

사전학습 언어모델과 GCN을 이용한 한국어 관계 추출

이제승^o, 김재훈

한국해양대학교, 컴퓨터공학과 및 해양인공지능융합전공
leeje1231@naver.com, jhoon@kmou.ac.kr

Korean Relation Extraction Using Pre-Trained Language Model and GCN

Je-seung Lee^o, Jae-hoon Kim

Dept. of Computer Engineering and Interdisciplinary Major of Maritime AI Convergence, Korea Maritime & Ocean University

요 약

관계 추출은 두 개체 간의 관계를 식별하는 작업이며, 비정형 텍스트를 구조화시키는 역할을 하는 작업 중 하나이다. 현재 관계 추출에서 다양한 모델에 대한 연구들이 진행되고 있지만, 한국어 관계 추출 모델에 대한 연구는 영어에 비해 부족하다. 따라서 본 논문에서는 NE(Named Entity)태그 정보가 반영된 TEM(Typed Entity Marker)과 의존 구문 그래프를 이용한 한국어 관계 추출 모델을 제안한다. 모델의 학습과 평가 말뭉치는 KLUE에서 제공하는 관계 추출 학습 말뭉치를 사용하였다. 실험 결과 제안 모델이 68.57%의 F1 점수로 실험 모델 중 가장 높은 성능을 보여 NE태그와 구문 정보가 관계 추출 성능을 향상시킬 수 있음을 보였다.

주제어: 관계 추출, 개체 위치 토큰, 개체명 표지, GCN

1. 서론

정보 추출(information extraction)은 비정형 텍스트를 구조화시키는 방법이다. 정보 추출을 통해 구조화된 데이터는 정보처리, 지식베이스 구축 등 다양한 분야에 응용될 수 있다. 그 중 관계 추출(relation extraction)은 문장 안에서 두 개체 간의 관계를 분류하여 정형화된 데이터를 만들어내는 작업인데, 이 때 생성되는 데이터는 주어, 술어, 목적어로 이루어진 데이터(트리플 데이터)이다. 아래 그림 1은 문장에서 두 개체(“아누스”와 “로마 제국”)의 관계를 나타낸다.



그림 1. 관계 추출 예시

관계 추출은 이처럼 두 개체가 각자 관계의 주어(“아누스”)와 목적어(“로마 제국”)으로 주어졌을 때, 문장의 맥락 속에서 어떤 관계인지 예측하는 것을 목표로 한다. 최근 관계 추출 연구에서 사전학습 언어모델을 사용해 문장에 개체(entity) 위치 토큰을 추가하여 해당 정보를 이용해 관계를 추출하는 방법[1]이 높은 성능보였고, NE(Named Entity)태그와 개체 위치 토큰을 이용한 방법인 개체 표지(Typed Entity Marker, TEM)와 문장부호를 이용한 개체명 표지(Typed Entity Marker (punct), TEM(punct))을 제안한 연구[2]등 많은 연구가 이루어졌으나, 이는 대부분 영어 기반 학습 말뭉치에서의 실험이다. 그러나 영어에 비해 한국어에서는 개체 위치 토큰을 이용한 관계 추출 연구가 많이 이루어지지 않았고, 또한

한국어에서 NE태그를 추가 정보로 사용하여 연구된 사례 또한 찾기 힘들다. 따라서 본 논문에서는 TEM(punct)[2]을 일부 변형하여 사용하였으며, 의존 구문 그래프와 그래프 기반 합성곱 신경망(Graph Convolution Network, GCN)을 이용해서 관계 추출 모델을 제안한다.

본 논문의 2장에서는 관련 연구를 간략히 소개하고, 3장에서는 제안 모델에 대해 자세히 기술한다. 4장에서는 실험 및 결과를 분석하고, 5장에서는 결론과 향후 연구 방향을 제시한다.

2. 관련 연구

사전학습 언어모델(BERT[3], ELECTRA[4], RoBERTa[5])은 현재 대부분의 자연어처리 작업에서 높은 성능을 보이는 모델이고, 최근의 관계 추출 연구에서는 대부분 사전학습 언어모델을 사용하고 있다. 본 장에서는 사전학습 언어모델 등장 이전과 이후의 관계 추출 연구를 소개하고, 이를 통해 본 연구의 필요성을 제시한다.

사전학습 언어모델이 등장하기 이전의 관계 추출 연구에서는 구문 구조 정보를 사용하는 방법이 주를 이루었다. 구문 최단 경로를 이용하여 필요한 정보만을 활용하려는 연구[6]와 두 개체 간의 최소 공통 부모를 중심으로 서브트리를 구축하여 그래프를 구성해 GCN[7]을 통해 인코딩하는 방법[8]등이 있다.

사전학습 언어모델의 등장 이후 구문 정보 없이 사전학습 언어모델을 사용하여도 이전의 연구보다 더 좋은 성능을 보여준 연구가 있었고[9], 사전학습 언어모델에서 개체의 시작과 끝을 나타내는 개체 위치 토큰을 사용하는 것이 성능이 높다고 알려졌으며[1], NE태그 정보를 사용하기 위해 개체 위치 토큰을 NE태그와 결합한 TEM 등의 모델을 사용한 연구[2]에서 기존 개체 위치 토큰만

을 사용한 모델보다 더 좋은 성능을 보였다. 한국어 관계 추출 연구에서 개체 위치 토큰을 사용한 연구는 한국어 특성에 맞게 개체에 부사를 포함하여 개체 위치 토큰을 다양하게 사용한 모델의 비교 연구[10] 등이 있지만, 영어 기반 연구에 비해 상대적으로 부족하다.

사전학습 언어모델 등장 이후 구문 정보를 이용한 연구는 활발히 연구되고 있지 않으며, 그 방법론에 대한 연구 또한 찾아보기 힘들다. 이에 본 논문에서는 사전학습 언어모델과 구문 정보를 함께 이용하는 방법을 제안한다.

3. 사전학습 언어모델과 GCN을 이용한 관계 추출

관계 추출 모델은 크게 전처리, 관계 추출 두 단계로 이루어지는데, 전처리 단계는 문장에 다양한 토큰을 추가하고 그래프를 생성하는 단계이며, 관계 추출 단계에서는 전처리 단계에서 생성된 문장 및 그래프를 이용하여 주어진 두 개체에 대한 관계를 얻는다. 전체 모델의 구조는 다음 그림 2와 같다.

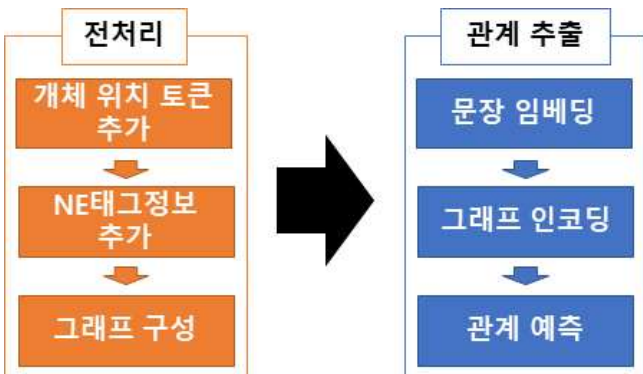


그림 2. 제안 모델의 구조

3.1 전처리 단계

전처리 단계는 (1)개체 위치 토큰 추가 단계, (2)NE 태그 정보 추가 단계, (3)의존 구문 정보를 바탕으로 그래프를 구성하는 그래프 구성단계로 이루어진다.

3.1.1 개체 위치 토큰 추가

개체 위치 토큰을 제안한 연구[1]에서 개체 위치 토큰을 사용하는 것의 성능이 높다고 알려졌으므로, 본 모델 또한 관계 추출 모델에서 개체의 위치 정보를 중심으로 학습할 수 있도록 문장에서 개체의 시작과 끝 위치에 토큰을 각각 그림 3과 같이 추가한다.

영상은 **[o]**로마 제국 **[/o]**의 **[s]**아누스 **[/s]**를 조명했다.

Object Subject

그림 3. 개체 위치 토큰이 추가된 문장

그림 3에서 “아누스”가 관계의 주어이고 “로마 제국”이 목적어이므로 “아누스” 단어의 양쪽에 주어를

나타내는 토큰 [s], [/s]을 추가하고, 목적어도 마찬가지로 목적어를 나타내는 [o], [/o]토큰을 추가한다. 본 논문에서 개체 시작 토큰은 [s], [o]를, 개체 끝 토큰을 [/s], [/o]를 의미한다.

3.1.2 NE태그 추가

관계 추출은 문장의 맥락 속에서 두 개체가 어떤 관계를 가지는지를 파악하는 것인데, 이를 위해 관계 추출 모델에서 개체에 대한 정보를 파악하는 것이 중요하다. NE태그는 관계 추출 모델에서 개체에 대한 추가적인 정보를 제공해줄 수 있으며, 개체의 중의성을 어느 정도 해결해 줄 수 있다[2]. NE태그별로 개체 위치 토큰을 따로 추가한 TEM과 NE태그를 단어로 바꾸고 새로운 토큰이 아닌 기존 문장부호 토큰으로 감싸서 추가한 TEM(punct)을 제안한 연구[2]에서 관계 추출시 NE태그 정보를 이용하여 성능을 향상시킬 수 있음을 보여주었다. 이에 본 논문에서는 [2]에서 가장 높은 성능을 보인 TEM(punct) 방법을 약간의 변형을 통해 사용하여 NE태그 정보를 추가 정보로써 이용한다. 이를 위해 먼저 NE태그를 한글로 변환하기 위해 NE태그의 의미와 그에 맞는 한글 단어를 찾는다. 실험에 사용한 KLUE-RE 말뭉치[11]에서의 개체에 대한 NE태그 데이터는 다음 표 1과 같이 변환된다.

표 1. 한글로 변환된 NE태그

NE태그	의미	한글 단어
PER	Person	인물
ORG	Organization	기관
LOC	Location	장소
POH	Other proper nouns	명사
DAT	Date and time	날짜
NOH	Other numerals	수량

이후 변환된 NE태그를 문장에 추가하기 위해 개체 위치 토큰을 추가한 것과 같이 NE태그 위치 토큰을 추가하는데, 주어(subject) NE태그의 앞과 뒤에 [st], [/st] 토큰들을 추가하고, 목적어(object) NE태그의 앞과 뒤에 [ot], [/ot]토큰들을 추가한다. 이후 문장 안에서 각 개체 끝 토큰인 [/s], [/o]뒤에 추가하여 최종으로 그림 4와 같은 문장이 만들어진다.

영상은 **[o]**로마 제국 **[/o]****[ot]**장소**[/ot]**의 **[s]**아누스 **[/s]****[st]**인물**[/st]**를 조명했다.

Object Object NE tag Subject Subject NE tag

그림 4. 개체 위치 토큰과 NE태그퐁치가 추가된 문장

본 논문에서 NE태그퐁치는 ([ot]...[/ot]), ([st]...[/st])를 뜻한다. 예를 들어 그림 4의 NE태그퐁치는 ([ot]장소[/ot]), ([st]인물[/st])를 의미한다. 또한 본 논문에서 NE태그 시작토큰은 [st]와 [ot]를, NE태그 끝 토큰은 [/st]와 [/ot]를 의미한다.

본 3.1.2절에서 TEM(punct)을 약간 변형하여 사용하였

다고 언급하였는데, 본 모델과의 차이점을 설명하기 위해 먼저 TEM(punct)을 간단하게 소개한다. TEM(punct)은 주어와 목적어 개체를 각각 “@”, “#”으로 감싸고, 주어와 목적어의 NE태그에 해당하는 단어를 각각 “*”, “^”으로 감싸 개체 시작 토큰 뒤에 넣는다. TEM(punct)으로 생성된 문장은 다음 그림 5와 같다.

영상은 # ^ 장소 ^ 로마 제국#의 @* 인물*아누스@를 조명했다.

그림 5. TEM(punct)방법으로 생성된 문장

TEM(punct)과 본 모델의 차이점은 총 3가지로 다음과 같다 : (1) TEM(punct)방법은 개체 시작과 끝 토큰이 모두 같고, 또한 NE태그 위치 토큰 또한 마찬가지로이다. 이는 모델이 개체의 시작과 끝 경계를 인식하는 데 어려움을 줄 수 있기 때문에 본 모델에서는 시작과 끝을 다른 토큰으로 처리하였다. (2) TEM(punct)방법에서는 새로운 토큰을 사용하지 않고 기존에 존재하는 문장부호 토큰을 사용하였는데, 문장에 동일한 문장 부호가 있는 경우 학습이 어려울 수 있기 때문에 모두 새로운 토큰으로 대체하였다. (3) TEM(punct)방법은 개체 시작 토큰 바로 뒤에 NE태그몽치가 배치되는데, 이는 임베딩 단계에서 개체 시작 토큰이 NE태그몽치에 더 많은 주의(attention)를 기울이게 할 수 있지만, 본 모델은 이후의 그래프 구성 단계에서 NE태그몽치 노드가 개체 시작 토큰 노드와 바로 연결되기 때문에 NE태그몽치를 개체 끝 토큰 뒤에 배치하여 개체 시작 토큰이 다른 정보에 더 주의를 기울일 수 있도록 하였다.

3.1.3 그래프 구성

2장에서 언급했다시피 사전학습 언어모델이 등장하기 이전의 관계 추출 연구에서는 의존 구문을 이용한 관계 추출 모델을 많이 사용하였는데, 이는 구문 정보가 관계 추출에 있어 중요한 역할을 한다는 것으로 해석할 수 있어 성능의 향상을 위해 의존 구문 그래프를 사용하였다.

KLUE-RE 말뭉치에서는 구문 정보를 제공하지 않아 토큰이 추가되지 않은 본래 문장을 공개된 의존 구문 분석 모델인 ¹⁾DependencyParserofKorean[12]을 사용하여 의존 구문을 얻는다. 그래프의 노드는 해당 모델의 구문 분석 단위인 어절로 이루어지고, 간선은 어절과 해당 어절의 head가 무방향으로 연결된다. 이후 개체 위치 토큰과 NE태그몽치가 다음과 같이 새로운 노드로 추가된다.

(1) 개체 시작 토큰 노드 추가

개체 시작 토큰에 해당하는 노드를 추가하고, 개체가 포함된 어절과 연결한다. 개체를 나타내는 어절이 2개 이상이면, 맨 앞쪽 어절과 연결한다.

(2) 개체 끝 토큰 노드 추가

개체 끝 토큰에 해당하는 노드를 추가하고, (1)과 같이 개체가 포함된 어절과 연결한다. 개체를 나타내는 어

절이 2개 이상이면, 맨 뒷쪽 어절과 연결한다.

(3) NE태그몽치 노드 추가

NE태그몽치에 해당하는 노드를 추가하고, 개체 시작 토큰 노드에 연결한다.

3.1.2절에서 생성된 문장(그림 4)의 문장에 대한 의존 구문 그래프는 다음 그림 6과 같이 구성되며, 모든 간선의 가중치는 동일하게 1로 설정하였다.

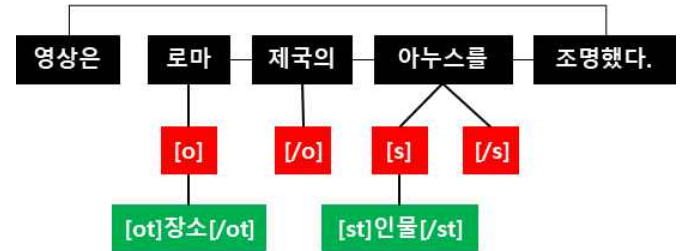


그림 6. 의존 구문 그래프 예시

3.2 관계 추출 단계

관계 추출 단계에서는 (1)문장을 사전학습 언어모델로 임베딩하는 문장 임베딩 단계, (2)이후 의존 구문 정보를 바탕으로 생성된 그래프를 GCN을 이용하여 인코딩하는 그래프 인코딩 단계, (3)마지막으로 GCN으로 인코딩된 개체 시작 토큰을 이용하여 관계를 예측하는 관계 예측 단계로 이루어진다.

3.2.1 문장 임베딩

3.1.2절에서 생성된 문장을 토큰화한 후 사전학습 언어모델을 이용해 문장의 토큰들을 임베딩한다. [s], [o] 등의 토큰은 기존에 존재하지 않는 토큰이므로 새로운 토큰으로 추가하고, 이를 통해 총 8개의 토큰이 새로 추가된다.

3.2.2 그래프 인코딩

하나의 노드는 여러 토큰으로 이루어질 수 있기 때문에 여러 토큰으로 이루어진 노드는 해당하는 토큰들의 벡터를 평균을 취하여 자질(feature)로 사용한다. 이후 각 노드의 자질(feature)과 3.1.3절에서 생성된 그래프로 인코딩하여 각 노드가 주변 노드의 정보를 반영하도록 한다.

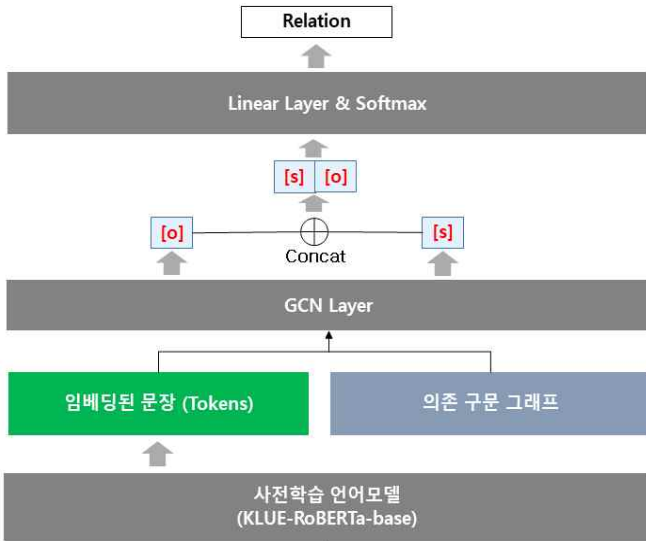
3.2.3 관계 예측

개체 위치 토큰을 제안한 연구[1] 이후 대부분 사전 학습 언어모델로 인코딩된 개체 시작 토큰만을 이용하여 관계를 추출하였고, 해당 방법이 가장 좋은 성능을 보였다. 따라서 본 논문에서도 관계 예측에 개체 시작 토큰인 [s], [o]에 해당하는 노드의 자질(feature)을 바탕으로

1) <https://github.com/yseokchoi/DependencyParserofKorean>

로 관계를 예측한다.

주어(subject)개체의 시작 토큰인 [s], 목적어(object)개체의 시작 토큰인 [o]에 해당하는 노드의 자질(feature)을 각각 h_s, h_o 라 할 때, h_s, h_o 는 연결연산(concat)을 통해 하나의 벡터로 합쳐지고, Linear layer를 통해 관계의 개수인 30차원의 벡터를 얻은 다음 softmax를 통해 관계별 확률을 얻는다. 그림 7은 관계 추출 단계의 전체적인 모델 구조이다.



영상은 [o]로마 제국 [o] [ot]장소 [ot]의 [s]아누스 [s] [st]인물 [st]를 조명했다.

그림 7. 관계 추출 단계의 모델 구조

4. 실험 및 평가

4.1 학습 및 평가 말뭉치

학습 및 평가에는 KLUE-RE 말뭉치[11]를 사용하였다. KLUE에서는 공식 평가 말뭉치를 공개하지 않아 개발용 말뭉치(development dataset)를 평가 말뭉치로 사용하였고, 학습말뭉치 32,470문장, 평가 말뭉치 7,765문장으로 실험하였다. 총 관계 수는 30개이며, KLUE-RE 말뭉치는 학습말뭉치에는 'no_relation' 클래스가 29%, 평가 말뭉치에서는 60%로 클래스 불균형 데이터셋(class imbalanced data)이다.

4.2 실험 환경

본 실험에서는 문장을 임베딩하기 위해 2)KLUE-RoBERTa-base를 사용하였으며, 임베딩된 각자의 토큰은 768차원을 가진다. 클래스 불균형 데이터를 효과적으로 학습하기 위해 손실 함수는 CB Focal loss[13]를 사용하였고, optimizer는 AdamW optimizer[14]를 사용하였으며, 본 실험의 초기 파라미터값은 표 2에 나타내었다.

2) <https://huggingface.co/klue/roberta-base>

표 2. 초기 파라미터

Parameter	Value
Epoch	10
Learning rate	2e-5
Batch size	16

4.3 실험 결과

각 모델의 성능은 KLUE-RE benchmark의 평가방법과 같이 AUPRC(Area Under the Precision-Recall Curve)와 'no_relation' 클래스를 제외한 미시 F1 점수(Micro F1 score)로 평가하였고, 학습 중 미시 F1 점수에서 최상의 성능을 내는 모델로 평가하였다. 실험 결과는 다음 표 3과 같다.

표 3. 실험 결과

모델	Micro F1-score	AUPRC
<i>개체 위치 토큰 모델</i>		
Hur <i>et al.</i> [10]	61.23	-
Base	65.91	62.06
↳with CLS	65.43	63.20
<i>NE태그 반영 모델</i>		
TEM(punct) [2]	66.65	61.86
Base + NE태그몽치	67.35	66.44
↳with CLS	67.61	67.69
↳with NE태그 시작 토큰	67.85	66.80
↳with NE태그 끝 토큰	67.97	68.61
제안 모델 (GAT)	66.51	62.63
제안 모델 (GCN)	68.57	65.22

표 3의 모든 모델은 개체 시작 토큰을 이용하여 결과를 예측하였고, with [토큰명]은 관계 추출 시 개체 시작 토큰과 함께 이용한 토큰으로, 개체 시작 토큰과 연결연산(concat)을 통해서 하나의 벡터로 만든 다음 Linear layer를 통해 관계 추출이 진행된 모델이다. 또한 CLS는 문맥 정보를 담고 있다고 알려진 사전학습 언어모델의 맨 처음 토큰인 [CLS]토큰을 나타낸다. 실험 결과 및 각 모델에 대한 설명은 아래에서 각자 절로 나누어 분석한다.

4.3.1 개체 위치 토큰 모델

개체 위치 토큰 모델은 3.1.1절의 개체 위치 토큰만 추가된 문장으로 실험한 모델이다. Hur *et al.* [10]의 연구와 Base모델은 같은 방식으로 엔티티 시작 토큰을

사용하였으나, 손실 함수와 같은 학습 방식과 평가모델 선택 방식 등의 차이로 인해 성능의 차이가 나타난 것으로 보인다.

4.3.2 NE태그 반영 모델

NE태그 반영 모델은 개체 위치 토큰 뿐 아니라 NE태그가 반영된 모델이다. TEM(punct)는 3.1.2절의 그림 5에 해당하는 문장으로 실험한 모델이고, Base+NE태그뭉치는 3.1.2절의 그림 4에 해당하는 문장으로 실험한 결과이다. Base+NE태그뭉치 모델에서는 개체 시작 토큰만을 이용하는 방식보다 다른 토큰을 함께 이용하는 방식이 더욱 높은 성능을 보여주었고, 그 중 NE태그 끝 토큰을 함께 사용하는 것이 가장 좋은 성능을 보여주었으며, TEM(punct)모델은 예상보다 낮은 성능을 보여주었다. 또한 모든 NE태그 반영 모델이 개체 위치 토큰 모델보다 높은 성능을 보여주어 한국어에서도 NE태그가 관계 추출 성능 향상에 도움이 된다는 것을 알 수 있다.

4.3.3 제안 모델

제안 모델 (GAT)는 3.2.2절의 그래프 인코딩 과정에서 GAT[15]를 사용한 모델이고, 제안 모델 (GCN)은 GCN으로 그래프를 인코딩한 모델이다. GAT와 GCN은 각각 3개의 Layer를 사용하였고, GAT의 head개수는 8개로 실험하였다. 실험 모델 중 제안모델 (GCN)의 성능이 가장 높았고, 이를 통해 사전학습 언어모델과 의존 구문 그래프를 함께 이용하여 관계 추출의 성능을 높일 수 있음을 보여주었다. 그러나 일반적으로 GCN보다 성능이 더 높다고 알려진 GAT가 GCN보다 떨어지는 성능을 보였고, 오히려 그래프를 사용하지 않은 base+NE태그뭉치 모델보다도 더 낮은 성능을 보여주었다. 이는 의존 구문 분석 오류가 GAT에 더 민감하게 반영된 것으로 유추된다.

5. 결론

본 연구에서는 영어 기반 연구에서 NE정보를 이용하여 높은 성능을 보인 TEM(punct)를 일부 변형하여 사용하고, 그에 더해 기존 연구에서 사용하던 의존 구문 정보를 이용한 그래프로 더욱 성능을 높이는 방법을 제안하였다. 실험 결과 NE태그를 사용하는 것이 개체 위치 토큰만을 사용한 모델보다 더 좋은 성능을 보였고, 그래프를 GCN으로 인코딩하여 사용하였을 때 가장 높은 성능을 보여 여전히 구문 구조 정보가 성능의 향상에 도움이 될 수 있음을 보였다. 추후 연구에서는 더욱 다양한 사전학습 언어모델을 이용하여 실험해 보고, 그래프를 토큰 단위로 구성하거나 노드의 연결 위치를 바꾸는 등 다양한 방식으로 구현하여 비교하고 실험 결과를 더욱 상세히 분석할 예정이다.

참고문헌

- [1] L. B. Soares, N. FitzGerald, J. Ling, and T. Kwiatkowski, "Matching the blanks: Distributional similarity for relation learning", Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 2895-2905, 2019.
- [2] W. Zhou and M. Chen, "An Improved Baseline for Sentence-level Relation Extraction", arXiv:2102.01373, 2021.
- [3] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding", Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pp. 4171-4186, 2019.
- [4] K. Clark, M. Luong, Q. V. Le, and C. D. Manning, "ELECTRA: Pre-training text encoders as discriminators rather than generators", Proceedings of International Conference on Learning Representations, 2019.
- [5] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and Veselin Stoyanov, "RoBERTa: A robustly optimized BERT pretraining approach", arXiv:1907.11692, 2019.
- [6] R. C. Bunescu and R. J. Mooney, "A shortest path dependency kernel for relation extraction", Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural, pp. 724-731, 2005.
- [7] T. N. Kipf and M. Welling, "Semi-Supervised Classification with Graph Convolutional Networks", Proceedings of International Conference on Learning Representations, 2017.
- [8] Y. Zhang, P. Qi, and C. D. Manning, "Graph convolution over pruned dependency trees improves relation extraction", Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pp. 2205-2215, 2018.
- [9] H. Wang, M. Tan, M. Yu, S. Chang, D. Wang, K. Xu, X. Guo, and S. Potdar, "Extracting multiple-relations in one-pass with pre-trained transformers", Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 1371-1377, 2019.
- [10] Y. Hur, S. Son, M. Shim, J. Lim, and H. Lim, "K-EPIC: Entity-perceived context

- representation in Korean relation extraction” , Appl. Sci. 2021, 11, 11472, 2021.
- [11] S. Park, J. Moon, S. Kim, W. Cho, J. Han, J. Park, C. Song, J. Kim, Y. Song, T. Oh, J. Lee, J. Oh, S. Lyu, Y. Jeong, I. Lee, S. Seo, D. Lee, H. Kim, M. Lee, S. Jang, S. Do, S. Kim, K. Lim, J. Lee, K. Park, J. Shin, S. Kim, L. Park, A. Oh, J. Ha, and K. Cho, “KLUE: Korean Language Understanding Evaluation” , arXiv:2105.09680, 2021.
- [12] Y. Choi and K. J. Lee, “Korean dependency parsing using stack-pointer networks and subtree information” , KIPS Transaction on Software and Data Engineering, 10(6) pp. 235~242, 2021.
- [13] Y. Cui, M. Jia, T. Lin, Y. Song, and S. Belongie, “Class-balanced loss based on effective number of samples” , Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 9268-9277, 2019.
- [14] I. Loshchilov and F. Hutter, “Decoupled weight decay regularization” , Proceedings of International Conference on Learning Representations, 2018.
- [15] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, “Graph attention networks” . Proceedings of International Conference on Learning Representations, 2018.