RESTful 웹 서비스를 위한 온톨로지 자동 구축 기법

이용주
경북대학교 이공대학 컴퓨터정보학부
e-mail:yongju@knu.ac.kr

Automatic Building Ontology Techniques for RESTful Web Services

Yong-Ju Lee
School of Computer Information, Kyungpook National University

요 약

최근 웹상에 이용 가능한 RESTful 웹 서비스들의 수가 급격하게 증가됨에 따라 사용자들이 적합한 웹 서비스를 찾는 것은 매우 중요한 이슈로 대두되었다. 그러나 기존의 키워드 기반 검색 방법은 나쁜 재현율과 나쁜 정확률 때문에 문제가 많다. 본 논문에서는 매개변수 클러스터링 기법에 패턴 분석 기법을 추가한 하나의 새로운 시맨틱 온톨로지 구축 방법을 제안한다. 이를 통해 온톨로지를 자동 구축하여 시맨틱 정보의 주석처리 부담을 줄일 수 있고, 보다 효율적인 웹 서비스 검색을 지원한다.

1. 서론

최근 웹 2.0의 등장과 함께 OpenAPI(Application Program Interface)와 매쉬업(mashup)이 발전되면서 기존의 SOAP 기반 웹 서비스에 비해 RESTful 웹 서비스의 활용이 크게 증가하고 있다[1]. 예를 들어, 구글, 아마존, 야후, 이베이, 네이버, 그리고 다음과 같은 기업에서는 웹 2.0의 새로운 패러다임에 발맞추어 자사의 정보 자원을 OpenAPI를 통해 외부 사용자에게 적극적으로 개방하고 있다. RESTful 웹 서비스란 HTTP와 REST의 원리를 사용하여 구현된 간단한 웹 서비스로 정의되며, GET, POST 등 HTTP 기본 기능을 적극 활용하여 XML 형식의 메시지 전송에 의해 보다 의미 있는 데이터셋을 생성하기 위하여 다수의 자원으로부터 데이터를 결합할 수 있게 지원한다. 이는 매쉬업과 비슷한 개념으로, 한편으론 매쉬업을 RESTful 웹 서비스의 조합으로 설명할 수 있다.

오늘날 웹상에 이용 가능한 RESTful 웹 서비스들의 수가 급격하게 증가됨에 따라 사용자가 적합한 웹 서비스를 찾는 것은 매우 중요한 이슈로 대두되고 있다. 그러나 정통적인 키워드 기반 검색 방법은 나쁜 재현율(recall)과 나쁜 정확률(precision) 때문에 문제가 많다. 이러한 키워드 기반 검색 방법의 한계를 극복하기 위한 기법으로서 시맨틱(semantic) 정보를 이용한 온톨로지(ontology) 활용 방법이 있을 수 있다[2]. 그렇지만 이러한 온톨로지는 대부분 전문가의 수작업으로 구축되고 있으며, 시간 및 인적 제약 때문에 실용적인 온톨로지를 구축하기가 어렵다. 또한, 시맨틱 정보를 위한 추가적인 레이어(layer)는 단순한 REST의 취지와도 맞지 않는다. 따라서 본 논문에서는 매개변수 클러스터링(parameter clustering) 기법에 패턴 분석(pattern analysis) 기법을 추가한 하나의 새로운 시맨틱 온톨로지 구축 방법을 제안한다. 이를 통해 온톨로지를 자동 구축하여 시맨틱 정보의 주석처리(annotation) 부담을 줄일 수 있고, 보다 효율적인 웹 서비스 검색을 지원한다.

본 논문의 구성은 다음과 같다. 2장에서 관련 연구들을 살펴보고 3장에서 RESTful 웹 서비스를 간단히 설명한다. 4장에서 RESTful 시맨틱 온톨로지 구축 방법을 제안하고 5장에서 결론을 내린다.

2. 관련 연구

온톨로지 구축 방법에 관한 연구는 크게 두 가지 방향으로 추진되고 있다. 첫째는 전문가의 수작업으로 웹 서비스 저장소(registry)에 추가적인 시맨틱 정보를 주석처리하여 온톨로지를 구축하는 방법이다. 전통적인 SOAP 기반 웹 서비스에서는 OWL-S, WSMO, 그리고 SAWSDL 방법이 있고, RESTful 웹 서비스에서는 SA-REST와 SBWS(Semantic Bridge for Web Services) 방법이 있다. 그러나 이러한 시맨틱 정보 주석처리 방법의 문제점은 전문가의 수작업으로 모든 것이 처리되어야 하므로 시간 및 인적 제약으로 인해 온톨로지를 구축하기가 쉽지 않다. 또한 현시점에서 웹 서비스 전체에 대한 주석을 다시 단다는 것은 거의 불가능하게 보인다.

두 번째 접근방법은 수작업으로 온톨로지를 구축하기가 어렵기 때문에 온톨로지 학습(ontology learning) 방법

이 논문은 2010년도 정부(교육과학기술부)의 재원으로 한국연구재단의 기초연구사업 지원을 받아 수행된 것임(No. 2010-0008303).

에 의해 자동 구축하는 방법이다. [3]은 웹서비스들을 자동 분류하기 위해 Naive Bayes와 SVM 머신 러닝 방법을 제안하였고, [4]는 연관성이 높은 웹 서비스 매개변수들을 같은 개념으로 묶는 클러스터링 메카니즘을 제안하였다. [5]에서는 온톨로지 학습을 위해 프로그램 소스, 도큐멘테이션, UML 다이어그램 등 다양한 소스를 고려하였고, 자연언어처리 기법을 이용한 웹 서비스 온톨로지 학습 프레임워크를 제안하였다. 그러나 이러한 온톨로지 학습 방법은 대부분 SOAP 기반 웹 서비스를 위한 방법이었고 RESTful 웹 서비스를 위해서는 이들이 잘 맞지 않는다. 왜냐하면, SOAP 기반 웹 서비스에서는 다양한 오퍼레이션들에 대한 시맨틱 처리가 중요한 반면에, RESTful 웹 서비스에서는 이들 오퍼레이션 대신에 체계적으로 구성된 URI에 대한 HTTP 기본 메소드만 수행되기 때문이다.

3. RESTful 웹 서비스

REST는 웹의 창시자 중 한 사람인 Roy Fielding의 박사학위 논문[6]에 의해 소개되었다. 그는 현재의 웹 아키텍처가 웹이 지닌 본래의 설계 우수성을 충분히 활용하지 못하고 있다고 판단하고, 웹의 장점을 최대한 활용할 수 있는 네트워크 기반의 아키텍처를 제안했는데 그것이 바로 REST다. 이런 REST 아키텍처 스타일에 따라 정의되고 이용되는 서비스나 응용을 RESTful 웹 서비스라 한다.

RESTful 웹 서비스의 기본 개념은 다음과 같다.

- 리소스의 URI 설정: RESTful 웹 서비스의 가장 큰 특징 중의 하나는 모든 대상을 리소스(resource), 즉, 자원으로 표현한다는 것이다. 이 리소스는 HTTP URI에 의해 표현되며, 웹 사이트, 블로그, 이미지 등 웹에서 다른 이들과 공유하고자 개방된 모든 자원을 의미한다.
- HTTP 메소드 사용: REST 구조에서의 리소스는 HTTP의 기본 메소드(method)인 GET, PUT, POST, DELETE만으로 접근할 수 있다. 리소스에 접근하기 위한 이러한 4개의 HTTP 메소드는 일반 CRUD (Create, Read, Update, Delete) 오퍼레이션에 각각 대응될 수 있다.
- 다양한 표현 방식: HTTP의 기본 메소드로 전달되는 리소스는 다양한 방식으로 표현(representation)되는데, 이는 XML, JSON, HTML, 텍스트, 이미지 등이 가능하며 클라이언트에서 원하는 형식으로 표현하면 서버에서 이를 처리하게 된다.
- 스테이트리스: HTTP의 특성을 상속하여 RESTful 웹 서비스 역시 스테이트리스(stateless) 특성을 가지게 되는데, 스테이트리스란 웹 서비스 제공 서버 측에서 클라이언트의 상태(state) 정보를 저장, 관리하지 않는 것을 의미한다. 즉, 클라이언트가 HTTP 요청(request)을 할 때 서버에 그 요청을 수행할 수 있는 모든 정보를 주어야 하며 이전의 요청에 의존해서는 안 된다.

최근 SOAP 기반 웹 서비스에 비해 가볍고 구현하기 쉬운 RESTful 웹 서비스가 많은 주목을 받고 있음에 따라, RESTful 웹 서비스의 인터페이스를 기술하기 위한 다양한 방법들이 제안되고 있다. 현재 RESTful 웹 서비스를 기술하기 위한 여러 가지 언어들이 제안되고 있으나 아직까지 주류는 형성되지 않은 상황이다. 기존의 WSDL에서도 2.0 버전에서는 RESTful 웹 서비스를 기술할 수 있는 HTTP 바인딩(binding) 확장 스펙이 제안 되었지만 너무 복잡하다는 이유로 많이 쓰이지는 못하고 있다. 이에 썬마이크로시스템은 간략하면서도 범용성이 뛰어난 WADL[7]을 발표하게 되었고, 간단하다는 장점 때문에 개발자들 사이에서 WADL의 사용은 점차 증가되고 있는 실정이다.

WADL에서 서비스는 *resource* 엘리멘트들로 기술되며, 이들 각각에는 *request*와 *response*를 기술하는 *method* 엘리멘터가 있다. *request* 엘리멘트에는 어떻게 입력을 표현할 것인가를 기술하고 있고, *response* 엘리멘트에는 서비스 결과의 *representation*과 상태(status) 정보를 기술하고 있다. 그리고 *request*와 *response*에는 매개변수를 나타내는 *param* 엘리멘트들이 있을 수 있다.

4. RESTful 시맨틱 온톨로지 구축

RESTful 웹 서비스를 위한 시맨틱 온톨로지의 구축은 많은 이점을 줄 수 있으며, REST 서비스와 관련된 수많은 문제점들을 해결할 수 있다. 비록 기존의 SOAP 기반 웹 서비스에 대한 시맨틱 온톨로지는 OWL-S, WSMO, 그리고 SAWSDL과 같은 많은 플랫폼들이 제안되어 있지만, RESTful 웹 서비스에 대한 시맨틱 온톨로지는 아직까지 구체적인 연구 결과가 없는 상황이다. 이는 REST 본래의 목적이 단순함이었기 때문에 RESTful 웹 서비스에서는 애초에 WSDL과 같은 기술 언어를 요구하지도 않았고, 이러한 기술 언어의 부재는 RESTful 시맨틱 온톨로지의 구축을 힘들게 만들었다.

본 논문에서는 RESTful 웹 서비스의 기술 언어로써 WADL을 채택하였다. 그러나 RESTful 시멘틱 온톨로지를 구축하기 위해 WADL이 꼭 필요한 것은 아니다. 다만, WADL이 온톨로지 구축 자동화에 도움이 줄 수 있는 수단이 될 수 있다. WADL은 WSDL 처럼 구문(syntactic) 정보는 제공하지만, 시맨틱 웹을 위해 설계되지 않았기 때문에 웹 서비스 리소스에 대한 시맨틱 정보를 정의할 수 있는 틀(placeholder)은 제공하지 않는다. 따라서 본 논문에서는 RESTful 시맨틱 온톨로지의 구축을 가능케 하기 위하여 매개변수 클러스터링 기법에 패턴 분석 기법을 추가한 새로운 시맨틱 온톨로지 구축 방법을 제안한다.

4.1 연관규칙 기반 클러스터링 기법

RESTful 웹 서비스의 매개변수들을 토큰화하여 용어들로 분리한 후, 관련성이 많은 용어들에 대해 클러스터를 형성하면 이 클러스터는 각각의 단어가 아닌 하나의 의미있는 개념(concept)을 나타낸다. 이러한 클러스터는 “매개변수들이 동시에 자주 나타난다면, 그것들은 같은 개념을 나타내는 경향이 있다”는 가정 하에 하나의 특별한 연관

규칙(association rules)[8]에 따라 만들어 진다.

연관규칙은 용어 A가 일어나면 용어 B가 일어난다는 의미로 A ⇒ B로 표현될 수 있으며, 여기서 트랜잭션(transaction)은 웹 서비스 입출력에 나타나는 용어들의 집합으로 볼 수 있다. 그리고 지지도(support)와 신뢰도(confidence)는 해당 규칙이 얼마나 유용한지를 나타내는 지표로서, 지지도는 용어 A와 B를 동시에 포함하는 트랜잭션의 확률을 표현하며, 신뢰도는 용어 A가 주어졌을 때 용어 B가 동시에 나타날 트랜잭션의 확률을 나타낸다.

$$support(A \Rightarrow B) = P(A \cup B)$$

$$confidence(A \Rightarrow B) = \frac{P(A \cup B)}{P(A)}$$

연관규칙을 찾는 과정은 기본적으로 빈발 용어 집합(frequent termset)을 찾는 단계와 연관규칙을 생성하는 두 단계로 구성된다. 빈발 용어 집합이란 후보 용어 집합(candidate termset) 중 최소 지지도(minimum support: *minsupp*) 이상의 값을 가진 용어 집합으로서 데이터 마이닝 기법에서 생성되는 연관규칙의 단위가 된다. 빈발 용어들이 생성되고 나면 이들로부터 최소 신뢰도(minimum confidence: *minconf*)를 만족하는 모든 연관규칙들을 찾는다.

본 논문에서는 계층적 클러스터링의 한 종류인 계층적 결합 클러스터링(agglomerative clustering) 방법[9]을 사용한다. 그러나 기존의 계층적 결합 클러스터링 방법은 대부분 저차원 데이터를 위해 설계되어 있어서, 데이터의 차원이 매우 큰(예, 수천 개의 상이한 용어들을 가지는 텍스트 문서) 경우 차원의 저주(curse of dimensionality) 문제가 대두된다. 이러한 문제를 해결하는 하나의 방법으로써 연관규칙이 적용될 수 있다. 본 논문에서는 연관규칙 탐사 과정에서 생성된 최종 연관규칙으로부터 먼저 신뢰도를 내림차순으로 정렬한 다음 지지도를 내림차순으로 정렬한 후, 각 단계에서 가장 최상위에 있는 규칙을 조사하여 만일 두 용어가 다른 클러스터에 속하면 이들을 결합한다.

결합하는 과정에서 우리는 가장 이상적인 클러스터를 형성하기 위하여 다음 두 가지 클러스터 평가 특성을 고려한다. (1) cohesion: 한 클러스터 내의 용어들과의 응집력, (2) separation: 다른 클러스터 용어들 간의 상호관계. 여기서, 클러스터의 cohesion은 높게 하고, 클러스터 간의 separation은 낮게 한다. 클러스터 C_1이 주어졌을 때 cohesion(C_1)은 한 클러스터 내에 서로 밀접하게 연관되어 있는 용어 쌍들의 분포 확률로 정의된다.

$$cohesion(C_1) = \frac{\| associationRule(i \Rightarrow j) \|}{\| C_1 \| \| (C_1 - 1) \|}$$

여기서, $i,j \in C_1$, $i \neq j$, associationRule(i⇒j) = {support(i⇒j)>*minsupp* & confidence(i⇒j)>*minconf*}이다. 한편, 클러스터 C_1과 C_2가 주어졌을 때 separation(C_1, C_2)은 클러스터 간 서로 밀접하게 연관되어 있는 용어 쌍들의 분포 확률로 정의된다.

$$separation(C_1, C_2) = \frac{\| associationRule(i \Rightarrow j) \| + \| associationRule(j \Rightarrow i) \|}{2 \| C_1 \| \| C_2 \|}$$

여기서, $i \in C_1$, $j \in C_2$이다.

최종적으로, 전체적인 클러스터 C의 품질을 측정하기 위하여 다음과 같은 클러스터링 점수(score)를 정의한다.

$$score(C) = \frac{avg(cohesion)}{avg(separation)} = \frac{(\| C \| - 1) \sum_{t_1} cohesion(C_1)}{2 \sum_{t_{12}} separation(C_1, C_2)}$$

여기서, t_1은 $C_1 \in C$, t_{12}는 $C_1, C_2 \in C$, $C_1 \neq C_2$로 표현된다. 이러한 과정에서 우리의 최종 목표는 가장 높은 점수를 갖도록 클러스터를 형성하는 것이다.

본 논문의 연관규칙 기반 클러스터링 기법은 각 용어들 간의 상호 연관성을 이용해 관련된 단어들끼리 클러스터링 함으로써 보다 효과적인 웹 서비스 검색을 가능하게 한다. 그러나 이 기법은 연관성 높은 단어들을 한 클러스터에 묶어 단지 동일한 개념처럼 취급할 뿐 계층관계에 따라 사용자의 요구사항을 정확하게 표현하는 온톨로지 기능은 제공하지 못하고 있다. 이러한 문제점을 해결하기 위해 본 논문에서는 다음의 패턴 기반 시맨틱 분석 기법을 추가로 제안한다.

4.2 패턴 기반 시맨틱 분석 기법

패턴 기반 시맨틱 분석 기법의 주된 목표는 매개변수 내의 각 단어들 사이의 상관관계를 취득하고, 비교되는 단어들이 서로 유사하고 상관관계가 조건에 일치한다면 그 비교를 매치하는 것이다. 이러한 기법은 "사람들이 단어를 조합하여 복합단어로 된 매개변수를 만들 때 일반적으로 비슷한 패턴을 사용한다"는 관찰로부터 시작한다[10].

RESTful 웹 서비스에서의 이러한 패턴들을 조사하기 위해 본 논문에서는 ProgrammableWeb 사이트[11]로부터 RESTful 웹 서비스 매개변수들을 다운로드 받아 실험을 수행하였다. 본 실험에서는 mapping, travel, weather 카테고리에 있는 168개의 RESTful 웹 서비스에 대해 실험을 수행하였다. 수집된 실험 데이터는 총 8209개의 매개변수들로 구성되었으며, 이에 대해 CRFTagger POS 형태소 분석기[12]를 적용시킨 결과 다음과 같은 결과가 나타났다.

- 단지 하나의 토큰으로 구성된 매개변수(예, city)가 3574개로 전체의 44%를 차지하였다. 이는 패턴 기반 시맨틱 분석 기법과는 관련 없는 매개변수들이다.
- 명사$_1$+명사$_2$(Noun$_1$+Noun$_2$) 형태의 매개변수(예, companyCode)가 2435개로 전체의 30%를 차지하였다. 이는 하나의 토큰 매개변수와 합치면 전체의 74%를 차지하는 가장 많이 나타나는 패턴이다.
- 형용사+명사(Adjective+Noun) 형태의 매개변수(예, highTemperature)가 752개로 전체의 9%를 차지하였다.
- 동사+명사(Verb+Noun) 형태의 매개변수(예, update-List), 명사$_1$+명사$_2$+명사$_3$(Noun$_1$+Noun$_2$+Noun$_3$) 형태의

매개변수(예, telephoneAreaCode), 명사$_1$+전치사+명사$_2$(Noun$_1$+Preposition+Noun$_2$) 형태의 매개변수(예, passwordOfAccount)가 각각 608개(7%), 472개(6%), 368개(5%)로 거의 비슷한 분포를 보였다.

- 그 외 26개(0.3%)는 어떠한 패턴을 찾을 수 없는 매개변수들로 나타났다.

본 기법의 첫 번째 단계는 매개변수 내의 각 단어들 사이의 상관관계를 취득하여 그들을 온톨로지에 저장하는 것이다. 위에 서술된 패턴 조사 결과로부터 RESTful 웹 서비스 매개변수에 대한 온톨로지 변환 규칙(transformation rules)은 다음과 같다.

- **규칙-1: 명사$_1$+명사$_2$(Noun$_1$+Noun$_2$) 형태**
  매개변수가 명사$_1$의 속성(property)이 된다.
  - Parameter **propertyOf** Noun$_1$
  예를 들면, companyCode는 규칙-1을 따르며 다음과 같은 형태가 된다.
  - companyCode → companyCode **propertyOf** Company
- **규칙-2: 형용사+명사(Adjective+Noun) 형태**
  매개변수가 명사의 자식관계(subClass)가 된다.
  - Parameter **subClassOf** Noun
  예를 들면, highTemperature는 규칙-2를 따르며 다음과 같은 형태가 된다.
  - highTemperature → highTemperature **subClassOf** Temperature
- **규칙-3: 동사+명사(Verb+Noun) 형태**
  매개변수가 명사의 자식관계가 된다.
  - Parameter **subClassOf** Noun
  예를 들면, updateList는 규칙-3을 따르며 다음과 같은 형태가 된다.
  - updateList → updateList **subClassOf** List
- **규칙-4: 명사$_1$+명사$_2$+명사$_3$(Noun$_1$+Noun$_2$+Noun$_3$) 형태**
  매개변수가 명사$_1$의 속성이 된다.
  - Parameter **propertyOf** Noun$_1$
  예를 들면, telephoneAreaCode는 규칙-4를 따르며 다음과 같은 형태가 된다.
  - telephoneAreaCode → telephoneAreaCode **propertyOf** Telephone
- **규칙-5: 명사$_1$+전치사+명사$_2$(Noun$_1$+Preposition+Noun$_2$) 형태**
  매개변수가 명사$_2$의 속성이 된다.
  - Parameter **propertyOf** Noun$_2$
  예를 들면, passwordOfAccount는 규칙-5를 따르며 다음과 같은 형태가 된다.
  - passwordOfAccount → passwordOfAccount **propertyOf** Account

위와 같은 규칙들을 사용하여 온톨로지가 구축되고 나면, 두 번째 단계는 두 개념 간 매칭을 시키는 것이다. 두 개의 온톨로지는 다음 조건을 만족하면 매치된다.

- 어떤 개념이 다른 개념의 속성일 경우
  예를 들면, companyCode **propertyOf** Company
- 어떤 개념이 다른 개념의 자식관계인 경우
  예를 들면, highTemperature **subClassOf** Temperature

이 알고리즘은 Greedy 방식으로 진행되는데, 만일 두 개의 개념이 조건에 만족되면 유사도 점수는 1이 되고, 두 개의 개념이 조건에 만족되지 않으면 유사도 점수는 0이 되며 이들은 결과로부터 제거된다. 패턴 기반 시맨틱 분석 기법은 관련 없는 개념들의 매치를 피할 수 있으므로, 매치되는 후보 집합들은 클러스터링 기법에 의해 생성되는 결과보다 더욱 정확한 매치를 얻을 수 있다.

5. 결론

본 논문에서는 연관규칙 기반 클러스터링 기법에 패턴 기반 시맨틱 분석 기법을 추가한 새로운 시맨틱 온톨로지 구축 방법을 제안하였다. 본 연구의 핵심 내용은 RESTful 매개변수들에 대해 의미적으로 같은 개념들을 클러스터링으로 묶고, 매개변수 내에 있는 각 단어들 간의 계층관계를 형성하여 자동적으로 시맨틱 온톨로지를 구축하는 것이다. 이러한 자동 구축 방법에 따라 기존에 수작업으로 수행되고 있는 온톨로지 구축 작업이 보다 수월하게 진행될 수 있다.

참고문헌

[1] L. Richardson and S. Ruby, RESTful Web Services, O'Reilly, 2007

[2] R. Battle and E. Benson, "Bridging the Semantic Web and Web 2.0 with Representational State Transfer (REST)," Journal of Web Semantics, Vol. 6, pp. 61-69, 2008

[3] A. Hess and N. Kushmerick, "Learning to Attach Metadata to Web Services," In Proceedings of the International Semantic Web Conference, 2003

[4] X. Dong, A. Halevy, J. Madhavan, E. Nemes, and J. Zhang, "Similarity Search for Web Services," In Proceedings of VLDB, 2004

[5] M. Sabou, C. Wroe, C. Goble, and H. Stuckenschmidt, "Learning Domain Ontologies for Semantic Web Service Descriptions," Journal of Web Semantics, 3(4), 2005

[6] R. Fielding, Architectural Styles and The Design of Network-based Software Architectures, PhD thesis, University of California, 2000

[7] https://wadl.dev.java.net/

[8] R. Agrawal, T. Imielinski, and A. Swami, "Mining Association Rules between Sets of Items in Large Databases," Proceedings of the 1993 ACM-SIGMOD International Conference Management of Data, 1993

[9] L. Kaufman and P. J. Rousseeuw, Finding Group in Data: An Introduction to Cluster Analysis, John Wiley & Sons, New York, 1990

[10] H. Guo, A. Ivan, R. Akkiraju, and R. Goodwin, "Learning Ontologies to Improve the Quality of Automatic Web Service Matching," Proceedings of IEEE International Conference on Web Services(ICWS), 2007

[11] http://www.programmableweb.com

[12] http://crftagger.sourceforge.net/