Check for updates

Journal of KIIT. Vol. 19, No. 8, pp. 17-26, Aug. 31, 2021. pISSN 1598-8619, eISSN 2093-7571
http://dx.doi.org/10.14801/jkiit.2021.19.8.17

자동분석기와 언어규칙 기반 한국어 형태-구문 부착 말뭉치 반자동 구축 방법

류법모*, 오효정**

Semi-Automatic Construction of Korean Morpho-syntactically Annotated Corpus using Automatic Analyzers & Linguistic Rules

Pum-Mo Ryu*, Hyo-Jung Oh**

이 논문은 2017년 과학기술정보통신부의 재원(NRF-2017M3C4A7068186)과 2019년 교육부의 재원(NRF-2019S1A5B6102698)으로 한국연구재단의 지원을 받아 수행된 연구임

요 약

본 논문에서는 복수의 자동 언어 분석기와 언어규칙을 활용하여 반자동으로 한국어 형태 및 구문 부착 말뭉치를 구축하는 방법을 제안한다. 약 6억5천만 토큰 규모의 한국어 원시말뭉치를 대상으로 한국전자통신연구원과 울산대의 형태 및 구문 분석기를 이용하여 자동으로 분석하였고, 자동 형태소 분석 결과를 언어규칙 및 예제기반으로 정규화한 후에 두 분석기의 결과가 동일한 문장을 추출하였다. 이후 자동 구문분석 결과의 일관성 오류를 수정한 후 분석결과를 비교하여 최종적으로 약 8,300,000문장 81,500,000토큰 규모의 대용량 형태 및 구문 부착 말뭉치를 CoNLL-U 형식으로 구축하여 GitHub에 공개하였다. 형태소 부착 말뭉치에 대한 수작업 평가에서 토큰 기준 99.12%, 문장 기준 92.00%의 정확도를 보였으며, 구문부착 말뭉치의 경우 LAS와 UAS 모두 토큰 기준 98.00% 이상의 정확도를 보였다.

Abstract

In this paper, we propose a semi-automatic method to construct a Korean Morpho-syntactically annotated corpus using automatic language analyzers and linguistic rules. We automatically analyzed the Korean raw corpus of about 650 million tokens using morpho-syntactic analyzers of ETRI and Ulsan University. After normalizing the results of morphological analysis using linguistic rules and example rules, sentences with the same results of both analyzers were extracted. After correcting the consistency errors of the automatic syntactic analysis results, the results were compared to construct a large-scale morpho-syntactically annotated corpus of about 8,300,000 sentences and 81,500,000 tokens. We finally converted the corpus into CoNLL-U format and published it on GitHub. In the evaluation of the morpheme annotated corpus, the accuracy was 99.12% on token basis and 92.00% on sentence basis. For syntactically annotated corpus, both LAS and UAS showed higher than 98.00% accuracy on token basis.

Keywords

Korean morpho-syntactically annotated corpus, raw corpus, morpho-syntactic analyzer, CoNLL-U format

* 부산외국어대학교 사이버경찰학과 교수(교신저자)
- ORCID ID: https://orcid.org/0000-0002-9777-9211
** 전북대학교 문헌정보학과 부교수
- ORCID ID: https://orcid.org/0000-0001-8067-2832

· Received: Jul. 26, 2021, Revised: Aug. 18, 2021, Accepted: Aug. 21, 2021
· Corresponding Author: Pum-Mo Ryu
Dept. of Cyber Security & Police, Busan University of Foreign Studies, Korea,
Tel.: +82-51-509-6782, Email: pmryu@bufs.ac.kr

Ⅰ. 서 론

자연언어처리(Natural language processing) 기술은 인공지능에서 핵심적인 요소이고, 대용량 고품질의 말뭉치를 활용하는 학습 방식을 적용하여 정확성과 커버리지를 향상시킬 수 있다. 특히, 최근에는 딥러닝을 비롯한 인공지능 기술의 비약적인 발전으로 인해 고품질의 학습데이터에 대한 필요성이 더욱 높아지고 있다. 이러한 필요성에 부흥하여 한국어 분석기술의 연구와 산업화를 목적으로 표준 말뭉치를 구축하고, 공유 및 확산하기 위한 국가차원의 노력이 진행되고 있다. 국립국어원에서는 2018년부터 다양한 유형의 대용량 말뭉치 구축 프로젝트를 진행하여 '모두의 말뭉치' 사이트를 통해서 공개하고 있다[1]. 또한 한국전자통신연구원(ETRI)에서는 'AIHub' 사이트를 통해서 다양한 종류의 음성 및 자연언어 데이터를 공개하고 있다[2].

한편 전 세계적으로는 여러 언어에서 각자의 방식으로 말뭉치를 구축하면서 나타나는 호환성 문제점을 개선하고, 여러 언어 간 말뭉치를 연계하여, 특정 언어를 처리하기 위해 개발된 모듈을 다른 언어 모듈에서도 쉽게 활용하기 위한 노력이 UD (Universal Dependencies) 프레임워크를 중심으로 진행되고 있다[3]. UD에서는 품사 태그와 의존 관계 태그 부착 수준에서 말뭉치 구축 가이드라인을 제공하고 연구자들이 자발적으로 참여하여 자신의 말뭉치를 공개할 수 있도록 유도하고 있다. 하지만 공개된 UD 기반의 한국어 말뭉치의 규모가 영어, 일본어 말뭉치에 비해서 작은 문제점이 있다.

본 논문에서는 국내에서 개발된 여러 개의 자동 언어 분석기를 활용하여 반자동으로 대용량 한국어 형태소 및 구문 부착 말뭉치를 구축하는 방법을 제안한다. KCC(Korean Contemporary Corpus) 원시말뭉치[4]를 대상으로 전처리 작업을 거친 후 ETRI와 울산대의 자동 형태소 및 의존 구문분석기를 적용하였다. 자동 형태소 분석 결과를 언어규칙 및 예제 기반으로 정규화한 후에 두 분석기의 결과가 동일한 문장을 추출하였고, 이후 구문분석 결과의 일관성 오류를 수정한 후 자동 매칭을 통해 최종적으로 CoNLL-U 형식[5]의 형태 및 구문 분석 일치 말뭉치를 구축하였다. 두 언어분석기의 분석 결과에서 차이가 많은 패턴을 수집하고, 수작업으로 검토하여 형태소 분석 정규화 규칙과 구문분석 일관성 오류 수정 규칙을 작성하였다. 또한, 형태소 분석 말뭉치의 경우 문맥 기반 자동검증을 통해 두 개의 언어분석기에서 동일한 오류를 발생시켰을 때 탐지하는 방법을 적용하였다. UD에서 제안하는 CoNLL-U 형식은 이미 많은 연구에서 공용으로 활용하고 있는 말뭉치 형식으로, 이에 맞게 대용량의 한국어 표준 형태 및 구문 분석 말뭉치를 구축하여 GitHub에 배포하였다.

본 논문의 구성은 다음과 같다. 2장에서는 관련 연구를 기술한다. 3장에서는 분석 대상이 되는 한국어 원시말뭉치에 대해서, 4장에서는 형태 구문 분석 말뭉치 반자동 구축단계에 대해서, 마지막으로 5장에서는 결론 및 향후 연구를 기술한다.

Ⅱ. 관련 연구

기존에 UD 사이트에 공개되어 있는 한국어 주석 말뭉치 관련해서 Chun의 연구에서는 Google Korean Universal Dependency Treebank(GKT), The Parallel Universal Dependency Treebank(PUD), KAIST Korean Universal Dependency Treebank(KTB)의 특징에 대해 구체적으로 논의하였고, 실제 한국어를 대상으로 구축할 때 선행되어야 할 토큰의 기준, 문장 부호 처리 기준, 인용문 처리 기준, 문장 성분 생략 문제, 대등 연결에서 지배소 지정 문제, 품사 태그셋 문제 등을 예시를 들어 설명하였다[6].

Lee의 연구에서는 UD의 의존관계 주석체계를 한국어에 적용할 때 발생하는 문제점을 분석하였다. 전체 UD 의존관계 리스트 중에서 영어권에서는 활용될 수 있지만 한국어에서는 사용되기 어려운 iobj(간접목적어), xcomp(개방보문), exp(허사), cop(계사), mark(절접속표지), dislocated(전위), clf(분류사), case(격표지), parataxis(무표지접속), orphan(생략)을 한국어 UD 말뭉치 구축에는 적용하지 않는 표지로 분류하였다. 이 연구는 언어학적 정합성에 초점을 두고 말뭉치 구축 방법론 제안에 그치고 있기 때문에 대용량 말뭉치 구축에서 발생할 수 있는 다양한 예외를 설명하지 못하는 한계가 있다[7].

Kim의 연구에서는 언어분석 모듈을 위한 학습 말뭉치를 구축하기 위하여, 다수의 자동 형태소 분석 모듈이 제시한 결과를 참조하고 오류를 분류하여 주요 유형을 반자동으로 보정함으로써 수작업을 최소화하는 방법을 제안하였다. CoNLL-U 형식으로 348,229문장, 총 9,455,930어절 규모의 한국어 형태소 분석 말뭉치를 구축하였다. 그러나 구문분석을 포함하지 않았고, 규모도 기존에 배포되고 있는 형태소분석 말뭉치와 비슷하다는 한계가 있다[8].

본 논문에서는 기존의 형태소 분석 수준의 반자동 말뭉치 구축 방법론을 의존 구문분석 단계까지 확장하고자 한다. 또한 영어권 언어 중심의 UD 의존관계를 한국어에 적용할 때 발생하는 문제점을 해결하기 위하여 기존의 연구를 참조하여 일관된 규칙을 제안하고 실제 말뭉치 구축과정에 적용하고자 한다.

III. KCC 원시말뭉치

영어의 경우에는 자연언어처리를 위한 목적으로 현대 영어를 기준으로 매년 2천만 어절 규모로 총 5억 6천만 어절의 원시말뭉치를 구축하고 있다. 구글과 마이크로소프트는 웹 데이터에 대한 대규모 n-gram 말뭉치를 구축하여 자연언어처리 연구에 활용하도록 배포되고 있다. 또한, 최근 자연언어처리 연구에서는 대용량 말뭉치를 기반으로 학습된 딥러닝 연구들이 진행되고 있으며, 대표적인 모델은 BERT[9]와 GPT-3[10]가 있다. BERT의 경우 학습에 사용된 말뭉치는 위키피디아 말뭉치와 책 말뭉치로, 위키피디아 말뭉치는 25억 개의 단어의 규모이며 책 말뭉치는 8억 개의 단어 규모를 가진다. GPT-3의 학습에 사용된 말뭉치는 총 5종류로, 위키 말뭉치, 책 말뭉치 2종류, 웹 텍스트 말뭉치, 그리고 크롤링 말뭉치이다. 이 말뭉치들의 총합은 4,990억 개의 단어에 이른다.

반면에 한국어는 대용량 말뭉치의 부재로 영어권에서 진행되는 연구만큼 대용량 말뭉치를 활용한 연구가 어렵다. 이와 같은 필요성을 인식하고 국민대학교에서 KCC 원시말뭉치를 구축하여 공개하였다. KCC 원시말뭉치는 뉴스 기사 문장들을 수집하여 구축한 말뭉치로, 표 1과 같이 총 4개의 말뭉치 약 4천9백만 문장, 7억4천만 어절로 구성되어 있다.

KCC 원시말뭉치의 전처리 과정에서 그림 1과 같이 어절 단위 이외에 추가로 ‘,’, ‘.’, ‘?’, ‘!’ 등 문장부호를 독립된 토큰으로 분리하였고, KCC940, KCCq28, KCC460 말뭉치의 경우 일부 인용문에서 인용부호 불일치 문제가 발생하여, 인용문에서 내포문장을 추출하여 단문으로 변경하거나, 내포 문장 추출이 불가능한 문장은 분석 대상에서 제외하였다. 전처리 후 표 1과 같이 약 4천4백만 문장, 6억5천만 토큰으로 구성된 구축대상 말뭉치를 준비하였다.

표 1. 전처리 전후의 KCC 원시말뭉치 통계
Table 1. Statistics of KCC raw corpus before and after preprocessing

Corpus	Before preprocessing		After preprocessing	
	Tokens	Sentences	Tokens	Sentences
KCC150	150,705,457	11,961,347	166,904,509	11,961,347
KCC940	93,210,333	6,263,454	81,123,597	5,607,899
KCCq28	28,785,803	1,337,721	18,113,978	1,355,280
KCC460	466,971,438	29,316,426	385,722,589	25,661,405
Total	739,673,031	48,878,948	651,864,673	44,585,931

그림 1. 전처리 단계에서 문장부호 토큰 분리 예제
Fig. 1. An example of token seperation in preprocessing phase

IV. 반자동 말뭉치 구축 단계

형태소 및 구문 부착 말뭉치 반자동 구축 단계는 그림 2와 같다. KCC 원시말뭉치에 전처리 작업을 거친 후 ETRI와 울산대의 형태소 및 구문 분석기를 적용하였다[11][12]. 자동 형태소 분석 결과를 규칙 및 예제 기반으로 정규화한 후 두 분석기의 결과가 동일한 문장을 추출하였고, 이후 구문분석 결과의 일관성 오류를 수정한 후 비교를 통해 최종적으로 CoNLL-U 형식의 형태 및 구문분석 말뭉치를 반자동으로 구축하였다. 두 언어분석기의 결과에서 차이가 많이 발생하는 패턴을 수집하고, 수작업으로 검토하여 형태소 정규화 규칙과 구문분석 일관성 오류 수정 규칙을 작성하였다.

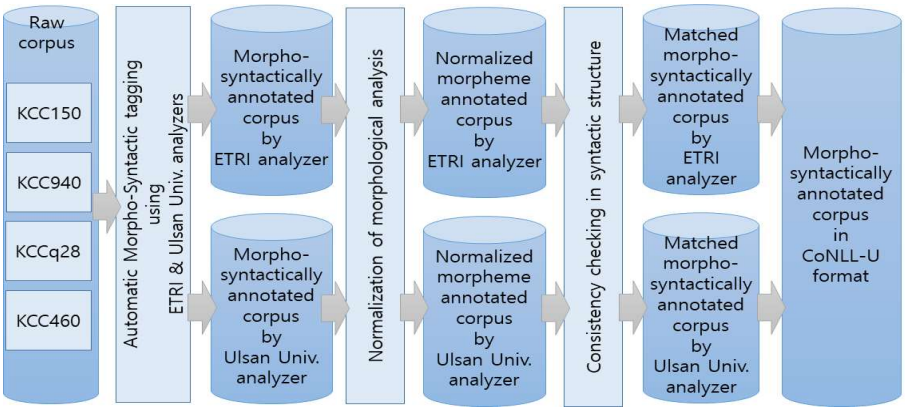

그림 2. 형태 및 구문 부착 말뭉치 반자동 구축 단계
Fig. 2. Semi-automatic process of constructing morpho-syntactically annotated corpus

4.1 형태소 분석 결과 정규화 및 비교

말뭉치 전처리 후 ETRI 형태소 분석기와 울산대 형태소 분석기를 이용하여 자동으로 분석한다. 형태소 분석 결과에서 형태론적으로는 타당하지만, 분석기의 특성으로 인해 서로 다른 결과가 생성되는 경우가 있기 때문에 규칙기반과 예제기반 방법을 적용하여 형태소 분석결과를 정규화한 후 비교한다.

4.1.1 규칙기반 형태소 분석 정규화

규칙기반 형태소 분석 정규화는 표 2의 예시와 같이 형태론적으로 문제가 없을 경우 개별 형태소의 길이가 길어지도록 결합하여 형태소 분석 복잡도를 낮추는 방향으로 진행한다.

첫 번째 규칙은 용언 파생 규칙으로, '명사+용언 파생접미사' 또는 '명사+용언'으로 분석되는 경우 단일 용언으로 묶는다. 두 번째 규칙은 고유명사 파생 규칙으로, '일반명사+고유명사', '접사+고유명사', '고유명사+접사', '고유명사+수사'로 분석되는 경우 단일 '고유명사'로 묶는다. 세 번째 규칙은 첫 번째 규칙을 확장한 형용사 파생 규칙으로, '부사+형용사 파생접미사', '어근+형용사파생접미사'로 분석되는 경우 단일 '형용사'로 묶는다. 네 번째 규칙은 두 번째 규칙을 확장한 체언 파생 규칙으로, '보통명사+보통명사', '보통명사+명사파생접미사', '체언접두사+보통명사'를 단일 '보통명사'로 묶어서 제시하고, 또, '숫자+수사'는 단일 '수사'로 묶어서 제시한다.

추가적으로 일관성 유지를 위해 두 가지의 오류 유형을 수정한다. 첫 번째는 토큰 '및', '혹은', '또는', '즉'이 'MAG'(일반부사) 또는 'MAJ'(접속부사)로 분석되는 경우가 있는데, 'MAJ'로 일괄 변경하며, 두 번째는 연결어미 및 종결어미 분석 오류와 관련하여, 문장 중간에 나타나는 토큰이 'EF'(종결어미)일 때 다음 토큰이 기호인 경우는 제외하고 'EC'(연결어미)로 변경하였고, 문장 마지막 바로 앞 토큰이 'EC'일 때 다음 토큰이 'SF'(문장기호)인 경우만 'EF'로 변경하였다.

표 2. 규칙기반 형태소 분석 정규화 예시(NNG: 일반명사, NNP: 고유명사, NR: 수사, VV: 동사, VA: 형용사, XSV: 동사파생접미사, XSA: 형용사파생접미사, XSN: 명사파생접미사, XPN: 체언접두사, XR: 어근, SN: 숫자, MAG: 일반부사)
Table 2. Examples of normalization of morphological analysis

	Normalization rules	Examples
1	NNG+XSV → VV	공부 + 하 → 공부하
	NNG+VV → VV	존중 + 받 → 존중받
	NNG+XSA → VA	적정 + 하 → 적정하
	NNG+VA → VA	독기 + 어리 → 독기어리
2	NNP+NNP → NNP	경기 + 포천 → 경기포천
	NNP+NNG → NNP	양주 + 시청 → 양주시청
	NNG+NNP → NNP	더본 + 코리아 → 더본코리아
	XPN+NNP → NNP	반 + 스탈린주의 →반스탈린주의
	NNP+SN → NNP	갤럭시노트 + 7 →갤럭시노트7
3	MAG+XSA → VA	짜릿 + 하 → 짜릿하
	XR+XSA → VA	치밀 + 하 → 치밀하
4	NNG+NNG → NNG	입당 + 원서 → 입당원서
	NNG+XSN → NNG	위원 + 장 → 위원장
	XPN+NNG → NNG	초+강수 → 초강수
	SN+NR → NR	500+만 → 500만

4.1.2 예제기반 형태소 분석 정규화

동일한 토큰에 대해서 두 개의 분석기가 형태론적으로는 타당하지만 서로 다른 분석결과를 제시하였고, 단순 규칙으로 정규화하기 어려운 경우, 예제 단위로 전문가가 수작업으로 판단하여 한쪽 결과를 선택하는 방식으로 예제기반 형태소 정규화 규칙을 정의하였다. 누적빈도 97% 이상에 해당하는 분석 결과 불일치 품사 태그 패턴 1,000여 개를 대상으로 판단하였다. 표 3의 규칙 1번은 '예민해짐에'의 분석결과가 서로 다른 경우 형태소 개수가 적은 ETRI 결과를 기준으로 정규화한다. 향후 하나의 토큰에 대한 분석 결과의 품사 나열이 규칙 1과 동일한 경우, 선택된 패턴으로 정규화한다. 규칙 2번은 ETRI의 분석결과에 오류가 있기 때문에, 향후 두 형태소 분석 결과의 품사 패턴이 이와 같을 경우 정규화하지 않는다.

표 3. 예제기반 정규화 규칙 예시
Table 3. Normalization rules based on analysis examples ([E]: ETRI analysis result, [U]: Ulsan Univ. analysis result)

1	- Chose the result of concise language analysis
	예) 예민해짐에 [E]예민하/VA+어/EC+지/VX+ㅁ/ETN+에/JKB [√] [U]예민/NNG+하/XSA+여/EC+지/VX+ㅁ/ETN+에/JKB [x]
2	- Normalization is not applied when linguistically different analysis results are presented.
	예) 아물지를 [E] 아물지/NNP+를/JX [x] [U] 아물/VV+지/EC+를/JX [x]

4.1.3 형태소 분석 일치 말뭉치 구축

형태소 분석 결과 정규화 이후 포함된 모든 토큰의 분석 결과가 일치하는 문장을 추출하였다. 표 4는 정규화 전과 후의 형태소 분석 결과가 일치하는 토큰과 문장의 개수와 비율을 보여준다. 표 1의 전처리 후 토큰 및 문장수와 비교하였다. 정규화 전에는 18.22%의 문장만 일치하지만 정규화 이후에는 74.96%의 문장의 형태소 분석 결과가 일치함을 알 수 있다.

표준 형태소 분석 말뭉치를 적용한 자동 평가가 어렵기 때문에 본 논문에서는 소규모의 수작업 평가를 진행하였다. 형태소 분석이 일치하는 문장 중에서 문장길이 별로 임의로 100문장을 선택하여 수작업으로 정확도를 평가하였다. 표 5와 같이 토큰 기준 정확도는 99.12%, 문장 기준 정확도는 92.00%를 달성하였다. 표 6은 형태소 분석 결과의 오류 예제를 보여주고 있다. 첫 번째 문장 1번 토큰 '목소리'는 일반명사이므로 'NNP'에서 'NNG'로 바뀌어야 한다. 두 번째 문장 5번 토큰의 '로비'는 인명이므로 'NNG'에서 'NNP'로 바뀌어야 한다.

표 4. 형태소 분석 정규화 전후 분석 일치 토큰 및 문장 수
Table 4. Number of matching tokens and sentences before and after normalization of morphological analysis results

	Before preprocessing		After preprocessing	
	Tokens	Sentences	Tokens	Sentences
KCC150	23,227,819	2,242,867	117,849,839	8,875,652
KCC940	10,195,189	928,976	56,938,877	4,157,230
KCCq28	2,875,703	277,697	13,419,423	1,050,866
KCC460	11,890,458	1,111,334	275,246,223	19,341,671
Total	36,298,711 (13.63%)	3,449,540 (18.22%)	463,454,362 (71.09%)	33,425,419 (74.96%)

표 5. 형태소 부착 말뭉치의 정확도
Table 5. Accuracy of morpheme annotated corpus

Num. Tokens	~10	~15	~20	~25	26~	Total (prec.:%)
Eval. sentences	41	34	15	5	5	100
Correct sentences	39	32	13	4	4	92 (92.00)
Eval. tokens	319	430	250	112	140	1,251
Correct tokens	317	427	246	111	139	1,240 (99.12)

표 6. 형태소 분석 오류 예시
Table 6. Examples of morphological analysis error

ID	FORM	LEMMA	XPOS	Error correction
1	목소리는	목소리+는	NNP+JX	NNG+JX
2	늘	늘	MAG	
3	온화하게	온화하+게	VA+EC	
4	유지했고요	유지하+었+고+요	VV+EP+EF+JX	
5	.	.	SF	

ID	FORM	LEMMA	XPOS	Error correction
1	홍길동	홍길동	NNP	
2	감독은	감독+은	NNG+JX	
3	가드인	가드+이+ㄴ	NNB+VCP+ETM	
4	리처드	리처드	NNP	
5	로비를	로비+를	NNG+JKO	NNP+JKO
6	뽑는	뽑+는	VV+ETM	
...	...	...	...	...

4.1.4 문맥기반 형태소 분석 말뭉치 자동 검증

형태소 분석 말뭉치의 검증은 구축만큼의 인력, 비용, 시간이 소요된다. 검증의 일관성과 신뢰성 등의 문제를 완화하기 위해 다양한 방법이 제안되고 있다. 본 논문에서는 Choi의 연구를 참고하여 형태소 분석 일치 말뭉치를 대상으로 문맥 기반 자동검증 방법을 적용한다[13]. 이 방법은 주어진 형태소에 대하여 문맥이 동일할 경우 동일한 품사가 부착되어야 한다고 가정한다.

주어진 형태소가 동일한 문맥 C에서 여러 개의 품사 태그 t_1, t_2, t_3, ..., t_n이 주어졌을 때, 각 품사 태그의 조건부 확률 $p_a = P(t_a|C)$를 계산하고, 이 확률을 내림차순으로 정렬한다. 부착된 품사의 확률 p_a와 가장 높은 확률 p_1의 차이가 기준값 Θ_1보다 클 경우 p_a에 해당하는 품사를 오류로 간주한다. 이 때 문맥이란 특정 형태소에 대해 앞뒤 n개에 해당하는 형태소 표상과 품사 표상이 연접한 것을 의미한다. 그림 3은 가설을 개념적으로 설명하고 있다. 실제 검증 과정에서 하나의 형태소를 기준으로 앞뒤 한 개의 형태소와 품사를 문맥으로 정의하였고, Θ_1은 0.01로 지정하였다.

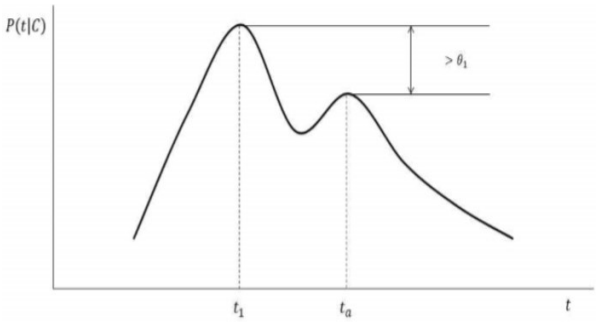

그림 3. 문맥기반 품사 검증에 대한 개념적인 그래프
Fig. 3. Conceptual graph explaining POS verification based on contexts

그림 4는 검증에서 자주 나타나는 오류 패턴의 일부를 설명한다. (a)에서 형태소 '다'의 앞뒤 문맥이 동일할 때 '다/EF(종결어미)'의 확률이 0.999977, '다/MAG(일반부사)'의 확률이 0.000023으로 $p_1 - p_a > \Theta_1$가 성립하여 '다/MAG'를 오류로 판단한다. (b)에서 형태소 '내'의 문맥이 동일할 때 '내/VX(보조용언)'의 확률이 0.982655, '내/VV(동사)'의 확률이 0.017345으로 $p_1 - p_a > \Theta_1$가 성립하여 '내/VV(동사)'를 오류로 판단한다. (c)에서 형태소 '과'의 문맥이 동일할 때 '과/JKB(부사격조사)'의 확률이 0.500000, '과/JC(접속조사)'의 확률이 0.500000으로 $p_1 - p_a < \Theta_1$가 성립하여 두 품사 모두 오류로 판단하지 않는다.

C_{i-1}	t_i	C_{i+1}
없/VA	다/EF	./SF
없/VA	다/MAG	./SF

ID	FORM	LEMMA	UPOS	XPOS
1	구체적으로	구체적+으로	NOUN	NNP+JKB
2	확인된	확인되+ㄴ	VERB	VV+ETM
3	부분이	부분+이	NOUN	NNG+JKS
4	없다	없+다	ADJ	VA+EF
5	.	.	PUNC	SF

(a) 형태소 '다'의 품사 검증
(a) POS verification of the morpheme 'da'

C_{i-1}	t_i	C_{i+1}
어/EC	내/VX	ㄴ/ETM
어/EC	내/VV	ㄴ/ETM

ID	FORM	LEMMA	UPOS	XPOS
1	독일	독일	PROPN	NNP
2	,	,	PUNC	SP
3	스웨덴	스웨덴	PROPN	NNP
4	등의	등+의	NOUN	NNB
5	나라도	나라+도	NOUN	NNG+JX
6	모두	모두	ADV	MAG
7	사회적	사회적	DET	MM
8	합의를	합의+를	NOUN	NNG+JKO
9	통해	통하+어	VERB	VV+EC
10	노동개혁을	노동개혁+을	NOUN	NNG+JKO
11	이뤄낸	이루+어+내+ㄴ	VERB	VV+EC+VX+ETM
12	것	것	NOUN	NNB

(b) 형태소 '내'의 품사 검증
(b) POS verification of the morpheme 'nae'

C_{i-1}	t_i	C_{i+1}
일가족/NNG	과/JKB	함께/MAG
일가족/NNG	과/JC	함께/MAG

ID	FORM	LEMMA	UPOS	XPOS
1	이후	이후	NOUN	NNG
2	일가족과	일가족+과	NOUN	NNG+JKB
3	함께	함께	ADV	MAG
4	양강도	양강도	PROPN	NNP
5	지역	지역	NOUN	NNG
6	농장원으로	농장원+으로	NOUN	NNG+JKB
7	배치됐다	배치되+었+다	VERB	VV+EP+EF
8	.	.	PUNC	SF

(c) 형태소 '과'의 품사 검증
(c) POS verification of the morpheme 'gwa'

그림 4. 문맥 기반 품사 검증 예시
Fig. 4. Examples of POS verification based on contexts

4.2 구문분석 결과 일관성 오류 수정 및 비교

의존관계 구문분석에서는 단어 사이의 '의존관계'를 표현하여 문장의 구조를 분석한다. 그림 5에서 화살표는 하나의 의존관계를 표현하고, 화살표 쪽 단어는 의존소(Dependent) 그 반대쪽 단어는 지배소(Governor)이다. 화살표 위의 레이블은 의존관계의 유형을 나타낸다. 예를 들어, 의존관계 ['chased' ->(nsubj)->'cat']에서 'chased'는 지배소, 'cat'은 의존소, 'nsubj'는 의존관계 '체언형 주어'를 나타낸다. 일반적으로 의존소는 하나의 지배소만을 가지고, 지배소는 여러 개의 의존소를 가질 수 있다.

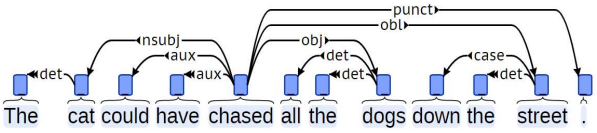

그림 5. UD 방식으로 구문분석한 영어문장 예시
Fig. 5. An example of UD analyzed English sentence

앞 단계에서 반자동으로 구축한 형태소 부착 말뭉치를 대상으로 ETRI, 울산대 자동 구문 분석 결과를 비교하여 의존관계 구문분석 말뭉치를 구축한다. 두 구문분석기는 기계학습 기반으로 개발되었기 때문에 비슷한 문장에 대해서 서로 다른 분석 결과를 도출하는 경우도 있다. 따라서 자주 발생하는 의존관계 오류에 대해서 규칙을 이용하여 일관성 있게 수정한 후 비교한다.

4.2.1 규칙기반 의존관계 일관성 수정

다음과 같은 세 가지의 규칙을 이용하여 의존관계의 일관성 오류를 수정하였다.

[규칙 1] 연속하는 2개의 토큰 t_i, t_{i+1}이 고정된 표현(fixed expressions)을 나타내는 경우, t_i는 t_{i+1}만을 지배소로 취하고 반대로 t_{i+1}은 t_i만을 의존소로 연결한다. 만약 자동 분석 결과에서 t_{i+1}이 t_i 이외에 전방의 다른 토큰 t_{i-k}를 의존소로 가지는 경우, t_{i-k}의 지배소를 t_i의 지배소 t_{i+j}로 수정한다.

표 7은 연속된 2개 토큰으로 구성된 고정된 표현의 예시를 보여준다. 1번-5번은 고정된 표현에 대한 의존규칙이고, 6번은 본용언-보조용언에 대한 의존규칙이다.

그림 6은 오류 수정의 예를 보여 준다. 자동분석 결과 중에서 '의원에 대해'에서 '대해'는 '의원에'만을 의존소로 가진다. 자동분석결과에서 '대해'가 추가적으로 '대표가'를 의존소로 가지는 경우, '대표가'의 지배소를 '대해'의 지배소 '들이댄다'로 변경한다.

[규칙 2] 3개의 연속된 토큰 t_i, t_{i+1}, t_{i+2}이 고정된 표현을 나타내는 경우, t_i는 t_{i+1}을 지배소로 가지고, 연속해서 t_{i+1}은 t_{i+2}를 지배소로 가진다. 반대 방향으로 t_{i+1}은 t_i만을 의존소로 가지고, t_{i+2}는 t_{i+1}만을 의존소로 가진다.

표 8은 연속된 토큰 3개의 고정된 표현을 나타내는 경우, 의존관계 설정 규칙을 보여준다.

그림 7은 오류가 수정된 예를 보여 준다. '실현할 수 있는' 표현을 ['실현할' <- (VP_MOD) <- '수' <- (NP_SBJ) <- '있는'] 구조로 분석하였다.

표 7. 연속된 토큰 2개의 고정된 표현에서 의존관계 규칙
Table 7. Dependency rules of fixed expressions in two consecutive tokens

	Dependent(t_i)	Rel.	Governor(t_{i+1})
1	~을/를	OBJ	통하+어, 위하+어,필두+로
2	~만/~기	OBJ	위하+어, 위하+ㄴ
3	~에	AJT	대하+어, 관하+어, 따르+아
4	~로/으로	AJT	인하+어, 인하+ㄴ, 말미암+아
5	~와/과	AJT	마찬가지+로
6	VV/VA (본용언)	VP	VX (보조용언)

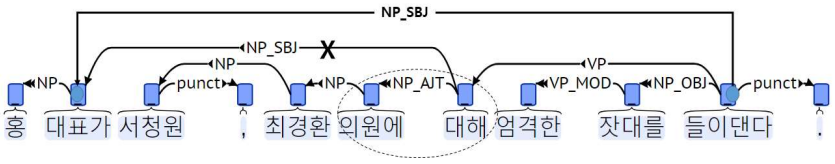

그림 6. 의존관계 일관성 오류 수정 예시
Fig. 6. An example of consistency error correction in dependency relation

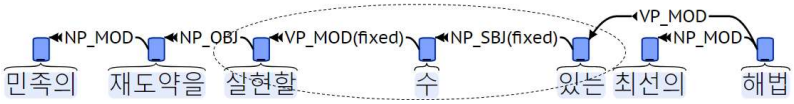

그림 7. 의존관계 일관성 오류 수정 예시
Fig. 7. An example of consistency error correction in dependency relation

표 8. 연속된 토큰 3개의 고정된 표현에서 의존관계 규칙
Table 8. Dependency rules of fixed expressions in three consecutive tokens

Pattern	Dep.(t_i)	Rel.	Gov./Dep(t_{i+1})	Rel.	Gov.(t_{i+2})
~ㄹ 수 있/없	~ㄹ	VP_MOD	수	NP_SBJ	있/없
~ㄴ 것 같	~ㄴ	VP_MOD	것	NP	같

[규칙 3] 일반적으로 한국어에서는 ‘,’, ‘.’, ‘?’, ‘!’ 등의 문장부호가 앞 토큰에 붙어서 나타나지만, 본 논문에서는 문장부호를 독립된 토큰으로 분리하였기 때문에 문장부호에도 의존관계를 부여하여야 한다. 한국어의 의존 구문분석은 지배소 후위의 원칙을 지키지만, 문장부호는 앞 토큰에 의존하게 하여 지배소 후위 원칙의 예외로 인정한다. 그림 6에서 ‘.’은 앞 토큰 ‘들이댄다’를 지배소로 가지고, ‘,’도 앞 토큰 ‘서청원’을 지배소로 연결한다.

4.2.2 의존관계 일치 말뭉치 구축

형태소분석결과 일치 문장을 대상으로 의존관계 일관성 오류를 수정 후, 서로 비교하여 의존관계 일치 토큰 및 문장을 추출하였다. 표 9에서 의존관계가 일치하는 토큰 및 문장의 수와 표 1의 전처리 후 토큰 수, 문장 수 대비 일치하는 토큰 수, 문장 수의 비율을 제시한다. 전체 문장 중 18.65%의 문장을 형태 및 구문부착 말뭉치로 구축하였다. 국립국어원 모두의 말뭉치에서 배포하는 300만 어절 규모의 구문분석 말뭉치와 비교하면 약 27배 큰 규모이고, GitHub 사이트를 통해 공개하고 있다[14].

의존관계가 일치하는 문장을 토큰의 길이에 따라 구분해 보면, 표 10과 같이 길이가 짧은 문장이 다수 추출되었고, 토큰 길이가 6~15인 문장이 전체의 85% 정도를 차지한다. 의존구조가 일치하는 100문장을 임의로 선택하여 수작업으로 평가하였다.

표 11에서 의존관계 단위 평가에서 UAS Unlabeled Attachment Score) 기준으로 99.39%, LAS(Labeled Attachment Score) 기준으로 98.91% 정확도를 보였다. 문장 단위 평가에서 UAS, LAS 기준으로 각각 95.0%, 91.0% 정확도를 보였다.

구문분석 오류의 예시를 보면, 그림 8에서 ‘속’의 지배소는 ‘미니멀리즘적’에서 ‘방식처럼’으로 수정되어야 한다.

표 9. 구문 분석 결과가 일치하는 토큰 및 문장 수
Table 9. Number of tokens and sentences that the dependency relation matches

Corpus	Tokens	Sentences
KCC 150	20,842,946	2,166,162
KCC 940	9,275,574	901,492
KCC q28	3,078,657	309,314
KCC 460	48,339,449	4,939,681
Total (Matched ratio)	81,536,626 (12.50%)	8,316,649 (18.65%)

표 10. 의존관계 일치 문장의 길이 분포
Table 10. Length distribution of sentences that the dependency relation matches

Len. sentences (Num. tokens)	Sentences	Ratio (%)
1~5	461,169	5.55
6~10	5,019,656	60.36
11~15	2,061,366	24.79
16~20	667,710	8.03
21~25	90,232	1.08
26~30	12,151	0.15
31~	4,365	0.04
계	8,316,649	100.00

표 11. 구문 일치 문장 정확도 평가
Table 11. Accuracy of sentences that the dependency relation matches

Num. tokens	Sent.	Rel.	UAS		LAS	
			Correct sent.	Correct rel.	Correct sent.	Correct rel.
~10	71	444	69	442	66	439
11~15	18	204	17	203	17	203
16~20	11	175	9	173	8	172
Total (Acc.)	100	823	95 (95.00)	818 (99.39)	91 (91.00)	814 (98.91)

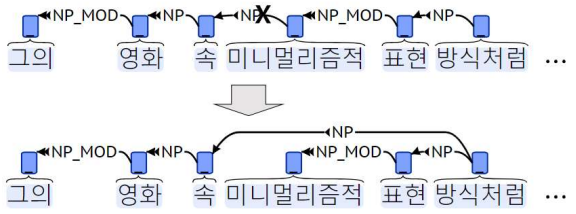

그림 8. 의존관계 오류 예시
Fig. 8. An example of dependency relation error

V. 결 론

본 논문에서는 약 6억5천만 토큰 규모의 한국어 원시말뭉치를 대상으로 기존에 개발된 다수의 형태, 구문분석기와 전문가가 검증한 규칙을 적용하여 반자동으로 약 8,300,000문장 81,500,000토큰 규모의 대용량 형태소 및 구문 분석 말뭉치를 구축하였고, GitHub 사이트에 공개하였다. 기존의 '국립국어원 모두의 말뭉치'에서 제공하는 3,000,000어절 규모의 형태소 및 구문분석 말뭉치와 비교하면 약 27배 정도 큰 규모의 대용량의 말뭉치를 효율적으로 구축한 것이다. 형태소 부착 말뭉치에 대한 수작업 평가에서 토큰 기준 99.12%의 정확도를 보였으며, 구문부착 말뭉치의 경우 LAS와 UAS 모두 토큰 기준 98.00% 이상의 정확도를 보였다. 향후 구축된 말뭉치를 이용하여 자동 형태소 분석 및 구문분석 시스템을 학습하고 성능을 평가할 계획이다.

References

[1] Korean Corpus of National Institute of Korean Language, https://corpus.korean.go.kr. [accessed: Jul. 25 2021]

[2] AI Hub, https://aihub.or.kr. [accessed: Jul. 25, 2021]

[3] Universal Dependencies, https://universaldependencies.org/. [accessed: Jul. 25, 2021]

[4] Korean Contemporary Corpus, http://nlp.kookmin.ac.kr. [accessed: Jul. 25, 2021]

[5] CoNLL-U Format, https://universaldependencies.org/format.html. [accessed: Jul. 25, 2021]

[6] J. Chun, N. Han, J. D. Hwang, and J. D. Choi, "Building Universal Dependency Treebanks in Korean", In Proceedings of the 11th International Conference on Language Resources and Evaluation, of LREC18, Miyazaki, Japan, pp. 2194-2202, May 2018.

[7] C. Lee, T. Oh, and H. Kim, "A Study on Universal Dependency Annotation for Korean", Language Facts and Perspectives, Vol. 47, pp. 141-175, May 2019.

[8] T. Y. Kim, P. M. Ryu, H. Kim, and H. J. Oh, "Unified Methodology of Multiple POS Taggers for Large-scale Korean Linguistic GS Set Construction", Journal of KIISE, Vol. 47, No. 6, pp. 596-602, Jun. 2020, http://dx.doi.org/10.5626/JOK.2020.47.6.596.

[9] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding", In Proceedings of NAACL 2019, Minneapolis, Minnesotapp, 4171-4186, Jun. 2019. http://dx.doi.org/10.18653/v1/N19-1423.

[10] T. B. Brown et. al., "Language Models are Few-Shot Learners", 4th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada., Vol. 4, Jul. 2020, https://arxiv.org/abs/2005.14165.

[11] ETRI Public AI Open API·Data service, http://aiopen.etri.re.kr. [accessed: Jul. 25, 2021]

[12] UTagger, http://nlplab.ulsan.ac.kr. [accessed: Jul.

25, 2021]

[13] M. Choi, C. Kim, H. Park, M. Cheon, H. Yoon, Y. Namgoong, J. Kim, and J. Kim, "Detecting Errors in POS-Tagged Corpus on XGBoost and Cross Validation", KIPS Transactions on Software and Data Engineering, Vol 9, No. 7, pp. 221-228, Jul. 2020, https://doi.org/10.3745/KTSDE.2020.9.7.221.

[14] BUFS-JBNU Corpus 2020, https://github.com/bufsnlp2030/BUFS-JBNUCorpus2020. [accessed: Jul. 25, 2021]

저자소개

류 법 모 (Pum-Mo Ryu)

2009년 : 한국과학기술원 전산학과 (공학박사)
2009년 ~ 2015년 : 한국전자통신연구원 지식마이닝연구실 책임연구원
2015년 ~ 현재 : 부산외국어대학교 사이버경찰학과 교수
관심분야 : 자연언어처리, 질의응답, 인공지능보안

오 효 정 (Hyo-Jung Oh)

2008년 : 한국과학기술원 컴퓨터공학과(공학박사)
2000년 ~ 2015년 : 한국전자통신연구원 지식마이닝연구실 책임연구원
2015년 ~ 현재 : 전북대학교 문헌정보학과 부교수
관심분야 : 정보검색, 텍스트마이닝, 빅데이