유사도 알고리즘을 활용한 시맨틱 프로세스 검색방안
Semantic Process Retrieval with Similarity Algorithms

이홍주 [1], Mark Klein[2]
[1] 가톨릭대학교 경영학부, hongjoo@catholic.ac.kr
[2]Center for Collective Intelligence, MIT, m_kline@mit.edu

Abstract

One of the roles of the Semantic Web services is to execute dynamic intra-organizational services including the integration and interoperation of business processes. Since different organizations design their processes differently, the retrieval of similar semantic business processes is necessary in order to support inter-organizational collaborations. Most approaches for finding services that have certain features and support certain business processes have relied on some type of logical reasoning and exact matching.

This paper presents our approach of using imprecise matching for expanding results from an exact matching engine to query the OWL MIT Process Handbook. In order to use the MIT Process Handbook for process retrieval experiments, we had to export it into an OWL-based format. We model the Process Handbook meta-model in OWL and export the processes in the Handbook as instances of the meta-model. Next, we need to find a sizable number of queries and their corresponding correct answers in the Process Handbook. We devise diverse similarity algorithms based on values of process attributes and structures of business processes. We perform retrieval experiments to compare the performance of the devised similarity algorithms.

1. 서론

데이터와 서비스가 의미를 포함하게 되어 조직간 응용프로그램의 연계와 자동화를 활성화할 수 있을 것이라는 시맨틱 웹의 목표를 이루기 위해 학계와 산업계의 많은 노력과 연구들이 이루어져 왔다. 시맨틱 웹의 목표를 이루기 위한 어려운 점들도 논의되고 있으나[McCool, 2005], 데이터의 표현에 있어서 시맨틱 웹 기술이 많이 활용되고이다. 조직내외의 비즈니스 프로세스를 연계하고 상호연동하기 위한 시맨틱 비즈니스 프로세스에 대한 연구도 수행되어 왔다[Ehrig, Koschmideret al., 2007]. 시맨틱 프로세스는 비즈니스 프로세스를 시맨틱 언어로 표현한 것을 뜻하며, 이로 인해 모호하지 형태로 프로세스가 표현되기 때문에 컴퓨터 추론이 가능하고 프로세스 구성자동화가 가능해진다[Ehrig, Koschmideret al., 2007]. 그러나, 조직들이 비즈니스 프로세스를 다양한 방식으로 표현하고 있기 때문에 조직간 협업을 활성화하기 위해서는 연계가능한 유사한 시맨틱 비즈니스 프로세스를 찾는 것이 필요하다.

시맨틱 웹의 발전과 함께 다양한 시맨틱 웹 리소스에서 적합한 정보를 찾기 위한 노력들이 진행되어왔다. 시맨틱 웹 리소스가 포함하고 있는 정보를 바탕으로 사용자의 질의어와 매칭되는 리소스를 찾아주기 위한 노력들이 있었으며, RDF 기반의 리소스를 검색하기 위한 질의 언어로 RQL, RDQL, SPARQL 등이 제시되어 왔다 [Haase, Broekstraet al., 2004]. URI, RDF 문장에 포함되어 있는 정보와 RDF 하위그래프 정보를 추출하여 질의어와 매칭되는 정보를 찾는 정확한(exact) 매칭 방안이나 추론방안을 활용하고 있다[Klusch, Frieset al., 2006]. 정확한 매칭방안이 문제시 되는 경우는 같은 의미이나 다른 용어로 표현된 경우와 유사정보를 찾기가 어렵다는 것이며, 이로 인해 정보검색에서 이야기하는 정확도(precision)는 높으나 상기도(recall)는 낮아지게 된다. 따라서 정확도를 유지하면서 상기도를 높이기 위한 방안으로 유사한(imprecise) 매칭방안을 정확한 매칭과 보완해서 사용하려는 노력들이 있어왔다 [Bernstein and Kiefer, 2006; Kiefer, Bernsteinet al., 2007; Klusch, Frieset al., 2005; Klusch, Frieset al., 2006].

시맨틱 프로세스를 표현하고 유사한 프로세스를 검색하는 연구는 현재까지 시맨틱 프로세스의 구조를 제시하고 이 구조에 맞게 인위적으로 생성된 프로세스 데이터를 활용하여 실험을 실시하여 왔으며, 프로세스 구조보다는 프로세스 설명이나 이름과 같은 데이터에 기반하여 유사한 프로세스를 찾아왔다.

본 연구의 목적은 실제 비즈니스 프로세스 구조에 바탕을 두어 시맨틱 프로세스를 표현하고, 효과적인 프로세스 검색을 위해 정확한 매칭방안의 결과를 확장하여 유사한 프로세스를 검색하는 유사도 기반의 검색 알고리즘을 제시하는 것이다. 이를 위해 MIT 프로세스 핸드북 프로젝트에서 구축해 놓은 기업들의 프로세스 데이터를 OWL 로 표현하였으며, 표현된 비즈니스 프로세스 데이터를 SPARQL 과 연계하여 검색하기위한 유사도 알고리즘을 제시하고 이들의 검색성과를 비교한다. 활용한 유사도 알고리즘들은 프로세스의 구조와 프로세스가 가지고 있는 속성들의 설명이나 이름을 활용하는 기본적인 방안들을 조합하여 활용하였다.

2. 시맨틱 프로세스 표현

MIT 프로세스 핸드북(Process Handbook)은 MIT Center for Coordination Science 에서 기업의 베스트 프랙티스(best-practice) 비즈니스 프로세스를 표현하고 축적하기 위해 구현한 프레임워크이며 이에 기반을 두어 실제 기업들의

비즈니스 프로세스를 축적하여왔다. 프로세스 핸드북의 목적은 기업의 비즈니스 프로세스 재설계, 새로운 프로세스의 설계, 그리고 기업 프랙티스들에 대한 아이디어를 공유하기 위해 고안되었다 [Malone, Crowstonet al., 1999]. 현재까지 약 8 천 개의 비즈니스 프로세스를 축적하였으며, 프로세스 검색, 분석과 편집을 위한 도구들을 구현하여 왔다 [Malone, Crowstonet al., 2003]. 프로세스 핸드북에서 비즈니스 프로세스는 프로세스 계층관리(specialization), 연관관계(dependency) 관리, 예외관리(exception handling)의 세가지 개념에 바탕을 두어 표현되고 있다.

프로세스 핸드북에 정의된 구조와 개념들에 바탕을 두어 프로세스 핸드북 온톨로지를 설계하였다. 그림 1 이 설계된 프로세스 핸드북 온톨로지의 개요이며, 온톨로지에 포함된 개체들과 기본적인 연관관계가 표현되어 있다. 설계된 프로세스 핸드북 온톨로지를 메타모델로 삼아, 프로세스 핸드북에 축적된 모든 비즈니스 프로세스들을 Web Ontology Language (OWL)로 표현하였다[1]. 프로세스 핸드북에 표현되어 있는 프로세스, 번들, 목적(Goal), 예외, 자원(Resource), 연관관계(Dependency) 그리고 Trade-off 등 모든 개체들을 OWL 클래스로 표현하였다.

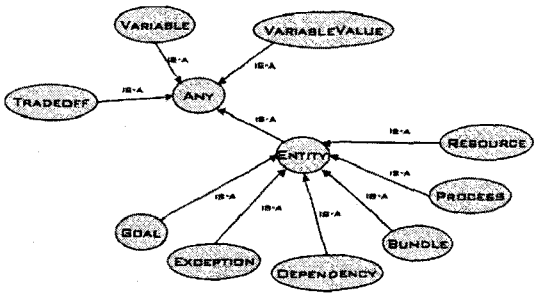

그림 1 프로세스 핸드북 온톨로지 구조

그림 2 는 OWL 로 표현된 비즈니스 프로세스 "E1024"이며,하나의 예외와 2 개의 프로세스 특수화 그리고 하나의 프로세스 일반화 관계()를 가지고 있다.

```
<?xml version="1.0" ?>
<!DOCTYPE RDF>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" xmlns:rdfs="http://www.w3.org/2000/01/rdf-sc
hema#" xmlns:ph="http://www.ifi.unizh.ch/ddis/ph/2006/08#" xmlns:process="http://www.ifi.unizh.ch/ddi
s/ph/2006/08/ProcessHandbook.owl#" xml:base="http://www.ifi.unizh.ch/ddis/ph/2006/08/E1024.owl">
  <processHandbook:Process rdf:ID="E1024">
  <processHandbook:name xml:lang="en">Determine cost</processHandbook:name>
  <processHandbook:description xml:lang="en">This is a general activity to determine the cost to the organization of pur
chase or production.</processHandbook:description>
  <processHandbook:hasException rdf:resource="http://www.ifi.unizh.ch/ddis/ph/2006/08/E17159.owl#E17159" />
  <processHandbook:hasSpecialization rdf:resource="http://www.ifi.unizh.ch/ddis/ph/2006/08/E6302.owl#E6302" />
  <processHandbook:hasSpecialization rdf:resource="http://www.ifi.unizh.ch/ddis/ph/2006/08/E8007.owl#E8007" />
  <processHandbook:hasGeneralization rdf:resource="http://www.ifi.unizh.ch/ddis/ph/2006/08/E3356.owl#E3356" />
  </process:Handbook:Process>
</rdf:RDF>
```

그림 2 OWL 로 표현된 비즈니스 프로세스

[1] http://www.ifi.unizh.ch/ddis/ph-owl.html 에 온톨로지 정보, OWL 파일, 실험집합 등 관련 정보들이 공유되어 있다.

3. 시맨틱 프로세스 검색방안

시맨틱 프로세스간의 유사도를 계산하기 위하여 iSPARQL 프레임워크[Kiefer, Bernsteinet al., 2007]을 활용하였다. iSPARQL 은 SPARQL 에 바탕을 두어 유사도 알고리즘을 통해 검색결과를 확장할 수 있도록 구현된 검색엔진이다. iSPARQL 에서는 SimPack[Bernstein, Kaufmannet al., 2005]에서 구현된 유사도 알고리즘을 모두 활용할 수 있다. 실험 데이터의 검색을 위해 일반적으로 활용할 수 있는 기본 유사도 알고리즘과 실험 데이터의 특성을 고려하여 설계된 유사도 알고리즘을 검색 실험에 활용하였으며, 표 1 에 활용된 알고리즘을 정리하였다. 유사도 알고리즘에서 유사도를 고려하는 기본적인 정보는 프로세스의 이름과 설명에 포함된 키워드 (TFIDF, LevN)와 프로세스가 포함하고 있는 개체 (Dice 와 Jaccard 척도)이다. 설계된 방안들은 기본방안들을 조합하여 활용하거나 프로세스가 여러 유형의 개체들을 포함하고 있기 때문에 이를 고려한 가중치 계산방안이 활용되었다.

표 1 활용된 유사도 알고리즘

1. TF-IDF
프로세스 설명부분에 포함된 키워드들을 활용하여 두 프로세스간의 TFIDF 값을 계산한다. 미리 계산된 각 키워드들의 빈도에 대한 정보를 활용한다. TFIDF 는 전통적으로 활용되는 코사인 척도를 확장한 개념이라고 볼 수 있다 [Baeza-Yates and Ribiero-Neto, 1999].
2. Lev
프로세스의 이름을 가지고 가지고 Levenshtein string edit distance 를 활용하여 유사도를 계산한다. Levenshtein string edit distance 는 하나의 문자열에서 다른 문자열로 전환하기 위해 필요한 삽입, 삭제 그리고 변환행위의 수를 가지고 거리를 계산한다[Levenshtein, 1966].
3. Dice-Parts
프로세스에 포함된 parts 를 비교하여 두 프로세스간의 Dice 유사도를 계산한다. Dice 유사도는 동일한 parts 의 수를 두 개의 프로세스 parts 의 수로 나누어 계산된다.
4. Jaccard-Parts
프로세스에 포함된 parts 를 비교하여 두 프로세스간의 Jaccard 유사도를 계산한다. Jaccard 유사도는 코사인 척도와 같이 벡터 공간 모델에 기반을 두고 있으며, Dice 척도보다 더 공통적인 값이 적은 경우에 많은 유사도의 감소가 있다. 두개의벡터 X, Y 의 Jaccard 유사도는 아래와 같이 계산된다. (X*Y) / (|X||Y|-(X*Y)), (X*Y)는 두 벡터의 내적이며, |X| = (X*X)^1/2 이다.
5. TreeEdit
프로세스에 포함된 parts 들이 트리 형태를 띄고 있기 때문에 이를 이용하여 프로세스들간의 트리 변경 거리(Tree edit distance)[Valiente, 2002]를 계산하였다. 트리 변경 거리가 작을 수록 유사한 프로세스로

판단할 수 있다.
6. Lev-TFIDF
프로세스 이름을 활용한 Levenshtein Level 2 유사도 계산방안과 프로세스 설명을 활용한 TFIDF 유사도 계산방안 결과를 조합하여 최종 유사도를 계산하는 방안이다. 프로세스 이름에 포함된 키워드를 활용하여 두 프로세스간의 Levenshtein string 유사도[Bernstein, Kaufmannet al., 2005]를 계산한다. Levenshtein string edit distance 는 기본적으로 하나의 문자열이 다른 문자열로 변하기 위해서 필요한 삽입, 삭제, 교환 작업의 수로 계산된다.
7. JaccardAll
4 번 방안인 Jaccard-Parts 는 프로세스가 가지고 있는 part 프로세스만 고려하여 유사도를 계산하는 것이라면, JaccardAll 은 프로세스에 포함된 모든 개체(part, 예외, 목적, 자원, input 과 output)를 활용하여 Jaccard 유사도를 계산하는 방안이다. 이때 각 개체별 유사도를 합칠때 개체별 가중치는 동일하다.
8. Weighted JaccardAll
프로세스에 포함된 모든 개체를 활용한 Jaccard 유사도 계산 방안이며, 각 개체유형의 수에 따라 매칭 여부에 대한 가중치를 부여하여 최종 유사도를 계산한다. 위의 7 번방안에서 개체별 가중치로 Part, 예외, 자원, input, output 에 0.5, 0.2, 0.1, 0.1, 0.1 를 부여하였다.
9. Lev-TFIDF-JaccardAll
프로세스 이름을 활용한 Levenshtein Level 2 유사도 계산방안, 프로세스 설명을 활용한 TFIDF 유사도 계산방안과, 프로세스 개체에 대한 Jaccard 유사도 계산방안의 결과를 조합하여 최종 유사도를 계산하는 방안이다.

위의 유사도 알고리즘은 iSPARQL 프레임워크에 구현되어 두 프로세스간의 유사도를 계산하였다.

4. 실험

3 절에서 생성된 프로세스 데이터를 가지고 프로세스 검색 실험을 수행하기 위해서는 검색하기 위한 질의 프로세스와 이에 대응되어 검색되어야만 하는 응답 프로세스들을 정의하여야 한다. 관련연구에서 언급된 것처럼 현재까지는 연구진들에 의해 분류되거나 인공적으로 생성된 의미 없는 프로세스들을 가지고 실험을 수행하였다. 본 연구에서는 의미가 보존되면서(semantic-preserving) 프로세스를 변화(process mutation)시키는 방안을 통해 실험 집합을 구성하는 방안을 활용하였다. 우리는 프로세스 핸드북에서 특수화나 일반화 관계가 구성되어 있는 프로세스들 중에서 80 개의 프로세스를 선정하여서 이를 목표집합(target set)으로 삼았다. 목표집합에 속한 프로세스 마다 형식적(syntactic)으로는 상이하지만 의미적(semantic)으로는 동일한 20 개의 돌연변이 프로세스를 생성하였다. 이 돌연변이들이 목표집합에 속한 프로세스와 유사한 프로세스를 검색하였을 때, 검출되어야 하는 정확한 응답집합이다. 목표 프로세스 별로 응답집합에 속한 프로세스들을 제외한 다른 프로세스들이 검색되는 경우에는 부정확한 결과로 계산하였다.

각 돌연변이들은 의미를 보존하는 변형방안들에 의해서 원래의 목표 프로세스에서 변화해 나가며, 응답집합에는 한번 변형한 경우부터 20 번 변형한 것까지의 돌연변이 프로세스들이 포함되었다. 응답집합 생성에 활용된 변형방안들은 다음 표 2 과 같다.

표 2 프로세스 변형방안

변형부분	이름	설명
프로세스 (part)	STEPSPLIT	하나의 part를 두 개의 형제 part로 변형
	STEPCHILD	하나의 part를 부모/자식 part 관계로 변형
	STEPMERGESIB	임의로 선정된 두 개의 형제 part를 하나로 병합
	STEPMERGEPARENT	하나의 part를 이의 부모 part와 병합
	STEPDELETE	하나의 part 삭제
표기내용	NAMEDELETE	프로세스 이름 중 하나의 단어 삭제
	DESCRIPTIONDELETE	프로세스 설명 중 하나의 단어 삭제

변형방안들은 프로세스를 모델링하는 다양한 다른 방안들에 바탕을 두어 선택되었다. 만약 우리가 식당의 비즈니스 프로세스를 모델링하는 경우에 패스트푸드 식당의 경우 '주문'과 '계산'을 하나의 행위로 표현할 수도 있고 다른 식당의 경우에는 나누어서 표현할 수도 있다. 위 두 프로세스는 표현양식은 좀 다를지라도 식당에서 주문과 계산이라는 의미를 모두 포함하고 있다. STEPMERGESIB 가 위의 경우에 해당하는 변형방안이며 반대로 STEPSPLIT 이 하나의 part 를 두 개의 part 로 나누는 변형방안이다. 물론 위의 변형방안에 상당한 임의(random)요소가 작용하고 있다. 연구자에 의해서 의미에 기반을 두어 part 가 나누어 지거나 합쳐진 것이 아니고 프로그램에 의해 임의로 선정된 part 들이 나누어 지거나 합쳐지고 일부 설명부분이 삭제되었다. 따라서, 목표 프로세스의 응답집합에 속한 돌연변이들이 사람들이 의미에 기반을 두고 생성한 돌연변이와는 차이가 있을 수 있다. 그러나 이러한 변형방안들이 큰 규모의 실제 비즈니스 프로세스를 가지고 테스트 데이터를 생성하는데 유용하리라고 믿는다. 목표집합에 속한 프로세스 마다 20 개의 돌연변이 프로세스를 생성하였다. 총 27,953 개의 프로세스와 80 개의 목표집합 프로세스의 돌연변이 1600 개를 대상으로 표 1 에 제시한 유사도 척도를 활용하여 각 목표집합에 속한 프로세스와의

유사도를 계산하였다. 각 유사도 척도 별로 유사도가 가장 높은 프로세스부터 100 번째로 높은 프로세스까지를 점검하면서 이중에 몇 개의 돌연변이 프로세스가 포함되어 있는지를 파악하였다. 성과 측정기준으로는 정확도(precision), 상기도(recall), F 척도를 활용하여 각 척도 별로 측정하였다.

그림 3, 그림 4, 그림 5 는 실험방안들의 정확도, 상기도, F 값을 보여준다. Tree edit 거리를 활용한 5 번째 방안의 정확도가 가장 낮았으며, 1 번째 방안인 TFIDF 와 Levenshtein string 거리와 TFIDF 를 조합한 6 번째 방안이 가장 높은 정확도를 보였다. 프로세스의 Parts 를 이용한 Jaccard 유사도, Dice 유사도를 활용하는 방안들보다는 프로세스에 대한 설명에 근거한 Levenshtein string 을 활용하는 1, 2, 6, 9 번째 방안이 전반적으로 높은 정확도를 보였다. 이는 프로세스 돌연변이가 주로 프로세스 구조를 변화시키는 경우가 많으며 이름이나 설명부분을 변화시키는 것이 적기 때문인 것으로 보인다. 그러나 이러한 프로세스 구조와 설명 부분을 모두 고려하는 9 번째 방안이 비교적 높은 정확도를 보였다.

실험방안별 상기도를 비교하여 정리한 그림 4 에서도 평균 정확도를 비교한 그림 7 과 유사한 결과를 얻었다. Tree edit 거리를 활용한 5 번째 방안이 가장 낮은 값을 나타냈으며, TFIDF 를 활용하는 1, 6, 9 번째 방안이 높은 상기도를 보였다.

일반적으로 정확도와 상기도는 정확도가 증가하면 상기도가 하락하고, 상기도가 증가하면 정확도가 하락하는 역의 관계에 있기때문에, 두 지표를 조합하여 하나의 성과지표로 삼기도 한다[Baeza-Yates and Ribiero-Neto, 1999]. 조화평균(Harmonic mean)의 값은 P 가 정확도, r 이 상기도라고 할때 $F = 2/(1/r + 1/P)$와 같이 계산된다. 조화평균은 0 과 1 사이의 값을 가지며 0 은 관련 문서가 검색되지 않은 경우이며, 모든 문서가 검색된 경우에는 1 의 값을 갖는다. 그림 3 과 그림 4 에 표기된 각 방안들의 정확도와 상기도를 조합하여 계산한 조화평균의 값이 그림 5 이다.

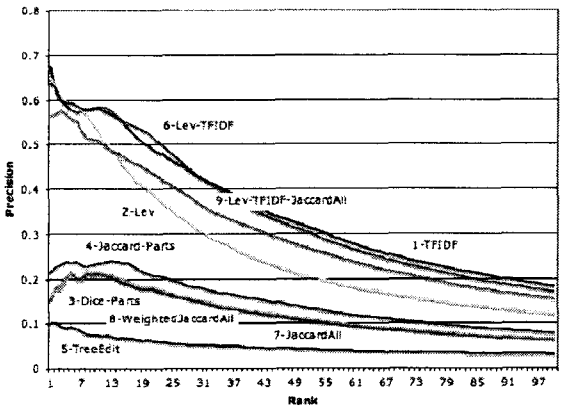

그림 3 실험방안별 평균 정확도

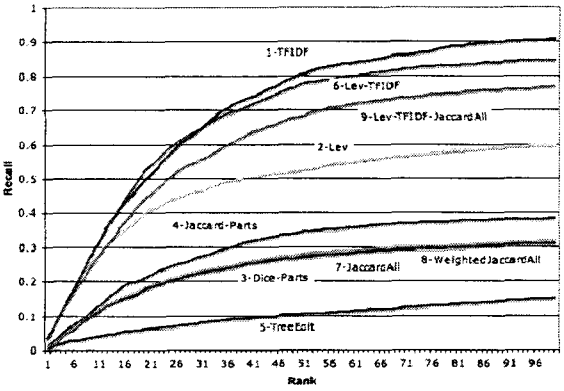

그림 4 실험방안별 평균 상기도(Recall)

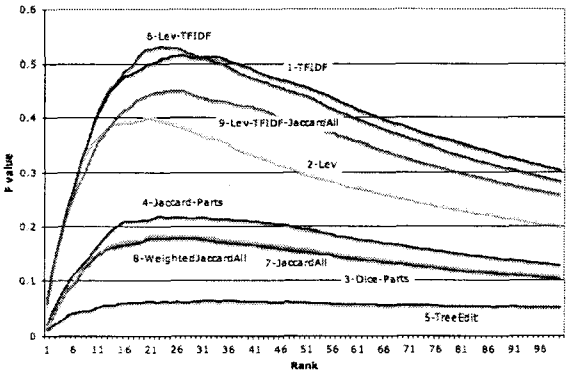

그림 5 실험방안별 조화평균

프로세스의 이름과 설명부분의 유사도를 고려하는 1 과 6 번째 방안이 가장 높은 값을 나타냈으며, Tree edit 거리 척도를 활용한 5 번째 방안이 가장 낮은 값을 보였다. 프로세스 구조와 설명을 모두 고려하는 9 번째 방안도 좋은 성과를 보였다.

또한, 돌연변이 프로세스들은 프로세스 변형 정도에 따라 1 부터 20 까지의 돌연변이 정도 값을 가지고 있다. 적은 수의 프로세스 변형이 이루어진

돌연변이 프로세스가 구조상으로는 좀더 원래 프로세스와 유사하다고 볼 수 있다. 따라서, 돌연변이 정도 순위와 각 유사도 알고리즘에 의해 판단된 유사도 순위를 비교할 수 있다. 동일한 수만큼 변형된 프로세스와 동일한 유사도 값을 가진 프로세스들이 있기 때문에 동일 순위(tie)를 고려한 순위상관계수인 Kendall's tau b 척도를 활용하여 유사도 알고리즘에 의한 유사도 순위와 프로세스 변형정도간의 상관관계를 측정하였으며, 결과는 표 3 과 같다. 조화평균 지표와 다르게 돌연변이 정도와 유사도 순위간의 관계에서는 프로세스 구조의 다양한 부분의 유사도를 평가하는 7, 8 번째 방안이 그렇지 않은 다른 방안들에 비해서 높은 순위상관계수 값을 나타냈으며, 프로세스 구조간의 유사도를 고려하는 3, 4 번째 알고리즘도 7, 8 번째 방안과 거의 유사한 결과를 나타내었다. 프로세스의 이름과 설명간의 유사도를 고려하는 1, 2, 5, 6 번 방안은 상대적으로 낮은 성과를 보였으며, 프로세스 구조와 설명을 모두 고려하는 9 번째 방안은 두 가지 방안유형들의 중간 값을 보였다.

표 3 순위상관계수

실험방안	Pearson	Kendall's tau b
1-TFIDF	0.439*	0.325*
2-Lev	0.343*	0.273*
3-Dice-Parts	0.653*	0.562*
4-Jaccard-Parts	0.659*	0.524*
5-TreeEdit	0.389*	0.351*
6-Lev-TFIDF	0.427*	0.317*
7-JaccardAll	0.680*	0.568*
8-WeightedJaccardAll	0.674*	0.563*
9-Lev-TFIDF-JaccardAll	0.568*	0.438*

* 유의수준 0.01 에서 유의

5. 토의 및 결론

정확한 매칭방안에 기반을 두고 있는 SPARQL 의 검색결과를 확장하기 위하여 우선 SPARQL 로 검색된 프로세스들을 찾고, 각 프로세스와 유사하지만 정확한 매칭방안에 의해서는 검색되지 않는 다른 프로세스들을 찾는 방식을 활용하였다. 유사한 프로세스를 찾기 위하여 활용된 프로세스의 정보는 프로세스 계층정보 및 속성정보와 같은 구조정보와 프로세스 이름, 설명과 같은 속성이 가지고 있는 값을 활용하였다. 일반적으로 검색에 활용되는 기본방안들과 이들을 혼합하여 활용하는 방안들을 설계하였으며, iSPARQL 을 통하여 각 방안들의 검색 성과를 비교하였다.

검색성과를 비교하는 정확도, 상기도, F 척도에서는 프로세스의 속성값을 활용하는 방안들이 좋은 성과를 보였으며, 돌연변이 프로세스의 돌연변이 정도와 유사도 순위간의 상관관계를 측정한 실험에서는 프로세스의 구조정보를 활용한 방안들이 높은 성과를 보였다. 실험데이터인 돌연변이 프로세스를 만드는 과정에서 속성값을 변형시키는 정도보다 구조정보를 변형시키는 경우가 더 많이 활용되었기 때문에 유사도를 측정함에 있어서는 비교적 덜 변형된 속성값 정보를 활용하는 것이 좋은 성과를 보였다고 판단된다. 또한, 속성값 정보와 구조정보를 동시에 고려하는 유사도 척도인 9 번째 방안이 검색성과와 돌연변이 정도 측정에서 모두 좋은 값을 보였다. 이는 프로세스의 유사도를 바라보는 측면이 다양할 수 있기 때문에, 다양한 측면을 고려한 유사도 알고리즘을 설계하는 것이 유사 프로세스 검색에서 유용하게 활용될 수 있음을 보였다고 할 수 있다.

향후 연구 방향으로는 본 연구에서는 하나의 데이터 집합을 활용하여 연구를 수행하였기 때문에 다른 도메인의 시맨틱 웹 자원을 활용하여 연구결과의 일반화 가능성을 확인하여야 할 필요가 있다. 또한 다양한 유사도 척도에서 얻어지는 유사도 결과들을 활용하는 것이 하나의 유사도 척도를 활용하는 방안보다 더 좋은 검색 성과를 얻을 수 있을 것이다. 따라서 여러 알고리즘에 의한 유사도 결과를 동시에 고려할 수 있는 검색방안에 대한 연구가 필요하다.

관련연구

R. Baeza-Yates and B.d.A. Ribiero-Neto, Modern Information Retrieval, ACM, 1999.

A. Bernstein, E. Kaufmann, C. Kiefer, and C. Bürki, *SimPack: A Generic Java Library for Similarity Measures in Ontologies*, Technical Report, Department of Informatics, University of Zurich, 2005.

A. Bernstein and C. Kiefer, "Imprecise RDQL: Towards Generic Retrieval in Ontologies Using Similarity Joins," in *Proceedings of SAC'06*, Dijon, France, 2006, ACM, pp. 1684-1689.

M. Ehrig, A. Koschmider, and A. Oberweis, "Measuring Similarity between Semantic Business Process Models," in *Proceedings of the 4th Asia-Pacific Conference on Conceptual Modelling (APCCM'07)*, Ballarat, Victoria, Australia, 2007, pp. 71-80.

P. Haase, J. Broekstra, A. Eberhart, and R. Volz, "A Comparison of RDF Query Languages," in *Proceedings of ISWC 2004*, 2004, pp. 502-517.

C. Kiefer, A. Bernstein, and M. Stocker, "The Fundamentals of iSPARQL - A Virtual Triple Approach For Similarity-Based Semantic Web Tasks," in *Proceedings of Proceedings of the 6th International Semantic Web Conference (ISWC)*, 2007.

M. Klusch, B. Fries, M. Khalid, and K. Sycara, "OWLS-MX: Hybrid OWL-S Service Matchmaking," in *Proceedings of AAAI'05*, 2005.

M. Klusch, B. Fries, and K. Sycara, "Automated Semantic Web Service Discovery with OWLS-MX," in *Proceedings of AAMAS 2006*, Hakodate, Hokkaido, Japan, 2006.

V.I. Levenshtein, "Binary Codes Capable of Correcting Deletions, Insertions and Reversals", Soviet Physics Doklady, vol. 10, 1966, pp. 707-710.

T.W. Malone, K. Crowston, and G. Herman, ed., *Orgznizing Business Knoweldge: The MIT Process*

Handbook, Cambridge, Massachusetts, USA: MIT Press, 2003.
T.W. Malone, K. Crowston, J. Lee, and B. Pentlad, "Tools for inventing organizations: Toward a handbook of organizational processes", Management Science, vol. 45, no. 3, 1999, pp. 425-443.
R. McCool, "Rethinking the Semantic Web, Part 1", IEEE INTERNET COMPUTING, vol. 9, no. 6, 2005, pp. 86-88.
G. Valiente, Algorithms on Trees and Graphs, Springer-Verlag, Berlin, 2002.