순위부여를 지원하는 웹 서비스 검색 엔진의 개발

손승범°, 황윤영, 이경하, 이규철
충남대학교 컴퓨터공학과
{sbsohn°, yyhwang, bart, kclee}@cnu.ac.kr

The Development of Web Services Search Engine supporting Ranking

Seung-Beom Sohn°,Yun-Young Hwang, Kyong-Ha Lee, Kyu-Chul Lee
Dep't. of Computer Engineering, Chungnam National University

요 약

현재 UDDI에 등록되어 있는 웹 서비스에 대한 검색은 키워드 검색을 기반으로 하고 있다. 그러나 독립된 웹 서비스의 상호 관련성을 통한 서비스의 조합에 대한 요구가 늘어남에 따라 기존의 키워드 기반의 검색으로는 이를 만족 시킬 수 없다. 본 논문에서는 WSDL과 UDDI의 비즈니스 정보에 대한 레이블링과 역파일을 생성하고 이에 따른 가중치 벡터를 생성하여 질의 벡터와 비교 연산을 함으로써 사용자가 조합하고자 하는 웹 서비스와 등록된 웹 서비스 사이의 유사성을 통한 검색 기법을 설명한다.

1. 서 론

XML[1]과 같은 웹 기술의 출현으로 서비스 제공자에 의하여 다양한 웹 서비스[2]들이 배포되어 지고 있으며 이를 등록할 수 있도록 해주는 UDDI[3,4]와 같은 등록소가 사용되고 있다. UDDI에 등록되어 있는 여러 웹 서비스들은 정보 통합에 이용되고, 이에 따라 통합하고자 하는 웹 서비스에 대하여 연관되어 있는 정보를 검색할 수 있는 방법이 요구되어 지고 있다. 하지만 현재의 웹 서비스의 검색은 키워드 기반에 의한 질의만을 지원하기 때문에 연관된 웹 서비스를 찾아내기 힘들며 정확하고 구체적인 사용자의 정보 요구를 만족시키지 못한다.

본 논문에서는 웹 서비스 검색을 위한 질의 인터페이스를 설계하고 UDDI와 WSDL[5]의 레이블과 역파일을 생성하고 IR[6,7,8] 기법을 적용하여 조합하고자 하는 WSDL과 연관된 WSDL를 효율적으로 찾고자 하였다.

본 연구의 구성은 다음과 같다. 2장에서는 지금까지 연구되고 있는 WSDL의 유사도 비교에 관련된 연구에 대해 살펴보고, 3장에서는 질의 인터페이스와 WSDL과 UDDI 의 레이블링 기법과 역파일 생성, 그리고 역파일을 기반으로 만들어진 벡터간의 비교를 통한 유사도 측정과 이를 취합하여 유사도 검색 기법을 설명한다. 4장에서는 결론을 제시하며 향후 연구에 대하여 기술한다.

2. 관련연구

웹 서비스 정보를 등록, 검색하기 위해 고안된 UDDI는 키워드 기반의 탐색만을 지원하므로 서비스간의 관계 또는 서비스간의 상호관련성을 표현할 수가 없다. 즉 조합하고자 하는 문서에 대하여 사용자의 개별적인 판별이 요구된다. 또한 UDDI에 등록된 문서는 제공자의 의도에 따라 분류되어 등록됨으로 제공자와 사용자의 생각에 따라 검색이 원활치 못하게 되는 경우가 발생한다. 이러한 한계를 극복하고자 웹 서비스 검색엔진에 관련된 연구가 진행되었다. 대표적으로 Woogle[9]과 같은 웹 서비스 검색엔진이 있다. 이는 웹 서비스에 대한 키워드 검색과 오퍼레이션의 템플릿 검색을 지원하는 웹 서비스 검색 엔진으로서 유사 웹 서비스 문서의 클러스터링 기술을 제공하여 카테고리 검색을 지원한다. 그러나 직접적으로 관련성 있는 웹 서비스 문서를 찾기 위한 선택적인 인터페이스를 지원하지 않으므로 원하지 않는 정보에 대한 직접적인 정보 획득이 힘들다. 또한 [10]에서는 데이터 타입, 메시지 구조, 매칭 오퍼레이션에 대한 유사정도를 수식화한 테이블을 만들어 가중치를 부여하여 WSDL 문서의 유사정도를 비교한다. 그러나 데이터 타입에 근거하여 WSDL 문서의 유사도를 비교함으로써 메시지 이름과 연관되어 있는 WSDL 문서를 찾기 힘들며 선택적인 인터페이스를 제공하지 않는다.

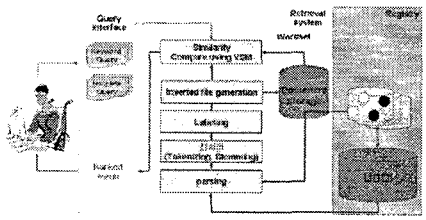

그림1 웹 서비스 검색엔진의 흐름도

3. 웹 서비스 검색엔진 설계

3.1 검색 엔진의 흐름도

그림1은 웹 서비스 검색 엔진의 흐름도를 보이고 있다. 파싱된 정보는 전처리 과정으로 UDDI에 등록되어 있는 WSDL 문서와 UDDI의 비즈니스 정보에 대하여 단어의 집합을 의미있는 단어로 분리하는 토큰화 과정과, 단어의 어근으로 변형해

주는 스테밍 단계를 거친다. 전처리 과정을 거친 후 각 요소(businessEntity, businessService service, operation, input, output)별로 레이블링을 한 후 역파일을 생성하고 tf*idf 값을 사용하여 용어에 대한 가중치를 한 n차원 벡터로 표현한다. 이렇게 가중치가 부여된 벡터와 질의 벡터 간에 대한 유사도 비교는 코사인 상관계수를 활용한 벡터 부합 연산방식으로 계산될 수 있다.

3.2 WSDL문서와 비즈니스 정보의 레이블링

웹 서비스 문서를 표현할 수 있는 정보로는 UDDI상의 businessEntity, businessService name과 description 그리고 WSDL 문서의 service, operation, message(input/output)의 name과 description 정보로 한정 되어 있다. 즉 웹 서비스 문서에 대한 검색을 하기 위해서는 UDDI 비즈니스 정보와 각 WSDL문서에 대한 요소들에 대해 ID가 부여된 정보를 가지고 역파일을 구성하여야 하고, UDDI에 등록된 비즈니스 정보에 대한 검색을 하기 위해서는 등록된 UDDI 문서에 대하여 ID(UUID)가 부여된 정보를 가지고 역파일을 구성하여야 한다. UDDI 비즈니스 정보에 대한 레이블링은 ID, object, type, term 순으로 구성된다. ID는 고유 번호를 의미하며 object는 businessEntity 정보인지 businessService 정보인지를 의미한다. type은 term이 Name 요소에 있을 경우 N, description 요소일 경우 D로 표기하고 term의 순서로 구성한다.

- businessEntity
  name : [documentID.BE, N, term]
  description : [documentID.BE, D, term]
- businessService
  name : [documentID.BS, N, term]
  description : [documentID.BS, D, term]

WSDL 문서에 대한 레이블링은 등록된 WSDL 문서의 고유아이디를 부여하고 각 WSDL 문서에 요소인 serviceID, operationID, Input/OutputID를 부여하여 "."로 구별한다. object는 Service인 경우 S, operation인 경우 O, Input인 경우 IN으로 표기하고 Name 요소에 term이 있을 경우 N, description 일 경우 D로 표기한다. WSDL요소의 레이블링은 다음과 같다.

- service:[documentID.serviceID, S, N, term]
- operation:[documentID.serviceID.operationID, O, N, term]
- input:[documentID.serviceID.operationID.inputID, IN, N, term]
- output:[documentID.serviceID.operationID.output ID, IO, N, term]

3.3 역파일 생성

그림 2은 대상 문서를 service와 operation, input, output 요소별로 각 단어에 대한 빈도와 포스팅수를 나타내는 역파일 구성을 보이고 있다. 역파일 생성은 레이블된 UDDI의 등록정보의 businessEntity, businessService와 WSDL의 service요소와 operation요소, 그리고 input, output 요소를 단어별로 정렬하고 단어 중복을 제거한 후 빈도수를 표시하여 구성한다. 포스팅은 object와 수는 각 단어가 나온 object와 해당 term, 그리고 해당 텀이 발생한 object의 숫자를 나타내는 posting 수로 만들어 진다. 단어 가중치 테이블은 역파일테이블과 포스팅 테이블을 기반으로 해당 단어의 가중치를 계산하고 id, object, type, term, 그리고 단어 가중치를 나타내는 weighted_value로 구성되어 진다.

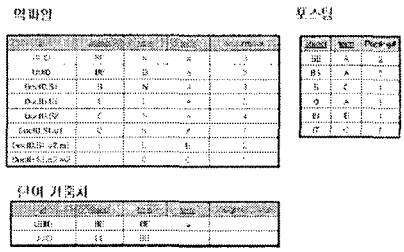

그림 2 대상 문서의 역파일

3.4 질의 선택

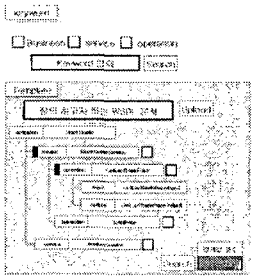

그림 3 질의 인터페이스

질의 선택은 그림 3과 같이 키워드 기반과 WSDL 기반의 템플릿 질의를 선택할 수 있다. 질의 키워드 입력 시에는 businessEntity의 검색을 할 경우 Business를 선택하고, service를 선택할 시는 비즈니스 정보의 businessService와 WSDL의 service에 대한 유사도 검색을 한다. operation 선택 시는 WSDL의 operation에 대한 유사정도를 계산하여 반환한다. 템플릿 검색 시는 WSDL을 업로드 하고 WSDL을 파싱하여 service name과 operation name을 사용자의 요구에 따라 선택할 수 있으며 service를 선택할 경우 service요소를 비교하고, operation 선택 시에는 operation, input, output의 유사도를 비교한 후 취합하여 결과를 반환한다. 이때 input 벡터는 output 벡터와, output 벡터는 input 벡터와 각각 유사도 측정을 한다. 각 유사도 측정은 코사인 내적 공식에 의하여 계산된다.

3.5 가중치 부여와 유사도 비교

단어의 가중치를 계산하기 위해 tf/idf를 사용하며 이때 term의 tf에 위치를 고려한 가중치를 적용하여 name type과 description type일 경우 각각 0.7과 0.3의 가중치를 적용한 후 각 type별로 단어에 대한 가중치를 계산한 tf와 해당 텀이 발생

한 object의 수인 idf를 적용하여 용어 가중치 부여(tf*idf)에 따라 term에 가중치를 부여한다. 가중치가 부여된 단어에 대해 각 요소별로 만들어진 단어의 집합과 비교하여 가중치가 부여된 벡터를 생성하여 낸다. 가중치 벡터와 질의 벡터를 코사인 상관계수를 활용하여 유사도 비교를 하게 된다.

3.6 유사도 측정결과 계산

businessEntity는 여러 businessSevice를 가지고 있다. businessService 는 UDDI의 자료 구조 상 bindingTemplate를 포함하고 있고 bindingTemplate는 accessPoint에서 어떤 형태의 문서를 가지고 있는지 결정하고 있는 useType 어트리뷰트에 accessPoint로 WSDLDeployment에서 WSDL의 위치정보를 등록하고 있다. 그러므로 전체적인 구조는 businessEntity-businessService-service-operation-message의 순으로 각각 하위 정보로 포함 하고 있다. 본 연구에서는 바로 아래의 하위 정보의 유사도 결과가 임계치보다 크면 상위 정보의 유사도를 높여주는 방식을 택하여 유사도 정확도를 증가 시키고자 하였다. 3.5절에서 언급한 유사도 비교의 결과는 아래의 식과 같이 취합되어 사용자에게 최종 유사도를 반환하게 된다.

키워드 검색에서 식(1)은 business 선택시 businessEntity의 유사도를 계산하고 하위 정보인 businessService의 유사도 값이 임계치 이상일 경우 businessEntity의 유사도 결과에 가중치를 주는 식이다. 유사도 결과는 소수점이므로 이에 대한 가중치를 두기 위해 제곱근을 사용 하였다. 식(2)는 Business 정보의 유사도를 계산하기 위해 businessService의 유사도와 service의 유사도를 가중치를 두어 취합 하고 businessService의 하위 단계인 service의 유사도가 임계치 이상일 경우 취합된 값에 제곱근을 적용하여 가중치를 부여한 계산식이다. 식(3)은 파싱된 WSDL문서의 service 선택 시 service와 비교하여 유사도값을 계산하고 하위 단계인 operation의 유사도가 임계값 이상일 경우 service에 가중치를 부여 한다. 식(4)는 operation 선택 시 operation의 유사도를 계산하고 하위 단계인 message의 유사도가 임계값 이상일 경우 operation에 가중치를 부여 하여 그 결과를 반환 한다.

$$Sim_{business} = Sim_{businessEntity \wedge NID}$$
$$if\ Sim_{businessService \wedge NID} \geq threshold$$
$$Sim_{business} = \sqrt{Sim_{business}} \quad (1)$$

$$Sim_{service} = \alpha * Sim_{businessService \wedge NID} + (1-\alpha) * Sim_{service \wedge NID}$$
$$if\ Sim_{service \wedge NID} \geq threshold$$
$$Sim_{service} = \sqrt{Sim_{service}} \quad (2)$$

$$Sim_{service} = Sim_{service \wedge NID}$$
$$if\ Sim_{operation \wedge NID} \geq threshold$$
$$Sim_{service} = \sqrt{Sim_{service}} \quad (3)$$

$$Sim_{operation} = Sim_{operation \wedge NID}$$
$$if\ Sim_{message \wedge NID} \geq threshold$$
$$Sim_{operation} = \sqrt{Sim_{message}} \quad (4)$$

4. 결 론

XML이 웹 환경에서 데이터를 공유하는 표준으로 자리 잡으면서 다양한 웹 서비스 문서들이 WSDL과 같은 기술을 통하여 작성되고 UDDI와 같은 레지스트리에 의하여 등록되고 있다. 본 논문에서는 다양한 웹 서비스 문서들에 대한 효율적인 검색을 위하여 기존의 키워드 검색의 한계를 극복하고자 선택 적인 인터페이스 제공과 비즈니스 문서와 WSDL 문서에 대한 레이블링 그리고 역파일을 생성하여 벡터화 하고 tf*idf 기법을 활용한 가중치 부과와 코사인 상관계수를 활용한 벡터간의 유사도 계산을 적용하고 포함관계에 따른 가중치를 적용하여 검색의 능력을 향상 시키고자 하였다.

향후 연구과제로는 단어 사전 등을 활용하여 유사한 의미간의 연관관계를 고려하여 웹 서비스간의 의미적 유사도를 계산할 수 있는 검색 방법 등이 있다.

참고문헌

[1] Tim Bray, C.M Sperberg-McQueen,Extensible Markup Language (XML)1. (Second Edition)",W3C Recommendations, http//w3.org/TR/REC-xml,2000

[2] W3C, "Web Services",http://www.w3.org/2002/ws

[3] Uddi.org, "Universal Description, Descovery, and Integration", http://www.uddi.org

[4] OASIS UDDI Specification TC, "UDDI Version 3.0 Specification",http://uddi.org/pubs/uddi_v3.html

[5] W3C.org,Web Services Description Language1.1, http://www.w3.org/TR/wsdl

[6] Arjen P. de Vries, Johan A and Henk Ernst Blok "the multi model dbms architecture and xml information retrieval," in Intelligent Search on Data :179-191, 2003

[7] William B. Frakes, Ricardo Baeza-Yates, Information Retrieval :Data Structures&Algorithms Prentice-Hall, Inc, 1992

[8] Hearst. "Current Topics in Information Access: IR Background",http://www.sims.berkeley.edu/courses/is296a-3/f98/lectures/ir-background/, 1998

[9] Xin Dong,Alon Havey,Jayant Madhavan,Ema Nem es, and Jun Zhang,"Similarity search for web services," In Proceedings of the 30th VLDB Conf erence, 2004.

[10]Yiqiao Wang, Eleni Stroulia."Flexible Interface Matching for Web-Service Discovery," Fourth International Conference on Web Information Systems Engineering(WISE'03), 2003

[11]Web Service Architecture, W3C working Group Note 11 February 2004