 수 있다. 평가 방법론에 따라서 다르지만, 앞서 언급했듯이 많은 매칭 측정 방법에서 정확도와 재현율은 측정 시 중요한 척도로서 작용한다. 이 두 척도는 위에 소개한 일반적인 매칭 프로세스 내의 매개변수를 이용하여 조정할 수 있다. 따라서 매개변수의 조정은 매칭 방법론의 성능과 적합성을 결정하는 중요한 요소 중 하나이다. 일반적으로 많은 연구에서는 매칭 방법 측정 결과가 높게 나올 수 있도록 매개변수를 조정하여 목표로 결정한 정확도 혹은 재현율 중 하나를 달성하지만 본 연구에서 구현하고자 하는 매칭 방법론에서는 매개변수 조정을 통해 각 환경에 맞도록 적절한 정확도와 재현율의 관계를 달성하는 것이 그 목표이다.

예를 들어 앞서 설명한 전자상거래 환경에서는 정확도와 재현율을 통합한 F-measure 값이 떨어지더라도 적절한 재현율을 유지하는 것이 더 중요할 수 있다. 사용자에 의해 요구된 상품에 대해 통

합검색을 하기 위해 두 쇼핑몰의 온톨로지를 매칭하는 과정에서 두 가지의 알고리즘을 사용했다고 가정해 보자. 첫째 알고리즘을 사용한 결과, 제안된 결과 15개 중 5개가 맞았고 매칭이 되어야 하는 대상은 10개였다고 하면 F-measure는 0.4가 된다. 둘째 알고리즘에서는 제안된 결과 4개 중 3개가 맞았으며 매칭이 되어야 하는 대상은 동일하다고 하면 이 때 F-measure 값은 0.428이 된다. 즉, F-measure 상으로는 둘째 알고리즘이 더 나은 알고리즘이지만 전자상거래 환경에서는 첫째 알고리즘이 오히려 더 나은 매칭 알고리즘이 더 부합한다고 볼 수 있다. 제시한 예는 정확도보다는 재현율을 더 중요시한 경우이지만, 상황에 따라 그 반대의 경우도 있다. 이 때 역시 매개변수 조절로 통해 환경에 최적화 된 온톨로지 매칭을 수행할 수 있다. 이처럼 온톨로지 매칭의 사용처에 따라 적절하게 매개변수를 조정하여 정확도와 재현율을 조정함으로써 온톨로지 매칭의 결과를 보다 효과적으로 사용할 수 있다. 매개변수의 조절과 관련하여 3.2절에서는 정확도와 재현율이 매칭 알고리즘에 의해 어떤 식으로 결정되는지에 대해서 설명하고, 4장에서는 반복실험을 통해 나타난 정확도와 재현율의 관계에 대해서 설명하고자 한다.

3.2 매칭 프로세스 및 알고리즘

본 논문에서 사용하고자 하는 매칭 프로세스는 크게 세 단계로 나뉜다. 첫째 단계는 소스 온톨로지에서 매칭을 할 대상에 대한 정확한 의미 파악이다. 이 단계를 거치는 이유는 소스, 목표 온톨로지에 동일한 의미로 존재하고 있는 클래스일지라도 서로 다른 명칭으로 사용되었을 경우나 동일한 명칭으로 사용되었으나 의미가 다를 경우를 정확히 구별하여 매칭을 수행하기 위함이다. 예를 들어 Car와 Vehicle은 같은 의미의 클래스이지만, 다른 명칭으로 사용되었기 때문에 온톨로지 매칭의 대상이 될 수 있고, 공책의 의미로 사용된 Notebook과 휴대용 컴퓨터인 Notebook은 같은 이름임에도 불구하고 다른 의미를 같고 있으므로 온톨로지 매칭의 대상이 되어서는 안 된다. 이를 위해서는 단순히 매칭 소스의 어휘 만으로는 안되며 매칭 소스의 경로를 사용하여야만 한다. 이 단계의 결과로 워드넷(Miller, 1995)으로부터 매칭 소스에 가장 근접한 확장의미가 선정된다.

둘째 단계는 워드넷으로부터 선정된 확장의미를 이용하여 매칭을 수행하는 단계이다. 매칭은 소스와 목표 온톨로지의 경로에 대해 이루어진다. 매칭을 하는 두 경로를 대상으로 동일 단어의 출현 빈도와, 출현 순서를 비교한다.

세 번째 단계는 매칭 유사도 산출 및 평가 단계이다. 두 번째 단계에 대한 결과에 두 개의 매개변수를 이용하여 매칭 유사도를 산출하고 이를 기반으로 유사성을 판별한다.

이상의 세 단계를 보다 이해하기 쉽도록 설명하기 위한 예제로, 본 논문에서는 ODP(Open Directory Project, www.dmoz.com)의 온톨로지를 소스 온톨로지로 사용하고 Amazon의 상품분류체계를 목표 온톨로지로 사용하였다.

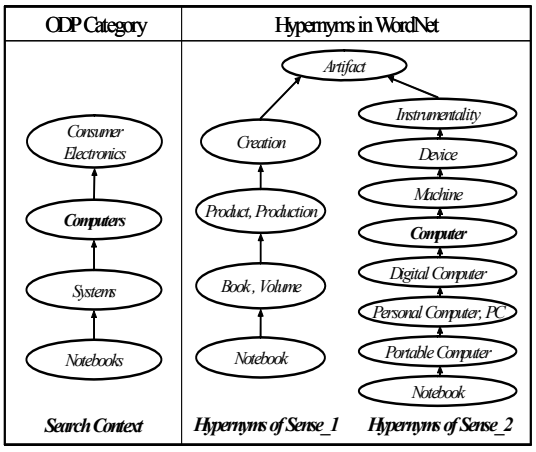
3.2.1 어휘의 의미 파악 및 확장

이 단계는 온톨로지 매칭 알고리즘의 첫 단계로서, 매칭할 소스 온톨로지 내의 클래스에 대한 정확한 의미를 파악하는 것이 목표이다. 이를 위하여 단어에 대한 하위어(hyponyms), 상위어(hypernyms), 동의어(synoyms), 속성 등을 정의하고 있는 영어 어휘 데이터베이스인 워드넷을 이용하였다(Avesani et al., 2005). 매칭대상이 될 소스 온톨로지에 있는 클래스의 이름에 대한 동의어를 워드넷을 이용하여 찾은 후 동의어 집합을 소스 온톨로

지의 클래스가 속한 경로와 비교하는 과정을 통해서 의미를 확장한다. 이러한 일련의 과정을 구현한 것이 *pathproximity* 함수이며 *cs* 함수와 *hypernymproximity* 함수로 이루어진다. *cs*는 다음 식과 같다.

$$cs(x, p)=\{h|h\in SYNSETS(x) and\ h\in hypernyms(p)\}$$
where x is an upper category of the product hierarch (2)

*cs*는 클래스 *x*와 워드넷의 센스(sense)를 입력받아 워드넷의 센스로부터 x와 매칭되는 p의 상위어들을 반환한다. 센스는 워드넷에서 주어진 단어의 다양한 의미들을 표현한다. 예를 들어 notebook이 앞서 언급한 두 개의 의미를 갖고 있다면, notebook에는 공책의 의미로 센스1, 컴퓨터 노트북의 의미로 센스2가 존재한다. 다음 그림의 예에서 notebook은 ODP에서 systems, computers, consumer electronics의 세 상위어를 갖고 있다. *cs* 함수는 이 세 개의 상위어와 그림의 오른쪽에 있는 두 개의 센스에 대해 적용될 수 있다. 예를 들어 *cs*(computers, sense_2)는 sense_2에 있는 상위어들 중에서 computer를 반환한다.



[그림 1] notebook의 ODP 경로와 워드넷 센스들

함수 cs를 이용하여 주어진 클래스와 워드넷 센스 간의 유사도를 다음 식과 같이 구할 수 있다.

$$hypernympoximity(x, p) = \left\{\frac{1}{\mathrm{Min\_dis}\,(cs(x,p), base)}\right\} \quad (3)$$

식에서 base는 센스에서 가장 아래 위치하는 노드이다. 따라서 *hypernymproximity*는 *x*와 매칭되는 센스의 상위어와 최하위 노드 간의 거리의 역수가 된다. 역수를 취하는 이유는 거리가 가까울수록 유사도는 높아지기 때문이다. 예를 들어 *hypernymproximity*(computers, sense_2)는 computer가 sense_2에서 notebook과 4단계 거리만큼 떨어져 있으므로 0.25가 된다. 만일 *x*와 일치하는 상위어가 *p*에 없으면 이 함수는 0을 반환한다.

함수 *cs*와 *hypernymproximity*를 이용하면 ODP의 클래스 경로와 워드넷의 센스들 간의 경로 유사도를 구할 수 있다. 이는 *pathproximity* 함수로 구현되며 다음 식과 같다.

$$pathproximity(p) = \frac{\sum_{x\in upper\_categories(base)} hyperproximity(x, p)}{n} \quad (4)$$

두 경로간의 유사도는 위 식에서 보는 바와 같이 ODP 클래스의 모든 상위어들에 대해 워드넷 센스와의 유사도를 구한 다음 이를 합산하여 ODP 클래스의 노드 수로 나눔으로써 구한다. 그림 1의 예에서는 sense_1의 경우, 일치하는 상위어가 없으므로 최종유사도는 0이다. 반면 sense_2는 computer의 값이 0.25이고 이를 다시 4로 나누어주므로 0.0625가 된다.

3.2.2 매칭 클래스 후보 검색

이 단계에서는 목표 온톨로지에서 현재 주어진

소스 온톨로지의 클래스에 대해 매칭되는 클래스 후보들을 검색한다. 다른 단계에 비해 이 단계의 알고리즘은 매우 단순하다. 우선 앞 단계에서 결정된 센스를 이용하여 목표 온톨로지로부터 대상이 될 수 있는 모든 클래스들을 검색한다. 이 과정에서는 워드넷 센스의 유사어 및 동의어가 사용된다. 클래스들이 검색되면 각 클래스에 대해 목표 온톨로지에서 경로를 찾아 매칭 후보로 저장한다.

3.2.3 온톨로지 내 경로의 유사성 판단

이 단계는 앞 단계에서 찾아진 후보들로부터 가장 적합한 후보를 선택하는 단계이다. 이를 위해 첫째 매칭하고자 하는 소스와 목표 온톨로지의 경로들에 대해 동일한 문자열이 얼마나 있는지를 계산하고, 둘째 동일한 문자열들이 경로상에서 동일 순서로 위치했는가를 비교한다. 이 두 과정은 각각 *Co-Occurrence*와 *Order_Consistency*로 구현되었다.

우선 *Co-Occurrence*함수는 *TermMatch, NodeMatch, MaxSim*으로 이루어지며, 이에 대한 상세한 설명은 다음과 같다. *TermMatch*는 주어진 두 개의 클래스에 대하여 문자열 비교를 하여 문자열 유사성을 계산한다. 다음은 *TermMatch*의 수식이다.

$$TermMatch(term_1, term_2)$$
$$\begin{cases} \dfrac{strlen(term_1)}{strlen(term_2)} & \text{if term1 is substring of term2} \\ \dfrac{strlen(term_2)}{strlen(term_1)} & \text{if term2 is substring of term1} \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

함수 *strlen*은 클래스 문자열의 길이를 계산한다. 예를 들어 *TermMatch*가 Electronics와 Consumer Electronics를 인수로 입력받으면 11/19가 두 문자열 간의 유사도로 계산된다.

*NodeMatch*는 *TermMatch*를 확장하여 현재 매칭하고자 하는 클래스에 대해 워드넷으로부터 구한 확장의미집합과 대상이 되는 클래스의 문자열 유사성을 구하는 함수이다. 이를 식으로 표현하면 다음과 같다.

$$Nodematch(term1, term2) = \max_{sterm \in ExtendedtermSet(term1)} TermMatch(sterm, term2)$$

where
term1 is a termbelongingto a source path
term2 is a termbelongingto a target path (6)

*MaxSim*은 소스 온톨로지 경로 중 하나의 클래스와 이전 단계에서 구한 목표 온톨로지의 클래스 경로 후보 사이의 유사성을 *NodeMatch*를 이용하여 구하는 함수이다. 다음의 수식과 같이 *MaxSim*은 매칭하고자 하는 클래스와 대상이 되는 경로의 클래스 사이의 문자열 유사성 중 최대값을 취하게 된다.

$$maxSim(cterm, cpath) = \max_{pterm \in cpath} NodeMatch(cterm, pterm) \quad (7)$$

수식 (8)에서 볼 수 있듯이 유사성을 구하기 위한 첫 단계의 최종값인 *Co-Occurrence*는 *MaxSim*을 이용하여 구할 수 있다. 우선 소스 온톨로지의 경로를 기준으로 하여 경로상에 있는 모든 클래스들에 대해 대상이 되는 목표 온톨로지와의 *Maxsim*을 계산하여 평균을 구한다. 다음에는 거꾸로 목표 온톨로지의 경로를 기준으로 하여 경로상에 있는 모든 클래스들에 대해 대상이 되는 소스 온톨로지와의 *Maxsim*을 계산하여 평균을 구

한다. 이 두개의 평균을 곱하면 *Co-Occurrence*가 된다. 결과적으로 목표 온톨로지의 후보 경로 중에서 경로상에 일치하는 클래스가 많은 경로가 더 높은 *Co-Occurrence* 값을 갖게 된다.

Co-Occurence(*source_path, target_path*)

$$= \left(\frac{\sum_{cterm \in target\_path} MaxSim(cterm, source\_path}{ns(target\_path)} \right) \quad (8)$$

$$\left(\frac{\sum_{sterm \in source\_path} MaxSim(sterm, target\_path}{ns(source\_path)} \right)$$

wherens(x)returnsthe number of elements of a set x

*Order_Consistency*는 매칭하려는 두 경로에서 유사한 문자열로 이루어진 클래스가 유사한 순서로 있는지를 측정하는 척도이다. *Order_Consistency*는 *Common, Prelset, Consistent* 로 이루어진다.

*Common*은 소스 온톨로지의 경로와 목표 온톨로지의 경로에 유사한 클래스가 포함되어 있는지를 확인하여 유사한 클래스의 쌍을 반환한다. 이를 이용하여 주어진 경로에서 공통으로 존재하는 클래스에 대한 순서를 반환하는 *Prelset* 함수를 정의한다. *Prelset*은 *Common*에서 주어지는 클래스에 대하여 주어진 경로에서의 순서쌍을 생성한다. 예를 들어 다음과 같은 두 개의 경로가 있다면,

Path1 = (A, B, C, D)
Path2 = (A', B', C', E) (A와 A'는 유사 문자열)

*Common*과 *Prelset*에 의해 반환되는 값은 다음과 같다.

Common(Path1, Path2) = {[A, A'], [B, B'], [C, C']}
Prelset(common(), Path2) = ([A', B'], [B', C'], [A', C'])

위 예에서 path2에서는 현재 공통되는 A', B', C' 클래스가 [A', B'], [B', C'], [A', C']의 순서로 나타나고 있다.

Consistent 함수는 *Prelset*의 순서쌍 중 하나와 다른 경로를 인수로 읽어 들인다. 만일 순서쌍의 순서가, 주어진 다른 경로에서도 유지되면 1을 반환하고 그렇지 않으면 0을 반환한다. 이를 식으로 표현하면 다음과 같다.

$$consistent\ ((t_p,\ t_s),\ C\_Path)$$
$$\begin{cases} 1 & \text{if } t_p' \ precedes\ t_s' \ in\ C\_Path \\ 0 & otherwise \end{cases} \quad (9)$$

*Order_Consistency*는 *Prelset*의 결과와 *Consistent* 함수를 함께 사용하여 구현된다. 매칭의 대상이 되는 목표 온톨로지의 경로에 대해 *Prelset*에서 구한 소스 온톨로지에서의 각 순서쌍이 목표 경로에서도 제대로 지켜지고 있는지를 *consistent* 함수를 이용하여 판단하고 제대로 지켜지고 있는 순서쌍의 비율을 계산하는 것이 *Order_Consistency*에서 이루어지는 작업이다. 이를 식으로 표현하면 다음과 같다.

Order_Consistency(*source_path, target_path*)

$$\frac{\sum_{pr \in prelset(ccts, source\_path)} consistent(pr, target\_path)}{ns(prelset)}$$

where ccts∈*common*(*source_path, target_path*) (10)

3.2.4 최종 유사성 판단

Co-Occurrence, *Order_Consistency*를 통합한 최종 유사성을 이용하여 소스 온톨로지의 경로와 목표 온톨로지의 경로 간의 매칭 여부를 판단하기 위하여 본 연구에서는 계산하는 최종 유사도는 다음과 같다.

$$Similarity = \alpha(Co\text{-}Occurrence) + (1-\alpha)\ Order\_Consistency$$

유사성을 판단하는 식에서는 앞서 언급한 매개변수들이 사용된다. 위 식을 보면 α와 t 두 개의 매개변수가 사용되고 있는 것을 볼 수 있다. α는 *Co-occurrence*와 *Order consistency*의 상대적 가중치이고, t는 쓰레숄드를 나타내는 매개변수다. α를 높일 경우, 최종 유사도에서 *Co-Occurrence*의 비중이 높아지게 되고 상대적으로 *Order Consistency*의 비중은 낮아진다. α를 낮출 경우는 반대의 현상이 나타난다. 이렇게 구해진 최종 유사도를 t와 비교함으로써 매칭의 여부가 결정된다. 이 과정에서 매개변수 t를 높이는 것은 매칭의 정확도를 높이기 위해 비교기준을 높이는 것이며 t를 낮추는 것은 재현율의 중요도를 더 높이는 결과를 가져온다. α에 비해 t는 보다 직접적으로 정확도와 재현율에 영향을 미치는 것을 볼 수 있다. 그러나 α 역시 두 척도에 영향을 미치고 있으므로 이 두 매개변수를 조정함으로써 원하는 정확도와 재현율의 비율을 설정할 수 있다.

4. 온톨로지 매칭 실험 및 결과

이 장에서는 실험을 통해 α와 t값을 변화시켜가며 정확도와 재현율의 변화를 살펴 봄으로써 매개변수들과 척도들의 관계에 대해 알아보고자 한다.

4.1 실험 개요

본 연구에서는 온톨로지 매칭을 수행할 때, α와 t 두 매개변수에 따른 정확도와 재현율의 변화를 살펴봄으로써 매개변수와 성능척도 간의 관계를 알아보기 위한 실험을 수행하였다. 이 결과를 이용하면, 정확도와 재현율의 최적 비율을 얻기 위한 매개변수 값을 결정하여, 특정 환경에 맞는 온톨로지의 매칭 방법을 설정하는데 도움을 줄 수 있다. 본 연구의 실험에 사용된 온톨로지는 아마존(Amazon.com)과 바이닷컴(Buy.com), ODP의 상품분류체계로 가장 많이 사용되는 온톨로지 언어인 OWL로 변환한 것이다.

일반적으로 온톨로지 매칭을 사용하는 상황에서는 실제 정답을 아는 것이 불가능하다. 그러나, 성능 척도는 정답을 아는 경우에만 계산이 가능하다. 따라서 실제로 온톨로지 매칭을 사용하는 시나리오에서 성능척도는 일부 온톨로지를 갖고 측정될 수 밖에 없으며 이 성능척도를 기반으로 다른 온톨로지에서도 비슷한 성능을 낼 것으로 기대하게 된다. 따라서 본 연구에서 가정하는 시나리오는 다음과 같다. 주어진 온톨로지 간의 매칭을 하기 위해서 우선 소스 온톨로지와 목표 온톨로지의 일부에 대해 매개변수와 성능척도에 대한 실험을 실시한다. 실험으로부터 얻어진 최적성능에 대한 매개변수가 결정되면 나머지 온톨로지에 대해서는 결정된 매개변수를 이용하여 온톨로지 매칭을 실시한다. 연구결과를 활용하기 위한 다른 방법은 유사한 도메인의 온톨로지에 이와 같은 방법을 확장하여 사용하는 것이다. 즉 다른 소스 온톨로지와 목표 온톨로지에서 만들어진 매개변수 값을 유사한 도메인의 온톨로지에서도 동일하게 사용하는 것이다.

본 실험에서도 이와 같은 활용방안을 그대로 이용하기 위해 위에서 언급한 세 온톨로지 중에서 일부 클래스들만을 추출하여 실험을 위한 환경을 구성하였다. 아마존과 바이닷컴 그리고 ODP에 공통으로 존재하는 상품분류 중에서도 전자제품을 중심으로 실험대상이 되는 클래스들을 추출하였

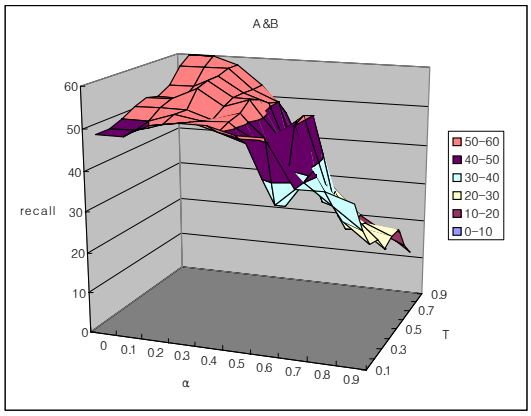
으며, 클래스들간의 경로가 온톨로지 매칭에서 사용되므로 각 클래스들 간의 관계 중에서 상하위 관계를 동일하게 추출하였다. 아마존은 1124개의 클래스를 추출하였고, 바이닷컴은 798개의 클래스를 추출하였으며, ODP는 1766개의 클래스를 추출하였다.

실험은 3개의 온톨로지를 각각 소스 온톨로지와 목표 온톨로지로 번갈아 가며 지정함으로써 총 6쌍의 소스 및 온톨로지를 만들어 실험을 수행하였다. 예를 들어, 처음에는 아마존을 소스 온톨로지로 하고 바이닷컴을 목표 온톨로지로 하여, 아마존에 있는 상품 경로들에 대해 매칭되는 바이닷컴의 상품 경로들을 찾고자 하였다. 실제로 매칭되어야 하는 상품 경로는 직접 확인을 통해 구했으며, 이 상품 경로와 알고리즘에 의해 구해진 상품 경로들을 비교함으로써 정확도와 재현율을 계산하였다.

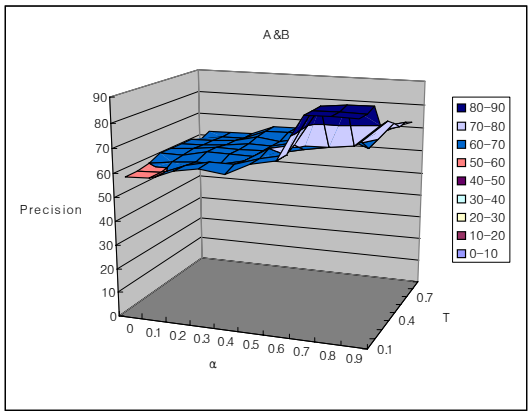
4.2 실험 결과 및 분석

각 온톨로지 매칭 실험에서는 α와 t의 값을 0부터 1까지 0.1씩 변화시켜가며 총 10구간으로 나누어 각 매개변수 쌍에 대한 정확도와 재현율을 계산하였다. 예를 들어 α와 t의 쌍이 (0.1, 0.1)인 상황부터 시작하여 (1, 1)이 되는 상황까지 실험을 총 100회 반복하였다. 온톨로지 쌍이 6종류이므로 총 600회의 실험이 반복하여 실시되었다. 각 소스 및 목표 온톨로지 쌍에 대하여 정확도와 재현율을 그래프로 표현해야 하므로 총 12개의 그래프가 결과로 생성되나, 본 논문에서는 그 중 의미가 있다고 판단되는 6개의 그래프만을 분석 대상으로 제시하였다. 다음은 온톨로지 매칭 결과이다.

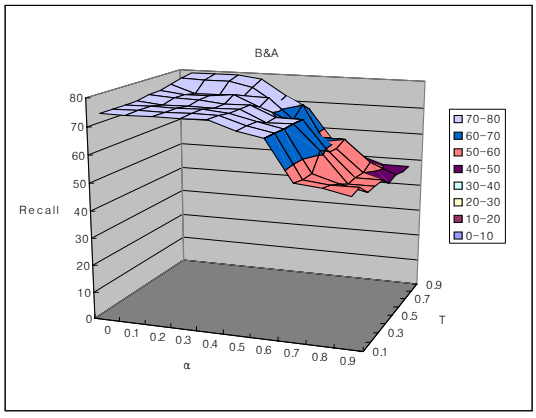
그래프를 보면 동일한 t값이 주어진 상태에서 α 값이 작을수록 재현율이 높게 나오는 것을 발견할



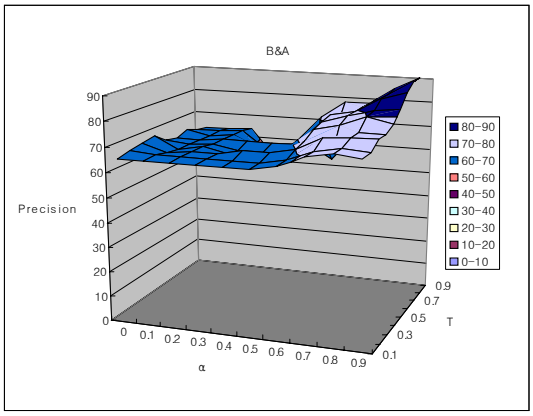
[그림 2] Amazon에서 Buy.com으로의 매칭 재현율



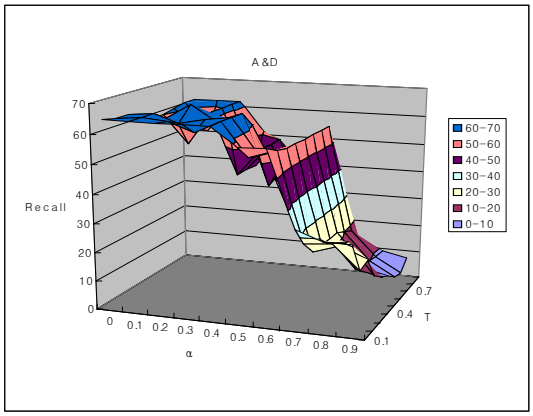
[그림 3] Amazon에서 Buy.com으로의 매칭 정확도



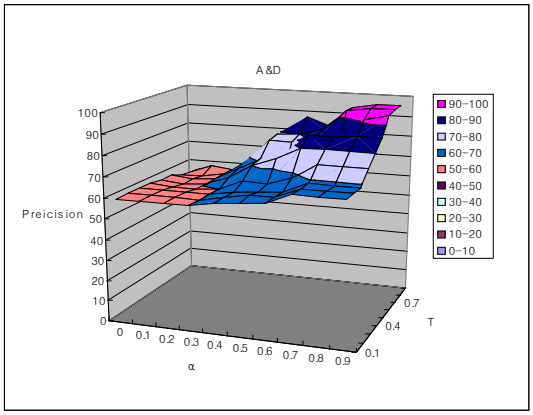
[그림 4] Buy.com에서 Amazon으로의 매칭 재현율



[그림 5] Buy.com에서 Amazon으로의 매칭 정확도



[그림 6] Amazon에서 ODP로의 매칭 재현율



[그림 7] Amazon에서 ODP로의 매칭 정확도

수 있다. 뿐만 아니라 몇몇의 그래프에서는 α값이 커짐에 따라 재현율이 급격하게 줄어드는 지점이 있음을 볼 수 있다. 이는 α값이 커지는 것이 t값이 커지는 것과 동일한 효과가 나타나는 것을 의미한다. 그 원인을 분석해 보면, 우선 α값이 커지면 *Co-Occurrence*의 비중은 높아지게 된다. *Co-Occurrence*의 비중이 클수록 재현율이 낮아지는 것은 유사도의 보다 큰 영향을 주는 성능척도가 *Co-Occurrence*임을 말한다. 따라서 일반적으로 *Order-Consistency*는 각 매칭 대상 클래스들 간에 큰 차이가 없으며, 대부분의 경우 높은 값을 보이고 있음을 알 수 있다. 반면, 동일 t에서 α값이 증가할수록 정확도는 증가함을 볼 수 있다. 그러나 변화 정도가 재현율 만큼 급격하지는 않기 때문에 α값의 변화는 정확도보다 재현율에 더 큰 영향을 미치고 있음을 알 수 있다. 이로부터 얻을 수 있는 시사점은 정확도를 기준으로 매개변수 값을 결정할 경우, 재현율에 대한 손해가 정확도로부터 얻는 이득에 비해 매우 커질 수 있다는 것이다. 주어진 그래프를 분석하면 적절한 정확도를 유지하면서 재현율을 최대화하기 위한 매개변수 값을 설정하는 것이 가능하다.

매개변수 t의 경우에는 예측한 바와 같이 동일한 α값에서 t값이 클수록 일반적으로 정확도는 향상되는 반면 재현율은 줄어드는 것으로 나타났다.

t의 경우 동일 α값에서 재현율에 미치는 영향이 α보다 적은 것으로 나타났다. 즉 이는 재현율을 조절하기 위해 α가 t보다 더 적절한 매개변수가 될 수 있음을 의미한다. 정확도에 대한 그래프들을 살펴 보면 정확도의 분포가 비교적 유사함을 볼 수 있다. 반면, 재현율은 소스 온톨로지와 목표 온톨로지 쌍에 따라 민감한 반응을 보인다. 즉, 재현율에 대한 고려가 더 많이 이루어져야 함을 알 수 있다.

본 실험에서 전반적으로 t값에 비해 α의 값이 재현율이나 정확도에 많은 영향을 끼치는 것으로 나타난 이유는 일부 온톨로지 자체의 특성으로 해석할 수도 있다. 실험에서 사용된 온톨로지들은 모두 상품 카테고리를 온톨로지화 한 것으로 대부분의 상품분류는 웹 사이트 혹은 쇼핑몰이 달라지더라도 상하위 관계가 유사하기 때문인 것으로 풀이된다. 즉 *Order-Consistency*가 항상 높게 나오기 때문에 결과적으로 *Co-Occurrence*에 비해서 *Order-Consistency*는 매칭 결과에 영향을 적게 미쳤음을 짐작할 수 있다. 그로 인해 α에 의한 성능척도의 변화가 더 크게 나타나게 되었다.

5. 결론

본 연구에서는 흔히 온톨로지 매칭 환경에 관계없이 매칭 성능만을 중요시 하는 지금까지의 온톨로지 매칭 방법과는 달리 온톨로지 매칭이 이루어지는 환경에 적합한 온톨로지 매칭 방법론을 제안하고자 하였고, 이를 정확도와 재현율을 변화시키는 방법으로 접근했다. 정확도와 재현율의 변화로 인해 특정환경에 적합한 온톨로지 매칭을 이루기 위한 시도 중 하나로 온톨로지 매칭에 사용되는 알고리즘 중 정확도와 재현율에 영향을 미치는 매개변수를 조정하여 매개변수와 정확도 및 재현율 간의 관계를 실험을 통하여 확인하였다. 실험을 통해 나타난 매개변수와 정확도 및 재현율의 관계를 활용함으로써 환경에 맞는 온톨로지 매칭을 위한 최적의 정확도와 재현율의 비를 구하기 위해 매개변수를 설정하는데 본 연구의 결과가 활용될 수 있을 것이다.

그러나, *Co-Occurrence*와 *Order-Consistency*가 매칭의 결과에 미치는 영향의 정도를 충분히 밝히지 못한 점과 이루어진 실험 결과를 다양한 도메인에서 해보지 못한 점, 그리고 일반적인 상황에서도 이와 같은 동일한 실험결과가 나올 것이라고 보장할 수 없다는 점은 본 연구의 한계점으로 지적될 수 있다. 추후 매개변수의 변화에 따른 온톨로지 매칭 결과에 대한 실제적인 검증을 통하여 매개변수의 조정을 통한 정확도와 재현율의 조정이 특정환경에 적합한 온톨로지 매칭을 이루는데 어떻게 기여하고 있는 지를 밝혀 내는 연구가 필요하다고 판단된다. 뿐만 아니라 전자상거래 환경의 온톨로지에서 보다 다양한 도메인의 온톨로지로 실험 환경을 확장하여 연구의 보편성을 확보할 필요가 있다.

참고문헌

[1] 김우주, 방시리, 박상언, “An Optimized Methodology of Ontology Driven Mapping for Product Search”, 지능 정보시스템 학회 논문집, 2006.

[2] 최남혁, “이질적인 쇼핑몰 환경을 위한 온톨로지 기반 상품 매핑 방법론”, 연세대학교, 석사학위 논문, 2006.

[3] 최대우, “에이전트와 쇼핑몰을 위한 의미 웹 서비스 기반 지능형 상품 정보 검색 프레임워크”, 전북대학교, 박사학위논문, 2004.

[4] Amazon.com, http://www.amazon.com.

[5] Avesani, P., F. Giunchiglia, and M. Yatskevich, “A large scale taxonomy mapping evaluation”, in Proceedings of International Semantic Web Conference (ISWC), LNCS 3729, 2005, 67~81.

[6] Benetti, H., D. Beneventano, S. Bergamaschi, F. Guerra, and M. Vincini, “SI-Designer: An information integration framework for e-commerce”, IEEE Intelligent Systems,

Vol.17, No.1 (2002), 18~25.

[7] Decker, S., Erdmann, M., Fensel, D., and R. Studer, "Ontobroker : Ontology based access to distributed and semi-structured information", In Meersman, R., Tari, Z., Stevens, S. M. (ed.), Proceedings of DS-8 : Semantic Issues in Multimedia Systems. Boston : Kluwer, (1999), 351~369.

[8] Doan, A, J., Madhaven, R, P. Dhamankar, Domingos, and A., Helevy, "Learning to match ontologies on the Semantic Web", Very Large DataBases Journal, Vol.12, No.4(2003), 303~319.

[9] Ehrig, M. and S. Staab, "QOM : Quick Ontology Mapping", Lecture Notes in Computer Science, No.3298, 683~697, 2004.

[10] Ehrig, M. and Sure, Y., "Ontology mapping -An integrated approach", Lecture Notes in Computer Science, No.3053(2004), 76~91.

[11] Giunchiglia, F., Shvaiko, P. and Yatskevich, M., "Semantic Schema Matching", Proceedings of the 13th International Conference on Cooperative Information Systems, 2005.

[12] Guarino, N., C. Masolo, and G. Vetere, "OntoSeek : Content-based access to the Web", IEEE Intelligent Systems, Vol.14, No.3(1999), 70~80.

[13] Kim, W., D. W. Choi, and S. Park, "Agent Based Intelligent Search Framework for Product Information Using Ontology Mapping", Journal of Intelligent Information Systems, Forthcoming, 2007.

[14] Magnini B., M. Speranza and C. Girardi "A Semantic-based Approach to Interoperability of Classification Hierarchies : Evaluation of Linguistic Techniques", Proceedings of COLING, 2004.

[15] Miller, G. A., "WordNet : A lexical database for English", Communications of the ACM, Vol. 38, No.11(1995), 39~41.

[16] Noy, N. F., "Semantic Integration : A Survey of Ontology-Based Approaches", Sigmod Record, Special Issue on Semantic Integration, Vol.33, No.4(2004), 65~70.

[17] Noy, N. F., and M. A. Musen, "The PROMPT Suite : Interactive tools for ontology merging and mapping", International Journal of Human- Computer Studies, Vol.59, No.6(2003), 983~1024.

[18] OAEI (Ontology Alignment Evaluation Initiative) http://oaei.ontologymatching.org/2007/.

[19] OWL Seb Ontology Language Reference available from http://www.w3.org/TR/owl-ref/.

[20] OWL Web Ontology Language Guide available from http://www.w3.org/TR/owl-guide/.

[21] Park, S., Kim, W. and Lee, S. and Bang, S. "An Ontology Mapping Algotitm between Hetegogeneous Product Classification Taxonomies", Proceedings of the IIWAS, 2006.

[22] Pecaut, D., M. Silverstein, and P. Stanger, Winning the online consumer : Insights into online consumer behavior. Boston: Boston Consulting Group Report, 2000.

[23] RDF Primer available from http://www.w3.org/TR/rdf-primer/

[24] Shvaiko, P. and Euzenat, J., "A Survey of Schema-based Matching Approaches", Journal on Data Semantics (JoDS), IV, LNCS 3730, (2005), 146~171.

[25] Veltman, K. H., "Syntactic and semantic interoperability : New approaches to knowledge and the Semantic Web", New Review of Information Networking, Vol.7(2001), 159~1841.

[26] Yatskevich, M., F. Giunchiglia, and P. Avesani, "A Large Scale Dataset for the Evaluation of Matching Systems", Proceedings of ESWC 2007.

Abstract

Adaptive Ontology Matching Methodology for an Application Area

Wooju Kim* · Sung Jun Ahn** · Juyoung Kang*** · Sangun Park****

Ontology matching technique is one of the most important techniques in the Semantic Web as well as in other areas. Ontology matching algorithm takes two ontologies as input, and finds out the matching relations between the two ontologies by using some parameters in the matching process. Ontology matching is very useful in various areas such as the integration of large-scale ontologies, the implementation of intelligent unified search, and the share of domain knowledge for various applications. In general cases, the performance of ontology matching is estimated by measuring the matching results such as precision and recall regardless of the requirements that came from the matching environment. Therefore, most research focuses on controlling parameters for the optimization of precision and recall separately. In this paper, we focused on the harmony of precision and recall rather than independent performance of each. The purpose of this paper is to propose a methodology that determines parameters for the desired ratio of precision and recall that is appropriate for the requirements of the matching environment.

Key words : Semantic Web, Ontology Matching, Ontology

* Department of Information and Industrial Engineering, Yonsei University
** Department of Information and Industrial Engineering, Yonsei University
*** Department of e-Business, School of Business, Ajou University
**** Division of Business Administration, Kyonggi University