웹 서비스 검색을 위한 시맨틱 매칭 엔진

양승훈° 이대욱 권준호 이석호
서울대학교 전기 컴퓨터 공학부
{ysh0089° ,dwlee, bluerain}@db.snu.ac.kr, shlee@snu.ac.kr

Semantic Matching Engine for Searching Web Services

SeungHoon Yang°, DaeWook Lee, JoonHo Kwon, SukHo Lee
School of Electrical Engineering and Computer Science, Seoul National University

요 약

인터넷망의 지속적인 발달과 함께 웹 애플리케이션 개발 방법으로 XML 기반의 웹 서비스가 부각되면서 많은 웹 서비스들이 개발되었고, 점차 더 많은 웹 서비스들이 개발될 것으로 예상된다. 이처럼 급격하게 늘어나는 웹 서비스들 중에서 사용자가 원하는 웹 서비스 찾는 것이 중요한 이슈로 부각되고 있다.

그러나 현재의 웹 서비스 검색 표준인 UDDI 레지스트리는 키워드 기반이기 때문에 검색 성능의 한계점을 갖고 있다. 최근에 이러한 한계를 극복하고자 하는 많은 연구가 진행되고 있지만 아직은 많이 부족한 상황이다. 따라서 본 논문에서는 비록 키워드가 일치하지 않더라도 사용자가 원하는 웹 서비스를 찾을 수 있도록 웹 서비스 표준인 UDDI 레지스트리에 시맨틱 매칭 엔진(semantic matching engine)이라는 추가적인 시맨틱 레이어를 추가하여 재현율(recall)과 정확률(precision)을 모두 향상 시킬 수 있는 시스템을 제안한다.

1. 서 론

1990년대 인터넷의 등장과 함께 웹은 양적으로나 질적으로나 급격한 성장을 해 왔다. 그러나 현재의 웹은 컴퓨터가 의미를 이해하고 자동으로 처리할 수 없고, 단지 사람이 보고 이해할 수 있는 HTML을 이용해 표현하고 있다. 이러한 문제점 때문에 1997년 W3C(World Wide Web Consortium)에서 컴퓨터가 정보의 의미를 이해하고 의미를 조작할 수 있는 웹인 시맨틱 웹(semantic web)[1]을 제안하였다. 또한 21세기 인터넷망의 지속적인 발달과 함께 웹 애플리케이션 개발 방법으로 XML 기반의 웹 서비스(web service)[2]가 부각되면서 활발한 연구가 진행되고 있다. 이러한 상황 속에서 웹 서비스와 웹 응용은 급격하게 증가하고 있으며, 이와 같은 추세라면 웹 서비스의 수는 기하급수적으로 늘어날 것으로 예상된다. 따라서 수많은 웹 서비스들 중에서 사용자가 원하는 적절한 웹 서비스를 어떻게 검색하는지가 가장 중요한 관심사가 될 것이다. 하지만 기존이 키워드 중심의 검색 시스템[3,4,5,6]은 여러 가지 이유로 한계를 드러내고 있다.

본 논문에서는 사용자로부터 키워드를 입력 받아 적절한 웹 서비스를 검색하는 것에 초점을 맞추고 있다. 키워드 기반 검색의 한계점을 극복하기 위해서 많은 연구들이 진행되고 있다. 그 중에 Washington 대학에서 개발된 Woogle[7,8]은 클러스터링(clustering) 방법을 사용한 웹 서비스 검색엔진을 제안하고 있다. 그러나 이 역시 웹 서비스에 포함되어 있는 단어들을 갖고 클러스터링을 하기 때문에 키워드 기반 검색의 한계점을 크게 벗어나지는 못하였다. 따라서 본 논문에서는 기존의 클러스터링 알고리즘에 추가적으로 온톨로지를 이용하여 보다 더 효율적인 검색을 할 수 있는 방법을 제안한다.

본 논문은 다음과 같이 구성된다. 2장에서는 관련 연구로서 웹 서비스 전반에 관한 설명과 Woogle에서의 클러스터링 방법을 사용한 웹 서비스 검색 방법에 대해서 기술하였다. 3장에서는 본 논문에서 제안하는 방법인 시맨틱 매칭 엔진에 대해서 기술하였고, 마지막 4장에서는 결론과 향후 연구에 관해 기술하였다.

2. 관련연구

2.1 웹 서비스(web service)

그림 1은 웹 서비스의 등록, 검색, 호출의 전체 구조를 나타낸다. 웹 서비스는 사용자가 원하는 정보를 동적으로 제공하기 위해 애플리케이션의 상호 운용을 가능하게 하는 기술이다. 웹 서비스 표준에는 SOAP, WSDL, UDDI가 있다. SOAP(Simple Object Access Protocol)[9]은 W3C의 XML 표준 프로토콜이고, WSDL(Web Service Description Language)[10]은 웹 서비스를 기술하기 위한 언어이다. UDDI(Universal Description Discovery and Integration)[11]는 웹 서비스에 대한 디렉토리 서비스를 지원하는 표준 레지스트리로, 웹 서비스의 등록과 검색이 가능하다.

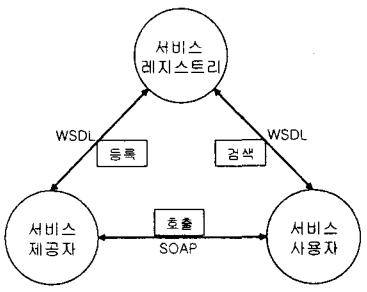

그림 1. 웹 서비스의 구성 요소

서비스 제공자는 제공하고자 하는 웹 서비스를 UDDI 레지스트리에 등록한다. 사용자는 원하는 웹 서비스를 UDDI 레지스트리를 통해 검색하고, 적절한 웹 서비스를 찾아 이용하게 된다.

이 때, 웹 서비스 검색의 결과에 대한 평가 척도로서 재현율(recall)과 정확률(precision)이 있다. 재현율은 사용자의 질의에 적합한 전체 웹 서비스 중에서 검색된 적합 웹 서비스의 비율을 나타낸다. 즉, 적합한 웹 서비스를 얼마나 검색했는지를 나타내며, 재현율이 높을수록 검색엔진이 적합한 웹 서비스를 찾는데 뛰어나다고 할 수 있다. 정확률은 검색된 웹 서비스 중에서 사용자에 질의에 적합한 웹 서비스의 비율을 나타낸다. 이는 검색 결과 중에서 사용자 질의에 적합한 웹 서비스들이 얼마나 되는지를 나타내며, 정확률이 높을수록 검색엔진이 불필요한 웹 서비스를 찾지 않는 능력이 뛰어나다고 할 수 있다.

$$\text{재현율} = \frac{|\text{검색된 적합 웹 서비스}|}{|\text{전체 적합 웹 서비스}|}$$

$$\text{정확률} = \frac{|\text{검색된 적합 웹 서비스}|}{|\text{검색된 웹 서비스}|}$$

그러나 웹 서비스 표준 레지스트리인 UDDI는 키워드 기반이라는 점 때문에 웹 서비스 검색에 있어서 몇 가지 문제점을 갖는다. 다시 말해서 사용자가 키워드를 갖고 원하는 웹 서비스를 찾고자 할 경우 재현율과 정확률 측면에서 모두 좋지 못한 결과를 나타내는데, 그 이유는 다음과 같다. 첫째, 사용자가 원하는 웹 서비스이지만 키워드가 포함되어있지 않다는 이유로 검색되지 않은 웹 서비스들이 존재한다. 둘째, 검색 결과에 사용자가 원하지 않지만 키워드가 포함된 수 없이 많은 웹 서비스들이 포함되어 있을 것이다. 따라서 사용자는 그러한 웹 서비스들 중에서 다시 원하는 웹 서비스를 찾아야 하는 불편함이 있다.

2.2 클러스터링 방법을 사용한 웹 서비스 검색

위에서 설명한 것과 같이 UDDI 레지스트리의 키워드 기반이라는 한계점을 극복하기 위해 많은 연구들이 진행되고 있다. 이와 같은 연구 중에 하나가 Washington 대학의 웹 서비스 검색 엔진 Woogle이다. 이 연구에서는 클러스터링 방법을 사용한 웹 서비스 검색 엔진을 제안하였다.

전통적인 검색 방식인 키워드 검색 방식은 웹 서비스의 파라미터 이름(parameter names)을 이용해서 검색을 한다. 그러나 단순한 파라미터 이름은 웹 서비스의 내포된 의미를 적절히 나타내지 못하고, 또한 개발자에 의존적이어서 다분히 주관적이며, 적용된 제목 규칙에 따라 다르기 때문에 사용자가 원하는 웹 서비스를 검색하는데 한계를 갖는다. 따라서 상호 연관성이 큰 단어들끼리 모아 클러스터를 형성하게 해서 웹 서비스를 검색 할 경우, 사용자가 입력한 키워드뿐만 아니라 그것이 포함되어있는 클러스터 내의 모든 단어를 이용해 검색함으로써 보다 의미 있는 검색이 이루어진다.

파라미터 이름은 주로 연속된 단어의 연결이므로 이를 각각의 단어로 분리하였다. 예를 들면 zip code를 입력하면 local time을 알려주는 웹서비스의 이름이 LocalTimeByZipCode라고 하였을 때, 이를 대문자를 기준으로 {Local, Time, By, Zip, Code}와 같이 나눠서 각각의 단어를 텀(term)이라 부른다. 각각의 텀들은 관련이 많은 텀들끼리 클러스터를 형성하고 이 클러스터는 각각의 단어가 아닌 하나의 의미 있는 개념을 나타낸다. 이러한 클러스터는 "파라미터 이름이 동시에 자주 나타난다면, 그것들은 같은 개념을 나타내는 경향이 있다."는 가정 하에 그림 2의 규칙[7,8,12]에 따라 만들어진다.

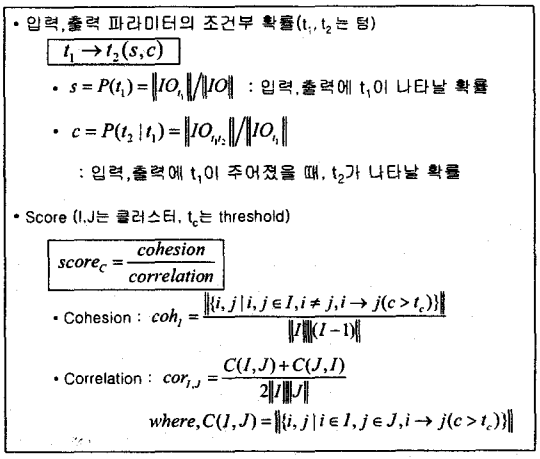

그림 2. 결합 규칙

위 그림에서 c가 threshold t_c보다 크면, 텀 t1은 텀t2와 밀접하게 연관되었다고 할 때($t_1 \rightarrow t_2(c>t_c)$), 결과적으로 높은 $score_c$를 갖도록 클러스터를 형성하는 것이 목표이다. 이때, 높은 $score_c$는 cohesion(한 클러스터 내의 텀들간의 연관)은 높고, correlation(다른 클러스터에 있는 텀들간의 연관)은 낮은 것을 의미한다.

이러한 클러스터링 기법을 이용한 웹 서비스 검색에서는 단순히 입·출력 파라미터 이름의 빈도수에 의존하는 것이 아니라 각 텀들간의 상호 연관성을 이용해 관련된 단어들 끼리 클러스터링 함으로써 보다 효과적인 웹 서비스의 검색이 가능하다. 그러나 클러스터링 방법만으로는 사용자가 원하는 모든 웹 서비스를 찾아내기에는 부족한 점이 많다. 클러스터링 방법을 보면 클러스터가 형성되는 과정에서 사용되는 텀들은 모두 웹 서비스들에 포함되어 있는 것들이다. 따라서 클러스터링 방법은 단순히 웹 서비스들에 포함되어 있는 텀들을 상호 연관성을 이용해 모아 놓은 것이기 때문에 비록 키워드 기반의 검색 엔진에 비해 검색 성능은 좋아졌을지라도 키워드 검색 방법의 한계점을 크게 벗어나지 못했다고 볼 수 있다. 다시 말해서 단어는 다르지만 의미상으로 같은 단어

를 포함하는 웹 서비스를 찾아내지 못 할 수도 있다는 것이다. 예를 들어 주소와 관련된 클러스터 {city, state, street, zip, code}가 있다고 가정할 때, 이를 이용한 검색은 단순히 하나의 키워드를 이용해 검색한 결과 보다 더 좋은 검색 결과를 나타낼 것이다. 그러나 이 클러스터는 location이나 address 같이 의미상으로 비슷한 웹 서비스를 검색해 줄 수 없고, 또한 클러스터에 있는 텀인 code의 경우, 사용자는 우편번호의 code를 원한 것인데, 컴퓨터 code와 같이 사용자의 의도와는 거리가 먼 웹 서비스들이 검색 결과로 반환될 수 있다.

3. 본 론

본 논문에서는 Woogle에서 사용된 파라미터 이름을 클러스터링하는 방법에 온톨로지를 적용하여 재현율과 정확률을 모두를 향상시킬 수 있는 시스템을 제안한다. Woogle의 장점은 그대로 살리면서 한계점으로 언급했던 사항을 보완하기 위해, 클러스터링 방법을 사용하되, 공유된 개념에 대한 시맨틱 매칭이 필요하다. 이를 위해서 클러스터링 방법에 추가적으로 온톨로지를 사용하여 보다 효율적인 검색이 가능하도록 하였다.

핵심 부분은 시맨틱 매칭 엔진(semantic matching engine)으로서 UDDI 레지스트리에 추가적인 시맨틱 레이어(semantic Layer)를 추가한 것이다. 이 시스템에서는 검색 키워드와 일치하지 않는 웹 서비스일지라도 의미적으로 같은 웹 서비스에 대하여 Woogle에서 검색하지 못한 웹 서비스까지 검색하기 위해서 클러스터링된 텀들과 온톨로지 클래스간의 매핑(mapping)을 통해 재현율을 높혔다. 또한 이에 따라 검색된 웹 서비스들 중에 사용자의 의도와 맞지 않은 웹 서비스 들을 제거하기 위해 온톨로지 라이브러리를 이용하였다. 관련된 온톨로지 리스트를 사용자에게 보여주고, 사용자가 원하는 온톨로지를 선택하면, 선택된 온톨로지에 해당하는 웹 서비스들만 검색함으로써 정확률을 높혔다.

3.1 시스템 구조

그림 3은 시맨틱 매칭 엔진 시스템의 전체 구조를 보여준다. 시맨틱 매칭 엔진은 UDDI 레지스트리에 시맨틱 레이어를 추가한 것으로, 사용자가 원하는 웹 서비스를 찾기 위해 사용자로부터 키워드를 입력받아 이와 관련된 웹 서비스들을 검색해 주는 시스템이다. 기존의 UDDI 레지스트리 위에서 동작하기 때문에 웹 서비스 표준을 따르고, 구조의 특별한 변경 없이 시맨틱 레이어의 추가만으로 시스템이 구성되어 간단하다. 이때, 사용자는 클러스터나 온톨로지에 관한 지식 없이 단순히 키워드만으로 원하는 웹 서비스를 검색할 수 있다.

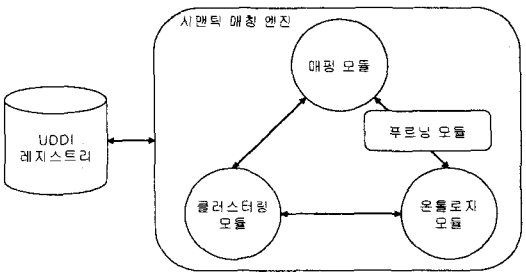

그림 3. 시맨틱 매칭 엔진의 시스템 구조

매칭 엔진은 다시 4가지 모듈로 구성된다. 클러스터링 모듈(clustering module)은 Woogle에서 사용된 알고리즘을 기반으로 하여 웹 서비스들의 파리미터 이름을 클러스터링한다. 온톨로지 모듈(ontology module)은 온톨로지와 온톨로지 라이브러리를 생성하고, 유지한다. 온톨로지 라이브러리는 시스템의 모든 온톨로지의 이름과 검색 가능한 속성들을 포함하고 있다. 매핑 모듈(mapping module)은 클러스터링 모듈의 각 클러스터 내부 텀과 온톨로지 모듈의 온톨로지 클래스를 매핑한다. 프루닝 모듈(pruning module)은 사용자의 의도에 맞지 않은 웹 서비스들을 제거하는 구조로, 매핑 된 모든 온톨로지 리스트를 사용자에게 보여주고, 그 중에서 사용자가 선택한 온톨로지에 해당하는 웹 서비스들만을 검색 결과로 보여준다.

3.2 재현율을 높이기 위한 방법

시맨틱 검색 엔진에서 클러스터링 모듈과 온톨로지 모듈, 매핑 모듈은 사용자가 웹 서비스를 검색할 때 재현율을 높이기 위한 구조이다. 기존 클러스터링 기법만으로는 찾아내지 못한 웹 서비스들을 찾아내기 위해 클러스터링 모듈에 온톨로지 모듈을 추가하였다. 매핑 모듈을 통해 클러스터링 모듈에 형성되어 있는 클러스터의 텀과 온톨로지 모듈의 온톨로지 클래스간에 매핑을 하였다.

기존 연구에서는 클러스터만을 사용하여 입력된 키워드가 포함되어 있는 클러스터의 단어들을 포함하고 있는 웹 서비스들을 검색하였으나, 온톨로지를 추가함으로써 클러스터 내에 포함되어있는 단어와 일치하지 않더라도 그 단어와 매핑되어 있는 온톨로지 클래스 내의 의미상으로 같은 웹 서비스들을 검색하게 되어 보다 높은 재현율을 얻을 수 있다. 그림 4와 5는 Woogle에서의 클러스터링 방법만을 사용한 시스템과 본 논문에서 제안하는 온톨로지를 추가하는 시스템의 검색 결과를 각각 나타낸 것이다.

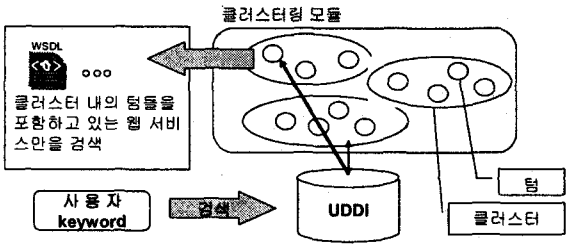

그림 4. 클러스터링 방법만 사용했을 때 검색 결과

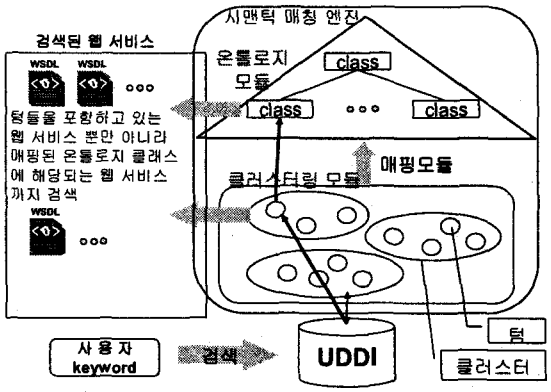

그림 5. 온톨로지를 추가했을 때 검색결과

시맨틱 매칭 엔진은 기존 클러스터링 방법의 장점을 그대로 사용하면서 온톨로지 매핑을 통해 입력된 키워드와 관련된 보다 많은 웹 서비스들을 검색할 수 있다. 단순히 클러스터링 방법만 사용한다면, 이는 클러스터 내의 텀들을 이용한 키워드 검색 방식에 불과하다. 그러나 온톨로지를 이용한 클러스터 내의 각 텀에 대한 의미상 매칭을 한다면, 클러스터링 방법만 사용할 때에 비해 보다 높은 재현율을 얻을 수 있다.

3.3 정확률을 높이기 위한 방법

시맨틱 매칭 엔진의 시스템 구조에서 위에서 설명한 세 가지 모듈이 재현율을 높이기 위한 구조라고 한다면, 프루닝 모듈은 정확률을 향상시키기 위한 구조이다. 재현율이 아무리 높다하더라도 검색 결과에 사용자의 의도와는 먼 웹 서비스들이 다수 포함되어 있다면 아무 소용이 없을 것이다. 물론 앞에서 제안한 방법을 통해 웹 서비스를 검색했다 하더라도 그 결과에는 사용자가 원치 않는 다수의 웹 서비스들이 존재할 것이다. 이러한 경우 사용자는 검색된 수많은 웹 서비스들 중에 자신이 찾고자 하는 웹 서비스를 찾아야 한다. 따라서 적합하지 않은 웹 서비스를 걸러내는 과정이 필요하다. 다시 말해 정확률을 높이기 위한 방법이 필요한데, 본 논문에서는 프루닝 모듈을 통해 적합하지 않은 웹 서비스들을 걸러내고 사용자가 원하는 것에 가까운 웹 서비스들만이 검색되도록 하는 방법을 제안한다.

사용자가 키워드를 입력하였을 경우 이를 포함하는 클러스터 내의 모든 단어가 여러 가지 온톨로지의 클래스와 매핑된다. 그러나 이러한 매핑은 의미상 매핑이 아닌 단순히 클러스터 내의 텀과 관련된 모든 온톨로지의 클래스와의 매핑이므로 이중에는 사용자가 전혀 의도하지 않은 여러 온톨로지들도 포함된다. 실제적으로 사용자가 원하는 웹 서비스는 특정 온톨로지의 클래스에 해당되는 웹 서비스들일 확률이 높다. 따라서 온톨로지 라이브러리를 미리 갖고 있어서 클러스터 내의 단어들이 매핑되었을 때, 해당되는 온톨로지들의 이름과 각 온톨로지에서 검색 가능한 속성들을 사용자에게 반환 해 준다. 사용자는 반환 된 온톨로지 리스트에서 원하는 온톨로지를 선택하고 그에 따른 웹 서비스들만 검색하게 함으로써 사용자가 원치 않는 웹 서비스의 검색을 크게 줄일 수 있어 보다 정확한 검색을 할 수 있게 된다.

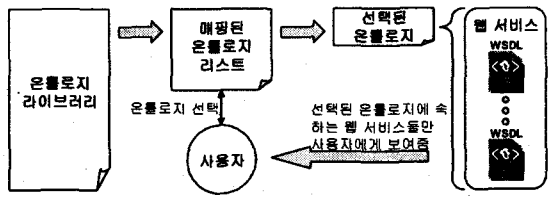

그림 6. 프루닝 과정

온톨로지 라이브러리는 시스템에 있는 모든 온톨로지들의 이름과 속성 정보를 리스트로 갖고 있다. 사용자는 반환 된 온톨로지의 이름과 속성들을 살펴봄으로써 자신이 찾고자 하는 웹서비스가 속하는 온톨로지를 쉽게 선택할 수 있다.

3.4 웹 서비스 검색 시나리오

사용자가 특정 지역의 날씨 정보를 알기 위해서 'zip'이라는 키워드를 이용해 날씨 관련 웹 서비스를 검색한다고 가정한다.

키워드로 'zip'이 들어오면 시맨틱 매칭 엔진에서 먼저 'zip'이란 텀이 포함되어 있는 클러스터를 검색한다. 검색된 클러스터가 주소와 관련된 클러스터({city, state, street, zip, code})라고 했을 때, 클러스터 내의 모든 텀에 대해서 매핑 된 온톨로지 클래스를 찾는다. 이때 해당 온톨로지의 이름과 속성정보를 온톨로지 라이브러리에서 찾아 사용자에게 반환 해 준다. 반환 된 온톨로지 이름이 {Airport, Delivery, Location, Weather ...}라고 가정하였을 때, 사용자는 반환 된 온톨로지 중에서 자신이 관심이 있는 Weather 온톨로지를 선택한다. 매칭 엔진에서는 각 텀이 매핑 된 온톨로지 클래스 중 Weather 온톨로지의 클래스에 속하는 웹 서비스들만을 사용자에게 결과로 보여준다. 그림 7은 위에서 설명한 일련의 과정을 그림으로 도시한 것이다.

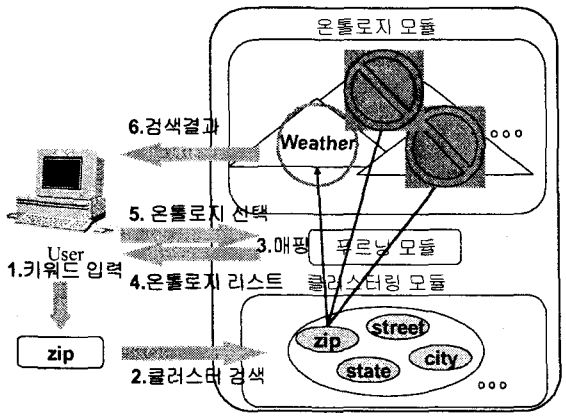

그림 7. 검색 과정

3.5 비교 분석

위 과정을 거쳐 검색된 웹 서비스들은 클러스터링 기법만 사용한 Woogle에 비해 재현율과 정확률 모두 뛰어난 성능을 보인다. 예를 들어 사용자가 'zip'을 검색했을 때 결과를 쉽게 비교하기 위해 간략화 해서 표현하면 그림 8,9와 같다. 5개의 웹 서비스(W1~5)를 사용해 예를 들었고 각각의 웹 서비스는 웹서비스 이름, 연산 이름, 입·출력 파라미터 이름으로 나타내었다.

W1 : GlobalWeather
- Operation : GetTemperature
- Input : **Zip**
- Output : Temperature

W2 : GetLocalTime
- Operation : LocalTimebyCityState
- Input : **City, State**
- Output : LocalTime

W3 : PlaceLookup
- Operation : **ZipToCityState**
- Input : **Zip**
- Output : **City, State**

그림 8. Woogle의 클러스터링 방법을 이용한 검색 결과

W1 : GlobalWeather · Operation : GetTemperature · Input : Zip · Output : Temperature	
W2 : GetLocalTime · Operation : LocalTimebyCityState · Input : City, State · Output : LocalTime W3 : PlaceLookup · Operation : ZipToCityState · Input : Zip · Output : City, State	프루닝 과정을 통해 제거된 웹 서비스
W4 : WeatherFetcher · Operation : GetWeather · Input : PostCode · Output : Temperature, WindChill W5 : GetWeather · Operation : GetWeatherByIP · Input : IPAddress · Output : Weather	온톨로지 매핑을 통해 추가로 검색된 웹 서비스

그림 9. 본 연구에서의 검색 결과

그림 8은 Woogle에서의 클러스터링 방법을 사용해서 검색한 결과이다. 여기에는 zip을 포함하는 클러스터 {city, state, street, zip, code}에 있는 텀들을 포함한 웹 서비스들이 검색된다. 본 논문에서 제안하고 있는 시맨틱 매칭 엔진을 통해 검색한 결과는 그림 9이다. 그림 8과 9를 비교해 보면, 본 논문에서 제시하는 시스템에서는 Woogle에서 검색할 수 없었던 웹 서비스들(W4, W5)을 검색할 수 있다. 그림 9에서 볼 수 있는 것과 같이 zip과 같은 클러스터 내에 있는 텀을 포함하고 있지 않지만 온톨로지 매핑을 통해 zip과 동의어인 'PostCode'로 날씨 검색을 할 수 있는 W3를 검색할 수 있다. 또한 zip과 같은 Weather 온톨로지의 클래스에 속하는 W4를 검색할 수 있어서 사용자는 비록 zip은 아니지만 ip address로 날씨를 검색할 수 있을 것이다.

뿐만 아니라 사용자에게 검색하고자 하는 웹 서비스와 관련된 온톨로지를 선택하고, 해당 온톨로지에 속하는 웹 서비스들만 결과로 보여주는 프루닝 과정을 통해 사용자가 원하지 않는 웹 서비스를 제거할 수 있다. 예에서는 사용자가 해당되는 온톨로지 리스트 중에서 Weather 온톨로지를 선택하여 W2, W3과 같이 키워드를 포함하고 있지만 사용자가 원치 않는 웹 서비스를 삭제할 수 있다.

4. 결 론

웹 서비스 사용이 급속히 증가함에 따라 사용자가 원

하는 적절한 웹 서비스를 찾는 문제가 점점 더 중요해지고 있다. 본 논문에서는 기존의 키워드 기반 검색 시스템의 한계를 극복하고자 클러스터링 방법과 온톨로지를 이용한 시맨틱 매칭 엔진을 제안하였고 이를 통해 사용자가 원하는 웹 서비스를 효율적으로 검색하는 방법을 설명하였다. 제안한 시스템에서는 기존의 연구 Woogle에 온톨로지 매핑 개념을 적용하여 재현율을 높였다. 즉 기존 연구에서는 찾지 못하지만 사용자가 원할 수 있는 웹 서비스들을 찾을 수 있다. 또한 프루닝 과정을 통해 사용자가 원치 않는 웹 서비스들을 제거하여 정확률을 높였다.

현재 제안한 시스템을 구현 중에 있으며, 향후 제안한 시스템의 구현을 완료하여 본 논문에서 제시한 방법에 대해 검증이 필요할 것이다. 또한 본 연구에 시소러스 또는 매치메이킹 개념[13,14]을 추가하여 보다 더 효과적인 검색을 할 수 있는 방법을 모색할 예정이다.

참고문헌

[1] S. A. McIlraith, T. C. Son, H. Zeng, "Semantic Web Service", IEEE Intelligent Systems, March/April 2001

[2] W3C Web Services WG, "Web Services Architecture", http://www.w3.org/TR/2004/NOTE-ws-arch-20040211/, W3C Working Group Note 11 February 2003

[3] Binding pint. http://www.bindingpoint.com/

[4] Grand central. http://www.grandcentral.com/directory/

[5] Salcentral. http://www.salcentral.com/

[6] Web service list. http://www.webservicelist.com/

[7] X. Dong, A. Halevy, Jayant. Madhavan, E. Nemes, J. Zhang, "Similarity Search for Web Services", VLDB, 2004

[8] Woogle. http://www.cs.washington.edu.woogle/

[9] N. Mitra, "SOAP Version 1.2 Part 0: Primer", W3C Recommendation, http://www.w3.org/TR/2003/REC-soap12-part0-20030624/, 24 June 2003

[10] R. Chinnici, "Web Services Description Language(WSDL) Version 2.0 Part 1: Core Language", http://www.w3.org/TR/2006/CR-wsdl20-20060327/, W3C Candidate Recommendation 27 March 2006

[11] UDDI. "The UDDI Technical White Paper", http://www.uddi.org/, 2000

[12] R. Agrawal, H. Mannila, R. Srikant, H. Toivonen, and A. Verkamo. "Fast discovery of association rules. Advances in Knowledge Discovery and Data Mining", 1996.

[13] ATLAS-Software Agents Group, "OWL-S/UDDI Matchmaker".

[14] M. Paolucci, T. Wawamura, T. R. Payne, and K. Sycara, "Semantic Matching of Web Services Capabilities", International Semantic Web Conference(ISWC), 2002.

Acknowledgement

본 연구는 2006년도 두뇌한국21사업과, 정보통신부 및 정보통신연구진흥원의 대학 IT연구센터 육성, 지원 사업 (IITA-2005-C1090-0502-0016)의 연구결과로 수행되었음.