강화된 지배소-의존소 제약규칙을 적용한 의존구문분석 모델 : 심층학습과 언어지식의 결합

신중민[01] 조상현[1] 박승렬[1] 최성기[1] 김민호[2] 김미연[3] 권혁철[1]
[1]부산대학교 정보융합공학과 [2]부산카톨릭대학교 소프트웨어학과 [3]주식회사 KT
tlswndals13@naver.com delosycho@gmail.com qkxk123@pusan.ac.kr view88@pusan.ac.kr
minho@cup.ac.kr miyeon.kim@kt.com hckwon@pusan.ac.kr

*Dependency parsing applying reinforced dominance-dependency constraint rule: Combination of deep learning and linguistic knowledge

JoongMin Shin[01] Sanghyun Cho[1] Seunglyul Park[1] Seongki Choi[1]
Minho Kim[2] Miyeon Kim[3] Hyuk-Chul Kwon[1]
[1]Dept. of Information Convergence Engineering Pusan National University,
[2]Dept. of Software, Catholic University of Pusan, [3]KT Corporation

요 약

의존구문분석은 문장을 의존관계(의존소-지배소)로 분석하는 구문분석 방법론이다. 현재 사전학습모델을 사용한 전이 학습의 딥러닝이 좋은 성능을 보이며 많이 연구되지만, 데이터셋에 의존적이며 그로 인한 자료부족 문제와 과적합의 문제가 발생한다는 단점이 있다. 본 논문에서는 언어학적 지식에 기반한 강화된 지배소-의존소 제약규칙 에지 알고리즘을 심층학습과 결합한 모델을 제안한다. TTAS 표준 가이드라인 기반 모두의 말뭉치로 평가한 결과, 최대 UAS 96.28, LAS 93.19의 성능을 보였으며, 선행연구 대비 UAS 2.21%, LAS 1.84%의 향상된 결과를 보였다. 또한 적은 데이터셋으로 학습했음에도 8배 많은 데이터셋 학습모델 대비 UAS 0.95%의 향상과 11배 빠른 학습 시간을 보였다. 이를 통해 심층학습과 언어지식의 결합이 딥러닝의 문제점을 해결할 수 있음을 확인하였다.

주제어: 의존구문분석, 딥러닝, 규칙 기반, 문형 정보

1. 서론

최근 초대용량 언어 자원의 확보와 기계학습, 딥러닝 알고리즘의 발전으로 언어처리 응용 시스템이 다양한 영역으로 확대되며 그 수요가 증가하고 있다. 언어처리 시스템 중 하나인 의존구문분석은 주어진 문장의 구성 성분을 의존관계(지배소-의존소)로 분석하는 구문분석 방법론이다[1]. 최근 구문분석에서 딥러닝 기반의 시스템이 좋은 성능을 보이며 많이 연구되고 있다.[20-25] 현재 딥러닝을 이용한 자연어 처리에는 사전학습모델을 기반으로 한 전이 학습(Transfer Learning) 방법론의 연구가 이루어지는데, 전이 학습에는 대용량의 unlabeled data를 활용해 데이터셋의 특성을 학습하는 자기지도학습(Self-Supervised Learning)의 사전학습(Pre-training)과 labeld data를 통해 학습시키는 지도학습(Supervised Learning)의 사후학습(Fine-tuning)이 활용된다. 그러나 이러한 지도학습의 딥러닝 모델은 정답 레이블을 통해 통계적으로 자질을 학습하므로 데이터셋에 의존적이라는 한계가 있다. 그렇기 때문에 데이터에 담기지 못한 정보는 학습할 수 없으며, 태깅에 에러가 있는 데이터로 학습과 평가를 한다면 학습과 예측에 오류를 발생시키는 과적합(overfitting)의 현상을 더욱 심화시킨다. 이러한 문제를 해결하기 위해 더 많은 데이터셋을 통한 학습과 더 좋은 품질의 데이터셋을 구축하는 것이 강조되고 있지만, 이를 위해선 더 많은 시간과 자원이 소요된다. 반면 규칙 기반 시스템은 언어학적인 분석과 구현에 많은 시간이 소요된다는 단점이 있지만, 데이터를 통한 자질 학습이 아닌 언어학적 지식을 기반으로 구현한 규칙을 사용하기 때문에 자료부족 문제(Data Sparsity problem)에 강건하며 데이터에 의존적이지 않다. 또한 학습을 위한 시간이나 컴퓨팅 자원이 필요하지 않으며, 모델의 출력에 대한 제어 및 설명이 가능하다는 이점이 있다.

본 논문은 언어 지식에 기반한 강화된 규칙 알고리즘으로 최종 attention score를 후처리 단계에서 제어하는 방식을 의존구문분석 딥러닝 모델에 적용하였다. 이를 위해 데이터셋 가이드라인[26] 및 보고서[27]와 규칙 기반 구문분석기[9-13,16-19]에 기반하여 지배소-

* 이 논문은 2022년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임 (No. 2013-2-00131, (엑소브레인-총괄/1세부) 휴먼 지식증강 서비스를 위한 지능진화형 WiseQA 플랫폼 기술개발)

의존소 관계 제어 알고리즘을 구현하였고, 이러한 결합을 통해 기존 의존 구문분석 시스템의 한계점을 극복하고 구문 분석의 성능이 향상됨을 보여주고자 한다. 또한 이를 통해 심층학습과 언어지식의 결합이 현재 딥러닝의 여러 문제점을 해결할 수 있음을 말하고자 한다.

논문의 구성은 다음과 같다. 2장에서 한국어 의존구문분석에 관한 연구에 대해 설명한다. 3장에서는 사용 데이터셋과 태깅에러에 대해 설명하고, 4장에서는 모델의 구조를 딥러닝과 규칙으로 나눈 뒤 딥러닝에 강화된 규칙을 적용하는 방법에 대해 설명한다. 5장에서는 실험환경과 4장의 규칙을 적용한 실험의 결과를 분석하여 기술하고, 6장은 결론 및 향후 연구에 관해 말한다.

2. 관련 연구

한국어 구문분석 연구는 크게 지배소-의존소 관계 자질을 규칙으로 정의한 규칙 기반 구문분석 [10-14], 대용량 구문분석 말뭉치에서 지배소-의존소 관계 자질을 통계적으로 추출하여 분석하는 통계/기계학습 기반 구문분석[4-5,14-15], 기본 자질(raw features)로부터 신경망 모델을 활용해 여러 단계를 걸쳐 복잡한 자질을 합성해 내는 딥러닝 구문분석[6-7,20-25]이 있다.

초기의 한국어 의존 구문분석은 규칙을 기반으로 한 차트 파싱 알고리즘 시스템이 연구되었다. 차트 파싱 알고리즘은 정점(vertex)과 에지(edge)에 의한 유방향 그래프로 표현한 파싱 알고리즘이다[9]. 그러나 차트 파싱 기반 시스템은 중의성을 가지는 문장에 대하여 모든 가능한 파스 트리를 생성하기 때문에 지배소-의존소 관계가 복잡해진다. [10]은 한국어 통사 규칙에 바탕을 두고 차트 파싱 알고리즘을 개선하여 불필요한 지배소-의존소 관계를 제거함으로 구문분석의 효율을 증가시켰다.

이후 지배소-의존소 관계의 복잡도를 최소화하기 위해 구 묶음(chunking)을 통한 연구가 진행되었다. 형태소 과생성을 유발하는 복합 동사구와 의존명사 등을 포함하는 어절에 대해 구문 형태소 단위로 처리하여 모호성을 줄이는 방법[11], 구문분석 단계 전에 수행될 명사구와 동사구 단위화 방법과 지배노드와 의존노드 사이의 범위를 제한하는 방법[12], 한국어 사전의 정보와 각 구문 사이의 문법 관계를 이용해 최장 묶음을 만드는 방법[13] 등 다양한 방법으로 연구되었다.

그 후 대량의 구문분석 말뭉치가 구축되면서 말뭉치로부터 유의미한 통계적 특성을 학습하는 기계학습 기반의 연구가 진행되었다. 기계학습은 사람이 디자인한 자질(feature)을 입력으로 받아 최고의 성능을 낼 수 있는 자질의 가중치(weight)를 찾아낸다. [14]의 어말-어두 공기 정보에 기초한 어휘 중의성 해소, [15]의 한국어의 지배 가능 경로 문맥을 이용한 수식 거리 확률 모델, [4]의 ME(Maximum Entropy)와 SVM(Support Vector Machine)을 이용한 결정적 구문분석, [5]의 주어나 목적어와 같은 의존 관계명 부착과 의존구조를 동시에 분석하는 CRFs 모델 등 다양한 연구가 이루어졌다. 그러나 기계학습은 사람이 직접 자질을 디자인해야 하므로 최적의 자질의 조합하는데 많은 시간과 노력 이 필요하다는 한계가 있다.

이러한 기계학습의 문제를 해결하기 위해 딥러닝 모델의 연구가 수행되었다. 딥러닝은 여러 은닉 계층(Hidden layer)을 포함하고 있는 신경망으로 비선형 변환(non-linear activation)의 조합을 통해 새로운 자질의 조합과 표현을 학습할 수 있는 장점이 있다. 딥러닝 기반 한국어 의존구문분석 연구는 크게 전이 기반 방식과 그래프 기반 방식으로 나뉜다. 전이 기반 의존 구문 분석은 입력(버퍼)과 스택으로부터 구문 분석 상태 표현을 얻고, 신경망을 이용하여 다음 전이 액션을 결정하는 방법이다. [6]는 스택과 버퍼의 형태소와 자질에 임베딩을 적용하고 피드 포워드 신경망(Feed-forward Neural Network)을 통해 전이 액션을 분류하였다. [7]는 스택, 버퍼 및 기존 액션 열에 Stack LSTM을 적용하여 단어 표상과 전이 액션의 효율을 향상시켰다. 그러나 전이 기반 딥러닝은 오류전파 현상이 크다는 한계가 존재한다.

그래프 기반 딥러닝은 입력 단어들에 대한 의존 관계의 점수(score)를 신경망 모델로 계산하는 방법이다. [20]은 어텐션(Attention)을 기반으로 포인터 네트워크 인코더-디코더(encoder-decoder) 모델을 의존 구문분석에 적용하였다. [21]는 어절 표현을 위하여 각 어절의 형태소 열에 Bi-LSTM을 적용한 후, 각 어절 벡터 열에 대해 Biaffine Attention을 적용하는 방법을 제안하였다. [22]는 어절 표현을 위한 합성곱 신경망(CNN)의 최대값 추출(max-pooling), 문맥 반영을 위한 양방향 순환신경망(bidirectional RNN), 지배소 및 레이블 인식을 위한 자가 집중(self-attention) 메커니즘의 3단계로 구분하였다.

최근에는 대용량 코퍼스를 언어모델로 학습한 BERT(Bidirectional Encoder Representations from Transformers)와 같은 사전학습 모델의 등장으로, 출력층을 추가하여 사후학습(Fine-tuning) 하는 전이 학습(Transfer Learning) 방법이 활발히 연구되고 있다. [23]는 BERT를 통해 단어표상을 얻고 그 위에 LSTM RNN과 어텐션층을 쌓는 모델을 제안하였다. [24]는 BERT 및 ELMO 임베딩를 이용하여 Biaffine attention 모델 및 좌우 스택-포인터 네트워크(Left-to-Right stack-pointer network)와 여러 모델을 결합한 앙상블 모델을 제안하였다. [25]는 사전학습 모델의 자가집중 메커니즘을 효과적으로 사용하기 위해 추가 순환신경망 없이 지배소 인식을 진행하였다. 그러나 이러한 사후학습도 Labled data를 통해 자질을 학습하는 통계기반 지도학습이므로 데이터셋에 의존적이며, 그러하므로 데이터셋의 자료부족 문제(Data Sparsity)와 과적합의 문제(Overfitting)가 발생한다.

이러한 문제를 해결하기 위해 크게 데이터셋을 늘리는 방법과 전처리 및 후처리에서 추가적인 언어지식을 사용하는 방법이 있다. 본 논문에선 선행연구[28]를 개선하여 후처리 단계에서 문형과 통사, 의미에 관한 언어지식을 활용해 불필요한 지배소-의존소 관계를 제거[9-13] 및 강화한[16-19] 규칙기반 의존관계 에지 생성 알고리즘을 구축하였다. 이러한 심층학습과 규칙 결합을 통해 성능 향상과 그 효과에 대해 검증하고자 한다.

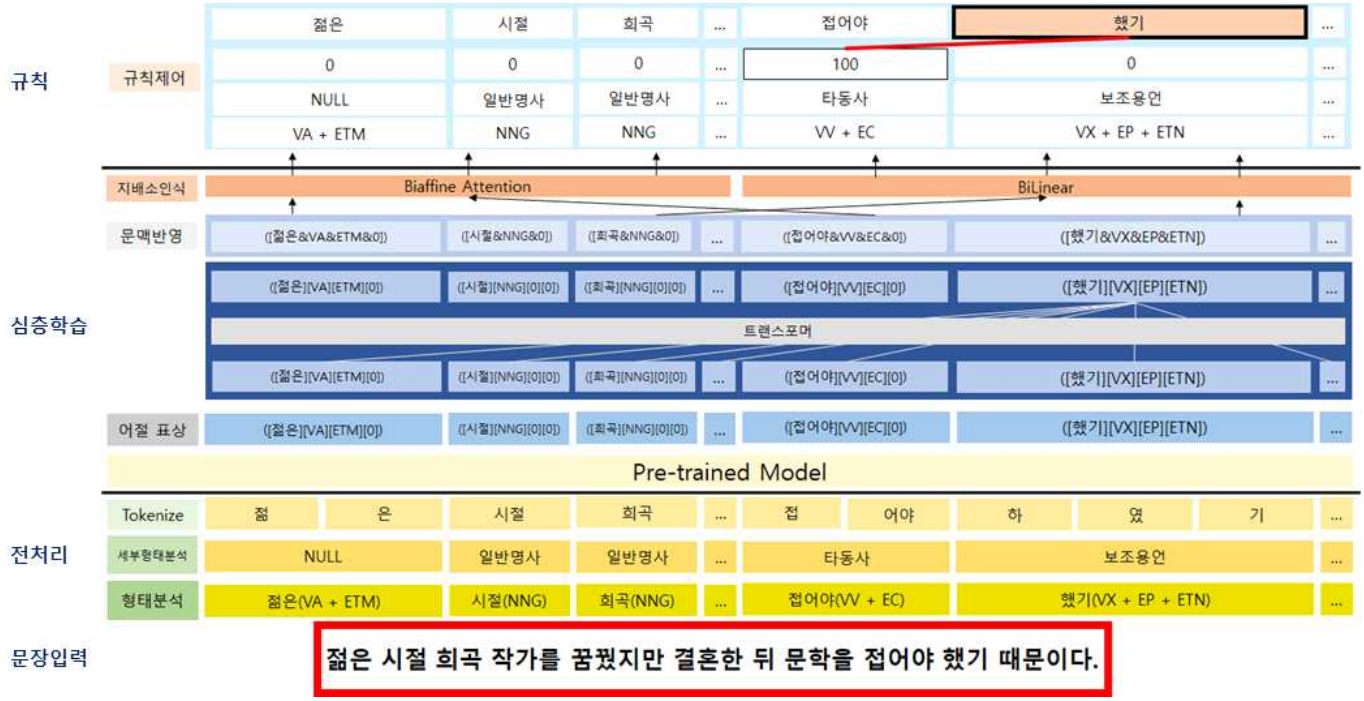

그림 1. 제안 모델 구조도

3. 지배소-의존소 제약규칙과 결합한 심층학습 구문분석 모델

본 논문에서는 데이터셋의 태깅 에러에 의한 모델의 과적합 현상을 해결하기 위해 기존 심층학습 모델에 한국어 언어학적 지식에 기반한 후처리 Attention 제어 Layer를 사용하였다. 본 절에선 모델 구조를 기존 딥러닝기반 모델과 의존관계 규칙으로 나누어 설명한다.

3.1 딥러닝 : 사전학습모델 + Transformer 문맥 임베딩 + Biaffine attention

3.1.1 어절 표상

각 입력 문장은 사전학습 언어모델을 통해 토큰 단위로 임베딩되어 벡터로 표현된다. 언어모델은 KoELECTRA을 사용하였다. 입력 문장을 토크나이저로 분절한 뒤, 언어모델에 입력하여 출력된 토큰단위 벡터들을 어절단위에 따라 맨 앞과 맨 뒤의 토큰을 결합한다. 이러한 벡터에 형태소 품사 임베딩 벡터를 결합하여 어절을 표상한다. 의존 구문 분석 모델의 성능은 형태소 분석에 큰 영향을 받는다. 본 논문에선 지배소의 종류(동사/명사 등)를 알 수 있는 첫 번째와 피지배소의 조사와 어미 등의 정보를 가진 마지막, 그리고 선어말 어미 등의 정보를 가진 마지막 앞 총 세가지의 형태소 정보를 사용하여

어절을 표상한다. 어절의 형태소가 3개 미만일 경우엔 해당 형태소를 NULL값으로 임베딩한다.

3.1.2 트랜스포머 인코더 기반 문맥 반영

어절 임베딩 벡터는 토큰 단위 임베딩 벡터에 형태소 임베딩 벡터를 결합(concatenation)한 형태이기에, 지배소 예측을 위해 다시 한번 형태소 정보를 포함하여 문맥을 학습할 필요가 있다. 어절 임베딩 벡터를 양방향 attention 기반 트랜스포머를 사용하여 형태소를 포함한 문맥을 반영해 어절-문맥 임베딩 벡터를 구한다.

3.1.3 지배소 및 레이블 인식 모델

문맥 반영된 각 어절 별 임베딩 벡터를 포인터 네트워크를 사용하여 지배소-피지배소 및 의존관계 레이블을 인식한다. 포인터 네트워크는 attention 기법을 통해 각 어절과 다른 모든 어절 간의 상호 의존성을 attention score 형태로써 구한다. 이때 사용 가능한 attention 기법은 scaled dot-product, multi-head, biaffine 등이 있으나 본 연구에선 어절의 지배소 결정에 가장 효과적인 biaffine attention을 활용하였다.

3.2 규칙 : 지배소-의존소 제약규칙(규칙기반 attention 제어)

3.2.1 지배소-의존소 에지 생성 알고리즘

지배소와 의존소의 의존관계(에지) 생성은 그림2와 같은 알고리즘의 형태를 따른다. 각 문장(batch)별로 Root 어절부터 첫 번째 어절까지 지배소를 중심으로 탐색한

```
for batch in allBatches do
  for Eojeol in allEojeiols do
    if Eojeol_pos =지배소 then
      if Eojeol_pos^specific =지배소 자질 then
        for Eojel in 어절 지배소 index do
          if 지배소 의존소 위반 Rule then
            edgeAttentionScore[ 피지배소 - 지배소] = 0
          end if
          if 지배소 의존소 정답 Rule then
            edgeAttentionScore[ 피지배소 - 지배소] = 100
          end if
```

그림 2. 지배소-의존소 제약규칙을 이용한 Attention 제어 의사코드

다. 특정 지배소를 찾으면 이러한 지배소의 전처리 단계에서 판별된 세부 분류(자질)을 확인한 뒤, 문장의 처음부터 앞 어절까지 탐색하여 규칙에 해당하는 의존소를 탐색한다. 규칙은 크게 위반과 정답 두 종류를 가지는데, 위반일 경우 해당 에지의 attention score값을 0, 정답일 경우 100을 부여한다. 예는 다음과 같다. 예문 : “젊은 시절 희곡 작가를 꿈꿨지만 결혼한 뒤 문학을 접어야(VV+EC) 했기(VX+EP+ETN) 때문이다.” [본용언+보조용언]의 문형이 나타날 경우, 문장성분은 보조용언이 아닌 본용언에 연결되는 것이 옳다. 또한 본용언은 보조용언과 연결되어야 한다. 이를 규칙화하면 다음과 같다. 위반Rule : Edge_attention_score[문장성분(꿈꿨지만)-보조용언(했기)] = 0 정답Rule : Edge_attention _score [본용언(접어야) - 보조용언(했기)] = 100

3.2.2 지배소-의존소 규칙

규칙은 지배소 후위를 포함하여 총 24가지를 사용하였으며, 규칙 구현에는 문형 정보 및 통사와 의미적인 부분까지 사용하였다. 형태소 정보는 국립국어원에서 배포한 형태분석 말뭉치 정보를 사용하였으나, 더 세부적인 분석을 위해 한국민족문화대백과사전과 KLParser[19]의 사전 정보를 사용하였다. 특히 동사의 경우, 목적어를 가지지 않는 ‘흐르다, 솟다, 피다’ 등의 자동사와 목적어를 가지는 ‘읽다, 쓰다, 먹다’ 등의 타동사의 분류가 목적어와 보어의 연결 여부가 되기 때문에 구문 분석에 매우 중요한 정보이지만 배포된 데이터셋의 형태분석엔 담기지 않았다. 또한 명사의 경우, 동작을 나타내는 동작성명사가 있으며 이는 의사보조용언(NNG + 중이다)이나 혹은 동사파생접미사(NNG + XSV)와 결합하여 동사의 역할을 대신하기도 한다. 이러한 정보 외에도 의존명사는 ‘것, 뿐, 중 등과 ‘번, 개, 원’ 등에 따라 의사보조용언과 단위의존명사로 나뉘며 적용되는 규칙 또한 달라진다. 이러한 정보들이 국립국어원 형태 분석에는 포함되지 않음으로, 추가로 전처리단계에서 지배소에 따라 문형 및 통사, 의미를 사용하여 세부 자질을 판별하고 추가 형태분석 및 분류를 진행하였다. 또한 이러한 분류를 기반으로 세부 규칙을 적용하였다.

표 1. 지배소-의존소 규칙 정리

지배소	자질	규칙
동사	-동사 공통	관형절은 용언에 연결되지 못함
	-문장 주동사	문장부사는 문장 주동사와 연결
	-자동사	목적격은 자동사와 연결되지 못함
	-불완전동사	주어를 제외하고 다른 문장성분은 연결되지 못함
	-동사+동사	본용언이 연속적으로 나타날 경우 주어를 앞에 위치한 서술어에 연결
	-부사형용언	‘~도록’ 과 같은 형태의 부사격으로 쓰이는 용언은 내포문의 주어가 아닌 문장주어가 연결되지 못함
보조용언	-보조용언	문장부사와 바로 앞 어절을 제외하곤 다른 문장성분은 연결되지 못함
		바로 앞 어절과 연결
인용	-직접인용	문장주어는 인용절 끝에 연결되지 못함
		인용절 끝은 문장 주동사에 연결
	-간접인용	직접인용과 같은 원칙, 문장주어는 인용절 끝에 연결되지 못함
		인용절 끝은 문장 주동사에 연결
의존명사	-의존명사공통	바로 앞에 지시관형사가 올 경우 연결
		바로 앞에 관형사가 올 경우 연결
		바로 앞에 명사가 올 경우 연결
		바로 앞에 명사파생접미사가 올 경우 연결
	-일반의존명사	바로 앞에 대명사가 올 경우 연결
		바로 앞에 명사가 올 경우 연결
	-단위의존명사	바로 앞에 수 관형사가 올 경우 연결
		바로 앞에 수사가 올 경우 연결
	-의사보조용언	문장부사와 바로 앞 어절을 제외하고 문장 성분이 연결되지 못함
		바로 앞 어절(관형형 ㄴ/ㄹ, 동작성명사)과 연결
일반명사	일반명사	연결어미는 목적격 조사를 가지는 명사절에는 연결되지 못함

4. 실험 환경 및 결과

4.1 실험 환경

본 논문에서 제안한 의존 구문분석 모델의 실험은 pytorch를 사용하였으며, 언어모델은 KoELECTRA-base를 사용하였다. 하이퍼 파라미터는 학습 횟수(epoch) 3, 학습률(learning rate) 5e-5, 배치크기(batch size) 16을 적용하였다. 평가 Metric은 의존 구문분석에 사용되는 UAS(unlabeled attachment score)와 LAS(labled attachment score)를 사용하였는데, UAS는 전체 어절 중 지배소를

올바르게 인식한 어절 정확도이고, LAS는 지배소와 의존 관계 레이블을 모두 올바르게 인식한 어절 정확도이다.

4.2 실험 데이터셋

본 연구의 데이터셋으로 국립국어원에서 배포하는 구문 분석 문어 말뭉치(모두의 말뭉치)를 사용하였다. 본 데이터셋은 21세기 세종계획 구구조 구문분석 말뭉치를 포함한 신문기사 200만 어절(14만 5천 문장)으로 이루어져 있으며 정답 태깅 가이드라인으로 정보통신단체표준(TTAS)[26]를 기준으로 하고 있다. 본 논문에선 학습(Train) 셋으로 12만 문장, 개발(Dev) 셋으로 1만 5천 문장, 평가(Test) 셋으로 1만 문장을 사용하였다. 그러나 본 데이터셋에는 정답 태깅 에러가 존재한다. 같은 국립국어원에서 조사한 보고서에 따르면, 추정 품질 평균은 93.57%로 추정 정답 태깅 에러 6.43%가 존재한다.[27] 이러한 데이터셋의 태깅에러는 모델의 학습과 예측에 오류를 발생시키는 과적합(overfitting)의 현상을 더욱 심화시킨다. 정답 태깅 에러의 예시는 다음과 같다.

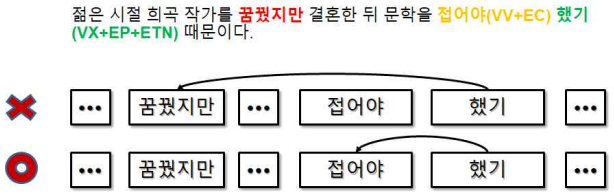

그림 3. 데이터셋 태깅 에러 예시

[본용언+보조용언]의 문형이 있을 때, 문장성분은 지배소로 본용언을 가지며 보조용언은 바로 앞의 본용언과 연결되는 것이 옳다. 그러나 말뭉치에서 이러한 원칙이 적용되지 않은 채로 정답 태깅이 잘못된 문장들이 존재하고, 이를 통해 학습한 결과 실제 모델의 예측에도 문장 성분이 보조용언을 지배소로 예측하는 과적합의 문제가 발생하였다. 본 논문에선 딥러닝에 규칙을 결합하는 방법을 통해 이러한 현상을 해결하고자 하였고, 방법의 증명을 위해 학습과 개발 셋은 그대로 두고 테스트셋의 태깅에러를 평가에서 제외한 것과 정답을 수정한 것 두 가지로 나누어 성능을 평가하였다.

4.3 실험 결과

표 2. 성능 비교 분석

	UAS	LAS
Bi-affine (baseline)	93.34%	91.02%
부산대 [28] (Bi-affine과 지배소 의존소 제약규칙)	93.85%	91.27%
제안모델 1 (태깅에러 제거 방식)	96.06%	93.11%
제안모델 2 (태깅에러 수정 방식, 강화된 지배소-의존소 제약규칙)	96.28%	93.19%

적용한 지배소-의존소 제약규칙이 구문분석에 얼마나 영향을 미치는지 평가하기 위해 모델을 2개로 나누었고, 모델마다 실험환경에서 제시한 모두의 말뭉치 1만 문장(13만 어절)을 대상으로 태깅에러 제외 방식과 정답수정 방식으로 성능을 평가하였다. Bi-affine(baseline)은 제약규칙을 사용하지 않은 딥러닝 의존구문분석기 모델이며, 부산대 구문분석기[28]는 지배소-의존소 제약규칙 중 [본용언+보조용언], [의사보조용언] 총 2개의 규칙을 활용한 선행연구의 모델이다. 제안모델 1은 부산대 구문분석기의 규칙과 에지 생성 알고리즘을 더 강화하여 총 24가지의 규칙을 적용한 제안모델로, 선행연구와 같이 태깅 에러를 제외하여 평가한 모델이다. 제안모델 2의 경우, 규칙을 적용하였을 때 모델 예측이 올바른지 확인하기 위해 데이터셋의 정답을 수정하여 평가한 모델이다. 제안모델 1의 결과를 통해 규칙을 적용하지 않은 Baseline 모델보다 UAS 2.72%, LAS 2.09%, 선행연구의 모델보다 UAS 2.21%, LAS 1.84% 향상된 것을 확인할 수 있으며, 이를 통해 본 연구에서 제안하는 방법이 모델의 성능에 효과적임을 알 수 있다. 또한 제안모델 2의 결과를 통해 학습 셋에서 태깅 에러가 있음에도 규칙을 통해 모델 예측을 옳게 제어할 수 있음을 알 수 있다.

표 3. 규칙 및 데이터량에 따른 속도와 성능 비교

	학습데이터량	학습시간	UAS	LAS
규칙적용X	12만 문장	56h 27m	93.34%	91.02%
규칙적용O	1만 5천 문장	5h 14m	94.29%	91.44%

표3은 규칙 적용 유무와 학습 데이터량에 따른 학습 시간 및 성능 비교이다. 본 논문에서 제시한 규칙과 분석 방법론을 적용한 모델은 학습데이터가 1/8로 줄었음에도 적용하지 않은 모델보다 학습시간은 11배 빠르면서 성능 또한 UAS 0.95% 높은 것을 확인할 수 있다. 이를 통해 본 논문에서 제시하는 방법이 적은 데이터셋에도 효과적이며, 학습에 필요한 시간과 자원을 줄일 수 있음을 알 수 있다.

5. 결론 및 향후 연구

본 논문에서의 실험 및 연구를 통해 심층학습에 언어지식을 결합하는 것이 딥러닝의 한계를 극복할 수 있음을 증명하고자 하였다. 제안 모델은 최대 UAS 96.28, LAS 93.19의 성능을 보였으며, 선행연구 대비 UAS 2.21%, LAS 1.84%의 향상된 결과를 보였다. 또한 적은 데이터셋으로 학습했음에도 8배 많은 데이터셋 학습모델 대비 UAS 0.95%의 향상과 11배 빠른 학습 시간을 보였다. 이를 통해 심층학습에 언어지식을 결합하는 것이 현재 심층학습이 데이터셋의 의존적이므로 발생하는 여러 문제를 해결할 수 있음을 알 수 있었다. 향후 연구에서는 규칙기반 모델과 심층학습 모델의 장단점을 실험을 통해 더 세부적으로 분석하고, 두 모델의 결합의 효과를 더 면밀히 살피고자 한다. 또한 복합명사와 복문의 문제, 특히 [문장주어-주동사]의 의존관계에서 언어학적 지식과 국립국어원의 가이드라인 원칙이 다른 문제에 대해 연구하여

더욱 강화된 에지 알고리즘을 개발할 예정이다. 또 최근 초거대모델의 수요가 높아진 만큼 파라미터 수를 줄이고 성능을 향상시키는 연구가 활발히 이루어지고 있는데, 규칙을 초거대모델에 적용한 효과적인 튜닝 방법을 연구할 예정이다.

참고문헌

[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. (in Korean)

[28] J. M. Shin, S. H. Cho, S. R, Park "Neural network-based dependency parsing with rules applied," KCC, 2022. (in Korean)