웹 서비스 발견을 위한 클러스터와 온톨로지 매칭 알고리즘

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

Cluster and Ontology Matching Algorithms for Web Services Discovery

Yong-Ju Lee
Dept of Computer Engineering, Kyungpook National University

요 약

본 논문은 클러스터링 탐색 방법과 온톨로지 학습 방법을 융합하여 보다 더 효율적인 검색 방안을 제안한다. 이를 통해 키워드가 정확하게 일치하지 않더라도 사용자가 원하는 웹 서비스를 검색할 수 있고, 반대로 키워드가 일치하지만 사용자가 의도하지 않은 웹 서비스는 검색 결과에서 제거할 수 있다. 주된 아이디어는 매개변수들 사이의 숨은 시맨틱 개념을 찾아내어 온톨로지를 학습하고, 확장된 키워드 탐색 방법과 온톨로지 활용 방법을 혼합 사용하여 보다 지능적인 웹 서비스 매칭을 수행하는 것이다.

1. 서론

웹 서비스는 동적으로 느슨하게 결합된(loosely coupled) 서비스 지향 구조(SOA: Service Oriented Architecture)로 되어있다. 이에 반해 기존의 비즈니스 시스템들은 하부 시스템과 단단히 결합된(tightly coupled) 시스템 의존적인 구조로 되어있어 이식성 및 유지관리 비용이 많이 든다. 향후 소프트웨어 개발은 단순히 한 컴퓨터에서 머무는 것이 아니라 웹 서비스 기술을 이용하여 필요한 모듈을 인터넷에서 구할 수 있고 개발된 소프트웨어를 인터넷에 공개할 수 있다. 이러한 서비스 지향 구조는 '소프트웨어를 서비스로 보는(software as a service)' 중요한 개념이 된다.

현재 웹 서비스의 사용은 사용자들이 이미 알고 있는 몇 개의 서비스들을 이용하거나, 웹 서비스 저장소(예, UDDI, Xmethods, Web Service List)에서 키워드 탐색에 의해 서비스들을 발견하고 있다. 하지만 이러한 키워드 탐색 방법은 다음의 두 가지 이유 때문에 문제가 있다. (1) 나쁜 재현율: 기존의 키워드 탐색 방법은 웹 서비스에 대한 의미적인 정보들을 활용하지 못한다. 기존의 방법에서는 키워드가 정확히 일치하는 웹 서비스일 경우에만 발견이 되므로 사용자가 원하는 웹 서비스일지라도 키워드가 일치되지 않는다는 이유로 검색되지 않은 웹 서비스들이 존재한다. (2) 나쁜 정확률: 키워드는 사용자의 요구사항을 정확하게 표현하지 못한다. 검색 결과 중에는 사용자가 원하지 않지만 키워드가 포함된 수많은 웹 서비스들이 포함될 수 있다. 따라서 사용자는 이러한 결과 중에서 다시 원하는 웹 서비스를 찾아야 하는 불편함이 있다.

이러한 키워드 기반 탐색 방법의 한계를 극복하기 위한 기법으로서 시맨틱(semantic) 정보를 이용하는 온톨로지(ontology) 활용 방법이 있을 수 있다[1]. 웹 서비스 저장소에 추가적인 시맨틱 정보(예, WSDL-S, OWL-S)를 주석처리(annotation)하여 키워드와 일치하지 않은 웹 서비스일지라도 의미적으로 연관성 있는 서비스들에 대해 확장 검색한다. 그렇지만 온톨로지는 대부분 전문가의 수작업으로 구축되고 있으며, 시간 및 인적 제약 때문에 실용적인 온톨로지를 구축하기 어렵다. 또한 현시점에서 웹 서비스 전체에 대해 주석을 다시 단다는 것은 거의 불가능하게 보이며, 이러한 문제는 오늘날 웹 서비스의 확산과 발전을 가로막는 큰 저해 요인이 되고 있다.

다른 기법으로서 클러스터링(clustering) 방법을 이용한 웹 서비스 유사성(similarity) 탐색 방법[2]이 있다. 이 방법에서는 상호 연관성이 높은 단어들을 함께 묶어 클러스터를 형성하게 하고, 웹 서비스를 탐색할 때 사용자가 입력한 키워드뿐만 아니라 그것이 포함되어 있는 클러스터 내의 모든 단어들에 대해 탐색을 수행함으로써 보다 의미있는 검색이 되도록 한다. 그렇지만 이 방법은 연관성이 높은 단어들을 단지 한 클러스터에 묶어서 동의어(synonyms)처럼 취급할 뿐 객체지향 모델과 같은 계층관계(hierarchy)에 따라 유사도를 결정하는 시맨틱 기능은 제공하지 못하고 있다.

본 논문에서는 클러스터링 탐색 방법에 추가적으로 온톨로지를 자동 구축하여 보다 더 효율적인 검색을 할 수 있도록 제안한다. 이를 통해 키워드가 정확하게 일치하지 않더라도 사용자가 원하는 웹 서비스를 검색할 수 있고, 반대로 키워드가 일치하지만 사용자가 의도하지 않은 웹 서비스는 검색 결과에서 제거할 수 있다. 본 논문의 주된 아이디어는 매개변수(parameter)들 사이의 숨은 시맨틱 개념을 찾아내어 온톨로지를 학습(learning)하고, 확장된

키워드 탐색 방법과 온톨로지 학습 방법을 혼합 사용하여 보다 지능적인 웹 서비스 매칭을 수행하는 것이다.

2. 웹 서비스의 구조

본 논문의 매칭 알고리즘 이점을 알아보기 위해 먼저 웹 서비스의 구조에 대해 간단히 살펴본다. 각각의 웹 서비스는 그 기능과 인터페이스를 기술하고 있는 하나의 WSDL 파일과 연관되어 있다. 일반적으로 하나의 웹 서비스는 UDDI 비즈니스 레지스트리(Registry) 내에 자신의 WSDL 파일과 서비스에 대한 간단한 설명을 등록함으로 공개된다. 각 웹 서비스는 (그림 1)과 같이 오퍼레이션들의 집합으로 구성되어 있고, 각 오퍼레이션은 다수의 입출력 매개변수들로 이루어져 있다.

Web Service(1)
W1: QueryCustomerInfo Operation1: CustomerInfo Input: CustomerName, State Output: CustomerID Operation2: QueryCompany Input: CustomerAddress, ZipCode Output: CompanyCode
Web Service(2)
W2: CheckClientInfo Operation1: ClientInformation Input: ClientName, Province Output: ClientIdentification Operation2: CheckCountry Input: PersonAddress, PostalCode Output: CountryCode

(그림 1) 웹 서비스의 예

(그림 1)에서 W1과 W2의 Operation1은 입출력 매개변수들에 대해 토큰화(tokenization), 단어 확장, 그리고 시소러스(thesaurus)에 의한 동의어를 적용하면 두 오퍼레이션은 동일한 것임을 알 수 있다. 예를 들면, CustomerID는 Customer와 ID로 분리되고, ID는 Identification으로 확장되며, Customer와 Client, State와 Province는 동의어로 처리된다. 그러나 Operation2는 동의어 사용만으로 이들 간의 유사성을 발견할 수 없다.

W1, W2의 Operation2는 실제적으로 입력은 같은 개념으로 매치되어야 하고, 출력은 다르게 판단되어야 한다. 하지만 입력도 Customer와 Person이 동의어가 아니므로 다르게 취급된다. 이런 매치 형태는 단지 동의어만 사용하여 결정할 수 없고, 이들 간의 상관관계(relationship)를 해석하기 위한 온톨로지 정보가 필요하다. 즉, Person과 Customer는 동일한 개념(예, equivalentClass(Customer, Person))으로 취급되어야 한다. 한편, 출력 부분에서는 CompanyCode는 Company에 관한 내용이고 CountryCode는 Country에 관한 내용으로 판단되어(예, isProperty (CompanyCode, Company), isProperty(CountryCode, Country)) 이들은 매치되지 말아야 한다.

위의 보기에서 입출력 매개변수를 대표하는 단어를 토큰화하고 동의어를 적용하는 구문 분석 방법(syntactic analysis method)은 키워드를 일반화하여 검색 범위를 넓혀주고, 온톨로지 정보의 사용은 상관관계를 표현하여 깊이 있는 탐색을 유도한다. 이러한 두 방법을 결합함으로써 재현율(recall)과 정확률(precision) 둘 다 향상 시킬 수 있는 기법이 될 수 있다.

3. 구문 분석 방법

오퍼레이션 입출력 매개변수들 간에 유사성을 발견하는 것은 쉬운 일이 아니다. 왜냐하면, 입출력 매개변수 이름은 복합단어, 약어, 개발자의 명명(naming) 습관 등으로 인해 매우 다양해 질 수 있다. 따라서 WordNet과 같은 전자 사전을 바로 적용하기 어렵다. 또한 웹 서비스 오퍼레이션 내에는 일반적으로 매개변수들이 몇 개 존재하지 않으며, 이에 대한 충분한 설명도 거의 제공하고 있지 않다. 따라서 단어 빈도수를 기반으로 하는 TF/IDF와 같은 전통적인 IR 기법들은 잘 적용될 수 없다.

구문 분석 방법에서는 먼저 복합단어로 구성된 매개변수들을 파싱하여 텀(term)으로 분리한다. 그리고 POS (part-of-speech)와 불용어(stop-word) 필터링이 수행되고, 필요 시 단어 내 약어들이 확장된다. 그 후 동의어 리스트를 발견하기 위해 시소러스가 사용된다. 각 단계에 대한 자세한 내용은 아래와 같다.

복합단어 토큰화: 웹 서비스를 파싱하여 모든 단어들을 뽑아낸 후에 복합단어는 여러 개의 텀으로 분리한다. 예를 들면 ClientName은 Client와 Name으로 나눈다. 단어를 토큰화하기 위해 프로그래머들에 의해 사용되는 일반적인 명명 규칙을 조사할 필요가 있다. 본 연구에서는 빈칸, 하이픈(-), 언더스코어 문자(_), 문자 내 숫자 등과 같은 구분 문자를 사용하여 복합단어를 분리한다.

POS와 불용어 필터링: POS는 접두사 또는 어미 등 어근에 붙어있는 부분을 제거하는 알고리즘으로서 단어를 어근으로 분리하므로 같은 단어이지만 접미사나 어미의 변화에 의해 다른 단어로 인식되는 것을 막을 수 있다. 또한, 미리 만들어진 불용어 리스트에 의해 불용어들이 필터링된다. 본 연구에서 사용되는 불용어는 상용 검색 엔진에서 사용되는 것과 비슷한 and, or, the, is 등이 된다.

약어 확장: 약어(abbreviation)는 완전한 단어로 확장된다. 예를 들면 CustomerInfo는 CustomerInformation으로 확장이 수행된다. 여러 개의 확장 단어 후보가 존재할 경우 복수개의 단어 확장도 가능하다. 따라서 CustPurch는 CustomerPurchase와 CustomaryPurchase 등으로 확장될 것이다.

동의어 탐색: 텀들에 대한 동의어 리스트를 발견하기 위해 WordNet 시소러스를 사용한다. 시소러스란 같은 의미를 갖고 있는 단어이지만 단어의 철자가 다른 경우 이를 해결하기 위해서 제안된 동의어 사전이다. 예를 들어 “bike”와 "bicycle"은 같은 의미를 같고 있으나 서로 철자가 틀리므로 다른 단어로 인식될 수 있다.

유사도 계산: 하나의 쿼리와 서비스 저장소로부터 매치되는 임의의 후보 매개변수 쌍을 (Q, S_k)라 하자. 매개변수 Q와 S_k에는 각각 m과 n개의 텀들이 있다고 가정하자.

$$Q = q_1, q_2, \cdots q_i, \cdots q_m$$

$$S_k = s_1, s_2, \cdots s_j, \cdots s_n$$

Q의 q_i와 S_k의 s_j 간의 매치를 고려할 때, 변개변수 Q와 S_k 사이의 유사성은 다음과 같이 계산된다.

$$Similarity(Q,\ S_k) = \frac{2 * \sum match(q_i, s_j)}{m+n}$$

여기서, $match(q_i, s_j) = \begin{cases} 1 & \text{if } sucess \\ 0 & \text{if } fail \end{cases}$

$$i = 1, 2, \cdots m,\ j = 1, 2, \cdots n$$

예를 들면, CustomerCare와 ClientSearch 사이의 유사성은 0.5이다. 왜냐하면 Customer와 Client는 동의어지만, Care와 Search는 매치가 실패하기 때문이다. 매치되는 서비스 개수를 p라 할 때, 주어진 쿼리에 대한 최상의 매칭 서비스는

$$BestService = max\{Similarity(Q, S_k)\} \text{ for all } 1 \le k \le p$$

로 주어지고 Similarity(Q, S_k) 값은 우선순위 리스트를 위해 정렬될 수 있다.

이러한 유사성은 State와 Province, CustomerInfo와 ClientInformation과 같은 매개변수 매치는 허용 하지만 Customer와 Person과 같은 동일한 개념의 매치는 허용하지 못한다. 이는 구문 분석 방법에서는 텀들 사이의 상관관계는 알 수 없기 때문에 발생되는 현상이다. 이러한 문제들은 다음의 온톨로지 학습 방법(ontology learning method)을 이용하여 해결할 수 있다.

4. 온톨로지 학습 방법

본 연구의 핵심 내용은 웹 서비스 매개변수들에 대해 의미적으로(semantically) 같은 개념들을 묶고(clustering), 각 텀들 간의 계층관계(hierarchy)를 구축하여 텀들 사이에 숨겨져 있는 시맨틱 개념을 활용하는 것이다.

웹 서비스의 오퍼레이션 내에는 일반적으로 매개변수들이 몇 개 존재하지 않기 때문에 기존의 전통적인 클러스터링 알고리즘들은 직접 적용할 수 없다. 왜냐하면, IR 응용에서는 동의어가 동일한 도큐멘트에 발생하는 경향이 높은 반면에, 웹 서비스에서는 하나의 오퍼레이션 내에 같은 입출력 매개변수는 거의 발생되지 않기 때문이다. 따라서 기존의 기법과는 다른 새로운 클러스터링 알고리즘의 적용이 요구된다.

매개변수들을 토큰화하여 텀으로 분리한 후, 관련성이 많은 텀들에 대해 클러스터를 형성하면 이 클러스터는 각각의 단어가 아닌 하나의 의미있는 개념을 나타낸다. 이러한 클러스터는 "매개변수들이 동시에 자주 나타난다면, 그것들은 같은 개념을 나타내는 경향이 있다"는 가정 하에 하나의 특별한 연관규칙(association rules)[2]에 따라 만들어 진다.

연관규칙 $\mathcal{R}$은 조건부와 결과부로 구성되며 텀 t_1이 일어나면 t_2가 일어난다는 의미로 다음과 같이 표현될 수 있다.

$$\mathcal{R}: t_1 \rightarrow t_2$$

따라서 연관규칙을 탐사하는 것은 적절한 텀 t_1과 t_2를 선택하는 문제로 볼 수 있으며 이를 위해 몇 가지 척도를 고려하고 있다. 우선 규칙 $\mathcal{R}$에 대한 지지도(support)와 신뢰도(confidence)는 각각 다음과 같이 정의된다.

지지도 = 입출력에 t_1이 나타날 확률

$$= \frac{\| t_1\text{을 포함하는 } IO\text{의 수} \|}{\| IO\text{의 전체 개수} \|}$$

신뢰도 = 입출력에 t_1이 주어졌을 때, t_2가 나타날 확률

$$= \frac{\| t_1, t_2\text{를 둘다 포함하는 } IO\text{의 수} \|}{\| t_1\text{를 포함하는 } IO\text{의 수} \|}$$

여기서 신뢰도가 임계치 δ보다 크면(즉 $t_1 \rightarrow t_2$(신뢰도$>\delta$)), 텀 t_1과 t_2는 밀접하게 연관되었다고 말할 수 있다. 이 알고리즘은 결과적으로 높은 점수(score)를 갖도록 클러스터를 형성하는 것이 목표이다. 이때 높은 점수는 cohesion(한 클러스터 내의 텀들과의 응집력)은 높고, correlation(다른 클러스터 텀들 간의 상호관계)은 낮은 것을 의미한다.

$$Score(C) = \frac{cohesion}{correlation}$$

$$cohesion = \frac{\| i \rightarrow j(\text{신뢰도} > \delta)\text{인 개수} \|}{\| C_1 \| \ \| (C_1 - 1) \|},$$

여기서, $i,j \in C_1$, $i \neq j$, C_1은 클러스터

$$correlation = \frac{\| i \rightarrow j(\text{신뢰도} > \delta)\text{인 개수} \| + \| j \rightarrow i(\text{신뢰도} > \delta)\text{인 개수} \|}{2 \| C_1 \| \ \| C_2 \|}$$

여기서, $i \in C_1$, $j \in C_2$, C_1, C_2는 클러스터

이러한 클러스터링 기법을 이용한 웹 서비스 탐색에서는 단순히 입출력 매개변수 텀들의 빈도수에 의존하는 것이 아니라 각 텀들 간의 상호연관성을 이용해 관련된 단어들 끼리 클러스터링 함으로써 보다 효과적인 웹 서비스의 검색이 가능하게 한다. 그러나 이 기법은 연관성 높은 단어들을 한 클러스터에 묶어서 단지 동일한(equivalent) 개념처럼 취급할 뿐 계층관계에 따라 사용자의 요구사항을 정확하게 표현하는 시맨틱 기능은 제공하지 못하고 있다.

계층관계 온톨로지 활용 기법은 사람들이 단어를 조합하여 복합단어로 된 매개변수를 만들 때 일반적으로 비슷한 패턴을 사용하는 경향이 있다[3]는 관찰로부터 시작한다. 이러한 패턴들은 다음과 같은 형태로 나타난다.

1. 명사(1) + 명사(2) (예,CompanyID)
2. 접두사 + 명사(1) + 명사(2) (예,virtualAccountID)
3. 형용사 + 명사 (예,virtualAccount)
4. 명사(1) + 전치사 + 명사(2) (예,passwordOfAccount)
5. 동사 + 명사 (예,enrollAccount)

첫 번째 단계로 각 텀들의 상관관계를 취득하여 그들을 온톨로지에 저장한다. 변환 룰(rule)은 <표 1>과 같다.

<표 1> 상관관계 변환 룰

룰	패턴 Example	상관관계 Example
1	명사(1) + 명사(2)	isProperty(매개변수, 명사(1))
	CompanyID	isProperty(CompanyID, Company)
2	접두사 + 명사(1) + 명사(2)	isProperty(매개변수, 접두사 + 명사(1) IS-A(매개변수, 명사(2))
	virtualAccountID	isProperty(virtualAccountID, virtualAccount) IS-A(virtualAccountID, ID)
3	형용사 + 명사	IS-A(매개변수, 명사)
	virtualAccount	IS-A(virtualAccount, Account)
4	명사(1) + 전치사 + 명사(2)	IS-A(매개변수, 명사(2))
	passwordOfAccount	IS-A(passwordOfAccount, Account)
5	동사 + 명사	IS-A(매개변수, 명사)
	enrollAccount	IS-A(enrollAccount, Account)

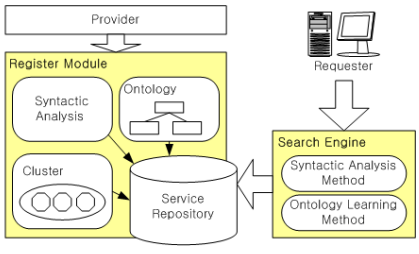
두개의 온톨로지 개념은 다음 조건이 만족되면 매치된다. (1) 어떤 개념이 다른 개념의 속성(property)일 경우(예, companyID는 company의 속성), (2) 어떤 개념이 다른 개념의 자식관계(subclass)인 경우(예, virtualAccount는 Account의 자식관계)

$$\text{Similarity}(Q, S_k) = \begin{cases} 1 & \text{if } sucess \\ 0 & \text{if } fail \end{cases}$$

위의 변환 룰을 적용함에 따라 관련없는 개념들 사이의 매치를 피할 수 있다. 예를 들면, companyID와 countryID는 각각 다른 개념의 속성이므로 매치에서 배제된다. 계층관계 온톨로지 학습 기법은 관련없는 개념들의 매치를 피할 수 있으므로, 매치되는 후보 집합들은 키워드 기반 탐색기법에 의해 생성되는 결과보다 더욱 정확한 매치를 얻을 수 있다.

5. 웹 서비스 매칭 알고리즘

본 논문에서 제안하는 매칭 알고리즘의 기본적인 원리는 (그림 2)와 같다. 먼저 사용자는 원하는 웹 서비스를 구문 분석 방법에 의해 서비스 저장소에서 찾는다. 그러나 이 방법에서는 텀들 사이의 상관관계를 알 수 없기 때문에 온톨로지 학습 방법을 추가하여 시맨틱 검색이 가능하도록 한다.



(그림 2) 구문 분석 및 온톨로지 학습 방법의 원리

온톨로지 학습 방법은 클러스터링 기법을 사용하여 클러스터 안의 모든 텀들에 대해 검색을 수행한다. 다음으로 계층관계 온톨로지 활용 기법에 의해 검색 키워드와 웹 서비스 간에 계층관계 조건이 체크된다. 따라서 검색 키워드와 웹 서비스 문서의 내용이 일치하지 않더라도 의미적으로 같은 웹 서비스를 검색할 수 있고, 검색된 웹 서비스들 중에서도 사용자가 원하지 않는 웹 서비스를 온톨로지를 통해 검색 결과에서 제거할 수 있다.

본 알고리즘은 이전에 우리가 제안한 SOA 기반 웹 서비스 조합[4]을 구현하는데 유용하게 사용될 수 있다. [4]는 쿼리 Q의 입력 매개변수를 사용하여 원하는 출력을 산출해 낼 수 있는 웹 서비스들을 찾는 것이다. 이를 위해서 선택되는 웹 서비스는 반드시 쿼리의 출력항목을 포함하고 있어야만 하고, 이 서비스의 입력 매개변수는 쿼리의 입력항목에 포함되어 있어야만 한다. 이러한 과정을 기반으로 웹 서비스 매칭 알고리즘을 작성하면 (그림 3)과 같다. (그림 3)에서 Discovery( ) 함수는 쿼리문 Q를 서비스 저장소에 있는 모든 웹 서비스 S들과 비교한다. 만일 매치가 발견되면 기록되고 우선순위에 의해 정렬된다. Match( ) 함수는 먼저 쿼리문 출력 매개변수 Q.Os를 웹 서비스 출력 매개변수 S.Os와 비교하여 유사도를 계산하고, 매치가 실패하지 않는다면 반대로 웹 서비스 입력 매개변수 S.Is와 쿼리문 입력 매개변수 Q.Is를 비교한다.

```
Algorithm: matching algorithm for web services
Input: query Q
Output: ranked list of matching services

Discovery(Q):
    for all S in ServiceRepository
        if Match(Q, S)
            then record.append(S)
        endif
    endfor
    return sorting(record)

Match(Q, S):
    outputMatch(Q.Os, S.Os)
    inputMatch(S.Is, Q.Is)
```

(그림 4) 웹 서비스 매칭 알고리즘

6. 결론

본 논문에서는 구문 분석 방법과 온톨로지 학습 방법을 혼합 사용한 보다 지능적인 웹 서비스 매칭 알로리즘을 제안하였다. 향후 과제로는 성능 분석을 통한 제안된 알고리즘의 우수성을 보이는 것이다.

참고문헌

[1] M. Paolucci, T. Kawamura, T. R. Payne and K. Sycara, "Semantic Matching of Web Services Capabilities," Proceedings of ISWC, 2002

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

[3] H. Guo, A. Ivan, R. Akkiraju, and R. Goodwin, "Learning Ontologies to Improve the Quality of Automatic Web Service Matching," Proceedings of ICWS, 2007

[4] 이용주, "SOA의 핵심 기술: 반자동 웹 서비스 조합 기법," 제29회 한국정보처리학회 춘계학술발표대회 논문집, 2008년 5월, pp. 393-396.