XML 웹 서비스 검색 엔진의 개발

손 승 범* · 오 일 진* · 황 윤 영* · 이 경 하** · 이 규 철***

Development of a XML Web Services Retrieval Engine

Seung-Beom Sohn* · IL-Jin Oh* · Yun-Young Hwang* · Kyong-Ha Lee** · Kyu-Chul Lee***

Abstract

UDDI (Universal Discovery Description and Integration) Registry is used for Web Services registration and search. UDDI offers the search result to the keyword-based query. UDDI supports WSDL registration but it does not supports WSDL search. So it is required that contents based search and ranking using name and description in UDDI registration information and WSDL. This paper proposes a retrieval engine considering contents of services registered in the UDDI and WSDL. It uses Vector Space Model for similarity comparison between contents of those. UDDI registry information hierarchy and WSDL hierarchy are considered during searching process. This engine suppports two discovery methods. One is Keyword-based search and the other is template-based search supporting ranking for user's query. Template-based search offers how service interfaces correspond to the query for WSDL documents. Proposed retrieval engine can offer search result more accurately than one which UDDI offers and it can retrieve WSDL which is registered in UDDI in detail.

Keywords : Web services, Information Retrieval, UDDI, WSDL

논문접수일 : 2006년 07월 14일 논문게재확정일 : 2006년 11월 22일

※ 이 논문은 정보 통신부 및 정보통신 연구 진흥원의 대학 IT 연구센터 육성 지원 사업(IITA-2005C1090-0502-0016)의 연구 결과로 수행 되었습니다.

* 충남대학교 컴퓨터공학과. e-mail : sbson@cnu.ac.kr

** 한국전자통신연구원 디지털 진흥 연구단 인터넷 서버 그룹

*** 교신저자, 충남대학교 전기정보통신공학부 컴퓨터공학 전공 교수

1. 서 론

웹 서비스(Web Services)는 인터넷과 같은 공개적인 네트워크 및 관련 표준을 통해 기업 내부 또는 기업간의 애플리케이션을 운영체제나 개발 언어에 상관없이 상호운영이 가능하도록 해주는 표준화된 소프트웨어 기술[1]이다.

웹 서비스는 XML 기술을 기반으로 기존의 웹 환경을 이용하여 소프트웨어 사이의 통합이 가능하다는 장점이 있다. 웹 서비스는 사용자가 다양한 인터페이스 정의와 교환 메시지 형식을 가지는 서비스를 개발하는데 있어 보다 효과적이고 단일화된 방법을 제공한다. 인터페이스 정의와 교환 메시지 형식은 WSDL[2]이라는 기술 언어를 통해 작성되며, 이 WSDL 문서를 통해 이용할 서비스의 인터페이스와 교환 메시지 형식을 파악하여 빠르게 해당 서비스를 이용할 수 있도록 한다.

이러한 웹 서비스의 등록과 검색을 위해 현재는 UDDI라는 레지스트리 방식을 이용한다. 개발된 서비스에 관한 설명 정보는 서비스 제공자에 의해 작성되어 레지스트리에 등록되며, 서비스 요청자는 레지스트리로부터 필요한 서비스를 검색하여 이용한다.

UDDI[3]는 웹 서비스를 위한 분산 레지스트리 표준으로 웹 서비스를 위한 등록과 검색 메커니즘을 제공한다. UDDI는 키워드 검색과 카테고리 검색을 지원한다. 하지만, UDDI의 키워드 기반 검색은 등록된 서비스의 이름에 대한 문자열 완전일치 검색만 지원한다는 한계가 있다. 또한 UDDI는 WSDL 문서에 대한 등록은 지원하지만 이에 대한 검색은 지원하지 못하는 단점을 가진다. 이러한 UDDI를 이용한 웹 서비스 검색에서의 문제점을 해결하기 위해서는 UDDI에 등록된 설명 정보와 WSDL 문서 모두에 대한 검색을 지원하고 검색 결과를 순위화(ranking)하여 제시할 수 있는 검색 엔진이 요구된다.

이 논문은 이러한 현재의 UDDI의 문제를 해결할 수 있도록 웹 서비스 등록 정보에 대한 검색을 지원할 수 있는 웹 서비스 검색 엔진을 제안한다. 제안한 검색 엔진은 UDDI 등록 정보의 기술정보 부분에 대하여 검색을 수행할 수 있도록 벡터 공간 모델을 활용한 유사도 비교 방법을 이용한다. 또한 UDDI 등록 정보 외에 실질적인 서비스의 인터페이스와 교환 메시지 형식에 대한 비교의 수행을 위하여 WSDL 문서에 대한 유사도 비교를 수행한다. 유사도 측정시 UDDI 등록 정보와 WSDL 문서와 같은 계층적인 문서 구조를 검색 결과에 반영할 수 있는 방법을 지원한다.

지원하는 검색 방법은 두 가지로 키워드 검색과 함께 템플릿 검색을 지원한다. 템플릿 검색은 서비스의 등록 정보 외에 사용자가 비교하고자 하는 WSDL과 검색하고자 하는 WSDL이 얼마나 일치하는지를 비교하기 위해 WSDL 문서에 대한 유사도를 비교할 수 있도록 한다. 이를 위해 이 논문에서는 사용자로부터 WSDL을 입력 받아 해당 서비스와 유사한 서비스를 검색할 수 있도록 하였다.

이러한 검색의 지원을 통해 제안한 웹 서비스를 위한 검색 엔진은 기존의 레지스트리를 이용한 검색 방법의 한계를 개선한 검색 결과를 제공한다.

2. 관련 연구

웹 서비스를 위한 기존의 연구 방법은 크게 레지스트리 기반 검색, 시맨틱 기술 정보를 이용한 검색, 유사도 기반 검색으로 분류할 수 있다.

2.1 레지스트리 기반 검색

레지스트리 기반 검색은 웹 서비스 제공자와 요청자가 웹 서비스에 대한 WSDL 문서와 설명 정보를 레지스트리에 등록하고, 검색하는 방법이다. 이를 위한 표준으로 UDDI가 있는데 이는 OASIS에서 지정한 레지스트리 표준이다. 웹 서비스는 UDDI 레지스트리에 등록되어 있고 이는 사용자가 등록하고 운영자가 관리한다.

UDDI 레지스트리는 키워드 검색과 카테고리 검색을 지원한다. 검색 할 수 있는 대상으로는 비즈니스, 서비스 및 기술모델이 존재하며 검색 대상을 지정하여 SQL의 LIKE 연산을 통하여 문자열이 완전 일치 하는 경우에 대하여 검색 결과를 반환한다.

키워드 기반 검색은 SQL LIKE 연산을 통해 비즈니스와 서비스의 이름에 대하여 부분 문자열이 일치하는지 검사하는 방식으로 이루어진다. 이러한 UDDI의 키워드 기반 검색은 등록된 서비스의 이름 이외의 기술 정보에 대한 검색을 지원하지 못하므로 효과적인 검색을 지원하지 못하는 단점을 가진다. 또한 UDDI는 WSDL 문서에 대한 등록은 지원하지만 이에 대한 검색은 지원하지 못하는 단점을 가진다. 따라서 서비스 요청자는 서비스 제공자가 작성한 서비스 설명 정보 중 이름만을 가지고 부분 문자열이 일치하는지 확인하는 제한된 검색만 할 수 밖에 없는 문제점을 가진다.

2.2 시맨틱 기술 정보를 이용한 검색

시맨틱 기술 정보를 활용한 검색 방법은 서비스의 검색과 이용을 위해 웹 서비스 표준 기술에 시맨틱 설명 정보를 추가하고 이의 추론을 통한 검색과 이용의 가능함을 목적으로 한다. 하지만, 이 방법은 시맨틱 정보의 추론과 처리에 많은 부하가 걸리고, 서비스 개발 시 서비스의 시맨틱 정보를 자동 생성하기 어렵다는 등 여러 문제점이 존재하여 아직까지 실제 도입에는 많은 문제점을 가진다.

2.3 유사도 기반 검색

레지스트리 기반 검색 방법의 문제점과 시맨틱 기술 정보를 이용한 검색의 문제점으로 인하여, 최근에는 오랜 기간 충분히 검증된 기존의 여러 정보 검색 기법들을 도입하여 웹 서비스 검색 방법을 풀고자 하는 시도가 이루어지고 있다. 이러한 연구들은 웹 서비스 또는 서비스 내의 연산자들에 대한 유사도를 계산함으로써 가장 유사도가 높은 서비스들을 추려 검색 결과로 제공한다. 하지만, 기존의 정보 검색 기법들은 평문(Plain text)을 대상으로 하고 있으나, 웹 서비스 정보는 XML 트리로 계층화되어 있고, 더불어 별도의 논리적 계층구조를 가지고 있어, 이를 바로 적용하는데 문제가 있다. 이에 따라 유사도 기반 검색의 연구 방향에서는 기존 검색 기법을 그대로 적용하지 않고, 이들을 변형하여 이용하는 방법을 택하고 있다.

WSDL의 유사도 기반 검색에서는 WSDL을 분석하여 유사도를 비교하여 유사도 순위별 검색 결과를 반환한다. 기존의 WSDL 유사도 기반 검색 방법에는 WSDL의 타입(type)을 정의하는 XML 스키마 타입 비교를 기반으로 한 유사도 비교 방법과 웹 서비스의 오퍼레이션 비교를 통하여 유사 웹 서비스를 클러스터링하여 정보 검색 기법을 이용하여 검색하는 방법이 있다.

yiqiao wang등[5]에서는 WSDL의 타입을 정의하는 XML 스키마 타입 비교를 기반으로 하여 WSDL의 유사도를 비교하였다. 두 WSDL 파일의 비교는 다단계 처리로 이루진다.

첫째, 메시지와 통신하는 오브젝트의 데이터 타입을 비교한다. 데이터 타입의 매칭에서는 XML 스키마에 대한 요소들의 동일성이나 유사도를 점수화 하고 이를 바탕으로 데이터 타입의 유사도를 계산한다.

두 번째로는 데이터 타입 비교 결과에 대해서 입/출력 메시지 오퍼레이션의 구조의 비교한다. 데이터 타입 유사도 점수가 계산되면 이를 기반으로 메시지 유사도가 계산된다. 이때 메시지 유사도 점수는 데이터 타입 점수의 최대값으로 계산한다. 오퍼레이션간의 유사정도는 메시지 유사도 점수의 합으로 계산된다.

세 번째로는 서비스에 의해 제공된 오퍼레이션 집합의 비교를 하여 점수화 한 후 전체 점수를 취합하여 계산하는 방법을 사용한다. 이렇게 단계별로 계산된 오퍼레이션 유사도 점수를 기반으로, 웹 서비스의 원본측과 목적측의 유사정도를 계산한다.

그러나 [5]에서는 WSDL 서비스, 오퍼레이션, 입출력 메시지의 이름에 대한 유사도를 고려하지 않아 유사한 WSDL을 찾기 위해 각 요소별 이름을 직접 확인하여야 한다.

woogle[6]에서는 유사한 상관관계를 가지고 있는 웹 서비스를 찾기 위해 웹 서비스에 대한 키워드 검색과 WSDL 메시지간 유사도를 비교하였다. 이를 위해 WSDL의 서비스와 오퍼레이션 그리고 입/출력의 각 단어에 용어 가중치를 부여한 후 각 컴포넌트들의 유사정도에 따른 클러스터링 기법을 제안하였다.

오퍼레이션 매칭을 통하여 웹 서비스의 유사한 오퍼레이션 리스트를 반환하였고 입/출력 유사도 비교를 통하여 주어진 웹 서비스의 입출력에 대한 리스트를 반환한다. 유사도를 결정하기 위해서 오퍼레이션의 텍스트 기술에 의한 웹 서비스 간 유사도를 통하여 웹 서비스의 이름들을 클러스터 하였다. 그러나 WSDL의 서비스, 오퍼레이션, 메시지의 이름은 개발자의 의도에 따라 결정되고 입/출력은 적은 단어를 가지고 있어 웹 서비스의 오퍼레이션의 입출력 매칭을 효율적으로 하기 힘들다. 이를 위하여 단어의 유사도에 따라 클러스터링 하였다. 웹 서비스를 클러스터링 하기 위해서 자주, 함께 발생하는 단어는 같은 확률상 같은 개념을 표현한다고 보고 웹 서비스의 오퍼레이션의 입출력 빈도를 이용하여 단어를 클러스터링 하는 결합 규칙을 정의 하였고 이를 위해 전체 입출력에서 단어를 포함하는 입출력이 나올 확률과 단어 1을 포함하는 입출력에서 단어 1과 단어 2를 모두 포함하는 입출력이 함께 나올 확률을 계산하였다.

[6]에서는 클러스터링을 통하여 WSDL 의 전체 유사정도의 계산을 통해 검색을 지원하지만 서비스, 오퍼레이션을 나누어 검색 대상으로 지정하지 않는다. 또한 WSDL 각 요소의 이름과 기술 정보에 나오는 단어의 위치에 대한 고려를 하지 않고 상 하위 요소의 구조적인 연관성을 고려하지 않는다.

3. 웹 서비스 검색 시스템

웹 서비스의 정보는 사용자에 의해 UDDI에 등록된 서비스 설명 정보와 서비스의 인터페이스를 정의한 WSDL 문서로 나뉘어 기술된다. UDDI의 자료 구조 중 BusinessEntity와 BusinessService는 각각 서비스를 제공하는 비즈니스 개체와 서비스 자체에 대한 설명 정보를 가지며, XML 문서로 정의된다. Service와 Operation, Message는 WSDL 문서에서의 요소들로 서비스 자체와 서비스가 가지는 연산자, 그리고 연산자들이 입출력으로 이용하는 XML 메시지의 규격을 정의한다. 이 논문에서의 검색 대상은 평문과 다르게 계층적인 구조를 가지고 있으며,

XML 엘리먼트로 구분되는 각 영역 내에서 단문 형태의 문자열들이 포함된 구조를 갖는다.

이 시스템에서 개발한 웹 서비스 시스템은 다음과 같은 특징을 갖는다.

(1) 유사도 기반의 검색을 지원함으로써 검색 결과에 대한 순위 부여를 지원하고, 또한 이에 따른 대체 후보군들을 순위에 따라 추천할 수 있는 기능을 갖는다.

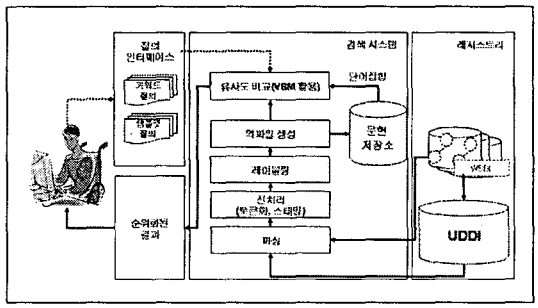
(2) 키워드 검색시 웹 서비스의 자료 구조 중 검색대상을 지정함으로써 사용자가 실제 검색하고자 하는 정보를 한정시킬 수 있다. 현재 검색 대상은 비즈니스와 서비스, 서비스 연산자를 선택할 수 있다.

(3) 키워드 질의 이외에 서비스 조합(Service Composition)기능의 지원을 위해 기 보유한 서비스와 유사하거나 연결될 수 있는 서비스의 검색을 지원한다. 이를 위해 WSDL 문서 자체를 입력으로 하여 연관되는 서비스를 검색 할 수 있도록 템플릿 질의라 명명한 질의 방법을 제공한다.

웹 서비스 조합[ ]은 임의의 프로세스를 수행하기 위하여 웹상의 여러 서비스들을 검색하고, 이들 서비스들 간의 인터페이스 연결을 통하여 프로세스 수행을 가능하게 하는 과정을 의미한다. 이를 위해서는 두 서비스 인터페이스간의 비교와 연결을 수행해야만 한다. 따라서 서비스 조합의 경우에는 키워드 질의 보다는 연결할 서비스 인테페이스 명세를 주고 이에 적합한 인터페이스를 가진 서비스를 검색하는 것이 보다 효율적이다.

이를 위하여 문헌과 질의 사이의 유사도를 계산하여 순위화된 결과를 제시한다. 이를 위해서는 문헌에 나오는 단어에 대한 정규화를 하고 이에 대한 가중치를 부과한 후 문헌을 벡터화한다. 문헌 벡터와 질의 벡터의 비교는 코사인 상관계수를 바탕으로 하는 벡터 부합 연산[7]을 사용하여 문헌과 질의 사이의 유사성을 계산할 수 있고, 이 유사성을 사용하여 문헌에 대한 순위를 부여할 수 있다.

3.1 웹 서비스 검색 엔진의 처리 흐름



<그림 1> 검색 엔진의 처리 흐름

<그림 1>은 이 논문에서 개발한 검색 엔진의 흐름도를 보인다. 검색 엔진은 파싱단계, 전처리 단계, 색인 단계, 역파일 생성단계, 벡터공간 모델(Vector Space Model)[7]을 활용한 유사도 비교 단계로 나누어진다.

파싱단계에서는 WSDL의 service, operation, message(input/output)을 파싱하여 이름(name)과 기술 정보(description)에 속한 단어와 단어의 개수, 단어위치 정보를 데이터 테이블에 넘겨준다.

전처리 단계에서는 UDDI 등록 정보와 WSDL에 기술되어 있는 단어들을 정형화하기 위해 스테밍(Stemming)과 토큰화(Tokenization)를 한다.

색인 단계는 검색 대상의 단어 위치를 식별하도록 하며 이는 용어 가중치를 부여하기 위한 역파일 생성에 사용된다.

역파일 생성단계에서는 BusinessEntity, Business Service, service, operation, input/output에 대

한 색인 정보와 단어의 빈도수 정보를 가지는 역파일 테이블을 생성하고, 포스팅 수 정보를 가지는 포스팅 테이블 그리고 각 요소별 단어 대한 용어 가중치 정보등을 가지는 가중치 테이블을 구성한다. 각 테이블은 문헌 저장소에 저장되며 유사도 비교 단계에서 사용된다.

유사도 비교 단계에서는 가중치가 부여된 단어에 대해 BusinessEntity, BusinessService, service, operation, input/ouput 별로 생성된 단어의 집합과 비교하여 n차원 벡터로 표현하고 벡터간 코사인 내적을 이용하여 유사도를 계산한다. 이후 상하위 관계를 고려하여 계산된 유사도를 취합하여 유사도 결과를 반환한다.

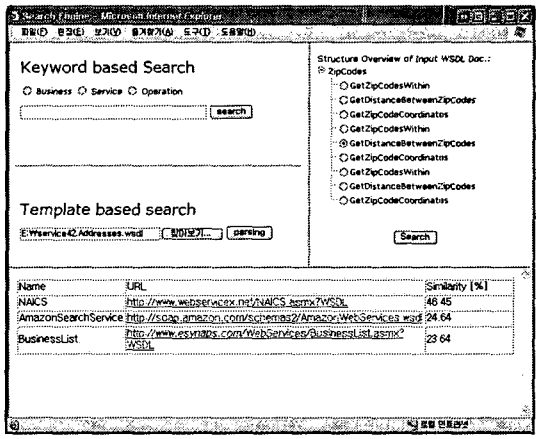
3.2 웹 서비스 검색을 위한 지원 질의 종류

이 논문에서 개발한 시스템에서 지원하는 질의는 키워드 검색을 하기 위한 텍스트 기반의 질의와 WSDL 유사도 기반 검색을 하기 위한 템플릿 질의이다.

키워드 검색은 사용자가 원하는 키워드에 대한 적절한 정보가 있을 경우 신속하고 정확한 접근을 할 수 있지만 일반적인 단어 검색 시 해당 키워드가 너무 많거나 적합 키워드가 없는 경우 검색 효율이 떨어진다. 또한 전문적이고 구조적인 정보를 단순 키워드로 검색하면 그 결과로서 불필요한 정보를 다수 획득하게 되는 단점이 있다.

이 논문에서는 이를 개선하기 위해서 검색하고자 하는 정보에 대해 검색의 범위를 사용자의 요구에 따라 선택할 수 있도록 하고, 검색 단어의 위치를 분석하여 단어에 대한 가중치를 부여함으로써 검색의 정확도를 개선하고자 하였다.

템플릿 기반 검색 방법은 설정된 값에 따라 보다 정확한 검색 결과를 얻을 수 있고 검색된 결과를 원하는 형태로 얻을 수 있으나 검색 대상에 대한 분석으로 인한 속도의 저하가 생긴다. 이 논문에서는 키워드 질의를 입력으로 받아 비즈니스 정보와 웹 서비스의 기술 정보 검색을 지원하고, 웹 서비스간의 구체적인 비교를 통한 검색을 위해서 템플릿 검색을 사용하였다.



〈그림 2〉 질의 인터페이스의 모습

<그림 2>는 이 논문에서 제안한 검색 엔진의 질의 인터페이스이다. 오른쪽 상단은 템플릿 질의 시 입력받은 WSDL 문서의 파싱 결과와 함께 검색 대상이 되는 연산자의 선택을 지정할 수 있는 입력 화면이며, 검색 결과는 하단에 표시된다.

3.3 웹 서비스 검색의 범위

기존 웹 서비스 검색 엔진은 검색범위를 지정하지 못함으로써 비즈니스 정보나 WSDL에 대하여 어떠한 부분이 얼마나 유사한지에 대한 검색을 지원하지 않아 불필요한 부분에 있는 정보에 대한 검색까지 해야 했다.

이 논문에서는 UDDI 비즈니스 정보와 WSDL 검색을 지원하고 검색 범위를 사용자에게 지정하도록 하였다.

이 논문의 검색의 범위는 다음과 같다.

(1) 키워드 검색

- business : BusinessEntity에 대한 유사도 기반 검색 결과
- service : BusinessService와 WSDL의 service에 대한 유사도 기반 검색 결과
- operation : WSDL의 operation에 대한 유사도 기반 검색 결과

(2) 템플릿 검색

- service : WSDL의 service에 대한 유사도 기반 검색 결과
- operation : WSDL의 operation에 대한 유사도 기반 검색 결과

3.4 검색 대상을 식별하기 위한 색인

검색하고자 하는 단어의 위치를 지정하여 검색하기 위해서는 검색 대상의 위치를 색인 하여야 한다. 이를 위해 각 단어가 위치한 정보를 식별하기 위해 단어에 대한 색인을 하였고 이는 역파일 생성시 사용된다. 이 논문에서 정의한 색인은 아래와 같이 ID, object, type으로 구성되어 있다.

(1) ID, object, type

ID는 고유한 번호를 식별하기 위해 필요하며 object는 단어가 속한 요소를 식별하기 위해 필요하다. type은 단어가 이름 또는 기술 정보중 어디에 위치 한 것인가에 대한 정보를 식별한다.

비즈니스 정보에서의 ID는 비즈니스 정보의 UUID를 사용한다.

BusinessEntity의 ID는 UDDI 데이터 정보 테이블의 UUID인 businesskey 값을 사용한다. BusinessEntity에 발생하는 단어의 object는 BusinessEntity의 약자로 표현하여 BE로 표기한다. type은 이름인 경우 N, 기술 정보인 경우 D로 표기한다. 단어는 term으로 표기하였다. BusinessEntity에 나타나는 단어의 색인은 다음과 같다.

(2) BusinessEntity에 발생하는 단어의 색인

- [UUID, BE, N, term]
- [UUID, BE, D, term]

WSDL에서의 ID는 각 WSDL의 도큐먼트와 이에 속한 service, operation, input/ouput에 임의의 넘버링을 통하여 고유 아이디를 부여하였다.

또한 object의 표기는 service인 경우 "S"로 operation인 경우 "O"로 input/output에 대하여는 각각 "IN", "IO"로 표기하였다. 타입은 이름일 경우 N, 기술 정보일 경우 D로 표기하였다.

다음은 service에 나타난 단어에 대한 색인을 보이고 있다.

(1) service에 발생하는 단어의 색인

- [documentID.serviceID, S, N, term]
- [documentID.serviceID, S, N, term]

3.5 역파일

역파일은 정보 검색 시스템에서 가장 널리 쓰이는 방법으로 탐색 작업의 속도를 향상시키기 위해 텍스트에 대한 역파일을 만들기 위한 단어 기반 메커니즘이다[9].

이 논문에서는 UDDI 정보 테이블을 기준으로 BusinessEntity와 Business Service에 대한 데이터 테이블을 생성하고 WSDL에 대한 데이터 테이블을 생성한 후 이를 활용하여 역파일과 포스팅 파일 그리고 각 단어의 가중치가 부여된 가중치 테이블을 만든다.

3.6 용어 가중치 부여

주어진 문헌의 데이터 집합이 i개의 서로 다른 용어를 사용한다고 하면 하나의 문헌은 하나

의 벡터(t1, t2, t3, ···, tn)로 표현될 수 있다. 여기서 ti의 값은 용어 i가 문헌이 있으면 1, 없으면 0이다. 벡터로 표현된 문헌에 대한 검색의 정확도를 향상시키기 위해서 각 벡터의 단어에 대한 용어 가중치[10]를 부여한다.

주어진 한 문헌에 나타난 단어의 빈도와 용어가 사용된 문헌의 빈도수를 활용하여 용어 가중치를 계산하는 가중치 공식은 식 (1)과 같다.

$$w_{ik} = \frac{tf_{ik}\log(N/n_k)}{\sum_{k=1}^{t}(tf_{ik})^2[\log N/n_k)]^2}$$

T_k = term k in document D_i

tf_{ik} = frequency of term T_k in document D_i

idf_k = inverse document frequency of term T_k in C

N = total number of documents in the collection C

n_k = the number of documents in C that contain T_k

$idf_k = \log(N/n_k)$

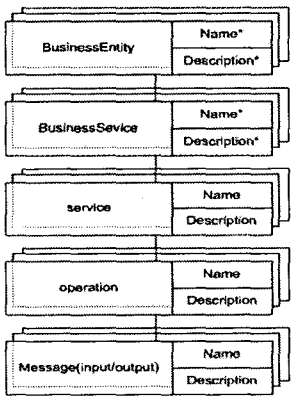
식 (1) 용어 가중치 부여 계산식

TF(Term Frequency)는 한 문서안에 존재하는 해당 단어에 대한 빈도(occurrence)를 나타낸다.

DF(Document Frequenty)는 해당 단어가 존재하는 문헌의 빈도를 나타낸다. idf는 공통으로 포함된 Term(단어)에 대해서 그 단어들이 희귀한 단어, 즉 빈도가 낮은 단어에 대해 높은 값으로 전환하기 위한 방법이다. 즉, 빈도가 낮은 단어들이 그 문서에 대한 특징을 잘 나타낸다고 본다. 이에 비해 빈도가 높은 단어는 문서들에 대한 특징이 잘 나타내지 않다고 본다. 즉 tf와 idf를 활용하면 각 단어에 대한 가중치를 구할 수 있다.

기존의 WSDL 유사도 기반 검색에서의 검색은 이름이나 기술 정보에 있는 단어의 위치 정보를 고려하지 않고 평문으로 보고 정보 검색을 하였다. 이러한 접근은 구조적인 문서에 대한 검색시 구조 정보를 고려하지 않으므로 정확한 정보의 검색을 할 수 없다.

이 논문에서는 UDDI 등록 정보와 WSDL에 나오는 단어의 위치를 이름(name)과 기술 정보(description)의 위치에 따라 차별화된 가중치를 부여함으로써 검색의 정확도를 높이고자 하였다. <그림 3>은 비즈니스 정보와 WSDL 정보에 대한 논리적인 계층을 보이고 있고 각각 이름과 기술 정보로 구성되어 있음을 보이고 있다.



<그림 3> 비즈니스 정보와 WSDL의 논리적 구조와 구성

[11]에서는 단어의 발생위치에 따라 따른 가중치를 차등부여 하여 XML과 같은 구조 문서에 대하여 순위를 부여하였다. 이 논문에서는 용어 가중치를 계산하기 위한 요소인 tf에 위치에 따른 가중치를 줌으로써 사용자에게 보다 정확한 검색을 할 수 있도록 하였다.

단어의 위치에 따른 가중치 부여 공식은 식 (2)와 같다.

비즈니스 등록정보와 WSDL은 각각 이름과 기술정보로 기술되어 있다. 이 논문에서는 서비

스를 표현하는 이러한 정보들에 대하여 차별된 가중치를 부여하였다. 이에 따라 서비스를 대표하는 이름에 속한 단어에는 높은 가중치를 부여할 수 있고, 이를 기술하는 기술정보에는 낮은 가중치를 두어 서비스를 검색할 때 정확성을 높이도록 하였다.

$$tf_{id} = \sum_{P=1}^{n} tf_{ipd}$$

$$tf'_{id} = \sum_{P=1}^{2} (C_p \times tf_{ipd})(C_1 = 0.7(name),$$
$$(C_2 = 0.3(description))$$

식 (2) 단어 위치에 따른 가중치 부여 계산식

C는 위치에 따라 부여되는 가중치를 두기 위한 상수 값이다.

3.7 문서 구조의 반영

서비스를 표현하는 상위 정보와 이에 포함된 하위 정보를 검색 할 때 상위 정보에 대한 정보는 사용자의 검색시 그 비중이 하위 정보에 비해 중요한 정보 대상이 되어야 한다. 이에 이 논문에서는 비즈니스 등록정보와 WSDL과 같은 구조적인 서비스 표현 문서에 대하여 상, 하위 관계에 따라 가중치를 차별화 하여 부여함으로써 사용자의 질의에 적합한 문서를 찾도록 하였다.

기존의 웹 서비스 검색에서는 비즈니스 정보의 BusinessEntity와 Business Service, Business Service와 WSDL의 service, 그리고 service와 operation간 연관성에 대한 고려가 없다.

<그림 3>과 같이 이 BusinessEntity는 여러 BusinessService 정보를 가질 수 있고 Business Service는 여러 WSDL을 가질 수 있다. WSDL은 여러 service들을 포함 할 수 있고 service는 operation들을 포함할 수 있고 operation은 message (input/output)들을 포함한다. 이를 트리구조의 노드라고 하면 상위 노드에 연관된 하위 노드(들)라고 볼 수 있다. 이 논문에서는 식 (3)과 같이 하위 노드의 유사도가 임계치(threshold) 이상이면 상위 노드의 유사도를 증가시킨다.

$$if\ Sim_{low\_level} \geq threshold$$
$$do\ Sim_{high\_level} = \sqrt{Sim_{high\_level}}\ else$$
$$do\ Sim_{high\_levell} = Sim_{high\_level \wedge NID}$$

식 (3) 단어 위치에 따른 가중치 부여 계산식

다음은 키워드 검색과 템플릿 검색에서 질의 범위를 한정하여 검색하였을 경우의 유사도 계산식을 보인다.

(1) 키워드 검색

• business 선택 시

BusinessEntity 정보의 이름과 기술 정보에 대한 유사도 계산과 Business Entity 하위 정보인 BusinessService와의 유사도를 계산한다. 이때 하위 정보인 BusinessService의 유사도가 임계치 이상일 경우 BusinessEntity와의 유사도를 계산한 결과에 제곱근(root)를 적용하여 유사도 값을 올려준다.

• service 선택 시($0 < \alpha < 1$)

BusinessService 정보의 이름과 기술 정보에 대한 유사도 값과 WSDL의 service요소의 이름과 기술 정보에 대한 유사도를 계산하고 각각 가중치 α, $(1 - \alpha)$를 적용하여 합산한다. α는 0과 1사이의 값이고 전체 가중치의 합은 1이다.

이때 Businessservice의 하위 정보인 WSDL의 service 요소의 유사도가 임계값 이상이면 계산된 정보에 제곱근을 적용하여 전체 유사도 값을 상승시킨다.

• operation 선택시

사용자가 operation을 선택하면 WSDL의 operation 요소의 이름과 기술 정보에 대한 유사도를 계산하고 message 정보(input/output)에 대한 유사도를 계산한다.

이때 하위 정보인 message 유사도 값이 임계값 이상일 경우 operation의 유사도 값에 제곱근을 적용하여 유사도를 상승시킨다.

(2) 템플릿 검색

사용자가 가지고 있는 WSDL에 대하여 이와 유사한 WSDL을 검색하고자 할때 템플릿 검색을 사용한다. WSDL 전체를 업로드 하고 질의 인터페이스를 통하여 사용자가 WSDL의 service와 operation을 선택하여 질의하면 등록된 WSDL의 service와 operation에 대한 유사도 기반 검색을 하여 사용자에게 유사도 결과를 반환한다.

• service 선택 시

service의 유사도와 operation의 유사도를 계산하여 operaiton의 유사도 결과가 임계값 이상일 경우에 service의 유사도 값에 제곱근을 적용하여 유사도 값을 상승시킨다.

• operation 선택 시

operation의 유사도와 message 유사도를 계산하여 message의 유사도 결과가 임계값 이상일 경우 operation의 유사도 값에 제곱근을 적용하여 유사도 값을 상승시킨다.

4. 웹 서비스 검색 엔진의 설계

이 장에서는 웹 서비스 검색을 위하여 UDDI 데이터 저장 테이블과 WSDL 데이터 저장 테이블을 생성하고 이를 활용하여 역파일 테이블과, 포스팅 테이블, 그리고 가중치 테이블 생성하고 비즈니스 정보와 WSDL에 대한 유사도 기반 검색 알고리즘을 보인다.

4.1 UDDI의 데이터 저장

외부의 UDDI의 비즈니스 정보에 대한 검색을 하기 위해서는 검색이 필요할 때마다 정보를 가져와야 한다. 이러한 경우에 검색의 속도가 저하된다. 이 논문에서는 UDDI 레지스트리의 등록정보에 대하여 검색에 필요한 부분을 별도의 테이블에 저장함으로써 검색의 효율을 높이고자 하였다.

UDDI의 데이터 저장을 위한 테이블은 역파일을 생성하기 위한 정보로 활용된다. 이는 비즈니스 정보 테이블로부터 가져올 수 있다.

BE_DATA 테이블은 BusinessEntity을 구성하는 테이블이며 UUID인 businesskey와 URL 정보인 discovery url로 구성된다.

BS_DATA 테이블은 BusinessService를 구성하는 테이블이며 UUID인 servicekey와 어느 BusinessEntity에 속하는지를 표현하는 businesskey로 구성되어 있다.

BusinessEntity와 BusinessService는 하나의 정보에 여러 개의 이름과 기술 정보를 가질 수 있으므로 BusinessEntity와 BusinessService 테이블과 name, description 테이블을 나눈 후 조인한다.

name테이블과 desc 테이블은 이름과 기술 정보를 구성하는 테이블이며 각 단어의 값(value)과 BusinessEntity와 BusinessService의 ID 정보를 가지는 parentObjectKey그리고 이중 어디에 속하는지를 표현하는 parentObject로 구성되어 있다. 기본키(PrimaryKey)는 밑줄로 표기하였다.

• TABLE BE_DATA [businessKey, dis-

covery_url]

- TABLE BS_DATA [serviceKey, business key references BE_DATA]
- TABLE Name [value, parentObjectkey, parentObject]
- TABLE desc [value, parentObjectKey, parentObject]

4.2 WSDL의 데이터 저장

다음은 WSDL 데이터 저장을 위한 테이블이다. WSDL_doc 테이블은 WSDL의 ID와 URL 그리고 어떤 BusinessService에서 등록하고 있는 WSDL인지를 표현하는 serviceKey로 구성되어 있다.

service와 operation, message 테이블은 각각 부여받은 고유 ID인 ServiceID와 operationID, messageID를 가지고 있고, 이름인지 기술 정보인지를 표현하는 type과 단어를 가지고 있는 value로 구성된다. WSDL의 ID는 docID로, service의 ID는 serviceID로, operation의 ID는 operationID로 표기하였다. WSDL은 service, operaiton, input/output에 각각 하나의 이름과 기술 정보만이 올 수 있다. 기본키는 밑줄로 표시하였다.

- TABLE WSDL_doc [docID, URL, service Key]
- TABLE Service [serviceID, type, value, docID references WSDL_DOC]
- TABLE Operation [operationID, type, value, serviceID references Service, parents_type references type]
- TABLE Message [messageID, type, message Type, value, operationID references Operation, parents_type references type]

4.3 유사도 계산을 위한 정보 모델

(1) 역파일 테이블

역파일 테이블은 id, object, type, term, occurrence로 구성되며 id와 object, type, term은 Primary Key이다. occurrence는 색인별 단어의 발생 빈도 정보이며 이는 단어가중치를 위한 단어의 빈도를 계산하기 위해 사용된다.

- TABLE inverted_file [id, object, type, term, occurrence]

(2) 포스팅 테이블

포스팅 테이블은 object와 단어 그리고 단어가 나온 오브젝트의 빈도를 나타내는 포스팅 수인 posting#(posting number)로 구성된다. 포스팅 수는 단어의 가중치를 계산하기 위해 필요한 idf(inversed document frequency)로 이는 단어 가중치를 계산할 때 사용된다. 아래와 같이 포스팅 테이블을 구성하였다. 포스팅 테이블의 기본키는 object, term이다.

- TABLE posting [object, term, posting#]

(3) 가중치

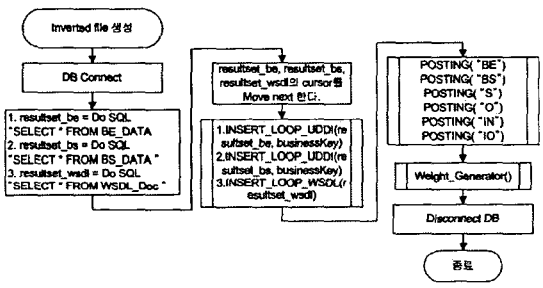
단어에 대한 가중치는 3절에서 설명한 것과 같이 용어 가중치 부여방식[8]을 사용한다. 비즈니스 정보와 WSDL에 대하여 각 ID에 속한 단어들에 대하여 이름과 기술 정보의 위치에 따라 가중치를 곱하여 tf를 계산하고 포스팅 수와 각 ID 별로 나온 단어의 총수로 idf를 계산하여 하나의 용어에 대한 가중치를 구하여 가중치 테이블에 저장한다. 가중치 테이블(weight Table)은 id, object, term 그리고 각 단어의 가중치를 계산한 가중치 값(weighted_value)을 가지고 있다. 이때 가중치 값은 각 단어가 속한 오브젝트의 타입(N(name), D(description))의 정보에 따라 위치에 따른 가중치를 차등 부여하여 계산한

다. 가중치 테이블의 기본키는 id, object, term 이다.

- TABLE weight[id, object, term, weighted _value]

가중치 테이블에 가중치 값을 계산하기 위해 역파일 테이블의 빈도(occurrence)에서 이름(N)과 기술 정보(D)에 대한 가중치를 계산하면 이름과 기술 정보에 나오는 단어에 대한 가중치가 부여된 tf(term frequency)값을 얻을 수 있다. 이 tf값과 포스팅 테이블의 포스팅 수를 식 (1)을 활용하여 계산하면 단어에 대한 가중치를 얻을 수 있다. 단어 가중치가 부여되면 가중치 테이블에 입력되어 DB에 저장된다.

<그림 4>는 역파일 테이블과 포스팅 테이블 그리고 다음에서 기술할 가중치 테이블을 생성하기 위한 순서도이다. DB에 저장된 BE_DATA 테이블과 BS_DATA 테이블 그리고 WSDL 테이블의 정보를 가져와 역파일 테이블 구조에 적합하도록 데이터 정보를 입력하여 DB에 저장한다.



〈그림 4〉 역파일, 포스팅, 가중치 테이블을 생성하기 위한 순서도

테이블 정의에 따라 BusinessEntity, Business Service, service, operaiton, message의 역파일과 포스팅 그리고 가중치 테이블의 정보를 생성한다. 다음은 이를 보인다.

- **BusinessEntity의 경우**
  - inverted_file [UDDI : <UUID>, BE, type ={N, D} <TERM>, occurrence]
  - posting [BE, <TERM>, posting#]
  - weight [UDDI : <UUID>, BE, <TERM>, weighted_value]

- **BusinessService의 경우**
  - inverted_file [UDDI : <UUID>, BS, type = {N,D}, <TERM>, occurrence]
  - posting [BE, <TERM>, posting#]
  - weight [UDDI : <UUID>, BS, <TERM>, weighted_value]

- **WSDL의 경우**
  - inverted_file [WSDL : docID.serviceID. operationID, messageID, object = {S, O, IN, IO}, type = {N,D}, <TERM>, occurrence]
  - posting [{object = {S, O, IN, IO}, <TERM>, posting#]
  - weight [WSDL : docID.serviceID.operationID, messageID, object = {S, O, IN, IO}, <TERM>,weighted_value]

4.4 유사도 계산

DB에 저장된 정보 테이블은 유사도 계산에 활용되기 위해 호출된다. 유사도 계산을 위하여 가중치 테이블에 있는 가중치가 부여된 단어는 BusinessEntity, BusinessServcie, service, operation, message의 각 단어집합과 비교하여 ID 별로 구분하여 벡터로 만들어 진다.

키워드 검색에서 키워드 질의가 입력되면 business일 경우 BusinessEntity에 있는 단어 집합과 비교하여 질의 벡터가 생성되고, service인 경우 BusinessService와 service에 있는 단어 집

합과 비교하여 질의 벡터가 생성된다. operation이 선택되면 operation에 있는 단어 집합과 비교하여 질의 벡터가 생성된다.

템플릿 검색에서 WSDL의 service 선택시 service에 나오는 단어 집합과 비교하여 질의 벡터가 생성된다. operation일 경우 operation에 나오는 단어 집합과 비교하여 질의 벡터가 생성된다. 이때 message에 나오는 단어 집합과 비교한 질의 벡터를 생성한다.

BusinessEntity, BusinessService, service, operation, input/output의 ID별로 생성된 벡터와 질의 벡터간 코사인 내적을 통하여 유사도가 계산된다.

4.5 문서 구조를 반영한 가중치 부여

4.6절에서 계산된 유사도는 상하위 관계에 따라 가중치가 부여될 수 있다.

BusinessEntity는 BusinessService의 상위요소, BusinessService는 service의 상위요소, service는 operation의 상위 요소, operation은 message의 상하위 요소이다.

키워드 검색에서 business 선택시 BusinessEntity와 BusinessService와의 유사도를 계산하여 BusinessService의 유사도가 임계값 이상일 경우 BusinessEntity의 유사도 값에 제곱근을 적용하여 유사도를 높여준다.

service 선택시는 BusinessService와 service의 유사도 값을 계산하여 각각 가중치를 부여하여 합한다. 이때 service의 유사도가 임계값 이상일 경우 합산된 값에 제곱근을 적용하여 유사도를 증가시켜 결과값을 반환한다.

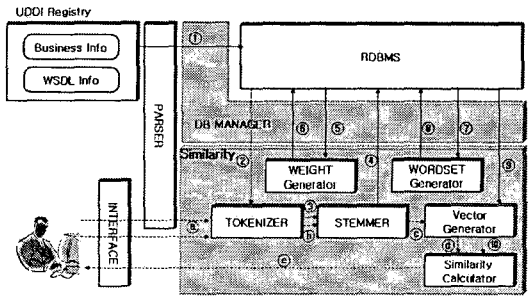
operation 선택시는 operation과 message의 유사도 값을 계산하여 message 값의 유사도가 임계값 이상이면 operatoin의 유사도 값에 제곱근을 적용하여 유사도를 높여준다.

템플릿 검색에서 WSDL의 service 선택시 질의 service의 단어들에 대해 service의 유사도와 operation의 유사도를 계산하여 operation의 유사도가 임계값 이상이면 service의 유사도에 제곱근을 적용하여 유사도를 증가시켜 결과값을 반환한다.

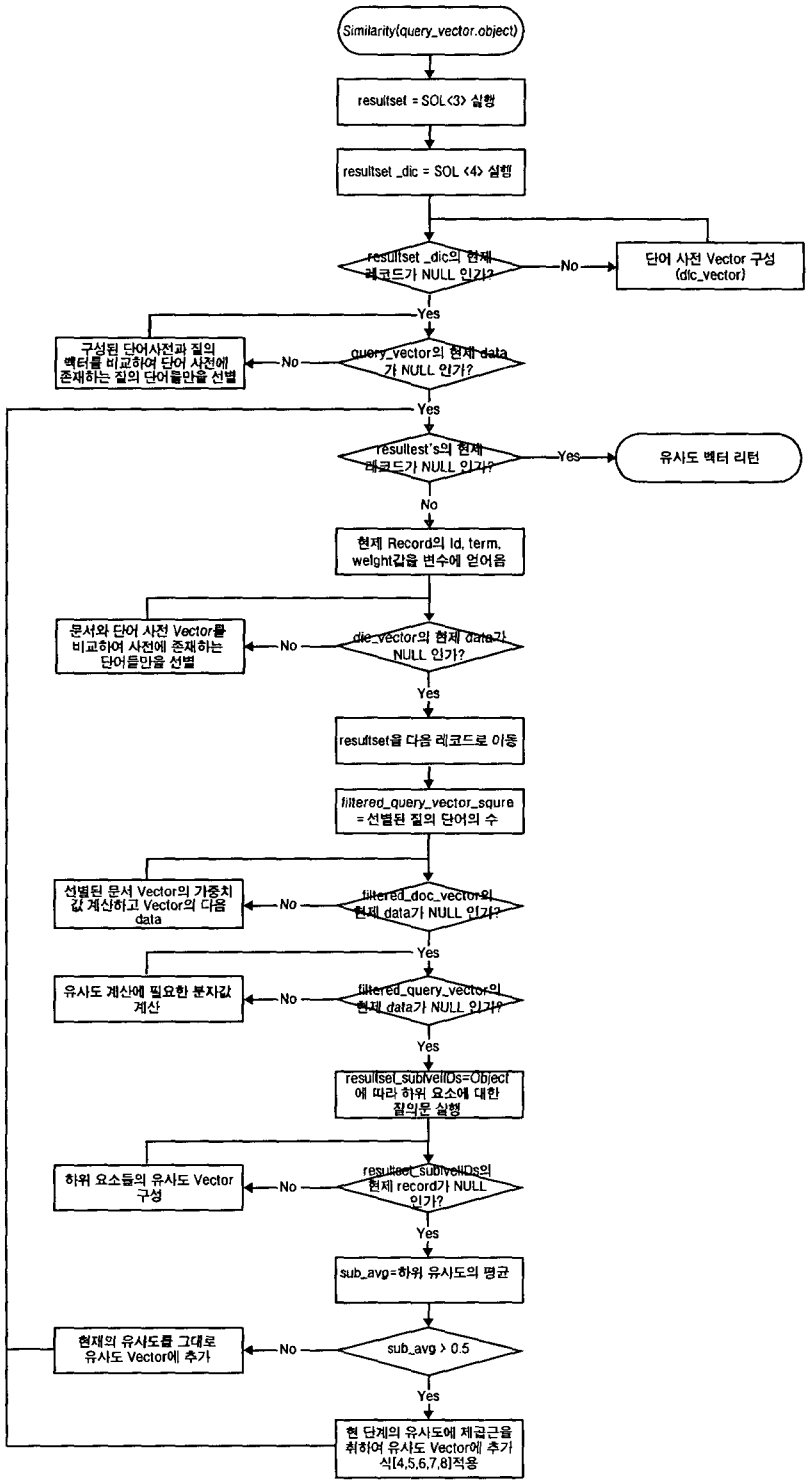
operation 선택시 질의 operation의 단어들에 대해 operation의 유사도와 message의 유사도를 계산하여 message의 유사도가 임계값 이상이면 operation의 유사도에 제곱근을 적용하여 유사도를 증가시켜 결과값을 반환한다. <그림 5>는 위의 SQL문을 활용하여 유사도를 계산하고 문서 구조를 반영하여 유사도 값을 계산하기 위한 순서도를 보이고 있다.

4.6 시스템 구조

<그림 6>은 전체 시스템 구조를 보이고 있다. UDDI의 비즈니스 정보와 이에 등록된 WSDL 정보를 검색하기 위해 비즈니스 정보와 WSDL의 저장 테이블을 생성하고 이를 활용하여 역파일 테이블, 포스팅 테이블, 그리고 가중치 테이블을 생성하고 이를 활용하여 유사도 계산을 한다. DB manager는 비즈니스 정보와 WSDL 정보에 대하여 검색에 필요한 테이블을 만들기 위한 DB 질의와 이를 통해 생성된 테이블의 단어에 대해 전처리 단계인 토큰(token)과 스테밍(stemming) 모듈(module)에 단어를 넘겨 주는



〈그림 6〉 시스템 구조



〈그림 5〉 문서구조를 반영한 유사도 계산

역할과 전처리 단계를 마친 단어들에 대하여 가중치를 부여하기 위한 대상 단어를 단어 가중치 부여 모듈인 weight generator 모듈에 넘기는 역할 그리고 가중치 테이블에서 만들어진 단어를 문헌별로 집합을 만들어 주는 모듈인 Wordset generator 모듈에 넘겨주는 역할 그리고 해당 문헌에 대한 벡터를 만들기 위한 vetor generator에 해당 정보를 넘겨주는 역할을 수행한다. ①이 수행되면 4.1절과 4.2절에서 보인 테이블들이 생성된다. ②~③은 단어에 대한 전처리 단계이며 ④가 수행되면서 역파일과 포스팅 테이블이 생성된다. ⑤~⑥은 단어 가중치 부여와 가중치 테이블 생성 단계이며 ⑦~⑧은 object별로 벡터를 만들기 위한 단어집합 생성 단계이다. ⑨는 문헌별로 벡터 생성을 하기 위한 단계이며 ⑩은 벡터 생성 단계를 거쳐 질의와 유사도 비교를 위해 유사도 계산 단계로 가는 것을 보이고 있다. ⓐ는 사용자가 인터페이스를 통해 키워드 질의와 템플릿 질의를 하는 것을 보이고 있다. ⓑ~ⓒ는 입력된 질의에 대한 전처리 단계를 거쳐 벡터 생성 단계로 가는 것을 보이고 있다. ⓓ는 질의 벡터 생성 후 문헌 벡터와 유사도 비교를 위해 유사도 계산 단계로 가는 것을 보이고 있다. ⓔ는 질의에 대한 유사도 계산 결과를 순위화하여 검색결과로 보인다.

4.7 관련연구와의 비교 및 고찰

<그림 7>은 UDDI 등록된 정보 내용과 WSDL 내용의 예를 보이고 있다.

<그림 7>은 논문 검색을 제공하는 DBLP 서비스이다. <그림 7>의 정보를 검색할 때 기존

〈표 1〉 키워드 검색의 성공 유무

키워드 질의입력			UDDI	Woogle[6]	이논문
business 정보	business Entity	Michael	검색성공	기능없슴	검색성공(유)
		lecturer computer science university trier	검색못함	기능없슴	검색성공(유)
	Business Service	DBLP	검색성공	기능없슴	검색성공(유)
		bibliographic computer science journal	검색못함	기능없슴	검색성공(유)
service 정보	WSDL service	DBLP	기능없슴	검색성공(유)A	검색성공(유)B
		DataBase computer science Logic	기능없슴	검색성공(유)A	검색성공(유)B
operation 정보	WSDL operation 1	author	기능없슴	검색성공(유)A	검색성공(유)C
		paper author	기능없슴	검색성공(유)A	검색성공(유)C
	WSDL operation 2	advanced search	기능없슴	검색성공(유)A	검색성공(유)C
		author title conference journal	기능없슴	검색성공(유)A	검색성공(유)C
	WSDL operation 3	author	기능없슴	검색성공(유)A	검색성공(유)C
		paper author	기능없슴	검색성공(유)A	검색성공(유)C

A : 질의가 들어있는 전체 WSDL 검색. B : WSDL의 service 요소에 한정하여 검색
C : WSDL의 operation 요소에 한정하여 검색. (유) : 유사도 기반 검색

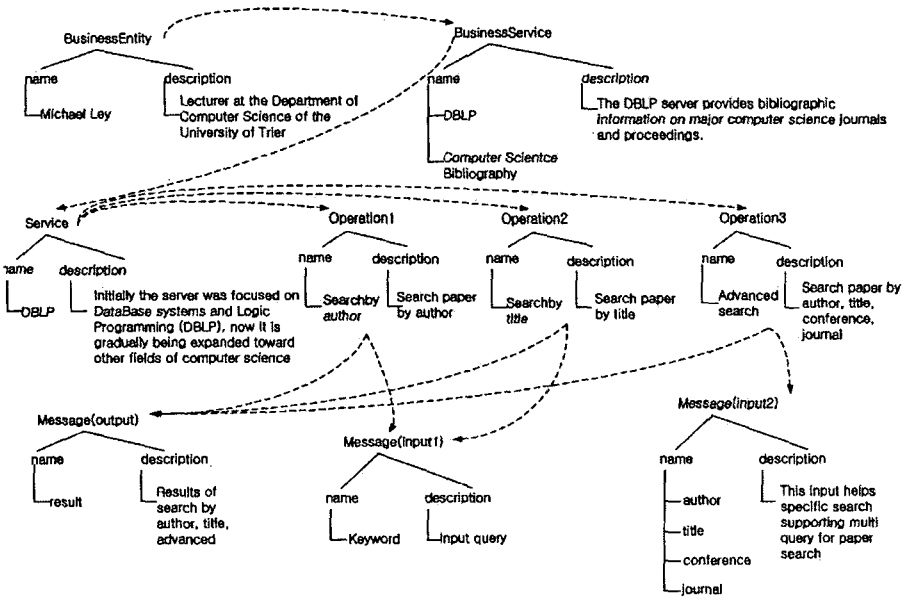
연구와 이 논문의 키워드 검색의 성공 유무를 <표 1>과 같이 정리하였다. 이 논문은 기존의 연구에서는 검색할 수 없었던 비즈니스 정보의 기술정보(description)에 대한 검색을 지원한다. 또한 WSDL 정보에 대한 검색을 지원하고 검색 대상을 지정함으로써 WSDL의 각 요소에 있는 정보에 대한 유사도 기반 검색이 가능하다.

또한 이 논문에서는 service 선택시 Business Service와 WSDL의 service의 유사도를 계산하여 합산한 결과를 순위화하여 보여줌으로써 UDDI 서비스 등록 정보뿐만 아니라 실질적인 서비스에 대한 고려를 함으로써 효과적인 검색이 가능하도록 하였다.

<그림 7>에 있는 정보에 대하여 이 논문은 유사도를 계산하여 순위화한 결과를 보일수 있다. <표 1>에서와 같이 비즈니스 정보에 대한 키워드 검색은 UDDI에서는 키워드에 완전 일치하는 정보를 name 요소에 한정하여 검색하여 줌으로 description 정보에 있는 단어에 대한 검색은 할 수 없다. 또한 UDDI에서는 Business Service 정보에 대해서는 name에 있는 단어에 대해서만 검색할 수 있지만 이 논문에서는 UDDI의 name, description 정보 뿐만 아니라 서비스의 인터페이스를 정의하는 WSDL 문서의 service에 대해서 유사도를 계산함으로써 빠르게 해당 서비스를 이용할 수 있다.

woogle[6]의 검색에서는 WSDL 문서 전체에 대한 유사도를 계산하여 검색 결과를 반환 하는데 비해 이 논문의 검색 엔진은 상위 정보와 하위 정보를 고려하고, WSDL의 검색 대상을 한정하여 검색하는 것을 지원함으로써 검색의 정확도를 높이도록 하였다.

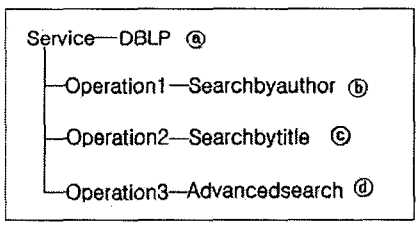
예를 들어 <그림 7>에서 "컴퓨터 과학 저널 목록이 있는 서비스를 검색하라"라는 질의에 대하여 UDDI 검색 시스템에서는 질의와 일치하는 단어가 존재하지 않으므로 검색이 불가능하다. 또한 Woogle[6]에서는 WSDL 문서의 모든 name과 documentation에 있는 단어에 대하여 검색하므로 서비스와 연관되지 않는 오퍼레이션이나 메시지에 해당 단어가 있으면 유사도가



〈그림 7〉 등록 정보 내용과 WSDL 내용 예

계산되어 해당 WSDL을 검색 결과로 반환하기 때문에 서비스에 한정된 정확한 검색을 할 수 없다. 그러나 이 논문의 검색 엔진은 이 질의에 대하여 서비스에 한정된 검색을 지원하고 있다.

<그림 8>은 템플릿 검색 예를 보이고 있다. a, b, c, d의 질의 선택에 따라 <표 2>는 <그림 7>의 WSDL 정보에 대한 질의 검색 결과를 보이고 있다. 예를 들어 사용자가 검색하고자 하는 WSDL 문서가 DBLP라는 서비스 이름과 유사할 때 woogle[6]에서는 DBLP가 서비스의 이름 뿐만 아니라 오퍼레이션이나 메시지에 나오는 정보에 대하여 검색을 하여 관련된 WSDL 모두를 검색 결과로 제시하기 때문에 정확한 검색을 하기 힘들다. 이 논문에서는 이러한 문제를 해결하기 위하여 검색 대상을 지정하도록 하였다.



〈그림 8〉 템플릿 검색 예

〈표 2〉 템플릿 검색의 성공 유무

	Yiqiao Wange[5]등	Woogle[6]	이 논문
ⓐ 선택	검색불가 (XML 스키마 타입 비교)	"DBLP"가 있는 모든 WSDL의 검색	service의 이름과 기술정보에 "DBLP"단어가 존재하는 WSDL 검색
ⓑ 선택		"search, author"가 있는 모든 WSDL의 검색	operation의 이름과 기술정보에 "search, author"단어가 존재하는 WSDL 검색
ⓒ 선택		"search title"이 있는 모든 WSDL의 검색	operation의 이름과 기술정보에 "search, title"단어가 존재하는 WSDL 검색
ⓓ 선택		"advance search"이 있는 모든 WSDL의 검색	operation의 이름과 기술정보에 "advance, search"단어가 존재하는 WSDL 검색

<표 2>에서 보이는 바와 같이 [5]에서는 WSDL 문서내의 단어에 대한 비교를 할 수 없으므로 비교 대상에서 제외되었다. woogle[6]에서는 WSDL의 name과 documentation 정보에 대한 검색을 지원하지만 질의에 해당하는 단어가 WSDL 문서내의 어느 요소에 있는지를 판별하지 않고 전체 WSDL을 검색 결과로 반환하므로 정확한 검색을 하기 어렵다.

〈표 3〉 제안 방식과 기존 시스템 비교 분석

	기 능		UDDI 검색	Woogle [6]	이논문
1	키워드 검색	name 정보 검색	Yes	No	Yes
		documentation 정보 검색	No	No	Yes
2	템플릿 검색	name 정보 검색	No	Yes	Yes
		documentation 정보 검색	No	Yes	Yes
3	문서 구조의 반영		No	No	Yes
4	검색 대상의 지정		Yes	Yes[1)]	Yes[2)]
5	순위화된 검색 결과		No	Yes	Yes[3)]

1) input, output documentation에 대해서 검색 대상 지정
2) UDDI의 business, service, WSDL의 service, operation에 대해서 검색 대상 지정
3) 유사정도를 보여줌

이 논문에서는 이름정보와 기술정보를 활용하여 UDDI 비즈니스 정보와 WSDL에 대한 키워드 검색을 지원하고 WSDL에 대한 템플릿 검색을 지원한다.

문서 구조의 반영은 이름(name)과 기술정보(description)에 가중치를 부여하였고, 상위 정보와 하위 정보에 대한 구조 정보를 고려하여 하위 정보에 대한 유사도가 높을 경우 유사도 값을 상승 시키도록 하였다. 또한 검색의 대상을 지정토록 하여 WSDL의 하위 요소(service, operation)에 대한 구체적인 검색이 가능토록 하였으며, 유사도를 계산하여 순위 부여를 하여 검색 결과를 제시하였다.

5. 실험 결과

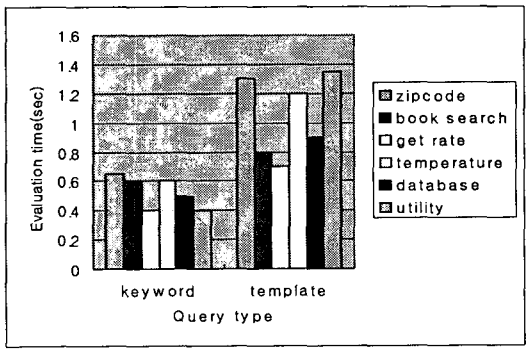
이 논문에서 개발한 시스템의 성능 측정을 위해 [12]의 테스트 대상과 동일한 WSDL 문서들을 UDDI 레지스트리에 등록하여 실험하였다. 검색 시스템은 Java로 개발되었으며, 운영 환경은 Pentium IV-2.7GHz, 1G RAM, RedHat Linux 9.0이다. 실험 대상인 WSDL 문서들의 통계 정보는 아래의 표와 같다.

〈표 4〉 실험 데이터의 통계 정보

service	operation	message
40	628	837
total file size : 1.0MB preprocessing time : 40 sec		

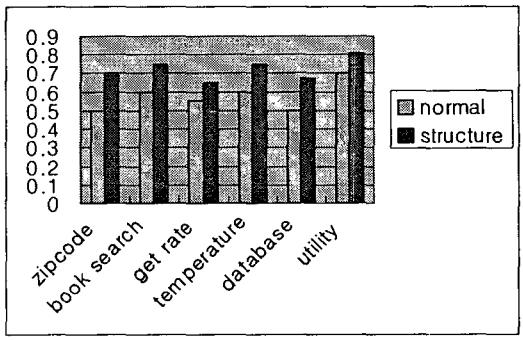
이 실험을 위해서는 총 40개의 서비스가 이용되었으며, 역파일을 생성하기 전까지의 전처리 단계에 40초의 시간이 소요되었다.

<그림 9>는 여러 질의어에 대한 키워드 질의와 이에 해당하는 템플릿 질의시 소요되는 시간을 측정하였다. 템플릿 질의시 처리 속도가 키워드 질의 시 처리속도보다 늦어짐을 확인할 수 있다. 이는 질의로 하는 입력 WSDL 문서의 파싱과 단어 추출에 많은 시간이 소요되기 때문으로 파악된다.



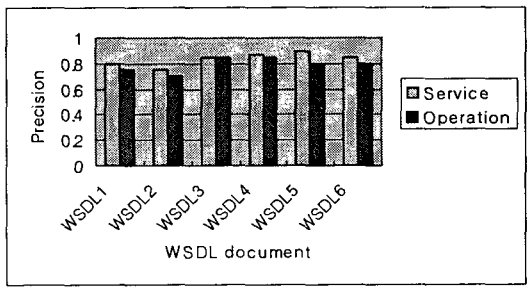
〈그림 9〉 질의별 수행 시간

아래는 여러 질의어에 대하여 서비스 검색의 정확도를 측정한 결과를 보인다. 여기에서는 질의를 두 가지 경우로 나누어 검사를 수행하였다. normal의 경우 식 (2)와 식 (3)을 적용하지 않고 단순히 TF-IDF 기반으로만 유사도를 검색한 경우이며 structure의 경우 식 (2), 식 (3)을 적용한 결과이다. 그림에서 보이는 바와 같이 문서 구조를 고려한 가중치 부여를 적용할 경우 정확도가 어느 정도 향상됨을 확인할 수 있다. (Top-K precision, K = 2인 경우에 대하여 정확도를 비교한다.)



〈그림 10〉 키워드 검색의 정확도 측정

<그림 11>은 템플릿 검색의 정확도를 측정한 것이다. 검색 대상은 서비스와 서비스 연산자 두 가지를 대상으로 하였으며, 검색 결과의 정확도는 구조 정보를 고려한 키워드 검색과 거의 동일한 결과를 얻음을 보인다.



〈그림 11〉 템플릿 검색의 정확도 측정

7. 결론 및 향후연구

이 논문에서 개발한 웹 서비스 검색 시스템은 기존의 레지스트리 기반의 서비스 검색의 한계를 극복하기 위해 기존 정보 검색 기법을 활용하여 보다 향상된 검색 결과를 순위를 부여하여 제공한다.

UDDI와 같은 레지스트리는 웹 서비스에 대한 정보를 등록하고 SQL LIKE연산을 통해 비즈니스와 서비스의 이름에 대하여 부분 문자열이 일치하는지 검사하는 방식으로 이루어진다. 이러한 UDDI의 키워드 기반 검색은 등록된 서비스의 이름 이외의 기술 정보에 대한 검색을 지원하지 못하므로 효과적인 검색을 지원하지 못한다. 또한 UDDI는 WSDL 문서에 대한 등록은 지원하지만 이에 대한 검색은 지원하지 못한다. 따라서 서비스 요청자는 서비스 제공자가 작성한 서비스 설명 정보중 이름만을 가지고 부분 문자열이 일치하는지 확인하는 제한된 검색만을 할 수 밖에 없는 문제점을 가진다.

또한 UDDI 등록 정보 외에 실질적인 서비스의 인터페이스와 교환 메시지 형식에 대한 비교의 수행을 위하여 WSDL 문서에 대한 유사도 비교를 수행하도록 하였다. 유사도 측정시 UDDI 등록 정보와 WSDL 문서와 같은 계층적인 문서 구조를 검색 결과에 반영할 수 있는 방법을 지원하였다. 제안된 검색 엔진은 기존의 연구에서는 검색할 수 없었던 비즈니스의 기술 정보에 접근하여 검색할 수 있다. 또한 WSDL 정보에 대한 검색을 지원하고 검색 대상을 지정함으로써 WSDL의 각 요소에 있는 정보에 대한 검색이 가능하다. 또한 문서 구조를 반영하여 유사도를 계산하여 검색하고자 하였다.

웹 서비스는 그 특성상 단순히 해당 단어가 특정 서비스 정의에 존재한다고 해서 옳다고만 할 수가 없으며, 실제 서비스 이용 시 추가되는 적용에 따른 비용을 최소화하는 검색 방법이 요구된다. 향후 이 검색 시스템은 이러한 도입 비용을 고려하는 웹 서비스 검색 엔진의 기반으로 활용될 것이고, 적용 비용의 최소화를 고려한 서비스 검색 방법이 연구되어야 할 것이다.

참 고 문 헌

[1] W3C.org, "Web Services Glossary", W3C Working Draft 14, http://www.w3c.org/TR/ws-g, Nov. 2002.

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

[3] "Web Services Description Language(WSDL) 1.1", W3C Note, World Wide Web Consortium, http://www.w3c.org/TR/WSDL, March 2001.

[4] "Simple Object Access Protocol(SOAP) 1.1", W3C Note, World Wide Web Consortium, http://www.w3c.org/TR/SOAP, May. 2000.

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

[6] Xin Dong, Alon Havey, Jayant Madhavan, Ema Nemes, and Jun Zh ang, "Similarity search for web services", In Proceedings of the 30th VLDB Conference, 2004.

[7] Dik.L. Lee, Huei Chuang, Kent Seamons, "Document Ranking and the Vector-Space Model", IEEE Computer Society Press, 1997.

[8] Ricardo Baeza-Yates and Berthier Ribeiro-Neto, "Modern Information Retrieval", Addision-Wesley Longman Inc, 1999.

[9] William B. Frakes, Ricardo Baeza-Yates, "Information Retrieval : Dat a Structures & Algorithms", Prentice-Hall, Inc. 1992.
[10] Hearst. "Current Topics in Information Access : IR Background", http://www.sims.berkeley.edu/courses/is296a-3/f98/lectures/ir-background/, 1998.
[11] Andrew trotman, "Choosing Document Structure Weights", an Inter national Journal archive Volume 41, 2003.
[12] Erhard Rahm, Philip A. Bernstein, A Survey of approaches to automatic schemamatching, VLDB Journal Vol. 10, No. 4, pp. 334-350, 2001.

□ 저자소개

손 승 범
충북대학교 컴퓨터공학과(공학사)
LG전자 유무선 단말연구팀
한국전자통신연구원 소프트웨어로봇연구팀
충남대학교 대학원 컴퓨터공학과(공학석사)
현재 국방과학연구소
관심분야 : 데이터베이스, 웹 서비스, 정보검색
E-mail : sbson@cnu.ac.kr

오 일 진
공주대학교컴퓨터공학과(공학사)
현재 충남대학교 대학원 컴퓨터공학과 재학중
관심분야 : 웹 서비스, 유비쿼터스 웹 서비스

황 윤 영
충남대학교 컴퓨터공학과(공학사)
충남대학교 대학원 컴퓨터공학과(공학석사)
현재 충남대학교 대학원 컴퓨터공학과(박사과정 재학중)
관심분야 : 서비스 지향 아키텍처, 유비쿼터스

이 경 하
충남대학교 정보통신공학과(공학사)
충남대학교 대학원 정보통신공학과(공학석사)
충남대학교 대학원 컴퓨터공학과(공학박사)
현재 한국전자통신연구원 디지털홈연구단, 인터넷서버그룹
관심분야 : 데이터베이스, XML, 정보 통합

이 규 철
서울대학교 컴퓨터공학과(학사)
서울대학교 컴퓨터공학과(석사)
서울대학교 컴퓨터공학과(박사)
미국 IBM Almaden Research Center 초빙연구원
미국 Syracuse University, CASE Center 초빙 교수
학술진흥재단 부설 첨단학술센터 파견 교수
현재 한국정보과학회 논문편집위원, 한국 ebXML 전문위원회 위원장
관심분야 : 데이터베이스, XML, 정보 통합, 멀티미디어 시스템, e-비즈니스 시스템, 유비쿼터스 웹 서비스

◈ 이 논문은 2006년 07월 14일 접수하여 2차 수정을 거쳐 2006년 11월 22일 게재확정되었습니다.