Pusan National University
Computer Science and Engineering
Technical Report 2022-04

세분화된 한국어 형태소 규칙에 기반한 의존구문분석 모델 개발

저자 1 201724443 김준기

저자 2 201724465 박기훈

저자 3 201724568 정대성

지도교수 권혁철

목 차

1. 서론

1.1. 연구 배경

의존구문분석에 대한 수요가 나날이 높아지고 있다. 구글의 BERT를 필두로 여러 의존구문분석 모델들이 여러 언어들의 시장의 파이를 나눠먹고 있는 상황인데 한국어 역시 BERT를 개선한 koBERT나 Klue/Roberta-Base(or Large), KoElectra-base 등의 의존구문분석 모델들이 자리를 차지하고 있다. 다만 이 모델들이 100% 완벽한 성능을 보여주는 것이 아니라 기존의 딥러닝 방식에서 다른 방식의 딥러닝 방식을 적용해 개선할 여지가 있다고 판단하여 해당 주제를 선정하게 되었다.

1.2. 기존 문제점

의존구문분석이란 자연어 문장을 지배소, 피지배소 의존 관계로 분석하는 구문 분석 방법론이다. 해당 방식을 통해 문장의 구조적 중의성을 해소 가능해 어순이 고정적이지 않고 문장 성분의 생략이 빈번한 한국어에 적합하다. 예로, '용감한 그의 아버지가 불길에 뛰어들었다.'라는 문장이 있으면, '용감한'이 '그'를 의미하는 것인지 아니면 '아버지'를 의미하는 것인지 모호해진다. 이때, 의존구문방식을 사용하면 '용감한'이 어느 단어의 피지배소인지 명확해져 의미를 확실하게 파악할 수 있게 된다.

이 방식은 분석 결과를 문장에서 차지하는 역할(예: 체언, 용언 등)인 구문 태그와 문장 내에서 지배소와 피지배소의 의존관계를 나타내는 기능 태그의 결합 형태인 의존관계 레이블(예: NP_SBJ(체언_주어), VP_MOD(용언_관형어))로 표현한다. 그림으로 예시를 나타내면 다음과 같다.

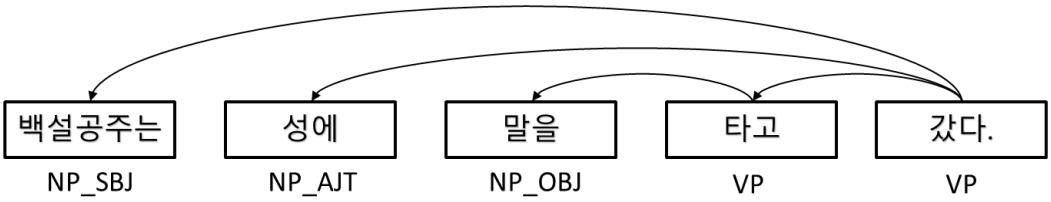


그림 1 의존구문분석 예시

그림 1의 예시에서 '백설공주는'과 '갔다'는 지배소와 피지배소의 관계이다. '백설공주는' 피지배소로서 '갔다'라는 지배소에게 의존적인 것을 표현한 것이다.

현재 의존구문분석은 전이학습(Transfer Learning: 사전학습 된 시스템에 추가 출력층을 활용하여 응용시스템을 구현한 것, 사전학습은 BERT 기반 Pretraining이며, 이와 반대되는 개념으로는 사후학습(Finetuning)이 있음)을 이용한 그래프기반 딥러닝 시스템이 주를

이루고 있다. 여기서 그래프기반 시스템은 각 노드 간의 에지(Edge)를 Score 형태로 출력하여 Score가 가장 높은 것을 최종 의존관계로 설정하는 것이다.

다만 이 방식에는 문제가 있는데, 데이터셋에 의존적인 방식이라 과적합(Overfitting) 문제가 발생한다는 것이다.

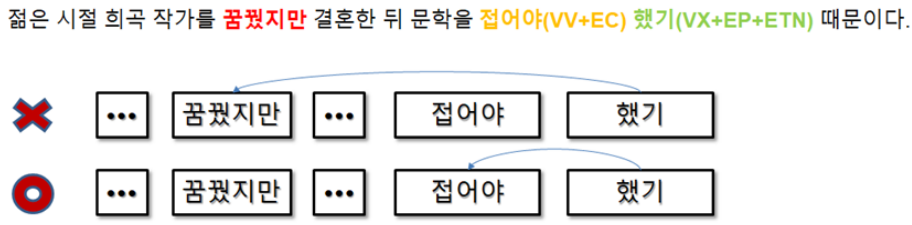


그림 2 과적합현상 예시

위의 그림 2의 경우, '했기'는 보조 용언이라 바로 앞의 '접어야'에 의존을 해야 하나 기존 시스템으로 분석할 경우 '꿈꿨지만'에 의존하는 과적합의 문제가 일어난 것을 볼 수 있다. 이러한 문제 외에도 데이터셋 구축에 많은 시간과 자원이 소요, 모델 학습에 많은 시간과 자원이 소요, 모델 예측 결과에 대한 제어 및 설명이 불가능하다는 문제들도 있다.

이 문제들을 해결하기 위해 부산대학교 AILAB 심층학습 및 규칙 결합 구문분석기를 사용했다. 해당 모델을 통해 심층학습 모델의 결과값을 언어학적 지식 기반 규칙으로 제어하였고, 최종 의존 관계 Score(Attention Score)를 0(위반 규칙)과 100(정답 규칙)으로 조정(위의 그림 2의 경우, Edge[꿈꿨지만 – 했기]는 Score = 0을 배정해 위반 규칙으로 보고, Edge[접어야 – 했기]는 Score =100을 배정해 정답 규칙으로 본다.)했다. 이를 통해 규칙을 통한 데이터셋 의존성(과적합 문제) 및 작은 데이터셋에 취약한 문제 등을 해결하였다.

그 결과, UAS(Unlabeled Attachment Score)가 96.28%, LAS(Labeled Attachment Score)가 93.19%가 나와 현재 한국어 의존구문분석의 SOTA(State-of-the-art: 현재 최고 수준의 결과를 가진 모델로, 현재 수준에서 가장 정확도가 높은 모델을 의미)를 달성함으로써 기존 시스템의 한계를 대부분 해결할 수 있었다. 다만, 해당 모델도 여전히 보완할 점은 남아 있었다.

1.3. 연구 목표

부산대학교 AILAB 심층학습 및 규칙 결합 구문분석기는 국립국어원의 Mecab 형태소 분석기를 기반으로 했는데 해당 분석기가 정확도가 93% 정도로 낮아 결과적으로 구문 분석 결과에서도 오류가 많고 세부 분류가 잘 일어나지 않는 모습을 보였다.

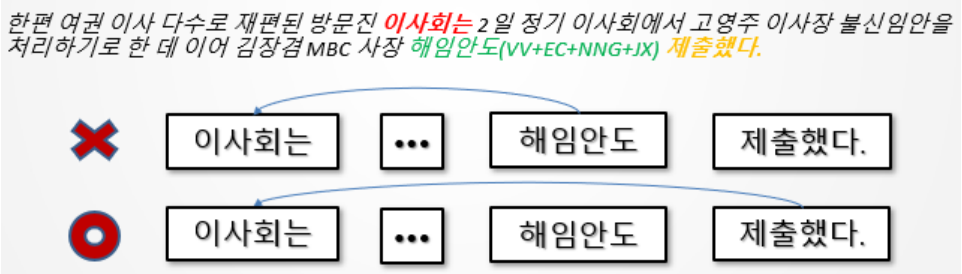


그림 3 형태소 에러 예시

그림 3의 예시에서 보면, '이사회'는 동사인 '제출했다'와 edge 관계를 맺어야 한다. 그러나 '해임안도'와 edge 관계를 맺고 있는 것을 볼 수 있는데 이는 해당 모델이 '해임안도'를 동사로 잘 못 해석하였기 때문이다. 이런 상황 때문에 더 높은 정확도의 형태소 분석기의 필요성을 느꼈다.

또한, 해당 모델은 국립국어원 모두의 말뭉치에만 실험을 해 특정 데이터셋에만 강건한 모습을 보일 가능성이 있기 때문에 다른 데이터셋의 실험 역시 필요함을 느꼈다.

2. 연구 내용

2.1. 구축 구문분석기 시스템

연구 목표에서 말한 점들의 개선 방안으로써 한국어 형태소 기반 KLTagger와 KLUE-DP 데이터셋을 사용하기로 했다.

2.1.1. 한국어 형태소 기반 KLTagger 사용

한국어 형태소 기반 KLTagger를 사용할 경우, 형태소들을 좀 더 세부적으로 나눌 수 있다. TTAS 표준이 형태소를 46개로 세분화한다면 KLTagger의 경우 형태소를 53개로 더 많이 세분화(동사는 일반동사, 자동사, 타동사, 자타동사로, 일반명사는 일반명사, 동작성명사, 상태성명사로, 의존명사는 일반의존명사, 단위성의존명사, 화폐단위)하기 때문에 더 높은 정확도를 보여준다. 이러한 형태소의 세분화는 앞서 본 그림 3과 같은 잘못된 형태소 분석이 일어나는 확률을 낮춰준다.

2.1.2. KLUE-DP 데이터셋 사용

데이터셋을 다양화하게 하기 위해 KLUE-DP를 사용했다. KLUE(Korean Language Understanding Evaluation)란 한국어 NLP(Natural Language Processing) 벤치마크 데이터셋으로 한국어 언어 모델 평가 및 비교를 위한 표준 데이터셋 구축을 목표로 개발되었다. Train은 10000문장, Dev는 2000문장(Test set 공개 X)의 분량으로 구성된 해당 데이터셋을 모델에 학습시켰다.

2.2. 전체 시스템 흐름도

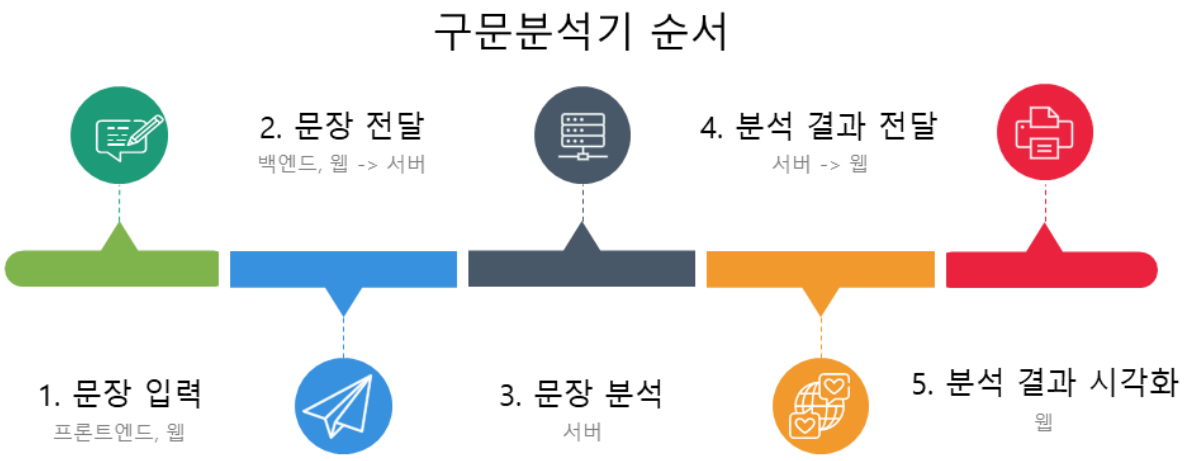


그림 4 구문분석기 순서도

목표로 한 구문분석기의 순서는 다음과 같다. 우선 사용자가 웹에 문장을 입력한다(프론트엔드: HTML, CSS). 해당 문장을 웹에서 서버로 전송한다(백엔드: Django). 서버에서 구문분석모델을 통해 문장을 분석하고, 분석 결과를 서버에서 웹으로 전달해 해당 결과를 그래프로 시각화하여 웹에 게시해 사용자에게 보여준다.

2.3. 구문분석모델

구문분석모델의 전체 구조도는 다음과 같다.

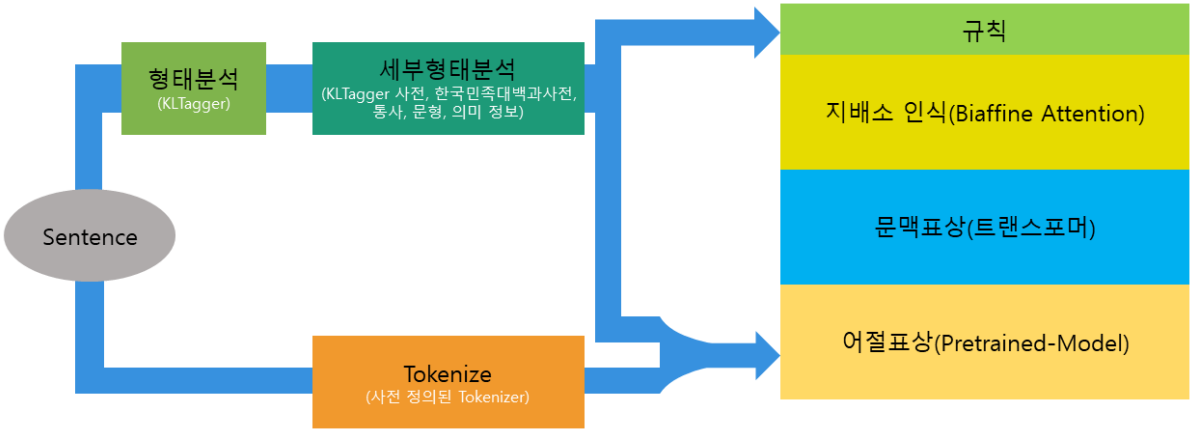


그림 5 구문분석모델 전체 구조도

구문분석모델이 작업을 수행하는 과정은 다음과 같다. 사용자가 입력한 문장이 형태소분석기 KLTagger를 통해 형태소 분석이 일어난다. 형태소 분석이 완료되면 해당 결과를 KLTagger 사전, 한국민족대백과사전, 통사, 문형, 의미 등의 정보를 활용해 세부형태분석 과정을 진행한다. 해당 결과는 추후에 설명할 규칙 제어 알고리즘에 입력값으로 들어가고 문장을 사전 정의된 Tokenizer를 이용한 Tokenize의 결과와 취합해 biaffine 기반 딥러닝 구문분석모델의 입력값으로 들어간다.

2.3.1. 구문분석 모델 구조

구문분석 모델의 세부 구조를 예시를 들어 설명하면 다음과 같다.

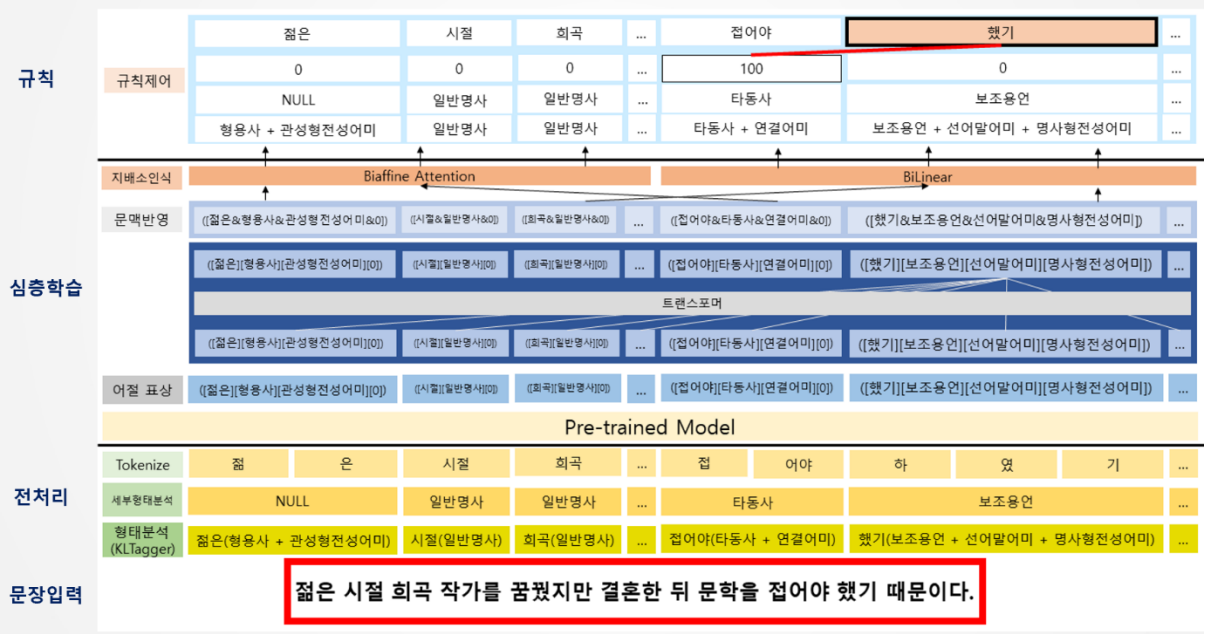


그림 6 구문분석모델 상세 구조도

'젊은 시절 희곡 작가를 꿈꿨지만 결혼한 뒤 문학을 접어야 했기 때문이다.'라는 문장을 입력하면, KLTagger에서 젊은(형용사 + 관성형전성어미), 시절(일반명사), 희곡(일반명사) 등으로 형태 분석을 진행한다. 규칙을 적용하기 위해 형태 분석 결과의 세부 자질을 한 번 더 분석한다. Tokenize된 결과를 Pre-trained Model에 넣어 나온 결과와 세부형태분석의 결과를 결합해 어절 표상의 결과를 얻어낸다. 어절표상은 트랜스포머를 통해 Attention을 활용하여 각 문장 내에서의 다른 어절들과의 관계를 학습해 문맥을 반영한 결과를 출력한다. 이후 해당 결과들을 각각 Biaffine과 BiLinear 연산을 통해 Attention Score를 계산해 형태소가 문장 내의 어떤 어절에 의존 관계를 가지는 지 Attention Score로 알려준다. 위 그림의 경우 '했기'는 보조 용언이기 때문에 바로 앞의 '접어야'에 Attention을 해야 한다. 그렇기에 문장 내의 어절 중 '접어야'만 Attention Score만 100으로 계산하고 나머지는 0으로 계산하여 Attention 대상을 '접어야'로 설정한다.

3. 연구 결과 분석 및 평가

학습 데이터, 평가 데이터는 KLUE-DP(Train 10000, Test 2000)로 고정하고 형태 분석 방식만 각각 KLUE 오픈소스(Baseline), 추가 모델링(형태소, 트랜스포머), KLTagger, KLUE 형태소 규칙, KLTagger 규칙 적용 모델로 다르게 하여 성능 비교를 해보았다(태깅 에러는 문장 평가에서 제외). 평가 메트릭은 클래스별 정확도의 평균, KLUE DP의 평가기준인 UAS/LAS Macro를 사용했다.

구분	학습데이터	평가데이터	Metric	UAS	LAS
(1) Baseline	KLUE-DP	KLUE-DP	Macro	93.48	87.82
(2) 추가형태소임베딩+트랜스포머	KLUE-DP	KLUE-DP	Macro	93.73	88.03
(3) KLTagger	KLUE-DP	KLUE-DP	Macro	**94.30**	**87.94**
(4) KLUE 형태소 규칙	KLUE-DP	KLUE-DP	Macro	94.22	88.39
(5) KLTagger 규칙	KLUE-DP	KLUE-DP	Macro	**94.88**	**88.48**

그림 7 모델 성능 비교(Macro)

위 그림에서 보듯이 KLTagger 규칙을 적용한 형태소 구분이 UAS와 LAS 각각에서 모두 SOTA를 달성함을 알 수 있었다.

한국어 형태소 기반 KLTagger 사용시 UAS가 0.57% 향상되었음(모델 2와 3을 비교)을 보아 한국어 형태소 세부 분석에서 Parser의 성능이 크게 향상되었음을 알 수 있다. 이는 곧 TTAS보다 더 세부적인 형태소 분석이 필요함을 알 수 있다.

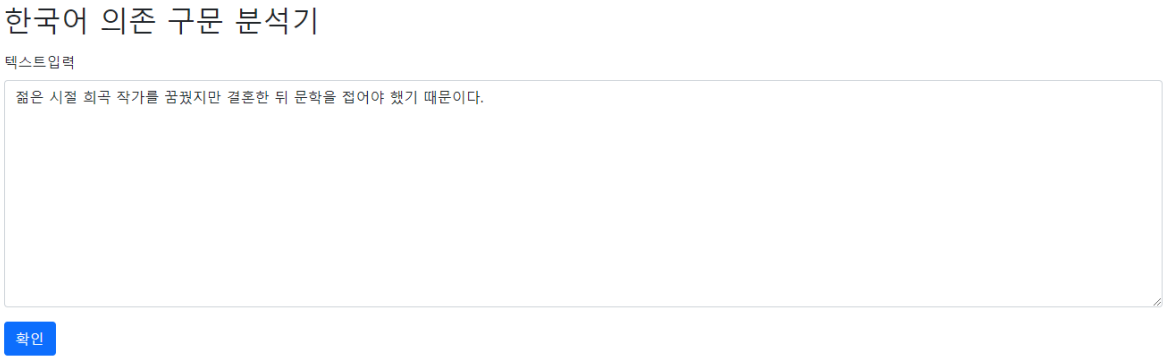
규칙 사용시 KLUE 형태소 0.74% 향상(모델 1과 4 비교), KLTagger는 1.4% 향상(모델 1과 5 비교)의 결과를 보아 형태소 분석에 상관없이 규칙이 효과적임을 알 수 있다.

이후, 평가방식만을 전체 정확도, 국립국어원 DP의 평가기준인 Micro로 바꾸고 해당 기준에서 SOTA인 건국대학교의 모델과 비교해 보았다.

구분	학습데이터	평가데이터	Metric	UAS	LAS
(1) 건국대학교모델	KLUE-DP	KLUE-DP	Micro	95.14	92.68
{2}KLUE 규칙	KLUE-DP	KLUE-DP	Micro	95.36	**93.66**
(3) KLTagger 규칙	KLUE-DP	KLUE-DP	Micro	**95.51**	93.49

건국대학교모델의 선행 연구결과보다 KLTagger 규칙보다 UAS는 0.37%, LAS는 0.81% 높은 성능(모델 1과 3 비교)의 결과를 보아 규칙을 적용하는 것이 효과적임을 알 수 있다. 다만 KLUE 규칙보다는 UAS가 0.15% 높았지만 LAS가 0.17%정도 낮았다.

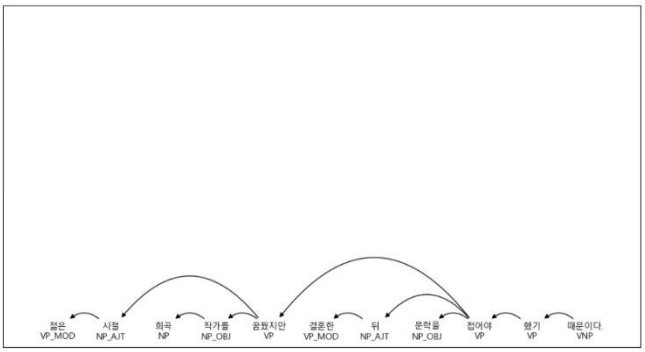
또한, 해당 구문분석모델을 웹으로 구현한 결과는 다음과 같다.



'젊은 시절 희곡 작가를 꿈꿨지만 결혼한 뒤 문학을 접어야 했기 때문이다.'라는 문장을 웹으로 구현한 한국어 의존 구문 분석기에 입력 후, 확인 버튼을 누르면 구문 분석 결과를 다음과 같이 출력한다.

입력문장 : 젊은 시절 희곡 작가를 꿈꿨지만 결혼한 뒤 문학을 접어야 했기 때문이다.

인덱스	단어형식	지배소인덱스	의존관계레이블
1	젊은	2	VP_MOD
2	시절	5	NP_AJT
3	희곡	4	NP
4	작가를	5	NP_OBJ
5	꿈꿨지만	9	VP
6	결혼한	7	VP_MOD
7	뒤	9	NP_AJT
8	문학을	9	NP_OBJ
9	접어야	10	VP
10	했기	11	VP
11	때문이다.	0	VNP



다시 분석하기

그림 8 웹 실행 결과

입력한 예시 문장과 문장의 구문분석결과를 표로 보여주고 또한 그래프로 시각화를 하고 보여줘 사용자의 이해를 돕기 쉽게 하였다.

4. 결론 및 한계점, 향후 연구 방향

위와 같은 결과를 통해 TTAS보다 더 세부적인 형태소 분석과 KLTagger 규칙의 방식이 더 높은 성능을 보여줌을 알 수 있었다. 단 여전히 개선할 점은 남아있었다.

모델의 경우 KLUE 추가 에러케이스 분석 및 규칙 작성을 통해 개선의 여지가 있다는 점, KLUE DP의 데이터셋에만 테스트를 진행하였으므로 국립국어원 모두의 말뭉치 KLTagger 분석 후 실험을 해야할 필요성을 느꼈다. 또한 학습 데이터는 동일하게, 테스트 셋은 다르게 실험을 해야 한다고 생각한다.

모듈의 경우 KLTagger 결과를 cmd를 통해 txt 파일로 추출하는 과정에서 특수문자 및 알파벳이 유니코드로 변환 저장되어 문장단위 비교가 불가능해 binary 값으로 변환 후 비교해야 하는 결과는 예외적으로 처리했다. 또한 Javascript 텍스트(http 및 www 포함된 문장)가 '<ELSS>'로 변경되는 경우는 '<ELSS>'를 http 등으로 치환했다.

서버의 경우 현재 로컬 도메인에서만 통신이 가능한데, 이는 이후 웹 서버를 구축한 후 공개할 예정이다.

5. 구성원별 역할 및 개발 일정

5.1.1. 구성원별 역할

학 번	성 명	구성원별 역할
201724443	김준기	모델 테스트 분석 결과 시각화 그래프 구현 보고서, 포스터, 시연 동영상 제작 Tech Week 시연 준비
201724465	박기훈	모델 구현 데이터 전처리 웹 소켓 구현 Tech Week 시연 준비
201724568	정대성	모듈 구현 데이터 전처리 웹 서버 구축 Tech Week 시연 준비

5.1.2. 개발 일정

5월			6월				7월					8월				9월			
15	22	29	5	12	19	26	3	10	17	24	31	7	14	21	28	4	11	18	25
딥러닝 스터디																			
			데이터 수집																
							데이터 전처리												
							품사 태깅												
										중간 보고서 작성									
												KLTagger 기반 사전학습모델 개발							
															모델 학습 및 최적화				
															웹 서버 구축				
																		최종 보고서 작성 및 발표	

6. 참고 문헌

논문 내용에 직접 관련이 있는 문헌에 대해서는 관련이 있는 본문 중에 참고문헌 번호를 쓰고 그 문헌을 참고문헌란에 인용 순서대로 기술한다. 참고문헌은 영문으로만 표기하며 학술지의 경우에는 저자, 제목, 학술지명, 권, 호, 쪽수, 발행년도의 순으로, 단행본은 저자, 도서명, 발행소, 발행년도의 순으로 기술한다.

[1] Michael A. Covington, "A dependency parser for variable-word-order languages," Research Reprot AI-1990-01, University of Georgia, 1990.
[2] J. Nivre, "An efficient algorithm for projective dependency parsing," Proc. Of IWPT, pp. 149-160, 2003.
[3] R. McDonald, K. Crammar, F. Pereira, "Online Large-margin Training of Dependency Parsers," Proc. Of ACL, pp. 91-98, 2005.
[4] Y.-H. Lee, J.-H. Lee, "Korean Parsing using Machine Learning Techniques," KIISE, Vol. 35, No. 1C, pp. 285-288, 2008. (in Korean)
[5] M. Choi, S. Jeong, H. Kim, "Dependency Structure Analysis and Dependency Label Annotation Using CRFs," Journal of KIISE, Vol. 41, No. 4, pp. 302-308, 2014. (in Korean)
[6] C. Lee, J. Kim, J. Kim, "Korean Dependency Parsing using Deep Learning," Proc. KIISE for HCLT, pp. 87-91, 2014. (in Korean)
[7] S.-H. Na, K. Kim, Y.-K. Kim, "Stack LSTMs for Transition-Based Korean Dependency Parsing, "KCC 2016, pp. 732-734, 2016. (in Korean)
[8] S.-Y. Hong, S.-H. Na, J.-H. Shin, Y.-K. Kim, "BERT and ELMo for contextualized word embeddings in Korean Dependency Parsing," KCC 2019, pp. 491-493, 2019. (in Korean)
[9] M. King, "Natural Language Parsing," pp. 58-87, Academic Press, 1983.
[10] H. Y. KIM, J. H. CHOI, S. J. LEE, "Improved Chart Parsing Algorithm based on Korean Syntactic Rules," KIISE, Vol. 17, No. 1, Apr. 1990. (in Korean)
[11] Y.-G. Hwang, H.-Y. Lee, Y.-S. Lee, "Using Syntactic Unit of Morpheme for Reducing Morphological and Syntactic Ambiguity," Journal of KIISE, Vol. 27, No. 7, pp. 784-793, 2000. (in Korean)
[12] M. Kim, S. Kang, J.-H. Lee, "Dependency Parsing by Chunks," KIISE, Vol. 27, No. 1B, pp. 327-329, Apr. 2000. (in Korean)
[13] S. K. Park, C. M. Jeong, J. M. Jo, S. J. Lee, "AnEffective Korean Syntatic Analyzer Using Longest Grouping Method," KIISE, Vol. 22, No. 1, pp. 961-964, Apr. 1995. (in Korean)
[14] H. Lee, "Korean Lexical Disambiguation using Tail-Head Co-occurrence Information," Journal for KIISE(B), Vol. 24, No. 1, pp. 82-89, 1997. (in Korean)
[15] Y.-M. Woo, Y.-I. Song, S.-Y. Park, H.-C. Rim, "Modification Distance Model for Korean Dependency Parsing Using Headible Path Context," Journal of KIISE, Vol. 34, No. 2, pp. 140-149, 2007. (in Korean)
[16] M.G. Jang, G.S. Yoon, and H.C kwon, "Korean Parsing System Based on Chart," KCC 1989.10, 571-574. (in Korean)

[17] J.-Ryu, "A rule-based Ambiguity resolution proposal for extensive Korean Parsing," Pusan National University Master's Thesis, 2018. (in Korean)
[18] A. Yoon, S. Hwang, E. Lee, H.-C. Kwon, "Construction of Korean Wordnet KorLex 1.5 ,"Journal of KIISE, Vol. 31, No. 1, pp. 92-108, 2009. (in Korean)
[19] S. T. Kim, M. H. Kim, H. C. Kwon "Rules-based Korean Dependency Parsing Using Sentence Pattern Information,"Journal of KIISE, Vol. 47, No. 5, pp. 488-495, 2020.
[20] C. E. Park, et al., "Korean Dependency Parsing with Multi-layer Pointer Networks," Proc. of the 29th Annual Conference on Human & Cognitive Language Technology, 2017.
[21] S. H. Na, et al., "Deep Biaffine Attention for Korean Dependency Parsing," Proc. of the KIISE Korea Computer Congress 2017, pp. 584-586, 2017. (in Korean)
[22] J.-H. Lim and H. Kim, "Korean Dependency Parsing using the Self-Attention Head Recognition Model," Journal of KIISE, Vol. 46, No. 1, pp. 22-30, 2019.
[23] C. Park, C. Lee, J.-H. Lim, and H.-k. Kim, "Korean Dependency Parsing with BERT," Proc. of the KIISE Korea Computer Congress (KCC) 2019, pp. 530-532, 2019. (in Korean)
[24] J. H. Han, Y. J. Park, Y. H. Jeong, I. K. Lee, J. W. Han, S. J. Park, J. A. Kim, and J. Y. Seo, "Korean Dependency Parsing Using Sequential Parsing Method Based on Pointer Network," Proc. of the 31th Annual Conference on Human & Cognitive Language Technology, pp. 533-536, 2019. (in Korean)
[25] J.-H. Lim and H. Kim, "Korean Dependency Parsing using Token-Level Contextual Representation in Pre-trained Language Model," Journal of KIISE, Vol. 48, No. 1, pp. 27-34, 2021.
[26] J. H. Lim, Y. J. Bae, H. K. Kim, Y. J. Kim, and K.C. Lee, "Korean Dependency Guidelines for Dependency Parsing and Exo-Brain Language Analysis Corpus," Proc. of the 27th Annual Conference on Human & Cognitive Language Technology, pp. 234-239, 2015. (in Korean)
[27] 국립국어원, "구문 및 무형 대용어 복원 말뭉치 연구 분석", 2021
[28] Gawlikowski, J., Tassi, C.R., Ali, M., Lee, J., Humt, M., Feng, J., Kruspe, A.M., Triebel, R., Jung, P., Roscher, R., Shahzad, M., Yang, W., Bamler, R., & Zhu, X. (2021). A Survey of Uncertainty in Deep Neural Networks. ArXiv, abs/2107.03342.
[29] J. M. Shin, S. H. Cho, S. R, Park "Neural network-based dependency parsing with rules applied"한국컴퓨터종합학술대회 논문집, 2022