대표 속성을 이용한 저자 개체 식별

Author Entity Identification using Representative Properties in Linked Data

김태홍*, 정한민**, 성원경**, 김 평**
과학기술연합대학원대학교*, 한국과학기술정보연구원**

Taehong Kim(kimtaehong@kisti.re.kr)*, Hanmin Jung(jhm@kisti.re.kr)**,
Won-Kyung Sung(wksung@kisti.re.kr)** , Pyung Kim(pyung@kisti.re.kr)**

요약

급격하게 성장하고 있는 오픈 리소스인 링크드 데이터는 최근 선진국 정부의 많은 관심 속에 데이터 공개 및 상호운용성 확보를 위한 방안으로 주목받고 있다. 그러나 신뢰할 수 있는 개체 식별 기술의 부재로 링크드 데이터의 양적 성장에 비해 개체 수 대비 링크의 수가 적은 현상과 일부 데이터 셋에 링크가 집중되는 현상을 보이고 있다. 본 연구에서는 이러한 링크드 데이터의 문제를 해결하기 위해 개체 간 관계(owl:sameAs, owl:differentFrom 등)를 이용하거나 Curation 방식을 사용하는 기존 링크드 데이터 기반 개체 식별 방식의 문제를 다중 온톨로지의 개체 식별이 가능한 자동화된 개체 식별 방식을 통해 개선하고 저자 개체의 대응 속성과 개체 유형의 논리적 특성을 활용하여 개체 식별 정확성을 검증할 수 있는 다중 온톨로지 기반의 실시간 저자 식별 방법을 제안하고 평가한다. 본인의 확인을 거친 29명의 저자 정보를 이용해 개체 식별 정확성 결과를 평가하여 평균 0.8533 (K measure)의 긍정적인 성능을 보였다.

■ 중심어 : 링크드 데이터 | 저자 식별 | 개체 식별 | OntoURIResolver |

Abstract

In recent years, Linked Data that is published under an open license shows increased growth rate and comes into the spotlight due to its interoperability and openness especially in government of developed countries. However there are relatively few out-links compared with its entire number of links and most of links refer a few hub dataset. These occur because of absence of technology that identifies entities in Linked data. In this paper, we present an improved author entity resolution method that using representative properties. To solve problems of previous methods that utilizes relation with other entities(owl:sameAs, owl:differentFrom and so on) or depends on Curation, we design and evaluate an automated realtime resolution process based on multi-ontologies that respects entity's type and its logical characteristics so as to verify entities consistency. The evaluation of author entity resolution shows positive results (The average of K measuring result is 0.8533.) with 29 author information that has obtained confirmation.

■ keyword : Linked Data | Author Identification | Entity Resolution | OntoURIResolver |

2009년 5월 미국의 연방정부 데이터 통합 저장소[1]가
오픈되면서 시작된 공공정보의 공개는 정보 공유의 중
요성을 부각시켰고 영국, 호주, 캐나다, 뉴질랜드, 노르
웨이, 에스토니아 등으로 확산되어 전 세계적 흐름으로
인식되고 있다[1]. 이처럼 정보 공유 및 연계의 중요성
이 부각되면서 최근 다양한 분야에서 데이터 연계 및
병합을 위해 표준화된 지식 표현 모델의 하나인 온톨로
지를 생성하고 서비스에 적극 활용하고 있다. 하지만
시맨틱 서비스가 쉽게 확산되지 못하는 이유는 시맨틱
기술 수준이 실용화 단계로 아직 정착되지 못했다는 것
과 함께 시맨틱 서비스를 제공하기 위한 지식 표현 모
델로서 사용되는 온톨로지의 구축이 쉽지 않기 때문이
다. 온톨로지는 시맨틱 웹(Semantic Web)을 지향하는
지식 표현 기법의 하나로 W3C에서 정의된 Resource
Description Framework(RDF)와 RDF Schema (RDFS),
Web Ontology Language(OWL) 등으로 정보를 기술한
다. 이렇게 기술된 정보는 모든 개체마다 Uniform
Resource Identifier (URI)가 부여되어 관리 및 공유가
이루어지며 URI 간 상호 연관 관계를 이용해 외부 온톨
로지의 개체와 연계가 가능해진다[2].

일 개체에 할당된 다양한 URI를 체계적으로 관리하는 것은 물론, 새로 생성된 인스턴스와 기존 인스턴스간의 연계가 필요하다.

본 연구는 시맨틱 정보의 연계 및 융합의 활성화를 위한 속성 기반 개체 식별 서비스 개발을 목표로 하고 있다. 이번 연구에서는 링크드 데이터에서 가장 많은 비율을 차지하고 있는 도메인인 Publication분야(295개 중 87개의 데이터 셋)의 개체 인식을 목표로 저자 유형의 개체를 대상으로 다중 온톨로지의 개체를 식별하고 활용할 수 있는 속성기반 개체 식별 방법을 제안하고 평가를 수행하였다. 제안된 방법은 온톨로지 간 정보 연계를 위한 개체 식별 서비스의 기반이 될 것으로 기대한다.

논문의 구성은 서론에 이어 II 장에서는 관련 연구를 비롯하여 링크드 데이터의 현황 정보에 근거해 링크드 데이터의 문제를 기술하고, III장에서는 속성 기반 개체 식별 방법을 상술하며, IV장에서 개체 식별 결과의 평가 방법, 평가 결과를 기술하고, V장에서 결론을 맺는다.

1. 링크드 데이터의 현황과 문제점

링크드 데이터는 RDF로 기술된 웹 데이터를 URI를 통해 공개하여 정보의 공유 및 연계를 촉진하기 위한 방법으로 데이터를 상호 연계하기 위한 출판(Publishing) 및 접근 원칙을 기반으로 동일 도메인의 데이터는 물론, 이질적인 도메인의 데이터 연계에 중요한 역할을 수행한다.

그림 1. 링크드 데이터 Cloud
(<http://linkeddata.org>)

1 <http://www.data.gov>

[그림 1]은 2011년 9월을 기준으로 링크드 데이터의 데이터 셋의 크기와 링크를 시각화 한 것이다. 링크드 데이터는 2007년 12개 데이터 셋으로 시작해서 매년 약 200%의 지속적인 양적, 질적 성장을 통해서 2011년 295개 데이터 셋, 316억 트리플 규모로 커졌으며, 매년 새로운 데이터 셋이 추가되면서 지속적인 성장을 계속하고 있다. 이러한 링크드 데이터는 시맨틱 데이터의 주된 배포 경로로서 온톨로지 연계 및 활용에 중요한 역할을 수행하고 있으며, 링크드 데이터에 출판되는 정보의 도메인이 확장되고 규모가 커짐에 따라 온톨로지 내의 개체 간 관계도 점차 복잡해지고 있다. 또한, 데이터가 점차 증가하면서 데이터에 대한 일관성, 품질의 신뢰성, 관리 및 활용을 위한 도구 개발에 필요성이 아울러 높아지고 있다.

표 1. 데이터 셋 참조 링크 통계

피참조 데이터 셋	참조 데이터 셋(n)	참조 Triple(Min.)
Bio2RDF	125	5,119,5500
Dbpedia	32	1,330,325
RKBexplorer	108	1,239,300
DBLP	5	1,003,100
Etc.	-	779,490
Total	416	6,256,3125

그림 2. 링크드 데이터 Cloud 의 데이터 셋 간 링크 연계 관계 시각화

데이터의 연계 및 활용 관점에서 링크드 데이터의 가장 큰 문제는 특정 데이터 셋으로 링크가 편중되는 것과 전체 개체 수 대비 상대적 상호 참조 링크가 부족함

현상이다. [그림 2]는 w3c.org에서 공개하고 있는 데이터 셋 간의 링크 정보[3]를 이용하여 데이터 셋 간 링크 연결 관계를 시각화한 것으로 시각화 도구인 Visione²를 이용하여 작성하였다. 데이터 셋 간 링크는 3개의 큰 군집(그림 2의 원)을 이루고 있는 것을 확인할 수 있으며, Dbpedia, RKBexplorer, Bio2RDF가 허브 역할을 하고 있는 것으로 나타난다. [표 1]은 외부 데이터 셋에서 참조되는 링크의 정량적인 정보를 정리한 것으로 참조 데이터 셋(n)은 피참조 데이터 셋을 링크로 가지고 있는 데이터 셋의 개수를 의미하며, 참조 Triple은 참조 링크의 최소 개수의 합이다. 앞서 언급한 3개의 데이터 셋이 가장 많은 참조 링크를 가지고 있으며, DBLP의 경우에는 참조하는 링크의 수는 많지만 데이터 셋의 수가 상대적으로 적다. 참조 링크의 수의 비율을 보면 상위 4개의 허브의 참조 링크의 개수는 전체 참조 링크의 88%를 차지하고 있는 것을 알 수 있다.

링크가 일부 데이터 셋으로 편중되게 되면 허브 데이터 셋의 오류가 다수의 데이터 셋으로 전파되어 정보의 신뢰성을 떨어뜨리는 문제가 발생하게 된다. 이러한 링크의 편중 현상은 다수의 온톨로지 데이터를 대상으로 정확한 개체 식별을 수행하는 것이 어렵고 많은 시간이 소요되어 주로 내부 데이터와 같은 도메인의 허브 데이터 셋만을 대상으로 개체 식별이 이루어지기 때문에 나타나는 현상이다. 한 가지 예를 들어보면, 링크드 데이터의 상호 참조 관계를 이용해 개체 식별을 하는 sameAs.org에서 “Alok N. Choudhary”로 동일 URI를 검색을 할 경우 RKBexplorer를 중심으로 1101 개의 URI가 동일 URI 그룹으로 제시된다. 이는 1101 개의 링크가 상호 참조 관계인 owl:sameAs, rdfs:seeAlso 등으로 연결되어 같은 개체를 지칭하는 URI로 간주된다는 의미이며 일반적으로 추론을 통해 같은 개체의 정보가 통합할 수 있다. 1101 개의 URI 정보를 분석한 결과, 1100 개의 URI 는 family name 속성으로 ‘Choudhary’라는 값을 가지고 있는데 반해 그중 1 개의 URI는 ‘Choudhary’ 외에 ‘Kandemir’란 잘못된 정보를 포함하고 있었다. 이런 경우에 1개의 오류 정보와 상호 참조 관계를 갖고 있는 1100개의 URI의 정보를 사용하는 다

² <http://visione.info>

수의 시스템은 family name 속성의 값으로 Choudhary와 Kandemir 두 가지 값을 모두 갖는 것으로 간주하게 되고, Kandemir의 속성 정보가 Choudhary의 속성 정보와 혼재되어 정보의 정확성을 떨어뜨리는 현상을 야기할 수 있는 것이다.

링크가 특정 데이터 셋으로 집중되는 것 이외에 다른 문제는 전체 링크 개수 대비 링크의 수가 현저히 부족한 것이다. 개체 식별의 어려움으로 인해 개체 식별은 다수의 데이터 셋이 아닌 소수의 데이터 셋만을 대상으로 이뤄지기 때문에 링크드 데이터의 전체 데이터 310억 Triple 중 링크는 0.016%인 5억 Triple에 불과하다[3]. 이러한 문제는 자동화된 링크드 데이터의 출판(Publishing) 방식에서 비롯된다. D2R Server[4]는 기존의 비시맨틱 데이터를 시맨틱 데이터로 변환 및 출판을 하기 위한 도구로 링크드 데이터의 정보가 급격하게 증가하는 데 큰 기여를 하였다. 이러한 도구들은 기존에 RDB와 같은 형태로 저장된 정보를 매핑 쿼리를 이용하여 시맨틱 데이터로 변환하는 기능과 함께 변환된 데이터에 접근하기 위한 SPARQL Endpoint 및 HTTP Negotiation 방식으로 RDF나 HTML 형식의 정보를 제공하는 기능을 가지고 있어 널리 활용되었지만 D2R Server는 개체 식별 기능을 포함하고 있지 않아 별도의 개체 식별이 요구되기 때문에 이러한 도구를 사용하여 생성된 시맨틱 정보는 대체적으로 개체 식별 과정이 소홀하게 처리되어 외부 온톨로지와의 연계가 미흡하다.

이러한 링크드 데이터의 문제 해결과 활성화를 위해서 2010년 시작된 LOD2 프로젝트에서도 개체 간 상호 참조 링크의 확대 및 관리와 오류가 없는 데이터를 확보하는 것을 주요 목표 중 하나로 삼고 프로젝트를 진행하고 있다. 하지만 아직까지 문제를 해결하기 위한 구체적인 방안을 제시하고 있지는 못하고 있다. 그렇기 때문에 링크드 데이터의 링크를 풍부하게 하고 개체 식별 정확성을 보장하여 데이터 출판 및 관리에 활용될 수 있는 다중 온톨로지 기반 개체 식별 방법의 필요성이 더욱 부각되고 있다.

2. 개체 식별 서비스

2.1 sameAs.org

영국 Southampton 대학에서 EU의 ReSIST 프로젝트의 일환으로 구축한 개체 식별 서비스인 sameAs.org는 서로 다른 온톨로지에서 키워드나 URI를 통한 동일 지시(Co-Reference) URI 검색을 지원한다[5].

sameAs.org는 RKBExplorer.com의 66개의 데이터 셋에 포함된 URI의 상호 참조 관계(owl:sameAs, owl:differentFrom, rdfs:seeAlso 등)를 이용하는 Consistent Reference Service(CRS)의 기반이 되는 개체 식별 서비스로 동일 지시 URI를 번들(Bundle)로 묶어 신규 URI를 부여한다. 이렇게 생성된 정보는 RKBExplorer.com에서 정보 통합 뷰[그림 3]를 제공하기 위해서도 사용된다. sameAs.org의 개체 식별은 기존의 URI 정보가 담고 있는 상호 참조 관계를 이용하기 때문에 참조 정합성이나 데이터 정합성 오류에 취약한 단점을 가지고 있다.

그림 3. sameAs.org 와 RKBExplorer.com

2.2 OKKAM

OKKAM은 EU 프로젝트로 분산된 다중 온톨로지 환경의 개체 식별 프레임워크이며, 개체 식별뿐만 아니라 URI를 부여하고 생성된 URI를 통하여 개체를 수집·관리하는 통합 프레임워크이다[6].

OKKAM은 개체 식별을 위해서 Jolly, The Feature Based Entity Match Algorithm(FBEM), Eureka 등 3가지 개체 식별 엔진을 모두 이용한다[7]. 각 개체 식별 엔진의 차이점을 살펴보면 Jolly 개체 식별 엔진은 String Metrics를 기반으로 개체명과 속성 유사도, 그리고 속성 가중치를 이용한다. FBEM 엔진 역시 속성과 값을 비교하지만, FBEM 엔진은 온톨로지 특성을 고려

하여 구현되어 시소러스나 번역어 사전을 통한 다국어 처리에 유리한 특성을 가지고 있다. 마지막으로 Eureka는 속성과 값의 분포를 색인하여 유사한 속성값을 가지는 후보군을 점진적으로 생성해가면서는 식별하는 방식을 사용한다. OKKAM은 검색 서비스에 사용자가 직접 엔진을 선택하고 결과를 비교해 볼 수 있는 UI를 통해 3가지 식별 엔진의 식별 결과를 보여주지만, 그 결과가 각각 달라서 다소 사용자에게 혼돈을 줄 수 있는 여지가 있다. 실제로 “seoul”에 대한 3가지 식별 엔진의 결과로 FBEM 엔진은 위치 정보를 담고 있는 소수의 URI만을 제시하며 Jolly 엔진은 Seoul의 위치 정보를 다수 그룹으로 식별하고 있으며 Seoul 이외의 다른 도시 정보가 혼재된 상태로 결과를 제시하고 있으며 Eureka 엔진은 단 한 개의 URI도 제시하지 않는다.

2.3 Identifiers.org

Identifiers.org는 European Bioinformatics Institute (EBI)에서 제공하는 개체 식별 서비스로 생물 정보학 (Bioinformatics) 분야 정보의 개체 식별 서비스이다[8].

그림 4. Identifiers.org의 Publication Pipeline

표 2. 기존 식별 서비스의 식별 방식의 특징과 문제점

서비스명	sameAs.org	OKKAM			identifier.org
식별 방법	CRS	FBEM	Eureka	Jolly	Curation & Annotation
식별 방식의 특징	데이터 상호 참조 관계 사용	시소러스 및 번역사전 사용	모든 데이터의 속성 분포 색인	모든 속성-값의 유사도 비교	전문가 검증
문제점	데이터의 오류 확산 가능성, 낮은 신뢰도	단어 변환을 통한 온톨로지의 잘못된 해석의 위험성	온톨로지 간 일괄 비교 방식 모든 정보를 분석해야함	속성 개수의 편차가 크고, 모든 속성을 비교해야함	고비용, 제한된 도메인, 제한적인 처리 속도

[그림 4]는 Identifiers.org의 정보가 시스템에 등록되는 과정을 보여주고 있는데 데이터가 시스템에 입력되기 전에 정합성 검사(Consistency Check)를 거쳐 Annotation models을 생성한다. 이러한 Curation과 Annotation과정은 모두 전문가(Curator)에 의해 이루어지기 때문에 데이터 오류가 적고 이를 통해 생성된 계산 모델(Computational Models)은 상호 참조, 용어 검색, 모델 군집화, 유전자 온톨로지 탐색, 형식 변환이 가능하다. 하지만 이러한 개체 식별 방법은 큰 비용과 시간이 요구되는 수작업에 의존하고 있을뿐더러 생물 정보학 분야의 정보만을 처리한다는 단점을 가지고 있다.

[표 2]는 기존 식별 방법의 특징 및 문제점을 정리한 것이다. sameAs.org는 기존 데이터 간 상호 참조 관계를 그대로 이용하는 방식을 사용하여 오류에 대한 필터링이 불가능하고 정확성이 낮으며, Identifiers.org에서 사용하는 전문가의 검증 방식은 정확성은 높지만 큰 비용과 시간이 필요할 뿐 아니라 식별 도메인이 한정되어 있다. OKKAM의 시소러스나 번역서 사전을 이용하는 FBEM은 온톨로지 속성을 단어 변환을 통해 이용해 동일 속성으로 간주하는 방식은 온톨로지의 잘못된 해석이 이루어질 가능성이 있으며 외부의 데이터를 이용하여 개체 식별 정확성을 높이는 것은 매우 어려운 방식이다. 또한, 모든 데이터 속성의 분포를 색인하는 Eureka는 데이터 셋 단위의 개체 식별에 초점을 두고 있기 때문에 다수 온톨로지의 대용량 정보의 식별에 적합하지 않다. Jolly는 속성-값의 유사도를 사용하는 방법은 URI와 URI를 1:1로 비교하게 되며, 이때 다수 속성을 가지고 있는 URI를 처리하기 위해서는 많은 연산이 필요할 뿐만 아니라 다수의 온톨로지의 URI 속성 개수의 편차가 크기 때문에 효율성을 보장하기 어렵다.

III. 속성 기반 자질 식별

제안하는 방법은 OKKAM의 Jolly와 같이 속성-값의 유사도를 사용하지만, Jolly는 정보의 모든 속성 정보를 비교하는 데 비해 개체의 유형 특성에 따라 식별 기준(Criteria)이 되는 대표 속성의 값만을 이용한다. 또한, URI 정합성을 보장하기 위해 개체 유형에 따른 논리적 특성을 이용하여 오류가 포함된 것으로 예상되는 URI는 식별에서 제외하여 잘못된 URI의 정보가 확산되는 것을 방지한다. 제안 방법은 데이터 셋과 데이터 셋 간의 개체 식별 방식이 아니라 개체와 개체 간의 식별을 진행하는 자동화된 점진적 개체 식별 방식으로 지속적으로 업데이트 및 추가되는 링크드 데이터의 개체 식별에 적절한 방법이다.

1. 개체 식별 프로세스

URI 관리 서비스는 개체 식별 및 URI 그룹별 정보 관리를 위해서 7개의 주요 프로세스를 거치며, 외부 서비스인 sameAs.org와 Sindice.com의 검색 결과를 활용한다. Linked Data에 연계된 온톨로지를 활용하기 위해서 SPARQL endpoint 서비스가 사용되며, 식별된 URI 그룹 및 편집 정보를 저장하고 추천하기 위해서 트리플 저장소를 이용한다. 개체 식별 서비스는 그림 5와 같이 구성되어 있으며 이해를 돕기 위해 [그림 5-그림 9]를 이용하여 수행하는 주요 기능을 설명한다.

그림 5. 개체 식별 프로세스

① URI, 개체명 검색기: URI, 개체명 검색기는 사용자의 검색어를 입력받아 URI 개체 식별의 필요성 여부를 판단하는 모듈로 기존의 검색 결과를 캐싱(caching)한다. 사용자가 입력한 검색어가 URI인지 개체명인지를 감지하여 외부 서비스(시멘틱 데이터 검색엔진: Sindice.com, sameAs.org)의 API 중 1개를 통해 URI 목록을 획득한다. 2개의 API를 모두 지원하여 외부의 API의 서비스 중단에 대비한다. 획득된 URI 목록은 기존의 식별 결과 목록의 해쉬 코드와 비교하여 기존의 식별 결과가 없거나 URI 목록의 변화가 있는 경우에만 URI 식별 모듈을 호출하고 그렇지 않은 경우에는 기존의 식별 결과를 캐쉬 서버에서 가져와 사용자에게 제시한다. 그림 6의 예제는 URI 목록을 얻기 위해 외부 서비스를 이용하는 과정을 보여주고 있다. sameAs.org나 Sindice.com의 Open API를 통해 “Alok N, Choudhary”를 질의하여 Json 형식의 식별 결과를 얻고 이를 파싱하여 식별 결과를 제외하고 단순 URI 목록만을 추출하게 된다. 이때 추출된 목록은 {URI 1, 2, 3, 4, 5, 6, 7}이다. URI 목록이 캐쉬 서버에 존재하지 않는다면 5개의 URI가 다음 단계로 전달되고, 존재한다면 기존의 식별 결과를 그대로 사용자에게 제시하게 된다.

그림 6. 외부 서비스를 통한 URI 목록 획득

② URI 속성 정보 수집: 수집된 URI 목록의 속성 정보를 Linked Data를 통해서 수집하는 모듈로 미리 정의된 Linked Data의 각 데이터 셋의 접속 정보를 이용하여 SPARQL Endpoint와 HTTP Negotiation을 통해

외부 온톨로지의 정보를 실시간으로 수집하고 내부의 시맨틱 저장소인 OntoReasoner에 Triple 형태로 저장한다. 수집과정에서 각종 HTTP 오류(Internal Server, File not found 등)가 있는 URI를 필터링하고, HTTP 오류는 없으나 기타 외부 온톨로지의 이유로 인해 사용되지 않는 URI(Deprecated or Redirect URI), Micro-format(Non-Rdf URI; 단순한 메타 정보만을 담고 있는 URI)와 비 시맨틱 데이터(HTML 등)를 구분하여 유효한 URI 만을 식별에 사용한다. [그림 7]은 URI 속성 정보를 설명하는 예제로 상위 단계에서 전달 받은 URI 목록(좌측의 URI 1~7)의 속성 정보를 링크드 데이터에서 수집하며, 이 과정에서 HTTP 오류와 사용되지 않는 URI로 분류된 URI 6과 7이 필터링 되어 식별 과정에서 제외된다. 유효한 URI인 URI 1~5의 유형, 이름 속성을 포함한 모든 정보가 수집되며 연결 관계를 가지고 있는 URI 목록인 8~10이 포함된다.

그림 7. URI 속성 정보 수집

③ 개체 유형별 그룹화: 시맨틱 저장소에 수집된 Triple 정보에서 SPARQL 질의를 통해 유형 속성을 추출하고 개체를 유형별로 그룹화한다. 그룹화된 URI 목록은 그룹별로 개체 식별이 수행되어 개체 식별은 유형의 수만큼 진행된다.

링크드 데이터의 Triple 정보는 데이터 셋마다 각기 다른 이형 서술어를 사용하여 개체와 개체의 속성을 표현하고 있기 때문에 동일 의미를 표현하는 이형 서술어를 처리할 수 있는 방안이 필요하다. 이형 서술어를 처리하기 위해서는 동일 유형의 정보를 수집하여 사용되

고 있는 온톨로지를 분석하는 절차가 선행되어야 한다.

표 3. 다양한 온톨로지에서 사용되는 이형 서술어

의미	이형 서술어의 예
이름, 명칭	foaf:label, rdfs:label, aktors:full-name
지리정보 위도,경도	dbpprop:long, geo:lat, dbpedia-owl:location
Wikipedia 링크	geonames:wikipediaArticle, dbpedia-owl:wikiPageRedirects

[표 3]은 다양한 온톨로지에서 사용되는 이형 서술어의 예이다. 다수의 이형 서술어를 처리하기 위해 매핑률을 적용하여 정의된 다수의 서술어를 하나의 대체 서술어로 처리한다. 대체 서술어를 이용하면 자동화된 SPARQL 질의 생성이 용이해져 개체의 속성 정보를 비교적 쉽고 유연하게 검색하고 개체 식별에 이용할 수 있다. 예를 들어, <http://sampleURI> 개체라는 이름 속성의 값을 구하기 위해서는 URI의 데이터 셋에서 사용하는 온톨로지를 분석하고 그에 맞는 서술어를 적용하여 사용되는 서술어가 foaf:label 일 경우에는 “SELECT ?name WHERE { <<http://sampleURI>> foaf:label ?name.}”와 같은 질의를 만들고 rdfs:label을 사용할 경우에는 “SELECT ?name WHERE { <<http://sampleURI>> rdfs:label ?name.}”와 같은 SPARQL 질의를 생성하고 수행해야한다. 하지만 대체 서술어를 이용하면 “SELECT ?name WHERE { <<http://sampleURI>> <Const.NAME> ?name.}”와 같은 대체 서술어를 통한 질의로 온톨로지 정보를 가져오고 그에 맞는 질의를 생성하는 과정을 거치지 않고 속성의 값을 가져올 수 있다. <Const.NAME>은 대체 서술어이다.

[그림 7]에서 URI 1~4와 URI 8~10은 ‘저작’이라는 동일 의미를 표현하는 다수의 서술어(Creator, Maker, Has-author)로 연결되어 있지만 대체 서술어를 통한 한 번의 질의를 통해 개체 식별을 위한 속성값 처리가 가능하다.

이 단계에서는 URI 1~5가 모두 Person 유형이므로 하나의 그룹으로 개체 식별이 수행된다.

④ 식별 대상 속성 정보 로딩: 시스템에 미리 정의된 유형별 식별 자질 및 규칙을 로딩하는 과정으로 대표 속성과 도메인 정보, 논리 정합성 규칙이 포함된다.

식별 자질 정보는 저자 유형 식별에 저자의 이름과 저작물 정보, 저작물 유형의 식별에 저작물 명과 저작자가 연관 속성으로 매핑되어 있으며, 도메인 정보는 복수의 값을 가지고 있는 속성의 경우(예: 이름)에 복수의 속성 간 유사도 및 일치도에 관한 규칙을 담고 있다, 논리 정확성은 연관 속성-값 비교에서 개체 식별 결과 검증에 의해 사용되는 규칙이다.

⑤ 개체명 기준 그룹화 : 개체는 다수의 개체명을 포함할 수 있기 때문에 정규화를 거쳐 사람이나 회사의 이름, 논문명과 같은 단문 형태의 문자열 비교에 적합한 String Metrics 방법 중 Jaro Winkler Distance[9]를 이용한 유사도 계산법을 사용하여 개체명 간 도메인 오류를 검사하고 객체 당 1개의 대표명을 선정한다. 이 과정에서 개체의 식별 속성에 따라 다른 임계값(상위 단계에서 로딩된 도메인 정보)을 사용한다. 예를 들어, 정규화를 거친 문서명은 저자명보다 문자열 길이가 길고 변화가 적기 때문에 저자명의 문자열 유사도의 임계값은 문서명의 임계값보다 낮은 값을 가진다.

그림 8. 개체명 기준 그룹화

[그림 8]의 URI 3과 5는 모두 복수의 이름 속성을 갖고 있어 도메인 오류 검사를 거치며 먼저 복수의 값을 가질 수 있는 속성인지를 확인한 후 문자열의 유사도를 검사한다. URI 5는 같은 속성의 문자열 유사도가 낮아 오류 URI로 분류되어 개체 식별 과정에서 제외된다. URI 3의 경우에는 동일 속성의 문자열 유사도가 임계값보다 높기 때문에 복수 속성 중 가장 긴 문자열을 대표 명으로 선정한다. URI 3의 이름 속성 대표 명은

‘Alok N. Choudhary’ 이다. URI 당 1개의 이름 속성이 정해지면 이름 속성의 유사도를 이용해 1차 그룹화를 수행한다. 1차 그룹화를 통해 {(URI 1), (URI 2, URI 3), (URI 4)}의 군집이 형성된다.

⑥ 연관 속성-값 비교: 연관 속성-값 비교단계에서는 URI의 연관 속성 정보-값을 수집하여 정확성을 판단한다. 예를 들어, 저자 유형의 개체는 논문이 연관 속성이므로 모든 저자 URI의 연관 속성 중 같은 논문을 속성으로 갖는 URI가 있는지 비교한다. 이 과정을 통해 이전 단계에서 다른 그룹으로 분류되었지만 같은 연관 속성을 갖는 그룹이나 같은 그룹이지만 연관 속성이 없는 URI를 선별한다.

그림 9. 연관 속성-값 비교

[그림 9]의 URI 1 ~URI 4의 연관 속성인 문서 유형의 정보를 이용해 상위 단계의 1차 그룹화 결과를 검증하게 된다. URI 8~10의 속성 정보를 링크드 데이터에서 수집한 후 식별 대상과의 관계를 분석한다. 동일 그룹인 URI 2와 3은 URI 9의 저작 관계로 연결되어 있으므로 동일 그룹으로 확정할 수 있으며 URI 1은 URI 8을 통해 URI 3과 연결되어 하나의 그룹으로 병합된다.

⑦ 개체 식별 및 결과 검증: URI 그룹 확정 및 식별 결과 검증은 이전 단계의 식별 결과를 재검증하는 역할을 한다. 개체명 기준 그룹화로 생성된 1차 그룹 URI 1, 2, 3이 같은 연관 속성-값 URI 7을 갖고 있기 때문에 역관계 속성(URI 7의 연관 속성-값)을 수집해 URI 1, 2, 3이 모두 속성으로 포함되어 있는지를 확인하는 과정을 거친다. 이 과정에서 URI 4는 URI 10과 연결되어 하나의 그룹의 후보가 되지만 URI 9와도 연결 관계가

있기 때문에 병합된 URI 1, 2, 3의 그룹과 충돌을 일으킨다. 이러한 문제를 해결하기 위해 논리 정합성 규칙이 사용되며 URI 8이 문서 유형의 정보이기 때문에 복수의 저자 속성이 허용되며, URI 4와 URI 1, 2, 3그룹은 공저자 관계로 판정되어 병합이 이루어지지 않는다.

이 과정에서 역관계 속성이 확인되지 않은 URI는 오류 URI로 분류되어 개체 식별 대상에서 제외된다. 그룹별 대표 개체명·URI 선정에서는 그룹화 되지 않은 URI와 데이터 정합성의 오류가 있다고 판단된 URI는 수집된 정보만으로 더 이상 식별이 어렵기 때문에 별도의 그룹으로 관리되어 추후 관리시스템에서 관리자가 처리할 수 있도록 관련 URI 정보를 트리플 스토어에 저장한다.

URI 정보는 가독성 낮아서 육안으로 식별하기 쉽지 않다. ①~⑦의 과정을 거쳐 식별된 정보는 사용자에게 편의성과 가독성을 높이고 식별 결과 활용의 효율성을 높이기 위해 식별이 완료된 그룹은 통계적 방법을 통하여 다수의 유효 속성 정보를 갖는 URI를 그룹의 대표 URI로 선정한다. 그룹의 대표 URI는 다수 그룹의 정보를 축약하는 일종의 메타 정보의 역할을 하며 식별된 결과에서 유효한 정보를 한눈에 확인할 수 있도록 요약 그룹을 생성하여 사용자에게 제시된다.

IV. 실험 및 평가

1. 평가 데이터

제안한 방법의 정확도를 측정하고 성능을 향상하기 위해서 검증된 정답셋을 이용한 실험이 최선이지만 아직 링크드 데이터 기반의 개체 식별 연구에 활용할 수 있는 표준 정답셋이 없는 실정이다. 기존 실험에서 링크드 데이터에 출판되어 있는 컴퓨터과학 분야의 서지 정보 사이트인 DBLP의 논문 수 상위 20명의 저자를 대상으로 개체 식별 성능 평가를 실험을 수행하였으나 사용된 정답셋의 정확성을 보장하기에 어려움이 있었다 [10]. 이를 극복하기 위하여 실세계 저자의 확인을 얻을 수 있는 정답셋을 다음과 같은 방법으로 구축하여 실험에 활용하였다.

시맨틱 검색 엔진을 통해 정부출연연구소 A의 B 본부 전체 연구자 134명의 영문 이름을 검색하여 최소 10개 이상의 URI가 검색되는 29 명을 식별 대상을 선정하였다. 선정된 29명의 각 URI 목록은 본인의 확인을 통해 동일 개체 여부를 직접 확인을 하여, 평가의 정답 셋으로 활용하였다. 평가 데이터의 저자 정보를 식별하기 위해서는 링크드 데이터에서 712개의 다양한 종류의 속성 값을 갖고 있는 247,937 개의 Triple을 수집해야 하고 이를 해석하기 위해 20개 이상의 온톨로지 분석이 필요하다. 평가 데이터는 488개의 URI로 구성되어 있으며 저자 당 평균 12.83개의 URI, 그룹 당 평균 2.44개의 URI를 가진다.

2. 평가 방법

개체 식별 결과의 평가는 군집화(Clustering) 평가와 매우 유사하다. 기존에는 Cluster F_1 , pairwise F_1 과 같은 다양한 평가 방법이 사용되어 왔으나 McCallum (2000)에 의하면 개체 식별은 일반적인 군집화의 일종이지만 비교적 군집의 크기가 작고, 개체 간의 차이가 명확하다는 차이점을 가지고 있기 때문에 기존의 평가 방법을 그대로 적용하기에 적절하지 않다[11]. David(2010)는 대표적인 개체 식별 평가 방법을 대상으로 개체 식별에서 발생할 수 있는 3가지 오류 유형(분할 오류(Broken entity error), 병합 오류(Glued entity error), 그룹 오류(Misplaced entity error))에 대한 평가 방법의 정확성을 비교하여 많은 개체 식별 연구의 평가에 사용되고 있는[12][13] K measure[14]가 가장 적절한 평가 방법임을 밝혔다[15].

K measure는 ACP(Average Cluster Purity)와 AAP(Average Author Purity), 그리고 K (AAP와 ACP의 기하 평균)으로 정의되며 모든 군집 쌍의 평균 유사도를 의미하며 ACP, AAP, K 값은 정보 검색에서의 정확률, 재현률, F measure와 대응되는 개념이라고 할 수 있다.

ACP는 제안 방법을 통해 생성된 군집의 평균 순도(Purity)를 나타내며 최소값은 0이고 평가셋과 완전히 일치할 경우 1의 값을 가진다. ACP는 수식(1)과 같이 나타낼 수 있다.

$$ACP = \frac{1}{N} \sum_{i=0}^q \sum_{j=0}^R \frac{n_{ij}^2}{n_i} \quad (1)$$

R : 정답 셋의 군집 개수

N : 전체 URI 의 개수

q : 식별된 군집 개수

n_{ij} : 대응되는 정답 셋과 식별된 군집의 URI의 일치 개수

n_i : 식별된 군집 i 의 URI 개수

AAP는 식별된 군집이 정답셋 기준 군집의 분리된 정도를 나타내며 최소 0에서 최대 1의 값을 가진다. 높은 AAP는 정답 셋 군집에 비해 분리된 군집이 적다는 것을 의미한다. AAP는 수식(2)와 같이 나타낼 수 있다.

$$AAP = \frac{1}{N} \sum_{i=0}^q \sum_{j=0}^R \frac{n_{ij}^2}{n_j} \quad (2)$$

n_j : 정답 셋 j 의 URI 개수

K measure는 AAP와 ACP의 기하 평균이며 $K = \sqrt{AAP \times ACP}$ 로 정의된다. K 값은 두 값이 모두 1일 경우 가장 이상적인 값을 가진다.

그림 10. 개체 식별 시스템

[그림 10]은 제안하는 개체 식별 방법을 구현한 웹 서비스의 식별 결과 화면이다. 웹 인터페이스를 지원하

며 사용자가 검색어를 입력하면 식별된 그룹의 정보를 바로 확인할 수 있도록 링크를 제공한다. 또한, 하나로 식별된 그룹의 모든 속성과 값을 하나의 화면으로 확인할 수 있는 통합 속성 뷰 기능도 지원한다.

평가의 객관성을 높이기 위해서 기존의 개체 식별 서비스를 대상으로 동일한 입력 조건에서 개체 식별 결과를 비교하였다. 관련 연구에서 언급한 개체 식별 서비스 중 identifiers.org는 식별 도메인이 한정되어 있고 수작업을 중심으로 이루어지기 때문에 비교 평가가 어렵고, OKKAM 역시 동일 입력을 통한 개체 식별 결과의 비교가 어렵기 때문에 본 실험에서는 동일한 입력에 대한 sameAs.org의 개체 식별 결과를 제안 식별 방법의 결과와 비교하였다. sameAs.org는 식별 결과를 수집할 수 있는 API를 제공하고 있어 API를 통해 수집된 식별 결과에서 URI 목록만을 제안 식별의 식별 대상 URI 목록으로 사용하면 동일한 식별 대상 목록을 이용하여 두 가지 방법의 식별 결과의 비교가 가능하기 때문에 III장에서 기술한 식별 프로세스의 “URI, 개체명 검색기”에 외부 시맨틱 검색엔진을 sameAs.org의 API로 설정하여 결과를 비교하였다.

3. 평가 결과

[표 4]는 평가 데이터를 이용한 개체 식별 결과로 제안 방법과 sameAs.org의 결과를 나타내고 있다. ID는 개체 식별에 사용된 검색어의 ID이며, N_c 는 생성된 군집의 개수를 의미한다.

29명의 저자명에 대하여 sameAs.org 서비스는 평균 0.35의 ACP, 0.43의 AAP, 그리고 0.38의 K 값을 나타냈다. 관련 연구에서 언급한 것과 같이 sameAs.org는 URI의 상호 참조 관계만을 이용하기 때문에 낮은 성능을 보이고 있는 것으로 해석할 수 있다. 본 연구에서 제안된 방법은 평균 0.85의 ACP, 0.85의 AAP, 0.85의 K 값으로 sameAs.org 보다 2배 이상 정확한 성능을 보이고 있으며 sameAs.org가 제안 방법보다 URI 군집을 평균적으로 더 많이 생성하고 있다.

표 4. 개체 식별 평가 결과

ID	sameAs.org				제안 방법			
	N_c	ACP	AAP	K	N_c	ACP	AAP	K
1	5	0.52	0.70	0.60	2	1.00	1.00	1.00
2	10	0.36	0.42	0.39	9	0.86	0.92	0.89
3	10	0.12	0.58	0.26	7	1.00	1.00	1.00
4	8	0.37	0.47	0.42	5	0.55	0.74	0.64
5	8	0.21	0.33	0.26	3	0.69	0.86	0.77
6	8	0.44	0.46	0.45	9	1.00	1.00	1.00
7	10	0.33	0.50	0.40	4	1.00	1.00	1.00
8	9	0.59	0.49	0.54	11	0.88	1.00	0.94
9	9	0.62	0.47	0.54	11	1.00	1.00	1.00
10	10	0.45	0.43	0.44	6	0.81	0.74	0.77
11	9	0.21	0.29	0.25	6	0.89	0.90	0.89
12	10	0.56	0.59	0.57	4	0.53	0.50	0.51
13	5	0.24	0.25	0.24	5	0.90	0.90	0.90
14	3	0.24	0.39	0.31	4	0.7	0.8	0.75
15	11	0.12	0.32	0.19	4	0.94	1.00	0.97
16	7	0.37	0.48	0.42	8	1.00	1.00	1.00
17	9	0.36	0.31	0.33	6	1.00	1.00	1.00
18	10	0.28	0.37	0.32	7	0.82	0.89	0.86
19	10	0.56	0.55	0.55	8	1.00	1.00	1.00
20	5	0.32	0.57	0.43	3	0.71	0.70	0.71
21	8	0.19	0.29	0.23	6	0.66	0.83	0.74
22	9	0.48	0.54	0.51	5	0.92	0.86	0.89
23	10	0.26	0.25	0.26	4	1.00	1.00	1.00
24	10	0.38	0.46	0.42	6	1.00	1.00	1.00
25	10	0.26	0.33	0.30	5	0.58	0.63	0.60
26	10	0.15	0.23	0.18	4	0.65	0.30	0.44
27	10	0.73	0.80	0.77	7	0.9	0.85	0.87
28	10	0.16	0.30	0.22	3	1.00	1.00	1.00
29	9	0.27	0.44	0.34	4	0.62	0.33	0.45
Avg	8.7	0.35	0.43	0.38	5.7	0.85	0.85	0.85

4. 결과 분석

sameAs.org는 개체를 식별하기 위해 새로운 군집을 생성하거나 분리하지 않고 URI에 정의된 관계를 하나로 통합하는 역할만 하는 데 비해 제안 방법은 개체의 속성 정보를 이용해 URI를 필터링하고 대응 속성 정보를 기반으로 개체를 식별하기 때문에 두 개의 방법의 식별 결과는 많은 차이를 보이며, 제안 방법의 평균 군집이 sameAs.org에 비해 적은 이유는 링크드 데이터의 개체 간 상호 참조 관계는 관련 연구에서 언급한 것처럼 특정 허브 데이터 셋을 중심으로 형성되어 동일 개체이지만 같은 개체로 인식되지 않았기 때문이다.

제안된 방법의 식별 결과를 분석 결과의 일부 개체 식별에서 낮은 성능을 보이고 있는 경우가 있어 이러한 오류 개체 식별 유형을 분석하였다. 분석된 유형은 유

사도 임계값 문제, 한국 저자 이름 처리 문제, 검증 규칙 문제의 3가지이다.

- 유사도 임계값 문제: 개체명의 유사도를 측정할 때 사용된 임계값 선정 문제로 동일 개체를 다른 개체로 식별하는 문제이다. 이 문제는 제안방법에 사용된 임계값 설정이 소수 데이터를 이용하여 휴리스틱(Heuristic)한 방법으로 설정되었기 때문에 발생한 것으로 이를 해결하기 위해서는 다수의 개체명 데이터 실험을 통한 적정 임계값 분석 및 적용이 필요하다.
- 한국 저자 이름 처리 문제: 주로 외자 이름에서 나타나는 현상으로 한국 이름의 외자(monosyllabic)는 Jaro Winkler Distance만으로 동일 여부를 판단하기에는 부족한 것으로 판단된다. 사용된 String metrics가 영어권 이름을 대상으로 연구된 것이기 때문에 한국 혹은 동양 이름을 처리할 수 있는 다른 방법이나 제한 규칙의 적용이 불가피한 것으로 분석되었고 좀 더 상세한 이름 처리 규칙의 정의가 요구된다.
- 검증 규칙 문제: 검증 규칙은 다수의 온톨로지를 모두 분석하지 못해 검증에 필요한 속성-값을 획득하지 못하여 생긴 문제이다. 이 문제를 해결하기 위해서는 링크드 데이터에 사용된 다양한 온톨로지를 분석하고 처리하기 위해 자동 온톨로지 분석 기법이 필요하다. 검증 규칙 문제 역시 다수의 온톨로지 분석을 통한 검증 규칙 보완이 필요한 것으로 사료된다.

제안된 방법은 평균 0.85의 성능(K 값)을 보였으며 본 연구를 통해 밝혀진 문제를 개선하여 저자 개체 식별에 충분히 활용될 수 있을 것으로 기대된다.

V. 결론 및 향후 연구

링크드 데이터를 활용한 대용량 정보원 간의 정보 공유 및 연계는 시맨틱 정보의 활용과 더불어 정보의 가

치를 증대시킬 수 있는 유용한 방법으로 많은 주목을 받고 있지만, 개체 식별 기술의 부재로 인해 정보 연계의 어려움이 있다. 본 연구에서는 기존 개체 식별 서비스의 단점을 보완하고 다중 온톨로지의 개체를 점진적으로 식별할 수 있는 방식을 통하여 개체 식별 정합성을 보장하고 타 온톨로지 간 개체 식별 정합성을 검증할 수 있는 개체 식별 방법을 제안하였다.

제안된 개체 식별 방식은 실시간으로 링크드 데이터의 정보를 수집하는 방식을 사용하기 때문에 다수의 URI 목록 중 사용되지 않는 URI나 HTTP 오류로 인하여 정보 획득이 불가능한 URI를 필터링하여 사용성(Usability) 측면에서도 유용할 것으로 예상되며, 향후 실험을 통해 밝혀진 문제점을 개선하여 개체 식별 성능을 높이고 자동화된 방법으로 다수의 온톨로지 분석을 통해 개체 식별 유형을 확대할 예정이다. 본 연구에서 제안한 링크드 데이터 기반 개체 식별 방법은 링크드 데이터의 특정 데이터 셋에 링크가 편중되는 문제를 해결하고 개체 수 대비 상대적으로 부족한 링크의 수를 풍부하게 하기 위한 기반 연구에 기여 할 것으로 기대한다.

참 고 문 헌

- [1] 이정아, “스마트 정부의 공공정보 개방과 이용활성화 전략”, CIO report, 제28권, 2010.
- [2] Pyung Kim, S. W. Lee, and B. J. You, “A same As Management Method based on URI,” 한국콘텐츠학회, KISTI-KOCON ICCS 2009, 제7권, 제2호, 2009.
- [3] <http://www.w3.org/wiki/TaskForces/CommunityProjects/LinkingOpenData/DataSets/LinkStatistics>
- [4] C. Bizer and R. Cyganiak, “D2R Server - Publishing Relational Databases on the Semantic Web,” Proceedings of the 5th International Semantic Web Conference, 2006.
- [5] H. Glaser, A. Jaffri, and T. Millard, “Managing Co-reference on the Semantic Web,” Proceedings of WWW2009 Workshop: Linked Data on the Web, 2009.
- [6] P. Bouquet, H. Stoermer, and D. Giacomuzzi, “OKKAM: Enabling a Web of Entities, Identity, Identifiers, Identification,” Proceedings of WWW2007 Workshop on Entity-Centric Approaches to Information and Knowledge Management on the Web, 2007.
- [7] E. Ioannou, “Intelligent Entity Matching and Ranking,” OKKAM report, D3.1, 2010.
- [8] C. Laibe, “Identifiers.org and MIRIAM Registry: Perennial Identifiers for Cross-referencing Purposes,” Proceedings of Nature, 2011.
- [9] W. E. Winkler, “String Comparator Metrics and Enhanced Decision Rules in the Fellegi-Sunter Model of Record Linkage,” Survey Research Methods of American Statistical Association, 2009.
- [10] 김태홍, 김 평, 정한민, 성원경, “다중 온톨로지의 속성정보를 이용한 점진적 개체 식별”, 한국인터넷정보학회, 2011년도 하계학술발표대회, 2011.
- [11] A. McCallum, K. Nigam, and L. Ungar, “Efficient clustering of high-dimensional data sets with application to reference matching,” Proceedings of KDD, pp.169-178, 2000.
- [12] A. Laender, M. Goncalves, R. Cota, A. Ferreira, R. Santos, and A. Silva, “Keeping a digital library clean: new solutions to old problems,” ACM Symposium on Document Engineering, pp.257-262, 2008.
- [13] R. Cota, M. Goncalves, and A. Laender, “A heuristic-based hierarchical clustering method for author name disambiguation in digital libraries,” Proceedings of Brazilian Symposium on Databases, pp.20-34, 2007.
- [14] A. Solomonoff, A. Mielke, M. Schmidt, and H. Gish, “Clustering speakers by their voices,” IEEE International Conference on Acoustics, Speech, and Signal Processing, pp.757-760,

1998.

- [15] M. David, E. Steven, and G. Hector, "Evaluating entity resolution results," Proceedings of the VLDB Endowment, Vol.3, No.1-2, pp.208-219, 2010.

저 자 소 개

김 태 홍(Taehong Kim)

준회원

- 2010년 2월 : 한밭대학교 컴퓨터 공학과(공학사)
- 2011년 2월 ~ 현재 : 과학기술연합대학원대학교 응용정보과학과(석·박사통합과정)

<관심분야> : 개체 식별, 시맨틱 웹, 분산 컴퓨팅

정 한 민(Hanmin Jung)

정회원

- 1994년 2월 : POSTECH 전자계산학(공학석사)
- 1994년 2월 ~ 2000년 7월 : ETRI 선임연구원
- 2000년 9월 ~ 2004년 6월 : (주)다이렉스트 연구소장/CTO

- 2003년 8월 : POSTECH 컴퓨터공학(공학박사)
- 2004년 7월 ~ 현재 : 한국과학기술정보연구원 소프트웨어연구실 실장
- 2005년 3월 ~ 현재 : 과학기술연합대학원대학교 겸임교수
- 2009년 1월 ~ 현재 : 한국콘텐츠학회 이사

<관심분야> : 시맨틱 웹, HCI, 자연어처리

성 원 경(Won-Kyung Sung)

정회원

- 1987년 2월 : 연세대학교 불어불문학과(학사)
- 1989년 2월 : 연세대학교 불어불문학과(석사)
- 1996년 12월 : 프랑스 파리7대학교 언어학과(공학박사)

- 1997년 ~ 1998년 : 한국전자통신연구원 Post-doc
- 1998년 ~ 2001년 : L&H Korea(주) 책임연구원
- 2001년 ~ 2003년 : (주)보이스텍 연구개발본부장/상무이사
- 2004년 ~ 현재 : 한국과학기술정보연구원 정보·소프트웨어연구센터 센터장
- 2004년 ~ 현재 : 과학기술연합대학원대학교 겸임교수

<관심분야> : 자연어처리, 시맨틱 웹

김 평(Pyung Kim)

정회원

- 1997년 2월 : 충남대학교 전산학과 학사
- 1999년 2월 : 충남대학교 컴퓨터과학과 석사
- 2004년 2월 : 충남대학교 컴퓨터과학과 박사

- 2000년 ~ 2003년 : (주)엔퀘스트 테크놀로지 개발실장 / 기술이사
- 2003년 ~ 2004년 : 충남대학교 강사
- 2004년 ~ 현재 : 한국과학기술정보연구원 선임연구원
- 2008년 ~ 현재 : 한국정보교육학회 이사

<관심분야> : 시맨틱 웹, 정보검색, 자연어처리