

맥락적 어휘 지식 그래프 추출 알고리즘의 설계

남상하[○], 최규현, 함영균, 최기선

한국과학기술원

{nam.sangha, wiany11, hahmyg, kschoi}@kaist.ac.kr

Design of a Contextual Lexical Knowledge Graph Extraction Algorithm

Sangha Nam[○], Gyuhyeon Choi, Younggyun Hahm, Key-Sun Choi

KAIST

요 약

본 논문에서는 Reified 트리플 추출을 위한 한국어 개방형 정보추출 방법을 제시한다. 시맨틱웹 분야에서 지식은 흔히 RDF 트리플 형태로 표현되지만, 자연언어문장은 복수개의 서술어와 논항간의 관계로 구성되어 있다. 이러한 이유로, 시맨틱웹의 대표적인 지식표현법인 트리플을 따름과 동시에 문장의 의존구조를 반영하여 복수개의 술어와 논항간의 관계를 지식화하는 새로운 개방형 정보추출 시스템이 필요하다. 본 논문에서는 문장 구조에 대한 일관성있는 변환을 고려한 새로운 개방형 정보추출 방법을 제안하며, 개체 중심의 지식과 사건 중심의 지식을 함께 표현할 수 있는 Reified 트리플 추출방법을 제안한다. 본 논문에서 제안한 방법의 우수성과 실효성을 입증하기 위해 한국어 위키피디아 알찬글 본문을 대상으로 추출된 지식의 양과 정확도 측정 실험을 수행하였고, 본 논문에서 제안한 방식을 응용한 의사 SPARQL 질의 생성 모듈에 대해 소개한다.

주제어: 온톨로지, 시맨틱웹, 개방형 정보추출, 질의처리

1. 서론

웹상에는 방대한 양의 정보가 비-구조화된 텍스트 형태로 존재하고, 텍스트로부터 지식 혹은 정보를 추출하고자 하는 다양한 연구들이 있었다. 이들 중 전통적인 정보추출 (traditional information extraction)은 몇 가지 미리 지정된 특정 서술어에 한정하여, 대상 온톨로지에 적합한 일부 정보만을 추출할 수 있었고, 지도학습 (supervised learning) 방식을 기반으로 하였기 때문에 고비용일 뿐만 아니라 텍스트의 많은 정보들을 놓치는 단점이 있었다. 최근에는, 이러한 단점들을 극복하기 위해 개방형 정보추출 (open information extraction) [1]에 관한 연구가 활발히 진행 중이다. 개방형 정보추출은 이러한 단점에서 벗어나 대상 온톨로지를 지정하지 않는 도메인 독립적 방식이고 비-지도학습 방식, 그리고 대용량 텍스트에 대한 확장성을 목표로 현재 다양한 방식으로 연구가 이루어지고 있다. 즉 코퍼스만을 입력으로 받아 관계를 스스로 발견하여 정보를 추출하는 방식이다.

초기의 개방형 정보추출은 문장 내에서 <논항-술어-논항>의 순서로 나타나는 단순 트리플(triple) 형태의 정보 추출만을 대상으로 하였으나, 문장의 40% 이상이 다항관계(n-ary)로 구성된 성질 때문에 [2] 초기의 개방형 정보추출 방법들은 불완전하고 불충분하다는 한계가 있다. 이를 극복하기 위해 다항관계 추출을 목표로 다양한 연구가 진행되었고, 대표적인 시스템으로 Kraken [3], OLLIE [4], ClausIE [5] 등이 있다. 이 연구들은 모두 의존구조분석 결과를 기반으로 한 문장 내에서 복수개의 술어와 논항간의 관계를 추출하였다. Kraken은 문장 내에서 사실관계를 나타내는 논항들을 발견한 후 두 논항 사이에 공통으로 존재하는 동사를 찾는 방법을 제안하였다. 그리고 OLLIE는 자신들의 선행연구인 ReVerb [1]의 이진관계 추출결과를 학습 데이터로 사용하여 부스팅 기

법을 통해 새로운 패턴을 발견한 후 다항관계를 추출하는 방법을 제안하였다. ClausIE는 의존구조 분석 결과를 미리 정의한 7개의 절(clause)에 따라 분류한 후 각 절의 패턴을 기반으로 다항관계를 추출하는 방법을 제안하였다.

그러나 위 시스템들은 공통적으로 문장내에 나타난 모든 정보를 추출하지 못하고, 또한 추출한 지식을 일관성 있게 표현하지 못하고 있다. 이는 지식베이스의 대표적인 응용 프로그램인 질의응답을 함께 있어 지식의 부족으로 인해 정답을 제대로 찾지 못하거나 SPARQL 질의문을 작성하기 어렵게 만들며, 또한 각 시스템으로부터 추출된 지식들간의 통합이 까다로워진다. 또 다른 관점에서, 위 연구들은 영어를 대상으로 하였기 때문에 문법과 어순이 달라지는 다른 언어에는 적용하기 어려운 한계점이 있고, 그 이유로 중국어에 대한 개방형 정보추출 연구 [6]가 진행된 만큼 한국어에 적합한 개방형 정보추출 연구가 필요하다.

본 논문에서는 SRDF (Sentence RDF)라 불리는 Reified 트리플 추출을 위한 한국어 개방형 정보추출 방법을 제시한다. 본 방법은 기존의 시스템과는 달리 입력 문장의 술어-논항 구조를 일관성있게 Reified 트리플 형태로 변환하는 방법이며, 문장에 나타난 모든 정보를 포함하는 지식 그래프를 생성하는 방법이다. 시맨틱웹 분야에서 지식은 가장 기본적으로 주어, 서술어, 목적어로 구성된 트리플로 표현한다. 그러나 문장은 트리플만으로는 표현하기 어려운 정보들을 많이 포함하고 있기 때문에, 이들 사이에 연결고리를 지을 수 있는 새로운 정보추출 방법 및 지식표현법을 제안하며, 본 논문의 지식 표현법은 문장의 의존구조를 반영할 뿐만 아니라 질의처리같은 시맨틱 웹 어플리케이션에 적용하기 용이하도록 설계하였다. 또한 본 논문에서 제안하는 방식은 개방형 정보추출

의 목적에 기반하여 말뭉치(corpus)만을 입력으로 받아들이며 특정 온톨로지에 종속적이지 않으며, 별도의 학습 과정이 필요치 않아 웹 스케일의 텍스트에 대한 확장성이 보장되는 방식이다.

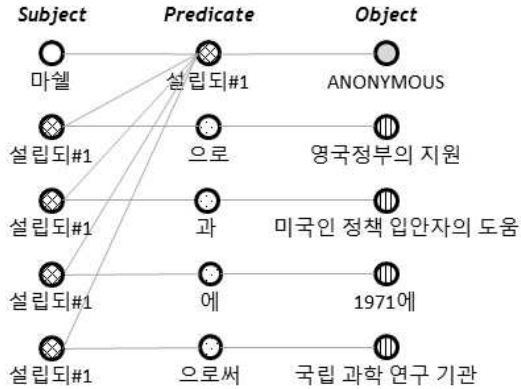


그림 1 SRDF 지식 그래프 예시

2. SRDF란?

텍스트로부터 특정 온톨로지에 적합한 지식을 바로 추출하는 작업은 복잡하고 많은 단계들, 예를 들어 개체연결 및 중의성 해소, 서술어 연결과 어휘의미망(Wordnet) 등의 다양한 자원들이 필요하다. 이러한 많은 단계들을 통합하여 텍스트로부터 온톨로지 지식을 직접 추출하는 방법은 복잡할 뿐만 아니라, 각 단계에서 발생하는 오류의 전파로 인해 전체 시스템의 성능이 만족할만한 수준에 도달하기까지는 어려움이 있다. 이러한 이유로 텍스트와 지식베이스 사이에 연결고리 역할을 하는 개방형 정보추출에 대한 연구가 필요하다.

SRDF의 목적은 텍스트로부터 Reified 트리플 즉 일종의 지식 그래프를 생성하여 텍스트와 온톨로지 지식 사이의 교량 역할을 하는 것이다. SRDF는 새로운 개방형 정보추출 시스템이자 문장 기반 어휘지식그래프 구조를 의미한다. SRDF 구조는 다음 3가지 목적을 위해 설계되었다. 첫째, 입력 문장의 의존구조에 기반하여 간단하고 간결한 규칙으로 일관성있게 Reified 트리플로 지식을 생성한다. 둘째, 개체중심의 지식과 사건중심의 지식을 모두 표현할 수 있도록 지식을 추출한다. 셋째, 기존의 많은 시맨틱웹 어플리케이션을 지원할 수 있도록 설계한다.

앞서 언급한 것처럼, 많은 시맨틱웹 어플리케이션들은 RDF 트리플로 지식을 표현하고 있지만, 한 문장에는 복수개의 술어와 논항이 존재하고, 개체중심의 지식뿐만 아니라 사건중심의 지식도 표현할 수 있어야하기 때문에 Reified 트리플로 지식을 표현하는 것이 적합하다. 본 논문에서는 다양한 RDF Reification 방법 중 Singleton Property [7] 방식을 이용하여 SRDF를 설계하였다. Singleton Property의 핵심 아이디어는 두 개체 사이의 관계는 보편적으로 고유하다고 할 수 있기 때문에 그 관계를 일종의 키(key)로 취급할 수 있다는 것이다. 이 방법은 기존의 RDF Reification 기술보다 중복되는 지식이 현저히 줄어들고 SPARQL과 같은 시맨틱웹 어플리케이션 등에서 더 높은 성능을 보이는 방법이다. 우리는 이 아

이디어를 개방형 정보추출과 지식표현법으로 응용하였다.

그림 1은 “마셜은 국립 과학 연구 기관으로써 1971년에 미국인 정책 입안자의 도움과 영국 정부의 지원으로 설립되었다.”라는 문장에 대한 SRDF 지식 그래프 예시이다. 그림 1에서 보는바와 같이, SRDF는 Reified 트리플 형식으로 지식을 추출하며 문장에 나타난 모든 술어와 논항간의 관계를 일종의 지식그래프로 표현한다. 이때 서술어는 고유한 아이디가 되기 때문에 “설립되#1”처럼 각 서술어마다 고유의 숫자를 첨가한다.

3. 한국어 개방형 정보추출 시스템의 설계

SRDF 시스템은 텍스트를 입력으로 받아 해당 텍스트로부터 Reified 트리플 형태의 지식 그래프를 출력한다. 본 시스템은 전처리기(Preprocessor), 기본 골격 트리 생성기(BST generator), 그리고 SRDF 생성기(SRDF generator)로 구성되며 시스템 아키텍처는 그림 2와 같다.

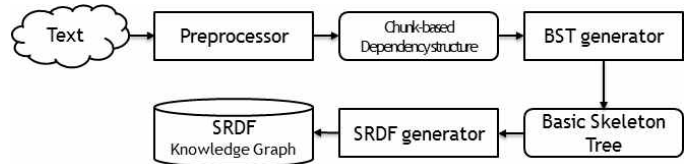


그림 2 SRDF 시스템 아키텍처

3.1. 전처리기

전처리기는 다시 ‘단문분할기’, ‘구 묶음 추출기’, 그리고 ‘구문구조 분석기’ 등의 3가지 서브 모듈로 구성된다. 먼저 단문분할기는 입력 문장을 접속조사를 기준으로 복문을 복수개의 단문으로 분할하는 역할을 하며, 본 시스템에서 사용한 분할 기준은 표 1과 같다. 이때 형태소 분석 결과 접속조사가 JC(세종품사태그)이면서 글자형태도 일치하는 접속조사만을 대상으로 분할하였다.

<표 1> 단문분할 기준

성질	예시
한 가지 이상의 일을 나열하는 것	-고, -(으)며
한 가지 이상의 일이 동시에 일어남을 보이는 것	-(으)면서
두 가지 일이 거의 동시에 잇달아 일어남을 보이는 것	-자
서로 상반됨을 보이는 것	-(으)나, -라도

또한 문장이 분할된 경우에는 분할된 각 부분 문장들이 주어를 가지고 있는지 확인한 후, 주어가 없는 경우에는 원 문장의 주어를 부착하는 방식으로 설계하였다.

“*암모니아는 대기 중에도 소량 포함되어 있으며, 천연수에 미량 함유되어 있기도 하다.*”

위 문장은 연결어미 ‘-으며/EC(in POS Tag)’를 기준으로 분할되며, 이때 두 번째 문장은 주어를 포함하고 있지 않기 때문에 ‘암모니아는’을 주어로 부착하여 아래와 같은 두 개의 문장으로 분할된다.

“암모니아는 대기 중에도 소량 포함되어 있으며,
“암모니아는 천연수에 미량 함유되어 있기도 하다.”

그 다음 구 묶음 추출기는 위 결과를 입력으로 받아 명사구(noun phrase)와 동사그룹(verb group)을 추출한다. 이때 명사구는 경우에 따라 관형어구(adnominal phrase)를 포함할 수 있으며, 동사그룹은 기본적으로 동사구(verb phrase)에서 보어구(complement phrase)와 명사구를 제외한 동사(verb)를 의미하며, 경우에 따라 부사구(adverb phrase)는 포함할 수 있다. 마지막으로 의존구조 분석기는 구묶음 기반 의존구조를 출력하며 예시는 그림 3과 같다.

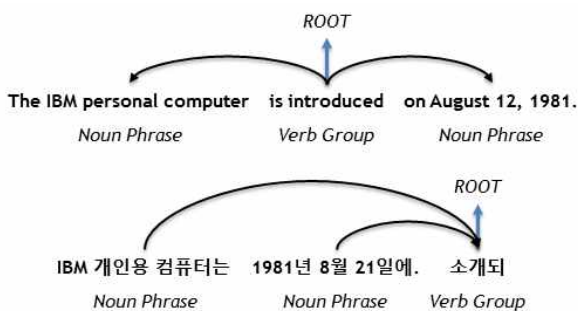


그림 3 구묶음 기반 의존구조 예시

3.2. 기본 골격 트리 생성기

기본 골격 트리 생성기는 전처리기의 출력을 입력으로 받아 SRDF 생성을 위한 기본 골격 트리를 생성한다. 그림 3에서 볼 수 있듯이 의존구조는 언어의 특성에 따라 서로 다른 구조를 가진다. 예를 들어, 영어와 중국어 그리고 한국어는 어순과 띄어쓰기 방식 등의 차이로 각각의 의존구조가 다르다. 그에 따라 SRDF 지식 그래프를 생성하기 이전에 의존구조를 일반화하기 위한 기본 골격 트리를 설계하였다.

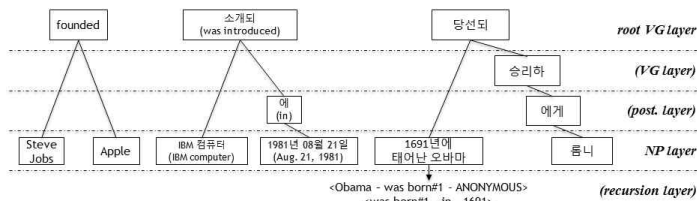


그림 4 SRDF 기본 골격 트리의 3가지 예시

그림 4는 SRDF 기본 골격 트리의 3가지 예시를 나타낸 그림이고, 기본 골격 트리는 다음과 같은 총 5개의 층(layer)로 구성된다; root-VG, VG, NP, Postposition, Recursion layer. 먼저 root-VG 층은 의존구조 상에서 root에 해당하는 동사그룹 하나만을 포함하는 최상위층이다. 그 다음 VG는 root-VG와 NP층 사이에 위치하는 층으로 root 이외의 동사 그룹들이 속하며, 의존구조 상에서 root로부터의 깊이에 따라 층의 개수가 결정되기 때문에 root로부터 깊이가 깊을수록 VG 층의 개수는 많

아진다. Postposition 층은 NP 층의 상단에 위치하며 NP들의 조사를 포함한다. NP 층은 문장에서 나타난 주어를 포함한 모든 NP들을 포함하는 층이고, 마지막으로 Recursion 층은 관형어구를 포함한 명사구가 존재할 때보다 세분화된 지식을 만들기 위한 층이다.

이와 같은 기본 골격 트리를 설계하여 SRDF 생성기의 입력 형식으로 사용함으로써 본 시스템이 다국어에 대해 적용이 가능하도록 하였다. 그리고 이와 같은 방법은 SRDF 생성기의 알고리즘을 간단하고 명료하게 해주는 장점이 있다.

3.3. SRDF 생성기

SRDF 생성기는 기본골격트리를 입력으로 받아 Reified 트리플 형태의 지식그래프를 생성하며, 그림 5는 SRDF 생성기의 알고리즘이다.

Algorithm 1 SRDF reified triple generation algorithm

```

1: procedure GENERATESRDF(G, SBJ, PRED, OBJQUEUE)
2:   for obj in objQueue do
3:     if obj is in NP layer then
4:       G ← G ∪ {<sbj,pred,obj>} ▷ Overwrite
        ANON object with the same sbj and pred
5:     else
6:       G ← G ∪ {<sbj,pred,ANON>}
7:       sbj ← pred
8:       pred ← obj
9:       generateSRDF(G, sbj, pred, obj.children)
10:  return G

```

그림 5 SRDF 지식 그래프 생성 알고리즘

본 알고리즘은 그래프(G), 문장의 주어(sbj), root-VG(pred), root-VG의 자식노드들(objQueue)을 입력으로 받아들이 확장된 그래프(G)를 반환한다. 알고리즘의 기본은 objQueue의 각 요소(obj)가 NP 층에 속하는지 아닌지를 검사한 후 (Line 3), 속한다면 트리플을 생성하여 그래프에 넣고 (Line 4), 속하지 않는다면 ANON 트리플을 생성하여 그래프에 넣은 후 Reification을 위해 sbj와 pred를 각각 변경한다 (Line 6-8). 그 다음 srdfGeneration 함수를 재귀 호출한다 (Line 9).

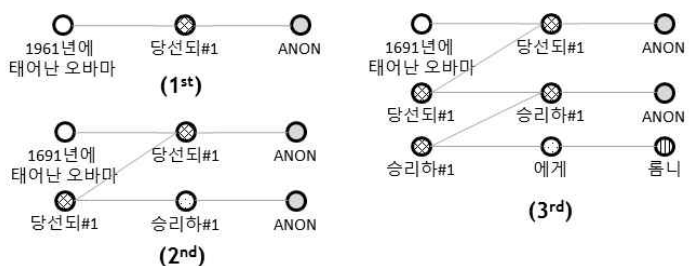


그림 6 SRDF 지식 그래프 생성 알고리즘 예시

그림 6은 본 알고리즘의 예시으로써 그림 4의 3번째 기본골격트리를 입력으로 사용한 그림이다. 먼저 문장의 주어 '1961년에 태어난 오바마'와 root-VG '당선되', 그리고 root-VG의 자식노드 '승리하'가 알고리즘의 입력으로 들어간다. 그 다음 '승리하'가 NP층에 속하지 않으므로 ANON 트리플 <1961년에 태어난 오바마, 당선되, ANON>을 생성한 후, sbj는 '당선되' pred

는 ‘승리하’로 변경한 후 ‘승리하’의 자식 노드인 ‘에게’를 objQueue에 삽입 후 srdfGeneration 함수를 재귀 호출한다(1st). 마찬가지로 ‘에게’가 NP층에 속하지 않기 때문에 ANON 트리플 <당선되, 승리하, ANON>을 생성한 후 sbj는 ‘승리하’ pred는 ‘에게’ 그리고 ‘에게’의 자식 노드인 ‘롬니’를 objQueue에 삽입 후 함수를 재귀 호출한다(2nd). ‘롬니’는 NP 층에 속하므로 트리플 <승리하,에게,롬니>를 생성한 후 <승리하,에게, ANON> 트리플이 그래프 G에 이미 들어있다면 이를 덮어쓴다(3rd). 마지막으로, 확장된 그래프를 반환하며 알고리즘이 종료된다.

4. 실험 및 응용

4.1. SRDF 성능 평가

본 논문에서 제안한 한국어 개방형 정보추출 방식의 성능을 분석하기 위해 한국어 위키피디아 알찬글 총 82개의 본문을 대상으로 약 140개의 문장을 랜덤하게 수집하여 실험 데이터로 사용하였고 이로부터 약 400개의 Reified 트리플을 추출하였다. ClausIE [5]에서 수행한 실험과 같이 본 논문에서도 추출된 트리플의 개수와 정확도 간의 상관관계를 측정하는 실험을 수행한 결과, 그림 7과 같은 결과를 얻었다. 먼저 실험 결과 문장당 약 2.8개의 트리플을 추출함을 알 수 있고, 그 정확도가가 약 0.75에서 안정됨을 확인할 수 있다. [5]에서는 영어에 대한 개방형 정보추출 방식으로써 약 0.7정도의 정확도를 보이는 것을 보아, 비록 언어와 테스트셋의 차이는 있지만 본 시스템이 다른 언어권의 개방형 정보추출 시스템과 비교하였을 때 비슷하거나 보다는 성능을 보인다는 것을 확인할 수 있다.

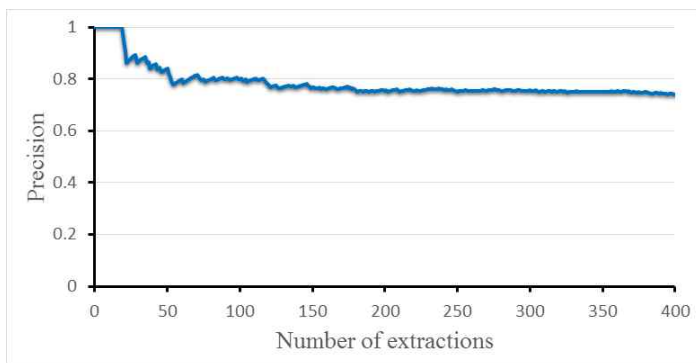


그림 7 SRDF 시스템 성능 평가 결과

4.2. SRDF 응용

본 논문에서 제안한 SRDF가 기존의 시맨틱웹 어플리케이션에서 응용될 수 있는지를 살펴보기 위해 OKBQA (Open Knowledge Base and Question Answering) [8]에서 사용될 수 있는 의사 SPARQL 질의문 생성모듈인 SenTGM을 개발하였다. OKBQA는 시맨틱웹 분야에서 전세계에 흩어진 다양한 자원들과 서로 다른 팀들이 개발한 모듈들이 조화를 이루게 하여 지식구축과 질의응답에 관련된 문제를 해결하고자 하는 해커톤이자 커뮤니티이다.

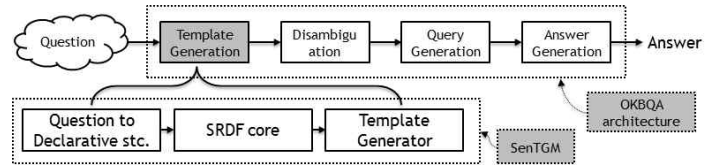


그림 8 SRDF의 응용인 SenTGM 워크플로우

그림 8은 OKBQA 아키텍처와 SRDF의 응용인 SenTGM의 워크플로우를 나타낸 그림이다. SRDF의 주요 목적 중 하나는 텍스트의 술어-논항 구조로부터 시맨틱웹 어플리케이션에서 사용될 수 있는 Reified 트리플을 생성하는 것이다. 따라서 OKBQA 아키텍처 중에서 자연언어 질의문을 입력으로 받아 트리플로 구성된 의사(pseudo) SPARQL 질의문을 생성하는 템플릿 생성 단계 (Template Generation)에서 SRDF가 유용하게 사용될 수 있다.

SenTGM은 자연언어 질의문을 입력으로 받아 3가지 단계를 거쳐 의사 SPARQL 질의문을 생성한다. SRDF는 긍정평서문을 입력으로 가정하기 때문에 첫 번째 단계는 자연언어 질의문을 긍정평서문 형태로 변환하는 단계 (Question to Declarative stc.)이다. 예를 들어, “1904년에 노벨생리의학상을 받은 러시아의 생리학자는 누구인가?”라는 질문에 대해 “누구는 1904년에 노벨생리의학상을 받은 러시아의 생리학자이다.”라는 형식으로 변환한다. 이때 질의문의 정답 타겟 워드는 ‘누구’, ‘어디’, ‘언제’, ‘무엇’ 4가지로 한정하였으며, 한국어의 특성상 정답 타겟 워드가 없이 “1904년에 노벨생리의학상을 받은 러시아의 생리학자는?”과 같은 형식으로 질문할 수 있기 때문에 정답 타겟 워드가 생략된 문장도 변환이 가능하도록 설계하였다.

두 번째는 본 논문에서 제안한 SRDF를 수행시키는 단계(SRDF core)로써 앞서 긍정평서문으로 치환된 질의문에 대한 SRDF 지식그래프를 생성한다. 질의를 위한 트리플은 질문에 등장하는 모든 명사들이 정답을 찾기 위한 주요 키워드로 간주될 수 있어서 SRDF 지식그래프 생성시 명사가 아닌 명사단위로 지식을 추출하도록 변형하였다. 앞선 질문 예시에 대한 SRDF 지식 그래프는 그림 9와 같다.

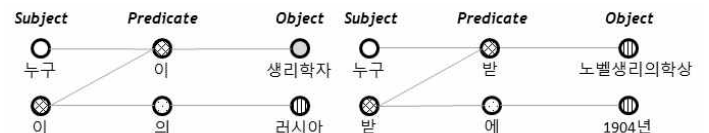


그림 9 SenTGM의 SRDF core 결과 예시

마지막으로는 SRDF 지식그래프를 OKBQA 프레임워크의 템플릿 [9] 생성 아웃풋 포맷으로 변환하는 단계 (Template Generator)가 수행되며, SenTGM의 최종 결과는 그림 10과 같다. 첫 번째 질의 트리플은 ‘누구’의 타입이 생리학자라는 것이고, 두 번째는 ‘러시아의 생리학자는 누구’다 라는 것이다. 여기서 ‘생리학자’가 때로는 class로, 때로는 property로 사용되었는데 이 부분은 템플릿 생성의 다음 단계인 중의성 해소 단계에서 각각 적절한 class와 property를 찾아 연결함으로써 둘

간의 차이를 구별한다. 특히 두 번째 질의트리플에서 생리학자가 서술어로 쓰인 이유는, 트리플의 한국어 해석상 ‘주어의 서술어는 목적어이다’가 간결하면서도 명확하기 때문이다. 세 번째는 ‘누구’가 노벨생리학상을 받았다는 것이고, 네 번째는 ‘누가’ 1904년에 받았다는 것을 나타내는 의사 SPARQL 질의문이다. 이러한 의사 SPARQL 질의문은 후에 중의성 해소 및 SPARQL 생성 단계를 거쳐 정답을 도출한다.

SenTGM은 현재 OKBQA 프레임워크 상에서 한국어 질의 처리 모듈의 템플릿 생성 단계에서 정상 작동중이며, 이를 통해 SRDF가 기존의 시맨틱웹 어플리케이션에 잘 응용될 수 있음을 확인할 수 있다.

```
SELECT ?v1 WHERE {
  ?v1 type 생리학자(C) .
  러시아(R) 생리학자(P) ?v1 .
  ?v1 받(P) 노벨생리학상(R/L) .
  ?v1 받(P) 1904년(R/L) .
}
C := rdf:Class
R := rdf:Resource
P := rdf:Property
L := rdfs:Literal
```

그림 10 SenTGM의 최종 결과 예시

5. 결론

본 논문에서는 Reified 트리플 추출을 위한 한국어 개방형 정보추출 방법을 제시하였다. 본 방법은 한국어 자연어 문장을 입력으로 가정하며, 이로부터 문장 내의 모든 논항과 서술어간의 관계를 추출할 뿐만 아니라, Reification 기법을 바탕으로 다항 관계를 트리플로 유지하면서도 지식 표현의 일관성을 갖추고, 시맨틱웹 어플리케이션에 쉽게 적용될 수 있도록 설계하였다. 본 논문에서 제안하는 방법의 우수성을 입증하기 위해 실제 시스템을 개발하였고, 한국어 위키피디아 알찬글 문장들을 이용해 높은 성능을 확인하였다. 또한 SenTGM을 개발하여 OKBQA 프레임워크에서 정상 작동하도록 함으로써 SRDF가 다양한 시맨틱웹 어플리케이션에 응용될 수 있음을 확인하였다. 향후에는 관형어구를 포함한 명사구를 보다 세분화하여 SRDF 지식그래프를 확장하는 연구와 SRDF를 이용한 개방형 질의처리에 관한 연구를 진행할 계획이다.

사사

이 논문은 2016년도 정부(미래창조과학부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임. (No. R0101-16-0054, WiseKB: 빅데이터 이해 기반 자가 학습형 지식베이스 및 추론 기술 개발)

이 논문은 2016년도 미래창조과학부의 재원으로 한국연구재단 바이오의료기술개발사업의 지원을 받아 수행된 연구임.

참고문헌

- [1] Fader, A., Soderland, S., & Etzioni, O. "Identifying Relations for Open Information Extraction", In Proceedings of EMNLP, 2011.
- [2] Christensen, J., Soderland, S., & Etzioni, O. "An Analysis of Open Information Extraction based on Semantic Role Labeling", In Proceedings of the sixth international conference on Knowledge capture, 2011.
- [3] Akbik, A., & Loser, A., "Kraken: N-ary Facts in Open Information Extraction", In Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction, 2012.
- [4] Schmitz, M., et al. "Open Language Learning for Information Extraction", In Proceedings of the 2012 Joint Conference on EMNLP and CNLL, 2012.
- [5] Del Corro, L., & Gemulla, R. "ClausIE: Clause-Based Open Information Extraction", In Proceedings of WWW, 2013.
- [6] Tseng, Y. H., et al. "Chinese Open Relation Extraction for Knowledge Acquisition", EACL, 2011.
- [7] Nguyen, V., Bodenreider, O., & Sheth, A. "Don't Like RDF Reification?: Making Statements about Statements Using Singleton Property", In Proceedings of WWW, 2014.
- [8] <http://www.okbqa.org/>
- [9] Unger, C., et al. "Template-based question answering over RDF data.", In Proceedings of the 21st international conference on WWW, ACM, 2012.