매쉬업을 위한 Open API 유사성 탐색 방법

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

Open API Similarity Searcing Method for Mashups

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

요 약

매쉬업은 공개된 Open API들을 이용하여 두 가지 이상의 서로 다른 자원을 섞어서 완전히 새로운 가치의 서비스를 만드는 것이다. 그렇지만, Open API 포털 사이트들은 매쉬업에서 사용 가능한 수많은 API들을 제공하고 있는데 이들에 대한 조합 가능한 API들을 탐색하고 발견하는 것은 매우 힘들고 많은 시간이 소비되는 작업이다. 본 논문에서는 다양한 Open API 타입들에 대한 API 유사성 탐색 방법을 지원하기 위해 계층적 결합 클러스터링 알고리즘과 계층관계 형태소 분석 기법을 제안한다. 본 논문에서 제안된 방법은 programmableWeb과 xmethods.net 사이트로부터 168개의 REST API와 50개의 SOAP API를 다운로드 받아 실험 분석을 수행하였으며 우리의 접근방법이 기존의 키워드 검색 방법과 Woogle 방법 보다 성능이 우수함을 보인다.

1. 서론

최근에 매쉬업은 미래 IT 융합 서비스의 효과적인 구현 방법으로써 그 관심도가 점점 높아지고 있으며, 그들의 활용도 엔터프라이즈 비즈니스로부터 과학적 e-사이언스 분야에 이르기 까지 매우 다양하다. 그렇지만 이러한 높은 관심에도 불구하고 Open API들을 매쉬업 속으로 결합할 때 여러 가지 이슈들이 있을 수 있다. 특히, Open API들이 SOAP, REST, JavaScript, 그리고 XML-RPC 등과 같은 다양한 프로토콜을 사용하고 있을 때 이는 더욱 심각해진다. 이러한 문제점들을 정리하면 다음과 같다.

첫째, 매쉬업을 제작하기 위해서는 그 목적에 맞는 적합한 Open API들을 먼저 찾아야 하는데, 이러한 작업은 인터넷 상에 Open API 들의 수가 기하급수적으로 증가됨에 따라 그렇게 쉬운 작업이 아니다. 현재 대부분의 Open API 포털 사이트(예, programmableWeb[1], SeekDa[2])들은 키워드 검색 또는 카테고리 검색만 지원하고 있다. 키워드 검색은 이미 여러 연구에서 밝혀진 바와 같이 나쁜 재현율과 나쁜 정확률 때문에 문제가 많으며, 카테고리 검색 결과도 매쉬업 개발자에게는 별로 관심이 없는 알파벳 순서나 최근 개발된 날짜 순서로만 정렬되어 있어 원하는 결과를 찾는데 쉽지 않다.

둘째, 현존하는 어떠한 매쉬업 포털 사이트들도 전통적인 SOAP 기반 웹 서비스 분야에서 제안되었던 것처럼 API들을 찾고 통합하는데 시맨틱 기법을 활용하는 사이트는 없다. SOAP 기반 웹 서비스들과는 달리 REST, JavaScript, XML-RPC API들은 그들의 인터페이스를 명시하기 위한 WSDL(Web Service Description Language)을 사용하지 않는다. 더군다나 이런 API들을 정의하거나 분류하기 위한 어떠한 표준도 논의된 바 없고 단지 개발자들이 수동적으로 웹을 탐색하여 원하는 API들을 찾고 있다.

본 논문에서는 이와 같은 이슈들을 해결하기 위해 먼저 Open API를 개발할 때 생성되는 신택틱(syntactic) 정보를 가지고 항목 간 숨어있는 시맨틱(semantic) 정보를 찾아내어 관련 온톨로지를 자동 구축하는 하나의 새로운 시맨틱 온톨로지 구축 방법을 제안한다. 그리고 이러한 신택틱/시맨틱 정보들이 어떻게 Open API들의 발견과 조합에 도움을 줄 수 있는지 보이고, 실험 분석을 통해 그 성능을 분석한다.

본 논문의 구성은 다음과 같다. 2장에서 관련 연구들을 살펴보고 3장에서 시맨틱 온톨로지 구축 방법을 제안한다. 4장에서 매쉬업을 위한 API 유사성 탐색 방법을 기술하고 5장에서 실험 분석을 수행한다. 마지막으로 6장에서 결론을 내린다.

2. 관련 연구

매쉬업에 대한 깊은 관심으로 지금까지 많은 연구들이 수행되어 왔으나, 이들 기술들과 개발 툴들은 주로 개인용과 기업용으로 크게 구별할 수 있다. 예를 들면, Pipes[3], Popfly[4], MashMaker[5]는 개별적인 개인사용을 목적으로 하고 있으며, Damia[6], MARIO[7], UQBE[8] 등은 주로 기업 인트라넷 환경에서 기업 비즈니스 생산성 향상을 목표로 하고 있다.

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

Pipes는 drag & drop 방식으로 인터넷 데이터 소스를 연결하고 흐름을 재구성하도록 매쉬업 아이디어를 일반화한 비주얼 툴이다. Popfly는 매쉬업을 손쉽게 할 수 있도록 실버라이트(Silverlight)를 이용하여 만들어진 웹 사이트이며, MashMaker는 사용자가 개별화된 매쉬업을 생성할 수 있도록 지원하는 브라우저의 부가기능이다. Damia에서는 Pipes에서 취급하는 URL 기반 소스뿐만 아니라 Excel, Notes, 웹 서비스, 그리고 XML 문서와 같은 기업 형태의 데이터 소스까지 확장하였으며, MARIO는 일반 사용자가 쉽게 데이터 매쉬업을 개발할 수 있는 태그(tag) 클라우드를 이용한 매쉬업 개발 도구이다. UQBE에서는 자동화된 스키마 매칭을 기반으로 마치 릴레이션 테이블 조인처럼 공통적인 속성들에 의해 분산된 데이터 소스들이 결합된다. 하지만 이러한 툴들은 제한된 키워드 기반 매쉬업 검색만 제공하고, 아직 시맨틱 개념이 지원되지 않아 순위화된 유사 서비스 탐색과 같은 방법은 없다.

3. 시맨틱 온톨로지 구축 방법

REST, SOAP, JavaScript, 그리고 XML-RPC API들은 공통적으로 일반 프로그래밍과 같은 입력과 출력을 가지고 있다. 이러한 프로그램적인 입/출력은 Open API에 대한 사용하기 쉽고 직관적인 방법은 제공하지만 이들의 단점도 신중히 고려하여야 한다. 일반적으로 입/출력 매개변수(parameter)들은 잘 통제되고 관리된 어휘만 사용하는 것이 아니라 개발자의 자유롭고 임의적인 판단에 의해 용어들이 선택되어 지므로 이러한 모호성으로 인해 매칭되는 API들 간의 불일치가 야기될 수 있다. 또한 이러한 매개변수들은 의미적으로 계층구조를 형성하지 못하고 단지 일차원적으로 나열되어 있는 형태이므로 수많은 매개변수들이 존재할 때 이들 중 호환되는 API들을 찾는 것은 매우 어려운 문제이다.

3.1 계층적 결합 클러스터링 알고리즘

본 논문에서는 매개변수들에 대해 의미적으로 같은 개념들을 묶는 계층적 결합 클러스터링 알고리즘을 제안한다. 즉, WSDL/WADL/HTML 파일에 존재하는 신텍틱 정보를 이용하여 그들 사이에 숨겨져 있는 시맨틱 개념(concept)을 얻기 위해 마이닝(mining) 알고리즘을 적용한다.

이를 위해 Dong[9]이 제안한 것과 비슷하게 오퍼레이션 매개변수에 숨겨져 있는 시맨틱 개념 정보를 추출한다. 일반적으로 입출력 매개변수들은 다수의 단어가 연결된 복합단어로 이루어져 있다. 이러한 매개변수 내에 존재하는 용어(term)들을 여러 개의 의미 있는 시맨틱 개념 속으로 클러스터링 시킬 때 우리는 용어들의 발생 빈도수를 고려한다. 이는 만일 용어들이 동시에 자주 나타난다면 그것들은 같은 개념을 나타내는 경향이 있다는 가정 하에 하나의 특별한 연관규칙(association rules)에 따라 만들어진다.

$T=\{t_1, t_2, ..., t_m\}$을 용어들의 집합이라고 할 때, 연관규칙은 용어 t_i가 일어나면 용어 t_j가 일어난다는 의미로 $t_i \rightarrow t_j$(여기서 $t_i, t_j \in T$)로 표현될 수 있다. 여기서 지지도(support)와 신뢰도(confidence)는 $t_i \rightarrow t_j$ 규칙이 얼마나 유용한지를 나타내는 지표로서 사용되는데, 지지도는 용어 t_i와 t_j를 동시에 포함하는 트랜잭션의 확률을 표현하며, 신뢰도는 t_i가 주어졌을 때 t_j가 동시에 나타날 트랜잭션의 확률을 나타낸다. 만일 규칙 $t_i \rightarrow t_j$의 지지도와 신뢰도가 최소 지지도와 최소 신뢰도보다 크다면 t_i와 t_j는 밀접하게 연관되어 있다고 말할 수 있다.

본 논문에서는 용어들의 집합 $T=\{t_1, t_2, ..., t_m\}$를 시맨틱 개념들의 집합 $C=\{c_1, c_2, ..., c_m\}$로 전환하기 위해 계층적 결합 클러스터링 알고리즘을 구현한다. 제안되는 알고리즘은 연관규칙 탐사 결과에서 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 가장 이상적인 클러스터를 형성하기 위하여 클러스터의 응집력(cohesion)은 높게 하고, 클러스터 간의 연관성(correlation)은 낮게 한다. 최종적으로 전체적인 클러스터 품질을 측정하기 위한 클러스터링 점수는 (응집력)/(연관성) 방식으로 계산되며, 이때 우리의 목표는 가장 높은 점수를 갖도록 클러스터를 형성하는 것이다.

3.2 계층관계 형태소 분석 기법

계층관계 형태소 분석 기법은 매개변수 내에 포함된 용어들 간의 상관관계를 취득하고, 만일 두 용어들이 서로 유사하며 상관관계가 조건에 일치한다면 그 매개변수를 매치하는 것이다. 이러한 접근방법은 사람들이 다수의 용어들을 가지고 매개변수를 만들 때 일반적으로 비슷한 패턴을 사용한다는 관찰로부터 유도되었다. 이러한 패턴들을 조사하기 위해 본 논문에서는 Open API의 대부분을 차지하는 REST와 SOAP API들에 대해 실험 데이터를 만들어 분석하였다. JavaScript는 WSDL 파일을 생성할 수 있으므로 SOAP API에 포함시킬 수 있고, XML-RPC는 차지하는 비율이 낮아 분석에서 생략하였다.

먼저 REST API는 programmableWeb으로부터 REST API 매개변수들을 다운로드 받아 사전 작업 처리과정을 거쳐 형태소 분석을 수행할 수 있는 데이터 파일을 준비하였다. 이 사이트는 본 논문의 실험 시점에 4913개의 Open API가 존재하고 있었으며, 이들 중 REST 방식으로 구현된 API는 3378개였다. 본 조사에서는 이들 모든 API들을 다 이용하지 않고 mapping, travel, weather 도메인에 있는 168개의 REST API들에 대해서만 실험을 수행하였다. 다음으로 SOAP API에 대해서는 xmethods.net에서 WSDL 파일을 다운로드 받아 실험 분석을 수행하였다. xmethods.net에서는 약 600여개의 WSDL 파일이 존재하고 있으나 본 조사에서는 zip, weather, address 도메인에 관한 50개의 파일에 대해서만 실험을 수행하였다.

이들 실험 데이터에 POS(part-of-speech) 형태소 분석

기를 적용시킨 결과 REST API에서는 단지 하나의 토큰으로 구성된 매개변수(예, City)가 전체의 43%로 가장 많았고, 명사+명사(30%), 형용사+명사(9%), 동사+명사(7%), 명사+명사+명사(6%), 명사+전치사+명사(5%), 그리고 기타(0.3%) 순으로 나타났다. SOAP API에서는 단지 하나의 토큰으로 구성된 매개변수는 38%를 차지하였으며, 명사+명사(37%), 명사+명사+명사(14%), 동사+명사(6%), 명사+전치사+명사(3%), 형용사+명사(1%), 그리고 기타(1%) 순으로 나타났다. 본 결과로부터 특이한 사항은 다른 프로토콜을 사용하거나, 다른 도메인을 선택하더라도 매개변수 패턴은 단지 출현 빈도의 순위에 다소간의 변동이 있을 뿐 도출된 5개의 패턴 종류는 동일한 사실을 알 수 있다. 따라서 매개변수에 대한 온톨로지 변환 규칙은 다음과 같이 5개의 규칙으로 정할 수 있다.

- 규칙1($Noun_1$+$Noun_2$): Parameter **propertyOf** $Noun_1$
- 규칙2(Adjective+Noun): Parameter **subClassOf** Noun
- 규칙3(Verb+Noun): Parameter **subClassOf** Noun
- 규칙4($Noun_1$+$Noun_2$+$Noun_3$):
  Parameter **propertyOf** $Noun_1$
- 규칙5($Noun_1$+Preposition+$Noun_2$):
  Parameter **propertyOf** $Noun_2$

위와 같은 규칙을 사용하여 온톨로지가 구축되고 나면, 다음 단계는 질의문에 의해 두 개념 간 매칭을 시키는 것이다. 두 개의 온톨로지 개념이 다음 조건을 만족하면 매치된다: (1) 어떤 개념이 다른 개념의 속성일 경우(즉, Parameter **propertyOf** $Noun_1$), (2) 어떤 개념이 다른 개념의 자식관계인 경우(즉, Parameter **subClassOf** Noun).

위의 조건으로부터 우리는 매개변수 유사성을 기반으로 한 매치를 발견할 수 있다. 예를 들면, 매개변수 CityName과 CodeOfCity를 비교한다고 하자. 이때 키워드 검색은 매치가 실패한다. 왜냐하면 두 개의 매개변수는 일치하지 않기 때문이다(즉, CityName != CodeOfCity). 그렇지만 City가 **propertyOf** 관계를 가지고 있다면 매칭 점수를 리턴할 것이다. 왜냐하면 이러한 두 개의 매개변수는 CityName **propertyOf** City와 CodeOfCity **propertyOf** City 관계에 의해 서로 밀접하게 연관되어 있기 때문이다.

4. 매쉬업을 위한 API 유사성 탐색 방법

본 장에서는 매쉬업을 위한 Open API의 유사도가 어떻게 측정되는지 기술한다. 3장에서 제안된 시맨틱 온톨로지 구축 방법의 장점은 신택틱 정보에 추가되는 시맨틱 온톨로지가 API를 정의하는 구조에 큰 변화를 주지 않는 것이다. 따라서 기존의 시맨틱 웹 서비스 탐색 알고리즘들이 API 매칭을 위하여 바로 적용될 수 있다. 본 연구에서는 매칭되는 API들을 효율적으로 발견하기 위해 신택틱과 시맨틱 정보를 혼합 사용하는 방법을 탐구한다. 먼저, WSDL/WADL/HTML 파일에 작성된 신택틱 정보를 파싱하여 토큰화(tokenization), POS와 불용어(stop-word) 필터링, 약어 확장, 그리고 동의어 탐색을 수행하고, 다음으로 시맨틱 온톨로지를 활용한다.

오퍼레이션은 벡터 $O=<d, Is, Os>$로 정의될 수 있는데, 여기서 하나의 오퍼레이션은 이름과 텍스트 설명 d와 최소한 하나의 입력 Is, 하나의 출력 Os로 구성된다. 각 입력과 출력에는 다수의 매개변수들이 집합으로 구성되어 있다. 따라서 두 개의 오퍼레이션이 주어졌을 때 전체 유사도는 각각의 개별 벡터 컴포넌트들에 대한 유사도의 합으로 결정된다. 먼저, 텍스트 정보(d)에 대한 유사도를 측정한다. 이는 전통적으로 IR(Information Retrieval) 분야에서 널리 사용되고 있는 TF/IDF 방법[10]을 사용하면 쉽게 계산될 수 있다. 다음으로, 입력과 출력 매개변수들의 유사도를 측정한다. 클러스터링을 고려한 입력은 형식적으로 벡터 $Is=<v_i, C_i>$로 묘사된다(출력도 이와 비슷하게 $Os=<v_o, C_o>$로 표현될 수 있다). 여기서 v_i는 입력 매개변수들의 집합이고, C_i는 v_i와 연관된 클러스터링 개념이다. 따라서 입력 유사도는 다음의 두 단계로부터 계산된다(출력도 비슷한 방법으로 처리된다): (1) v_i에 대해 토큰화, POS, 동의어와 같은 전처리 과정을 먼저 수행한다, (2) 이로부터 나온 각 용어들에 대해 관련된 개념들을 찾아 교환하고 이에 대한 유사도를 계산한다.

유사도는 다음과 같이 매개변수 쌍들의 평균값으로 계산된다. 하나의 질의와 저장소로부터 매치되는 임의의 후보 오퍼레이션 쌍을 (Q, O)라 하고, Q와 O에는 각각 m과 n개의 매개변수들이 있다고 가정하면, Q=$(q_1, q_2, \cdots q_i, \cdots q_m)$와 O=$(o_1, o_2, \cdots o_j, \cdots o_n)$로 표현될 수 있으며 Q의 q_i와 O의 o_j 간의 매치를 고려할 때, 클러스터링 기반 매개변수 유사도(ParaSim)는

$$ParaSim = 2 * \sum_{i=1}^{m} Match(q_i, o_j)/(m+n)$$

여기서, $Match(q_i, r_j) = \max\{TF/IDF(q_i, o_j)\}$ *for all* $1 \le j \le n$, $i = 1, 2, \cdots m$ 이다.

위 식을 이용하여 입력과 출력 매개변수 유사도를 각각 계산할 수 있으며, 전체적으로 질의와 저장소에 있는 임의의 오퍼레이션 간의 유사도(Similarity)는 다음과 같이 계산된다.

$$Similarity = w_1(ParaSim_x) + w_2(ParaSim_i) + w_3(ParaSim_o)$$

여기서, $ParaSim_x$는 텍스트 유사도, $ParaSim_i$는 입력 매개변수 유사도, 그리고 $ParaSim_o$는 출력 매개변수 유사도이며, w_1, w_2, w_3는 각각의 가중치이고 $\sum w_i = 1$이다. 유사도 결과 값은 0과 1 사이의 실수 값을 리턴한다.

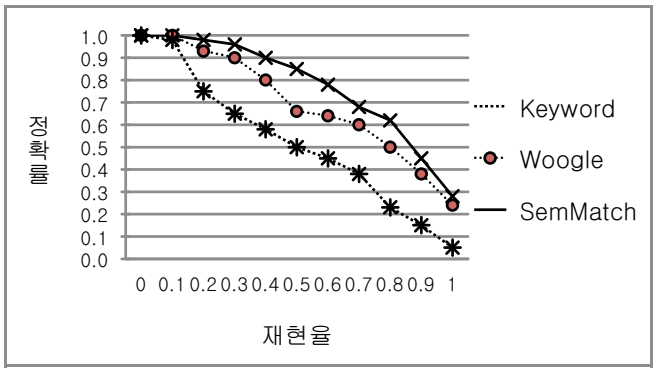
그러나 위의 유사도 측정 방법에서는 아직까지 형태소 분석 기법은 고려하지 않았다. 이를 통한 온톨로지를

활용함으로써 검색 결과의 정확률을 향상시킬 수 있다. 클러스터링만 사용한 API 유사성 탐색에서는 연관성 높은 단어들을 한 클러스터에 묶어서 단지 동일한 개념처럼 취급할 뿐 용어들 사이의 계층관계를 무시함으로써 사용자의 의도와는 관계없는 매칭(즉, false positives)이 발생할 수 있다. 따라서 매개변수 유사도를 계산할 때 계층관계 온톨로지 개념에 위배되는 매개변수 쌍들을 배제하여 사용자가 만족할 수 있는 품질 좋은 것만 선택할 수 있는 정제과정이 수행된다.

5. 실험 분석

본 장에서는 제안된 API 유사성 탐색 방법의 성능을 평가하기 위한 실험을 수행한다. 실험 분석을 위한 자료는 programmableWeb과 xmethod.net 사이트로부터 168개의 REST API와 50개의 SOAP API들을 다운로드 받아, 이를 파싱한 후 전처리 과정을 거쳐 우리의 알고리즘들이 적절히 수행될 수 있는 데이터 파일을 준비하였다.

본 논문에서 제안하는 유사성 탐색 방법의 우수성을 분석하기 위해 (그림 1)과 같은 R-P 커브(curve)를 작성하였다. R-P 커브는 IR 분야에서 검색 엔진의 효율성을 가장 잘 표현하는 그래프로써 알려져 있다. 이 그래프에서 X축은 재현율(recall)을 표시하고 Y축은 정확률(precision)을 나타내며, 가장 높이 있는 커브가 가장 좋은 성능을 나타낸다. 실험은 기존의 전통적인 키워드 검색 방법과 클러스터링 기법을 적용한 Woogle[9] 방법을 우리의 방법과 비교하여 제안된 방법의 우수성을 보여준다.



(그림 1) R-P 커브

(그림 1)에서 알 수 있듯이 기존의 키워드 검색 방법은 가장 낮은 성능을 보이고 있다. Woogle 방법은 키워드 검색보다는 상당한 성능 향상을 보이는데, 이는 연관성이 높은 용어들을 한 클러스터에 묶어 동일한 개념처럼 취급함으로써 검색의 재현율을 향상시킬 수 있기 때문이다. 그러나 재현율이 향상된 만큼 비례적으로 사용자의 의도와는 관계없는 false positive 매칭이 발생됨에 따라 정확률의 증가는 크게 기대할 수 없다. 우리의 접근방법(SemMatch)은 계층적 결합 클러스터링 알고리즘에 계층관계 형태소 분석 기법을 추가하여 검색 결과들 중 부적합한 API들을 정제함으로써 재현율과 정확률을 함께 증가시킬 수 있다. 결론적으로 Woogle과 우리의 접근방법은 기존의 키워드 검색 방법보다는 더 나은 성능을 보여주고 있으며, 본 논문에서 제안하는 방법은 이들 중 가장 좋은 성능을 보여주고 있다.

6. 결론

본 논문에서는 REST, SOAP, JavaScript, 그리고 XML-RPC와 같은 다양한 API 타입들을 지원하는 매쉬업을 위한 Open API 유사성 탐색 방법을 제안하였다. 이를 위해 먼저 시맨틱 온톨로지를 구축하기 위해 WSDL/WADL/HTML 파일에 존재하는 신텍틱 정보를 활용하고 그들 사이에 숨겨져 있는 시맨틱 개념을 얻기 위해 마이닝 알고리즘을 적용하였다. 계층관계 형태소 분석 방법은 API 매개변수 내에 포함된 용어들 간의 상관관계를 취득하고, 만일 두 용어들이 서로 유사하고 상관관계가 조건에 일치한다면 그 매개변수를 매치한다. 본 연구에서 제안된 알고리즘들은 관련 포털 사이트로부터 실제 사용되고 있는 자료를 다운로드 받아 실험 분석을 수행하였다. 실험 결과 기존의 키워드 검색 방법과 Woogle 방법보다 재현율/정확률 측면에서 상당한 성능 향상을 보였다.

참고문헌

[1] http://www.programmableweb.com
[2] http://sebservices.seekda.com
[3] http://pipes.yahoo.com/pipes
[4] http://www.popfly.ms
[5] http://software.intel.com/en-us/articles/intel-mash-maker-mashups-for-the-masses/
[6] D. E. Simmen, M. Altinel, V. Markl, S. Padmanabhan, and A. Singh, "Damia: Data Mashups for Intranet Applications," In Proceedings of the ACM SIGMOD International Conference on Management of Data, pp. 1171-1182, 2008
[7] A. V. Riabov, E. Bouillet, M. D. Feblowitz, Z. Liu, and A. Ranganathan, "Wishful Search: Interactive Composition of Data Mashups," In Proceedings of the ACM International Conference on World Wide Web, pp. 775-784, 2008
[8] J. Tatemura, S. Chen, F. Liao, O. Po, K. S. Candan, and D. Agrawal, "UQBE: Uncertain Query By Example for Web Service Mashup" In Proceedings of the ACM SIGMOD International Conference on Management of Data, pp. 1275-1280, 2008
[9] X. Dong, A. Halevy, J. Madhavan, E. Nemes, and J. Zhang, "Similarity Search for Web Services," In Proceedings of VLDB, pp. 372-383, 2004
[10] G. Salton and C. Buckley, "Term Weighting Approaches in Automatic Text Retrieval," Information Processing and Management, Vol.24, No.5, pp. 513-523, 1988