

# 의존 구문 분석을 활용한 자연어 추론

김슬기<sup>0,1</sup>, 김홍진<sup>2</sup>, 김학수<sup>1,2</sup>

건국대학교 컴퓨터공학과, 건국대학교 인공지능 학과

[cloudyju11@konkuk.ac.kr](mailto:cloudyju11@konkuk.ac.kr), [jin3430@gmail.com](mailto:jin3430@gmail.com), [nlpdrkim@konkuk.ac.kr](mailto:nlpdrkim@konkuk.ac.kr)

## Natural Language Inference using Dependency Parsing

Seul-gi Kim<sup>0,1</sup>, Hong-Jin Kim<sup>2</sup>, Hark-Soo Kim<sup>1,2</sup>

Konkuk University Department of Computer and Communications Engineering<sup>1</sup>,  
Konkuk University Department of Artificial Intelligence<sup>2</sup>

### 요 약

자연어 추론은 두 문장 사이의 의미 관계를 분류하는 작업이다. 본 논문에서 제안하는 의미 추론 방법은 의존 구문 분석을 사용하여 동일한 구문 정보나 기능 정보를 가진 두 개의 (피지배소, 지배소) 어절 쌍에서 하나의 어절이 겹칠 때 두 피지배소를 하나의 청크로 만들어주고 청크 기준으로 만들어진 의존 구문 분석을 사용하여 자연어 추론 작업을 수행하는 방법을 의미한다. 이러한 의미 추론 방법을 통해 만들어진 청크와 구문 구조 정보를 Biaffine Attention을 사용하여 한 문장에 대한 청크 단위의 구문 구조 정보를 반영하고 구문 구조 정보가 반영된 두 문장을 Bilinear을 통해 관계를 예측하는 시스템을 제안한다. 실험 결과 정확도 90.78%로 가장 높은 성능을 보였다.

주제어: 의미 추론 방법, 의존 구문 분석, 자연어 추론, 청크(Chunk)

### 1. 서론

자연어 추론(Natural Language Inference)은 자연어 이해(Natural Language Understanding)를 기반으로 모델의 추론 능력을 평가하는 작업으로 두 문장 사이의 의미 관계를 함의(Entailment), 모순(Contradiction), 중립(Neutral)으로 분류하는 문장 쌍 분류(Sentence-Pair Classification)의 일종이다. 두 문장은 전제(Premise)와 가설(Hypothesis)로 나누어지는데 전제를 참이라고 가정할 때 가설의 내용이 참(함의)인지, 거짓(모순)인지, 혹은 알 수 없는지(중립)에 따라 두 문장의 관계가 분류된다.

예를 들어, 한국어 NLI 데이터 세트 KLUE-NLI<sup>1)</sup>에는 전제 문장이 "부산은 블록체인 기술을 활용한 실증사업이 금융분야까지 확장하고, 대전은 바이오 스타트업에 병원체 공용연구시설을 공유하는 등 일부 기초 특구에도 사업이 추가됐다."이고 가설 문장이 "부산과 대전 등 기존 특구에는 사업들이 모두 폐지됐다."라는 문장 쌍이 있다. 전제 문장에서 ("부산은", "대전은", "일부 기존 특구에도", "사업이 추가됐다.")를 통해 가설 문장과 전

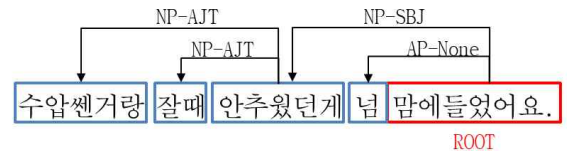


그림 1 의존 구문 구조 예시

제 문장이 서로 모순됨을 알 수 있다. 본 논문에서는 KLUE-NLI 데이터의 추론 방법을 표 1과 같이 나눈다. 표 1의 여러 추론 방법 중 의미 추론은 전제와 가설 문장 간의 의미를 정확히 파악해야 해결할 수 있는 경우를 뜻한다. 본 논문에서는 문장의 의미를 더 정확하게 파악하기 위해 의존 구문 분석을 활용한다. 먼저, 의존 구문 분석의 결과 중 각 어절(피지배소)이 지배소와 가질 수 있는 의존 관계 태그는 구문 정보와 기능 정보로 나뉜다. 이 중 구문 정보는 각 어절이 체언인지, 용언인지, 부사인지 등을 나타낸다. 기능 정보는 각 어절이 지배소와 가지는 관계가 주격인지, 목적격인지, 관형격인지 등을 나타낸다. 의존 관계 태그는 그림 1에서와같이 “-”를 기준으로 앞은 구문 정보, 뒤는 기능 정보가 표기되

1) 한국어 데이터 세트 KLUE-NLI, <https://aistages-prod-server-public.s3.amazonaws.com/app/Competitions/000068/data/klue-nli-v1.1.tar.gz>

어 있다. 이와 같은 구문 정보, 기능 정보와 (피지배소, 지배소) 어절 쌍을 활용하여, 전제와 가설 간의 의미를 정확하게 파악하기 위해 두 개의 어절 쌍을 하나의 청크로 표현하는 방법을 제안한다.

표 1 KLUE 데이터 셋의 추론 방법과 예시

방법	전제	가설	라벨
숫자 추론	3명이 시작해서 현재 연구원은 5명, 인턴은 3명으로 구성되었습니다.	현재 연구원과 인턴은 총합 8명입니다.	합의
지식 추론	생활력도 책임감도 없는 남편과 결혼한 여주인공 고마키는 초등학교 딸 노리코와 함께 집을 나온다.	고 마 키 는 기혼자이다.	합의
시간 추론	7월말부터 8월말까지 약 한달정도를 지냈어요.	7월 초부터 지냈어요.	모순
장소 추론	광주전남에서 유일하게 진행되는 이 공연은 서울시와 강진군의 문화교류 사업으로 무료 초대권을 6월 20일부터 광주 아트홀 매표소에서 배부한다.	이 공연은 서울시와 강진군에서 볼 수 있다.	모순
의미 추론	수업센터랑 잘때 안추웠던게 녀 맘에들었어요.	잘 때 흡진 않았어요.	합의

청크 표현 방법은 두 개의 (피지배소, 지배소) 어절 쌍이 문장 내에 연속적으로 등장하고, 각 어절 쌍의 피지배소의 구문 정보가 같거나 기능 정보가 모두 보어(CMP), 체언 수식어(MOD), 용언 수식어(AJT) 중 하나이면 두 어절 쌍의 피지배소를 하나의 청크로 만들어 준다. 이렇게 만들어진 청크와 남은 지배소를 하나의 쌍으로 묶은 것을 청크 쌍(Chunk Pair)이라 정의한다. 전제와 가설에서 만들어진 청크 쌍을 이용하여 전제와 가설 간의 의미를 정확하게 파악하면, 두 문장의 관계를 분류하는 것에 도움이 된다.

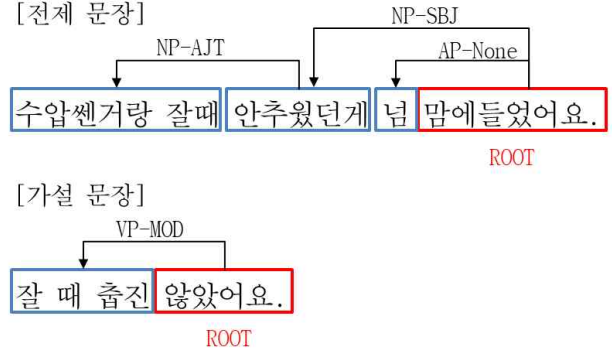


그림 2 청크 기준 의존 구문 구조 예시

예를 들어, “수업센터랑 잘 때 안추웠던게 녀 맘에들었어요.” 라는 전제 문장의 의존 구문 분석 결과는 그림 1과 같이 나타낼 수 있다. (피지배소, 지배소) 어절 쌍인 (“수업센터랑”, “안추웠던게”)와 (“잘때”, “안추웠던게”)는 모두 “안추웠던게”를 지배소로 가지고 있고, 문장 내에서 연속적으로 등장하며 이 두 개의 어절 쌍의 피지배소는 같은 구문 정보 “NP”와 같은 기능 정보 “AJT”를 가지고 있다. 따라서 두 어절 쌍에서 피지배소인 “수업센터랑”과 “잘때”는 하나의 청크 “수업센터랑 잘때”가 되어 최종적으로 하나의 청크 쌍 (“수업센터랑 잘때”, “안추웠던게”)가 된다. 전제 문장 “수업센터랑 잘때 안추웠던게 녀 맘에들었어요.”와 가설 문장 “잘 때 흡진 않았어요.”의 의존 구문 분석 결과는 그림 2와 같이 청크 기준으로 표현될 수 있다.

그림 2와 같이 어절 단위의 의존 구문 분석 결과를 청크 단위로 변경한 다음, 청크 단위의 구문 정보를 반영하기 위해 전제와 가설에 대해 각각 구문 정보 트리플(Triple)을 구성한다. 전제와 가설 문장과 각 문장의 구문 정보 트리플을 Biaffine Attention[14]을 이용하여 구문 정보를 반영한 후에 Bilinear를 통해 전제와 가설 간의 의미 관계를 예측하는 시스템을 제안한다.

## 2. 관련 연구

기존의 자연어 추론은 어휘·의미 중첩[1]이나 패턴 기반 관계 추출[2] 등을 기반으로 수행했다. 하지만 [1, 2]는 문장의 의미 구조를 파악하지 못하는 문제가 있었고, 이를 보완하기 위해서 [3]에서는 종속성 트리플 의미 구조의 프록시(Proxy)를 사용하여 어휘 및 구조적 일치 모두 통합하는 시스템을 제안했다. [4-5]는 두 문장에 대한 설명을 생성하여 이를 활용하면 두 문장의 예측에 대한 정확성 향상에 영향을 미친다는 것을 보여준다. [6]에서는 각 라벨(합의, 모순, 중립)별 설명 생성

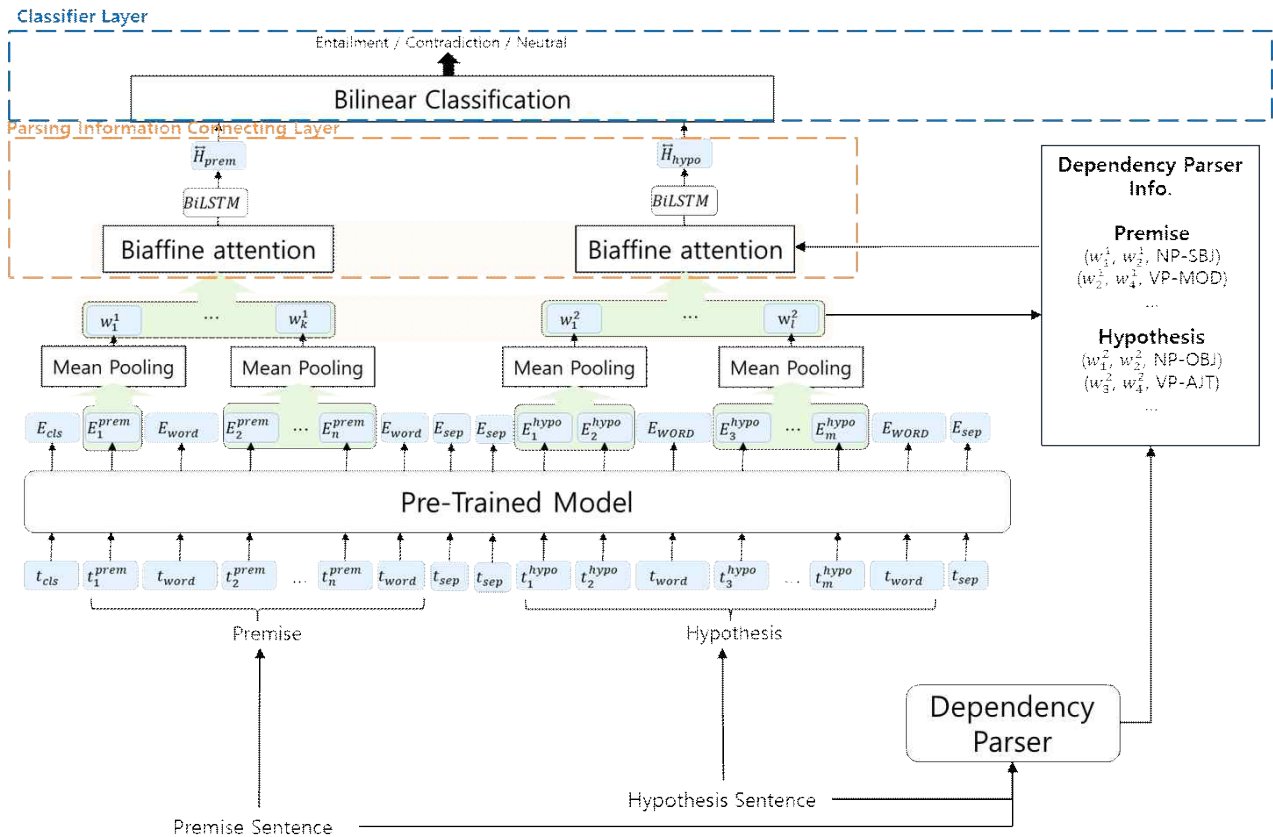


그림 3 모델 구조도

기를 통해 두 문장의 관계를 예측하는 방법, [7]은 전제 문장에 대해 가설 문장에서 라벨에 대한 근거를 추출하고 전제 문장과 근거를 통해 설명을 생성하는 방법을 사용한다. 하지만 이러한 방법은 모델을 학습하기 위해 데이터 세트에 라벨에 대한 설명이 존재해야 한다는 문제가 있다. [8]은 두 문장의 관계를 오분류한 MultiNLI[9] 데이터의 50%는 외부지식이 필요함을 언급한다.

최근 자연어 추론은 SNLI[10], MultiNLI[9]와 같은 대규모 데이터 세트를 사용하여 연구됐다. [11]은 단어가 중복되면 다른 의미를 가지더라도 두 문장에 대해 서로 수반된다고 예측하는 문제를 언급했다. [12-13]는 가설 문장에 not, never과 같은 특정 부정적인 어휘가 존재하면 두 문장의 관계를 모순으로 분류하는 문제를 언급하고 있다. 따라서 자연어 추론에서는 특정 단어보다는 전제와 가설 문장의 의미를 정확히 파악하는 것이 중요하다. 본 논문에서는 전제와 가설 간의 의미를 정확하게 파악하기 위해 의존 구문 분석의 결과를 활용하여 두 문장의 관계를 분류하는 방법을 제안한다.

### 3. 청크와 의존 구문 구조를 반영한 자연어 추론 모델

본 논문에서 제안하는 전체 모델 구조도는 그림 3과 같다. 제안 모델은 크게 RoBERTa, 구문 정보 반영 계층, 문장 쌍 분류 계층으로 구성되어 있다. RoBERTa에서 문장 쌍 입력 데이터를 받아 단어 표현 벡터를 생성한다. Mean Pooling을 통해 각 문장에 대한 청크 표현 벡터를 만들어 준다. 구문 정보 반영 계층에서 청크와 구문 정보 트리플을 Biaffine Attention[14]을 이용하여 한 문장에 대한 구문 구조 정보를 반영해 준다. 문장 쌍 분류 계층에서는 각 문장을 표현하는 벡터를 Bilinear[15]를 통해 두 문장 간의 관계를 예측한다.

#### 3.1 청크 표현 벡터 생성

입력 단계에서 두 문장과 [16]의 의존 구문 분석을 사용하여 만든 청크와 구문 구조 정보를 입력한다. 전제 문장 Sequence를  $t^{pre} = \{t_1^{pre}, \dots, t_n^{pre}\}$ , 가설 문장 Sequence를  $t^{hypo} = \{t_1^{hypo}, \dots, t_m^{hypo}\}$ 라 할 때, 입력 Sequence는  $\{t_{cls}, t^{pre}, t_{sep}, t_{sep}, t^{hypo}, t_{sep}\}$ 이다. 각 청크 사이를 구분하는 토큰으로  $t_{word}$  토큰을 넣어  $\{t_{cls}, t_1^{pre}, t_{word}, \dots, t_n^{pre}, t_{word}, \dots, t_1^{hypo}, \dots, t_m^{hypo}, t_{word}, t_{sep}\}$ 와 같이 각 청크를 구분해준다. 사전 학습된 언어 모델

RoBERTa를 통해 문맥 정보를 학습한 토큰 표현 벡터  $\{E_{cls}, E_1^{prem}, \dots, E_n^{prem}, E_{word}, \dots, E_1^{hypo}, \dots, E_m^{hypo}, E_{word}, E_{sep}\}$ 를 출력한다. 각 문장의  $j$ 번째 청크 표현 벡터는 청크를 구성하는 토큰 벡터들의 평균으로 생성되며 수식은 다음과 같다.

$$w_j^{sen} = \frac{1}{a+1} \sum_k^{k+a} E_k^{sen} \text{ for } sen \in \{prem, hypo\} \quad (1)$$

### 3.2 구문 정보 반영 계층

각 문장에 대한 구문 정보 트리플과 청크 단위의 단어 표현 벡터를 통해 지배소 청크  $w_{head}^{sen}$ , 피지배소 청크  $w_{mod}^{sen}$ , 두 청크의 태그 임베딩  $M_{w_{head}, w_{mod}}^{sen}$ 을 합하여 청크 간의 구문 구조 정보 벡터  $T_{w_{head}, w_{mod}}^{sen}$ 를 만들어 준다. 수식은 다음과 같다.

$$T_{w_{head}, w_{mod}}^{sen} = (w_{head}^{sen} + w_{mod}^{sen} + M_{w_{head}, w_{mod}}^{sen}) \quad (2)$$

전제 문장의 청크  $W^{prem} = [w_1^{prem}, w_2^{prem}, \dots, w_j^{prem}]$ 와 구문 구조 정보 벡터  $T^{prem} = \{T_{w_{head}, w_{mod}}^{sen}\}$ 를 이용하여 Biaffine Attention을 통해 전제 문장의 청크 간의 구문 구조 정보가 반영된 벡터  $B^{prem}$ 를 출력한다. 가설 문장에 대해서도 동일하게 수행하여  $B^{hypo}$ 를 출력한다.  $U_{sen}, V_{W^{sen}}, V_{T^{sen}}$ 는 가중치이고  $b_{sen}$ 는 편차이다.  $sen \in \{prem, hypo\}$ 일 때, 수식은 다음과 같다.

$$B^{sen} = W^{sen} U_{sen} T^{sen} + W^{sen} V_{W^{sen}} + T^{sen} V_{T^{sen}} + b_{sen} \quad (3)$$

구문 구조 정보가 반영된 청크 단위의 표현을 Bi-LSTM에 입력하여 정방향 마지막 은닉 상태와 역방향 마지막 은닉 상태를 연결하여 각 문장  $sen \in \{prem, hypo\}$ 을 표현하는 벡터  $\vec{H}_{sen}$ 를 출력하며, 수식은 다음과 같다.

$$\begin{aligned} \vec{h}_{sen,i} &= BiLSTM(B^{sen}, \vec{h}_{sen,i-1}) \\ \vec{H}_{sen} &= [\vec{h}_{sen,1}; \vec{h}_{sen,k}] \end{aligned} \quad (4)$$

### 3.3 문장 쌍 분류 계층

두 문장의 관계를 예측하기 위해서 두 문장의 출력 벡터  $\vec{H}_{prem}, \vec{H}_{hypo}$ 를 Bilinear Classification을 통해 두 문장이 합의, 모순 혹은 중립적인 관계인지를 다음 수식과 같이 예측한다.  $U_l, V_{prem}, V_{hypo}$ 는 가중치이다.

$$l = \vec{H}_{prem} U_l \vec{H}_{hypo} + \vec{H}_{prem} V_{prem} + \vec{H}_{hypo} V_{hypo} \quad (5)$$

## 4. 실험 및 결과

표 2 데이터 KLUE-NLI 설명

source	Total	Avg. Prem.	Avg. Hypo.
WIKITREE	4,288	52.81	26.86
POLICY	4,283	56.73	32.93
WIKINEWS	4,274	64.17	29.11
WIKIPEDIA	4,230	57.45	23.70
NSMC	5,499	27.48	21.49
AIRBNB	5,424	24.28	18.65
Overall	27,998	47.15	25.46

본 논문에서 사용한 KLUE-RoBERTa-base는 대용량 한국어 말뭉치를 이용해 사전 훈련된 언어 모델이다. 실험으로 사용하는 데이터는 WIKITREE, POLICY, WIKINEWS, WIKIPEDIA, NSMC 및 AIRBNB를 사용한 KLUE-NLI 데이터 세트를 사용하였다. 학습 데이터로 24998개의 문장 쌍, 평가 데이터로 3000개의 문장 쌍을 사용하였다. 표 2는 KLUE-NLI 데이터 세트에 대한 설명으로 Total은 최종 데이터로 각 코퍼스에서 몇 개의 데이터를 사용했는지를 나타내고 있다. Avg Prem.은 전제 문장의 평균 길이, Avg Hypo.는 가설 문장의 평균 길이를 나타낸다. 본 논문에서 사용한 파라미터로 Learning Rate는  $1e-5$ , Epoch는 5, Batch Size는 64이다. 본 논문의 실험에서 나온 성능은 3번 평가한 후 나온 정확도의 평균이다.

표 3 KLUE-RoBERTa-base 활용에 따른 성능 비교

Model	Accuracy
KLUE Baseline [17]	84.83
Ours Baseline	89.91
Mean Pooling + BiLSTM + Bilinear	90.45
+ Dependency Parser Triple	90.63
+ Biaffine Attention	<b>90.78</b>

표 3은 본 논문에서 사전학습한 KLUE-RoBERTa-base의 활용에 따른 KLUE-NLI 데이터의 성능이다. 표 3에서 Our Baseline은 문장 쌍을 입력했을 때 RoBERTa Encoder 출력 중 CLS 벡터에서 FNN을 적용하여 두 문장의 관계를 예측한 모델이다. KLUE의 baseline의 정확도 84.83보다 5.08 더 높은 정확도를 보인다. Mean Pooling + BiLSTM + Bilinear은 KLUE-RoBERTa-base Encoder 출력을 청크 기준으로 나누고 Mean Pooling을 사용하여 청크 표현 벡터를 생성한다. 청크 표현 벡터를 BiLSTM으로 각 문장의 문맥 정보 만들어 준다. 각 문장의 문맥 정보를 Bilinear을 통해 두 문장의 관계를 예측해 주는 모델이다. Mean Pooling + Dependency Parser Triple + BiLSTM

+ Bilinear은 Mean Pooling + BiLSTM + Bilinear 모델의 청크 표현 벡터와 구문 구조를 Bilinear을 통해 각 문장의 구문 구조 정보를 반영해 준 다음, Bilinear을 통해 두 문장의 관계를 예측해 주는 모델이다. Mean Pooling + Dependency Parser Triple + Biaffine Attention + BiLSTM + Bilinear 모델은 청크 표현 벡터와 구문 구조를 Biaffine Attention을 통해 각 문장의 구문 구조 정보를 반영해 주고 Bilinear을 통해 두 문장의 관계를 예측해 주는 모델이다. 실험 결과, 본 논문에서 제안하는 모델이 기존 성능보다 5.95만큼, Ours Baseline보다 0.87만큼 더 좋은 성능을 보였다. 따라서 청크와 청크에 대한 구문 정보를 Biaffine Attention을 통해 각 문장에 대한 청크 간의 정보를 반영해 주는 방법이 효과적임을 보인다.

표 4 청크 표현 기준에 대한 데이터 성능 비교

Model	Accuracy
구문 정보를 기준으로 청크 생성	90.56
기능 정보를 기준으로 청크 생성	90.45
구문 정보나 기능 정보를 기준으로 청크 생성	<b>90.78</b>

표 4는 청크 기준에 따른 성능 비교이다. 일반적으로 구문 구조는 구문 정보와 기능 정보로 나눈다. 구문 정보를 기준으로 청크를 생성하는 경우는 같은 구문 정보를 가진 두 개의 어절 쌍에서 피지배소를 하나의 청크로 만드는 경우이다. 기능 정보를 기준으로 청크를 생성하는 경우는 기능 정보가 MOD, AJT, CMP일 경우 두 개의 어절 쌍에서 피지배소들을 하나의 청크로 만드는 경우이다. 그리고 구문 정보나 기능 정보를 기준으로 청크를 생성하는 경우는 위의 두 기준을 합친 경우이다. 표 4를 통해 구문 정보와 기능 정보를 모두 고려하여 청크를 생성할 때 가장 효과적임을 알 수 있다.

표 5 청크 표현 방법에 대한 성능 비교

Model	Accuracy
단어 표현 조합 사용	90.50
Mean Pooling	<b>90.78</b>

표 5에서 청크 표현 방법은 토큰 표현 벡터를 청크 표현 벡터로 변경해주는 방법을 의미한다. 두 가지의 청크 표현 방법을 이용하여 비교 실험을 진행한다. 첫 번째는 하나의 청크에 해당하는 토큰 표현 벡터를 Mean Pooling을 통해 청크 표현 벡터로 만들어 주는 방법이다. 두 번째는 청크에 해당하는 토큰 표현 벡터 중 청크의 시작과 끝에 해당하는 토큰 표현 벡터에 대한 단어 표현 조합

$((E_{start}, E_{end}, |E_{start} - E_{end}|, E_{start} * E_{end}))$ 을 이용하는 방법이다. 표 5의 성능 결과를 통해 Mean Pooling을 통해 청크 표현 벡터를 만들어 주는 방법이 청크 단위의 의미를 표현하는 것에 더 효과적임을 알 수 있다.

표 6 의존 관계 태그 정보 유무에 따른 성능 비교

Model	Accuracy
지배소와 피지배소의 정보만 제공	90.36
지배소와 피지배소의 의존 관계 태그 정보를 제공	<b>90.78</b>

표 6은 청크 간의 구문 구조 정보 벡터( $T_{ij}$ )를 표현할 때 두 청크(지배소, 피지배소)에 대한 의존 관계 태그 임베딩,  $M_{ij}$ 의 포함 여부에 따른 성능 비교이다. 표 6의 성능 결과를 통해 지배소와 피지배소의 의존 관계 태그 정보를 주는 것이 문장의 구문 구조를 반영하는 것에 더 효과적임을 보인다.

표 7 의존 구문 분석 모델에 따른 성능 비교

Model	Accuracy
KoalaNLP <sup>2)</sup>	90.60
[16]의 의존구문 분석	<b>90.78</b>

표 7은 의존 구문 분석 모델에 따른 성능 비교이다. 표 7은 의존 구문 분석 모델에 따라 자연어 추론 모델의 성능이 달라진다. 본 논문의 제안 모델은 의존 구문 분석 모델과 자연어 추론 모델이 분리되어 있고 의존 구문 분석 모델의 출력이 자연어 추론 모델에 영향을 주므로 의존 구문 분석 모델의 성능에 따라 자연어 추론 모델의 성능이 달라짐을 알 수 있다.

## 5. 결론 및 향후 연구

본 논문에서는 자연어 추론에서 전제와 가설로 주어지는 각 문장의 의미를 효과적으로 파악하기 위해서 의존 구문 구조를 사용하여 구문 정보나 기능 정보를 기준으로 청크를 만든다. 생성된 청크를 기반으로 청크 단위의 의존 구문 구조를 사용하여 두 문장의 구문 구조 정보를 반영한다. 그 후, 전제와 가설 문장에 대한 관계를 분류하는 방법을 제안한다. 실험 결과, 본 논문에서 제안하는 모델은 정확도 90.78의 가장 높은 성능을 보였다. 향후 연구로, 본 논문에서 청크 간의 구문 구조 정보 벡터를 표현하는 방법으로 Graph Attention Network를 사용

2) Open Library KoalaNLP,

<https://koalanlp.github.io/python-support/html/>

하여 한 문장 내 모든 청크 간의 연결 정보를 반영하도록 실험할 예정이다.

### 감사의 글

이 논문은 2020년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임(No. 2013-0-00131, (엑소브레인-총괄/1세부) 휴먼 지식 증강 서비스를 위한 지능진화형 WiseQA 플랫폼 기술 개발). 또한, 이 논문은 2020년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임(No.2020-0-00368, 뉴럴-심볼릭(neural-symbolic) 모델의 지식 학습 및 추론 기술 개발)

### 참고문헌

- [1] Dagan, Ido, Oren Glickman, and Bernardo Magnini., “The pascal recognising textual entailment challenge”, In Machine Learning Challenges Workshop Springer (pp. 177-190)., 2005.
- [2] Lehmann, Erich Leo, Joseph P. Romano, and George Casella, “Testing statistical hypotheses”, springer, Vol. 3., 2005.
- [3] De Marneffe, Marie-Catherine, Bill MacCartney, and Christopher D. Manning., “Generating typed dependency parses from phrase structure parses”, Lrec, Vol. 6, (pp. 449-454). 2006.
- [4] Zachary C Lipton., “The Mythos of Model Interpretability” arXiv, 2016.
- [5] Supriyo Chakraborty, Richard Tomsett, Ramya Raghavendra, Daniel Harborne, Moustafa Alzantot, Federico Cerutti, Mani Srivastava, Alun Preece, Simon Julier, Raghuvver M Rao, et al., “Interpretability of Deep Learning Models: A Survey of Results”, IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computed, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation, (pp. 1-6), 2017.
- [6] Kumar, Sawan, and Partha Talukdar, “NILE: Natural language inference with faithful natural language explanations”, arXiv, 2020.
- [7] Zhao, X. and Vydiswaran, V.G. “LIREx: Augmenting language inference with relevant explanation”, arXiv, 2020.
- [8] Bauer, L., Deng, L. and Bansal, M. “ERNIE-NLI: Analyzing the Impact of Domain-Specific External Knowledge on Enhanced Representations for NLI”, In Proceedings of Deep Learning Inside Out (DeeLIO): The 2nd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures, (pp. 58-69). 2021.
- [9] A. Williams, N. Nangia, and S. R. Bowman, “A broad-coverage challenge corpus for sentence understanding through inference”, NAACL, 2018.
- [10] Bowman SR, Angeli G, Potts C, Manning CD, “A large annotated corpus for learning natural language inference”, arXiv, 2015.
- [11] McCoy, R.T., Pavlick, E. and Linzen, T. “Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference”, arXiv, 2019.
- [12] Gururangan S, Swayamdipta S, Levy O, Schwartz R, Bowman SR, Smith NA, “Annotation artifacts in natural language inference data”, arXiv, 2018.
- [13] Poliak A, Naradowsky J, Haldar A, Rudinger R, Van Durme B, “Hypothesis only baselines in natural language inference”, arXiv, 2018.
- [14] Timothy Dozat and Christopher D Manning, “Deep biaffine attention for neural dependency parsing”, In Proceedings of the International Conference on Learning Representations, 2017.
- [15] E. Kiperwasser and Y. Goldberg, “Simple and accurate dependency parsing using bidirectional LSTM feature representations”, TACL, 2016.
- [16] 박성식, 오신혁, 김홍진, 김시형, 김학수, “ELMo와 멀티헤드 어텐션을 이용한 한국어 의존 구문 분석”, 제30회 한글 및 한국어 정보처리 학술대회, (pp. 8-12), 2018.
- [17] Park, Sungjoon, Jihyung Moon, Sungdong Kim, Won Ik Cho, Jiyeon Han, Jangwon Park, Chisung Song et al., “KLUE: Korean Language Understanding Evaluation.”, arXiv, 2021.