한국어 의존 파싱을 이용한 트리플 관계 추출

곽수정[O*], 김보겸[†], 이재성[†]
충북대학교 정보산업공학과[*], 디지털정보융합과[†]
crystalk@cbnu.ac.kr, bogyum@cbnu.ac.kr, jasonlee@cbnu.ac.kr

Triplet Extraction using Korean Dependency Parsing Result

Sujeong Kwak[O*], Bogyum Kim[†], Jae Sung Lee[†]
Dept. of Information & Industrial Engineering[*], Dept. of Digital Informatics and Convergence[†],
Chungbuk National University

요 약

자연언어 문서에서 지식 추출은 QA 시스템을 비롯한 여러 분야에서 필수적이다. 트리플은 가장 일반적인 지식 추출 형식으로 문장 내부의 지식 정보를 주어, 서술어, 목적어의 관계로 표현한다. 본 논문에서는 한국어 의존 파서로 문장을 분석하고, 그 결과에서 트리플을 추출하는 방법을 제안했다. 제안된 트리플 추출기는 21개 문장에서 추출된 78개의 트리플 정답 집합과, 64개의 준정답 집합에 대해서 각각 60.75%와 66.67%의 F-measure 성능을 보였다.

주제어: 트리플 추출, 지식 추출, 관계 추출, 의존 구문 분석기

1. 서론

신문이나 블로그, 위키피디아와 같이 자연언어처리에서 활용 가능한 텍스트 문서가 증가하면서 자연언어 문서에서의 지식 추출에 대한 요구가 증가하고 있다[1]. 추출되는 지식 정보를 이용하면 문서에서 중요한 개체와 그들간의 관계를 파악할 수 있기 때문에 지식추출은 QA 시스템을 비롯한 다양한 시스템의 필수적인 요소가 될 수 있다[2]. 트리플은 가장 일반적인 지식 추출 형식으로 문장 안에 나타난 주어, 서술어, 목적어를 표현한 정보이다[3].

본 논문에서는 한국어 문장 또는 문서에 대하여 의존 파싱을 수행하고, 그 결과를 이용해 트리플을 추출하는 방법을 제안하고자 한다. 2장에서는 트리플 추출과 관련된 논문들을 소개하고, 3장에서는 의존 파싱 결과로 트리플을 추출하는 방법을 설명한다. 4장에서는 설명된 방법으로 구축된 트리플 추출기의 성능을 측정 및 평가하고, 5장에서 결론을 맺는다.

2. 관련 연구

자연언어 문서로부터 지식을 자동으로 추출하고 추론하는 것은 대개 한 문장에서 트리플 정보인 주어, 서술어, 목적어 관계를 추출하는 것으로부터 시작한다[3]. [4]에서는 공개된 파서(스탠포드 파서[5], 링크 파서[6] 등)의 파싱 결과로부터 규칙을 이용하여 트리플을 추출하는 방법을 제시하였다. 또한 [7]에서는 문장에 나타난 모든 중요 단어들을 트리플 관계로 조합하고 이를 SVM을 이용하여 유효한 트리플 관계를 판정하고 추출하였다. [8]은 의존 파서를 통한 구문 분석 결과를 이용해 트리플을 추출하는 방법을 설명한다. 특히 트리플에서 한 요소의 범위에 대한 문제, 능동과 수동 구조의 문장에 대한 처리, 병렬 구조의 문장에 대한 처리, 문장 내부의 종속절에 대한 처리 등 파서의 결과를 이용해 트리플을 추출할 때 생길 수 있는 여러 가지 고려사항에 대해 언급하고 있다. [9]에서는 영어의 술어-논항 구조를 분석하고 이를 정규화한 어휘 패턴으로 생성한 후, 스트링 커널 방법으로 트리플 관계를 추출하였다. 특히 [9]에서는 서술어를 38개의 관계로 한정하여 처리하였다. 이렇게 자연언어 문서에서 트리플을 추출하는 연구가 기존에도 존재하지만 모두 영어를 대상으로 하기 때문에 한국어에 바로 적용하기 어렵다.

본 논문에서는 한국어 의존 파싱의 결과를 이용하여 한국어 자연언어 문서에서 트리플을 추출하기 위한 방법을 제안한다.

3. 트리플 추출

3.1 트리플 추출 과정

의존 파서를 이용한 트리플 추출은 [그림 1]과 같은 과정으로 진행된다.

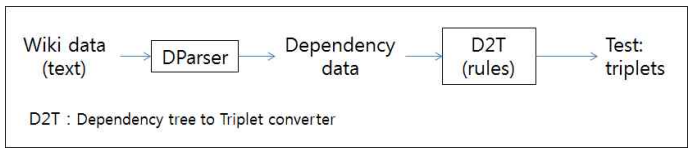


그림 1 트리플 추출 흐름도

먼저 입력 문서가 주어지면 세종 구문 분석 표지[10]를 이용한 ETRI 의존 파서로 문장의 구문 구조를 분석한다. 분석된 결과를 트리플 변환기(D2T)에 입력하면 의존 구조로 나타난 문장의 주어, 서술어, 목적어의 관계가 규칙에 따라 트리플 형식으로 추출된다. 이때 추출되는 주어와 목적어는 조사를 제거하고, 서술어는 기본형으로 표기하

여 저장한다. 이는 트리플을 정규화하여 표현함으로써, 추론 단계에서 트리플 사용을 간편하게 하기 위함이다.

3.2 트리플 추출 규칙

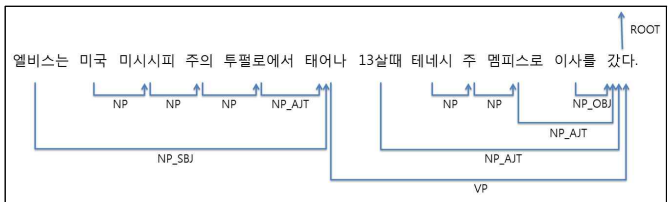
의존 문법에서 주어와 서술어는 각각 SBJ 의존 관계의 tail과 head가 되고 목적어와 서술어는 각각 OBJ 의존 관계의 tail과 head가 된다. 따라서 *_SBJ 레이블과 *_OBJ 레이블이 동일한 서술어를 head로 가지고 있는 경우 *_SBJ의 tail을 주어로, head를 서술어로, *_OBJ의 tail을 목적어로 하는 트리플을 구성할 수 있다. 이때 '*'는 wild card로 *_SBJ는 NP_SBJ, VP_SBJ 등을 포함한다.

서술어를 설명하는 부사는 서술어와 함께 주어에 대한 정보가 될 수 있기 때문에 *_AJT관계를 *_OBJ 관계와 같은 방식으로 처리한다. 그리고 문장에 동사가 여러 개 있어 주어를 공유한 경우에 동사와 동사는 VP 관계로 레이블되므로 주어가 없는 구에서 VP 관계를 통해 문장 전체의 주어를 찾을 수 있다. 이것을 트리플 추출에 대한 기본 규칙으로 [그림 2]와 같이 정의한다.

1. *_SBJ 레이블과 *_OBJ(혹은 *_AJT) 레이블이 연결된 서술어는 트리플로 묶는다.
2. VP 의존 관계를 통해 백트래킹 후 *_SBJ 의존 관계가 존재하면 해당 명사구를 주어로 사용한다.

그림 2 트리플 추출 기본 규칙

[그림 3]은 트리플 추출의 예로, ①은 기본 규칙 1에 의해 ②,③,④는 기본 규칙 1과 2에 의해 추출된 트리플이다.



추출 트리플

①(엘비스, 태어나다, 투펄로) ③(엘비스, 가다, 멤피스)
②(엘비스, 가다, 이사) ④(엘비스, 가다, 13살)

그림 3 트리플 추출 예

한국어는 주어의 생략이 빈번하게 발생하기 때문에 기본 규칙만 이용해 트리플을 추출하면 문제가 발생한다. 따라서 [그림 4]의 주어 생략 보완 규칙을 정의하여 적용한다.

1. 문장에 주어가 없는 경우 이전 문장의 주어를 사용해 트리플을 구성한다.

그림 4 주어 생략 보완 규칙

4. 실험 및 평가

4.1 평가 집합 구축

트리플 추출기의 성능 평가를 위한 데이터로 엘비스 프레슬리에 대한 위키피디아 문서[11]를 이용했다. 해당 문서는 총 21개의 문장으로 구성되었으며 각각의 문장에 대하여 트리플을 추출하여 총 78개의 정답(Gold answer) 트리플과 64개의 준정답(Silver answer) 트리플 평가 집합을 만들었다.

정답 집합은 원본에서 직접 추출한 트리플 집합으로, 의존 파서를 이용한 트리플 추출기 전체의 성능을 평가할 수 있는 평가 집합이다. 준정답 집합은 정답 집합에서 의존 파서의 오류에 의한 트리플 추출의 오류를 제거한 집합으로, 정답 집합의 82.05%로 구성된 부분 집합이다. 이것을 이용하면 파서의 오류에 의한 트리플 추출기의 성능 저하를 무시하고, 트리플 변환기의 성능만 평가할 수 있다. [그림 5]는 정답과 준정답 그리고 추출기에서 뽑힌 트리플의 차이를 설명한다.

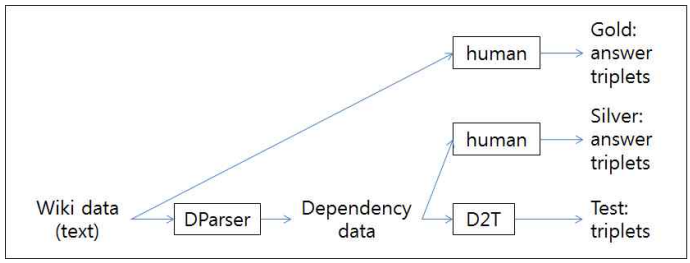


그림 5 정답, 준정답, 출력 결과의 차이점

트리플 정답을 구성할 때, 주어와 목적어의 범위를 정하는 기준이 문제가 될 수 있다. 본 논문에서 제안하는 트리플 추출기는 현재 이 범위 문제를 고려하지 않고 트리플을 추출하고 있으나 추후 확장된 트리플 추출기의 평가를 위해 주어와 목적어의 범위를 "트리플이 의미를 가지게 하는 최소의 범위"로 정의하여 평가 집합을 구성하였다. 예를 들어 "엘비스는 미국 미시시피 주의 투펄로에서 태어났다."라는 문장에서 목적어는 "미국 미시시피 주의 투펄로", "미시시피 주의 투펄로", "투펄로"가 될 수 있다. 그중 "투펄로"는 지명으로 추출하고자 하는 트리플에서 충분히 의미가 되기 때문에, 이를 목적어로 하였다. 다른 예로 "엘비스는 많은 음악 명예의 전당에 올랐다."라는 문장에서 목적어는 "음악 명예의 전당", "명예의 전당", "전당"이 될 수 있는데, "음악 명예의 전당"을 목적어로 하여 트리플을 추출하였다.

4.2 평가

정답 집합과 준정답 집합이 서로 다르기 때문에 트리플 추출기의 성능을 각각에 대해서 측정했다. [표 1]은 정답 집합에 대한 트리플 추출기의 성능이고, [표 2]는 준정답 집합에 대한 트리플 추출기의 성능이다. 이때 기본 규칙만 이용해 추출된 트리플은 38개이고, 기본 규칙과 보완 규칙을 함께 적용해 추출된 트리플은 80개이다.

표 1 정답 집합에 대한 트리플 추출기의 성능

구분	Recall	Precision	F-measure
기본 규칙	35.89%	73.68%	48.26%
기본+보완 규칙	61.53%	60.00%	60.75%

표 2 준정답 집합에 대한 트리플 추출기의 성능

구분	Recall	Precision	F-measure
기본 규칙	43.75%	73.68%	54.90%
기본+보완 규칙	75.00%	60.00%	66.67%

기본 규칙만 이용해 구성한 트리플 추출기는 정답 집합과 준정답 집합 모두에서 정확률이 낮고, 재현율이 높은 것을 확인할 수 있었다. 이것은 기본 규칙만 이용하면 추출되는 트리플의 수가 적기 때문에 나타나는 현상이다. 하지만 주어 생략 보완 규칙을 통해 추가적으로 생성되는 트리플이 정답(혹은 준정답) 집합에 저장되어 있는 경우가 많았다. 따라서 재현율이 떨어지는 정도보다 정확률이 올라가는 정도가 높아 F-measure의 성능이 향상되는 것을 확인할 수 있다.

4.3 트리플 추출기 오류 분석

준정답 집합의 트리플 중 추출되지 못한 트리플은 대부분 병렬 구조 문장의 것이었다. [그림 6]의 예를 보면 "함께"로 연결된 트리플 '(엘비스, 하다, 빌 블랙)'과 '(엘비스, 하다, 스코티 무어)'를 추출기에서 뽑지 못하였다.

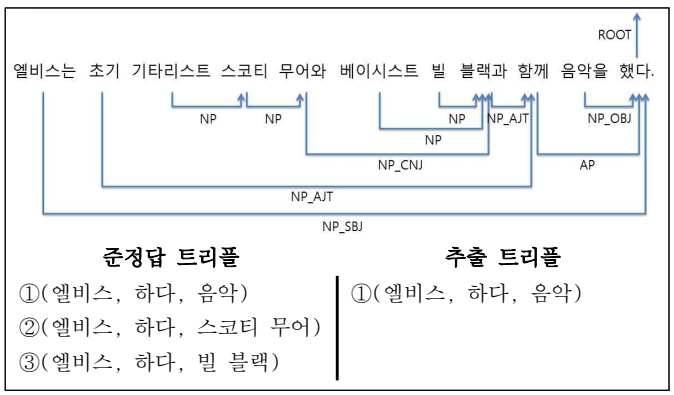


그림 6 병렬 구조 문장 트리플 추출 오류 예

그리고 추출된 트리플 중 정답이나 준정답에 없는 트리플은 의존 파서의 오류를 그대로 트리플로 추출한 것이거나, 주어가 아닌 성분을 수식하는 종속절에서 잘못된 주어를 가지는 트리플이 추출된 경우이다. [그림 7]의 예를 보면 '(음반 판매고, 많다, 팝 역사상)' 트리플이 주어를 제대로 찾지 못해 '(엘비스, 많다, 역사상)'과 같이 잘못된 트리플로 뽑힌다.

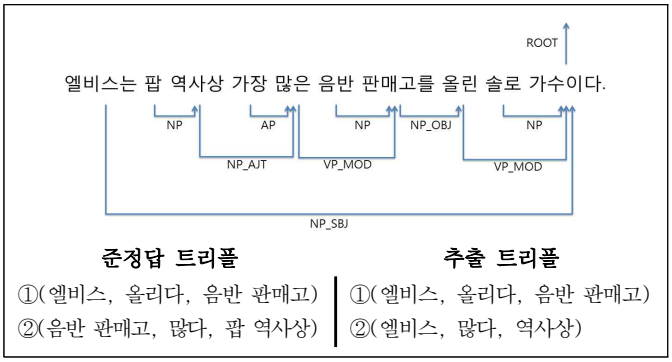


그림 7 종속절에서 추출된 오류 트리플

4.4 트리플 추출기 미고려 사항

현재 트리플 추출기는 주어와 목적어의 범위에 대한 규칙이 없다. 따라서 트리플 추출기 평가 시 이를 고려하지 않기 위해 정답에서 제시한 주어(혹은 목적어)의 head가 추출된 트리플의 주어(혹은 목적어)의 head와 같다면 옳은 것으로 간주했다. 4.1절의 예를 빌어 설명하면, 문장 '(엘비스, 오르다, 전당)'과 '(엘비스, 오르다, 음악 명예의 전당)'을 같은 것으로 간주하고 있다. 하지만 추후에는 주어와 목적어의 범위에 대한 규칙을 추가하고, 평가할 필요가 있다.

또한, 트리플의 정규화를 위해 목적어의 조사를 생략해 표기하고 있다. 이 때문에 목적어 성분이 문장에서 어떤 역할을 하는지 파악하지 못하는 경우가 있다. [그림 3]의 트리플 '(엘비스, 가다, 멤피스)'를 예로 보면 조사 '로'가 생략되어 엘비스가 멤피스에서 간 것인지 엘비스가 멤피스로 간 것인지 트리플이 정확한 정보를 가지지 못한다. 따라서 목적어의 역할을 분명히할 수 있는 구분을 트리플에 추가해 표기할 필요가 있다.

5. 결론

본 논문에서는 한국어 의존 파싱 결과를 이용하여 트리플을 추출하기 위한 방법을 제안했다. 트리플 추출기는 수동으로 구축된 정답 집합과 준정답 집합으로 평가했다. 평가를 위해 엘비스 프레슬리에 대한 위키피디아 문서를 이용해 평가 집합을 구성했다. 평가 결과 정답 집합에 대한 F-measure는 60.75%, 준정답 집합에 대한 F-measure는 66.67%의 성능을 보였다. 향후에는 현재 해결하지 못한 병렬 문장에서의 트리플 추출, 종속절에서의 정확한 트리플 추출, 트리플 구성 요소(주어, 서술어, 목적어)의 범위 설정 문제 등을 해결하고, 목적어 성분의 역할을 분명히 할 수 있도록 구분 정보를 추가해야한다. 또한 현재 평가에 사용한 평가 집합보다 더 많은 양의 집합을 이용해 추출기의 성능을 보다 정확히 평가할 필요가 있다. 또한 평가 문장이 적어 확인하지 못했던 트리플 추출 규칙의 부족한 부분을 확인하고 추가할 필요가 있다.

감사의 글

본 연구는 미래창조과학부 및 한국산업기술평가관리원

의 산업융합원천기술개발사업(정보통신)의 일환으로 수행하였음. [100445577, (1세부) 휴먼 지식증강 서비스를 위한 지능진화형 Wise QA 플래폼 기술 개발]

참고문헌

[1] A. Culotta, J. Sorense, "Dependency tree kernels for relation extraction," In Proceeding of the 42nd annual Meeting on Association for Computational Linguistics, 2004.
[2] J. Cowie, W. Lehnert, "Information extraction," communication of the ACM, vol.39, no.1, pp.80-91, 1996.
[3] J. Leskovec, M. crobelnik, N. Milic-Frayling. "Learning sub-structure of document semantic graphs for document summarization," In Proceeding of the 7th International Multiconference Information Society IS 2004, vol.B, pp.18-25, 2004.
[4] D. Rusu, L. Dali, B. Fortuna, M. Grobelnik, D. Mladenic, "Triplet extraction from sentences," In Proceeding of the 10th International Multiconference Information Soociety IS 2007, pp.8-12, 2007.
[5] Stanford Parser web page : http://nlp.stanford.edu/software/lex-parser.shtml
[6] Link Parser web page: http://www.link.cs.cmu.edu/link/.
[7] L. Dail, B. Fortuna, "Triplet extraction from sentences using SVM," In Proceedings of SiKDD, 2008
[8] D. Choi, K. Choi, "Automatic relation triple extraction by dependency parse tree traversing," Poster and Demo, pp.23-24, 2008.
[9] 정창후, 전홍우, 송사광, 홍순찬, 정한민, 최성필, "술어-논항 구조의 어휘 패턴을 이용한 스티링 커널 기반 관계 추출," 정보과학회 논문지: 소프트 웨어 및 응용 제39권, 제12호, pp.927-993, 2012.
[10] 국립국어원, 21세기 세종계획 최종 성과물(2011년 12월 수정판), 2011.
[11] 한국어 위키피디아 web page : http://ko.wikipedia.org/