의존 구문 분석에 손실 함수가 미치는 영향: 한국어 Left-To-Right Parser를 중심으로

이진우[0], 최맹식, 이충희, 이연수

㈜엔씨소프트

aquamrn@ncsoft.com, mschoi@ncsoft.com, forever73@ncsoft.com, yeonsoo@ncsoft.com

Effects of the Loss Function for Korean Left-To-Right Dependency Parser

Jinu Lee[0], Maengsik Choi, Chunghee Lee, Yeonsoo Lee

NCSOFT Corp.

요 약

본 연구는 딥 러닝 기반 의존 구문 분석에서, 학습에 적용하는 손실 함수에 따른 성능을 평가하였다. Pointer Network를 이용한 Left-To-Right 모델을 총 세 가지의 손실 함수(Maximize Golden Probability, Cross Entropy, Local Hinge)를 이용하여 학습시켰다. 그 결과 LH 손실 함수로 학습한 모델이 선행 연구와 같이 MGP 손실 함수로 학습한 것에 비해 UAS/LAS가 각각 0.86%p/0.87%p 상승하였으며, 특히 의존 거리가 먼 경우에 대하여 분석 성능이 크게 향상됨을 확인하였다. 딥러닝 의존 구문 분석기를 구현할 때 학습 모델과 입력 표상뿐만 아니라 손실 함수 역시 중요하게 고려되어야 함을 보였다.

주제어: 의존 구문 분석, Left-To-Right Dependency Parser, 손실 함수

1. 서론

의존 구문 분석은 문장 구성 성분들의 문법적 의존 관계를 분석하여 문장을 하나의 구조화된 트리로 표현하는 작업이다. 의존 구문 분석을 위한 방법은 크게 전이 기반(Transition-based) 방법과 그래프 기반(Graph-based) 방법으로 나누어진다. 전이 기반 방법은 현재 상태를 기준으로 다음 전이 행동을 결정하는 지역적 탐색 방식이며, 그래프 기반 방법은 모든 의존 관계에 대한 확률을 계산한 뒤 최적 트리를 구하는 전역적 탐색 방식이다[1]. 딥 러닝을 이용한 전이 기반 모델로는 Stack-Ptr Net 모델[1] 등이 있으며, 그래프 기반 모델로는 Deep Biaffine 모델[2] 등이 알려져 있다.

본 연구에서는 한국어 문장을 위한 전이 기반 Left-To-Right Pointer Network 의존 구문 분석 모델[3]을 학습시킬 때, 적용하는 손실 함수에 따라 의존 구문 분석의 성능에 차이가 있음을 밝힌다.

2. 관련 연구

2.1. 의존 구문 분석과 손실 함수

기계 학습을 이용한 의존 구문 분석 연구는 새로운 학습 모델을 도입하거나 입력에 이용되는 단어 표상을 개선함으로써 성능을 향상시키는 것이 대부분이다. 그러나 기계 학습의 핵심적인 요소인 손실 함수가 의존 구문 분석에 미치는 영향에 대해서는 관련된 연구를 비교적 찾기 어렵다.

영문 의존 구문 분석에서 현재까지 손실 함수의 영향을 가장 정밀하게 분석한 연구는 Zhang, Ma & Hovy(2019)[4]이다. 그래프 기반 Deep Biaffine 모델에 다양한 손실 함수를 적용하여 성능을 비교하였고, 그 결과 손실 함수에 따라 최대 0.2%p 정도의 유의미한 성능 차이가 있음을 확인하였다.

2.2. 의존 구문 분석 모델

본 연구에서 구현한 의존 구문 분석 모델은 Left-To-Right Pointer Network[3]이다. Left-To-Right 모델은 Pointer Network를 기반으로 한 여러 모델 중 구조가 간단한 편이지만, 영어[3], 한국어[5] 등에서 의존 구문 분석 성능이 우수하다.

Left-To-Right 모델은 [그림 1]과 같이 문장의 앞쪽에서부터 순서대로 어절들을 입력하며 각 어절의 지배소를 예측하는 모델이다. 같은 Pointer Network 기반 모델인 Stack-Ptr Net 모델과는 의존 관계를 예측하는 방향이 반대이며, Left-To-Right 모델은 디코딩 순서가 순차적이므로 입력 순서를 결정하기 위한 별도의 스택을 요구하지 않는다는 차이점이 있다.

문장 x는 각 어절들의 표상(embedding)의 수열인 $\{x_{<root>}, x_1, x_2, \ldots, x_N\}$으로 정의된다. 주어진 문장은 먼저 Bi-LSTM 인코더에 입력되어 각 어절 x_i에 대한 은닉 표상(e_i)을 생성한다. 인코더의 마지막 상태를 이용하여 디코더의 시작 상태를 초기화한 뒤, 어절을 앞에서부터 순차적으로 디코딩한다. 어절 x_i에 대한 디코더 은닉 표상(d_i)은 모든 어절의 인코더 은닉 표상과 Biaffine

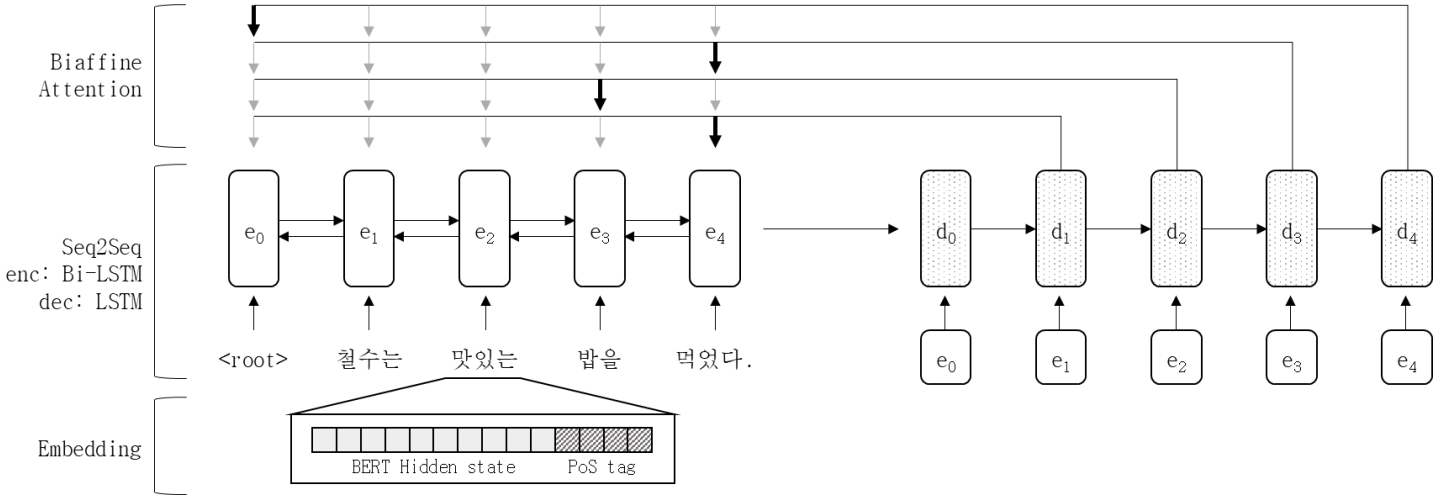

그림 1. Left-To-Right 의존 구문 분석 모델.

Attention이 계산되며, softmax 함수를 이용해 x_i가 특정 어절 x_j을 지배소로 가질 확률 a_i^j 역시 구해진다.

$$e_j = Bi - LSTM(x)_{t=j}$$
$$d_i = LSTM(x)_{t=i}$$
$$v_i^j = e_j \mathbf{W} d_i + \mathbf{b}$$
$$a_i = softmax(v_i)$$

[3]에서는 의존 구문 트리의 형태적 조건을 만족하면서 가장 확률이 높은 몇 개의 subtree를 매 step마다 갱신하여 보관하는 Beam Search 방식을 사용하였다. 본 연구에서는 Beam Search로 생성되는 구문 분석 트리가 지배소 후위 원칙, 지배소 유일 원칙, 투사성 원칙[6]을 항상 만족하도록 개선하여 적용하였다.

또한, 의존 관계 라벨(Dependency Label) 예측을 위하여 별도의 Biaffine Classifier를 학습시켰다. 이 모델은 모든 가능한 의존 관계에 대해 라벨을 예측하는데, 이 중 정답 의존 관계에 대해서만 역전파가 발생한다.

3. 실험 설계

3.1. 손실 함수

본 연구에서는 Left-To-Right 모델의 학습을 위한 여러 손실함수를 제안하고 손실 함수에 따른 학습 결과를 비교하였다. 본 연구에서 사용한 손실 함수는 Maximize Golden Probability[1][3], Cross Entropy[7], Local Hinge 총 3가지이다.

[그림 2]의 (a)와 (b)는 세 가지 손실함수의 차이를 보여주는 예시이다. 가상의 p_i 벡터 두 가지에 대하여 세 손실 함수를 각각 계산한 결과, 정답과 오답의 확률 분포에 따라 손실 함수의 값이 달라짐을 확인할 수 있다. Maximize Global Probability는 두 경우에서 같은 값을 가지고, Cross Entropy는 (a)의 경우에서 더 작으며, Local Hinge는 (b)의 경우에서 값이 더 작다.

세 손실 함수들의 상세한 수식적 정의 및 특징은 다음

a.

$L_{MGP}(x_1) =$ $-\log(0.5)$ $= 0.693$

$L_{CE}(x_1) =$ $-\log(1.0) - \log(1.0) - \log(0.9) - \log(0.6) - \log(0.5)$ $= 1.309$

$L_{LH}(x_1) =$ $\log(0.4) - \log(0.5) + 1 = 0.777$

p_1 | 0.0 | 0.0 | 0.1 | 0.4 | 0.5 |

b.

$L_{MGP}(x_1) =$ $-\log(0.5)$ $= 0.693$

$L_{CE}(x_1) =$ $-\log(0.9) - \log(0.8) - \log(0.9) - \log(0.9) - \log(0.5)$ $= 1.350$

$L_{LH}(x_1) =$ $\log(0.2) - \log(0.5) + 1 = 0.084$

p_1 | 0.1 | 0.2 | 0.1 | 0.1 | 0.5 |

그림 2. 가상의 예측 $\boldsymbol{p_i}$에 대한 MGP, CE, LH 손실 함수의 비교. 회색 칸은 주어진 의존소에 대해 지배소 어절을 올바르게 예측할 확률, 흰색 칸들은 나머지 어절들을 지배소로 잘못 예측할 확률이다.

과 같다.

3.1.1. Maximize Golden Probability (MGP)

MGP는 기존 Stack-Pointer Net 모델[1]과 Left-To-Right 모델[3]에서 사용한 손실 함수이다. 어떤 어절 x_i의 정답 지배소를 x_g라고 하였을 때, MGP 손실 함수는 다음과 같은 수식으로 정의된다.

$$L(x_i) = -\log(a_i^g)$$
$$L(x) = \sum_i L(x_i)$$

MGP는 실제 정답을 지목할 확률을 최대화하기 위한 손

실 함수이다. 따라서 오답에 대한 예측 결과의 분포는 고려되지 않는다는 특징이 있다.

3.1.2. Cross Entropy (CE)

텍스트 분류 작업에 주로 이용되는 Cross Entropy 손실 함수[7]를 Left-To-Right 모델에 적용하였다. CE 손실 함수는 정답에 대한 손실 함수 값은 MGP와 동일하지만, MGP와는 다르게 각 오답을 예측할 확률 역시 손실 함수의 값에 반영된다는 특징이 있다.

어떤 어절 x_i의 정답 지배소를 x_g라고 하였을 때, CE 손실 함수는 다음과 같은 수식으로 정의된다.

$$L(x_i) = -\log(a_i^g) - \sum_{j \neq g} \log(1 - a_i^j)$$
$$L(x) = \sum_i L(x_i)$$

3.1.3. Local Hinge (LH)

Hinge(혹은 Marginal) 손실 함수는 일반적으로 점수의 절대적인 크기가 아닌 상대적인 크기를 바탕으로 값이 결정되는 함수이다[7].

전이 기반 모델의 학습을 위하여 Left-To-Right 모델에 맞게 Hinge 함수를 재정의한 Local Hinge 손실 함수를 도입하였다. LH 손실 함수는 어떤 어절 x_i의 지배소를 예측할 때 정답 x_g를 올바르게 예측할 확률(a_i^g)과, 오답의 확률 중 최댓값($max_{j \neq g}(a_i^j)$)의 차이를 일정 수준 이상 벌리는 것만을 목표로 한다. 따라서 LH는 모든 오답의 확률이 0이 될 때까지 학습하는 CE보다 안정 상태를 달성하기 더 쉽다.

어떤 어절 x_i의 정답 지배소를 x_g라고 하였을 때, LH 손실 함수는 다음과 같은 수식으로 정의된다.

$$L(x_i) = \max\left(0, k - \log\left(\frac{a_i^g}{max_{j \neq g}(a_i^j)}\right)\right)$$
$$L(x) = \sum_i L(x_i)$$

수식에서 확인할 수 있듯이 LH 함수는 정답과 모든 오답의 확률 차이를 e^k배 이상으로 벌리면 학습이 일어나지 않는다. k는 Hyperparameter로서 임의의 양수이며, 본 연구에서는 편의상 $k = 1$로 사용하였다.

3.2. 어절 표상(Word embedding)

형태적으로 복잡한 언어인 한국어에서 어절의 표상을 구현하기 위하여 어절의 하위 자질들(Subword)의 표상들을 다양하게 조합하여 사용하였다. 사용한 하위 자질들의 종류는 다음과 같다.

3.2.1. BERT

사전 학습된 Wordpiece 기반 BERT-base 모델[8]을 사용하여 표상을 얻었다. BERT 모델의 사전 학습에는 한국어 Wikipedia, 인터넷 뉴스, 야구 뉴스 등 총 133M 개의 문장을 사용하였다[9]. BERT에 토큰화된 문장을 입력한 뒤, BERT 인코더의 가장 마지막 4개의 층을 가중 평균하여 Wordpiece들의 표상을 결정하였다. 어절의 가장 앞과 뒤에 위치한 Wordpiece 표상을 연결하여 총 1536 차원의 벡터를 얻고, 이를 어절 표상으로 사용하였다.

3.2.2. GloVe

한국어 Wikipedia 코퍼스를 사용해 사전 학습시킨 GloVe 모델[10]을 형태소 표상으로 도입하였다. 모델의 각 벡터는 300차원이며, 어절의 표현을 위해 가장 마지막 실질형태소와 가장 마지막 형식형태소의 표상을 연결한 600차원의 벡터를 사용하였다. 이 방법은 첫 형태소와 마지막 형태소를 잇는 방법이나 CNN을 이용하는 방법[3]보다 어절의 문법적 역할을 정확하게 반영할 수 있고, 사전 실험에서도 실제로 약 0.3%p의 성능 향상을 보였다. 해당 방법 및 이를 위한 실질형태소와 형식형태소의 구분은 선행 연구[11]를 따랐다.

3.2.3. 형태소 태그(Part of Speech; PoS)

형태소 태그 각각에 대하여 100차원의 표상 벡터를 무작위로 초기화하고, 이를 모델과 같이 학습시켰다. GloVe와 마찬가지로 가장 마지막 실질형태소와 가장 마지막 형식형태소의 표상을 연결하는 방식을 사용하였다.

3.2.4. 문자(Character; Char)

미등록어 문제를 보완하기 위하여 문자 표상을 도입하였다. 전체 모델과 같이 학습되는 Character-CNN 모델[12]을 이용하여 100차원의 표상 벡터를 얻었다.

4. 실험

4.1. 실험 환경

4.1.1. 코퍼스 정보

본 연구에는 세종 구구조 말뭉치를 변환한 의존 구문 말뭉치[13]를 사용하였다. 해당 말뭉치는 총 59,659 문장으로 구성되어 있으며, 이 중 90%를 학습 데이터로, 나머지 10%를 평가 데이터로 이용하였다.

4.1.2. 학습

본 연구의 Hyperparameter는 영어 Left-To-Right 모델[3]을 대부분 참고하였다. 단, 학습에 있어서 매 Epoch마다 Dev set의 손실 함수 값이 감소했는지를 기준으로 추가 학습 여부를 결정하는 Early Stopping 모델

(patience = 5)을 사용하였다.

4.2. 실험 결과

총 3가지의 자질 조합에 대하여, 각 손실 함수로 모델을 학습시킨 뒤 의존 구문 분석 모델의 성능을 비교하였다. 각 조건에 대한 UAS/LAS 성능 비교 결과는 [표 1]과 [그림 3]에 나타나 있다.[1]

모든 어절 표현 조합에서 손실 함수에 따른 모델의 성능 차이가 나타남을 확인할 수 있다. 기존 연구[3]에 사용된 MGP 손실 함수에 비해 새로 도입한 CE와 LH 손실 함수의 정답률이 더 높으며, 특히 LH의 경우 MGP에 비해 UAS와 LAS에서 모두 0.7~0.9%p가량 성능이 향상되었다.

표 1. 자질 조합과 손실 함수에 따른 평가 데이터의 UAS/LAS 성능.

	BERT+PoS		BERT+PoS+Char		GloVe+PoS+Char	
	UAS%	LAS%	UAS%	LAS%	UAS%	LAS%
MGP	90.92	88.56	90.98	88.56	89.79	86.53
CE	91.45	88.87	91.41	88.98	89.85	86.61
LH	**91.78**	**89.43**	91.72	89.40	90.53	87.41

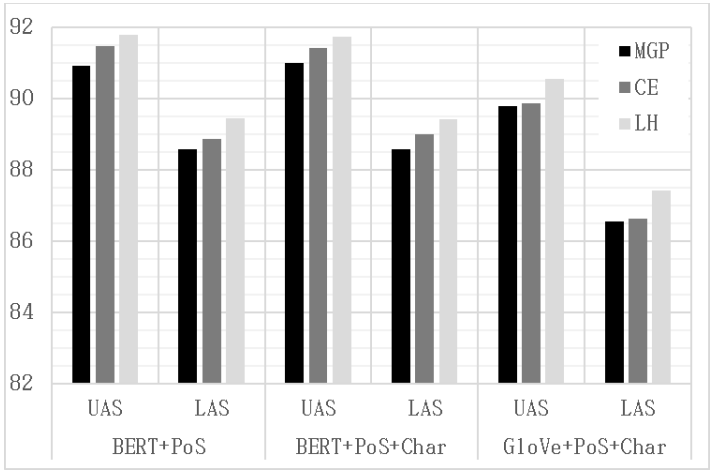

그림 3. 표 1의 결과.

[그림 4]는 문장의 길이에 따른 정확도이다. 문장의 길이(어절 수)를 5어절 단위로 구분하여 구간 별로 각각 UAS를 측정하였다. 대부분의 길이 구간에서 LH > CE > MGP 순서로 정확도가 높았다.

[그림 5]는 의존소와 지배소 사이의 거리에 따른 정확도이다. 의존 거리를 3어절 단위로 구분하여 구간 별로 각각 UAS를 측정하였다. 문장의 마지막 어절은 지배소가 항상 <root>이므로 별도로 구분하였다. 의존 거리가 4어절 이상일 때 문장의 길이와 마찬가지로 LH > CE > MGP 순서로 정확도가 높았다.

세 손실 함수 모두 문장이 길어지고 의존 관계가 멀수록 정답률이 낮아지는 경향을 보인다. 그러나 전역적인 확률 분포를 고려하는 손실 함수(CE, LH)들이 정답에 대한 정보만을 학습하는 MGP 손실 함수에 비해 학습된 모델이 의존 거리에 영향을 적게 받음을 확인할 수 있다.

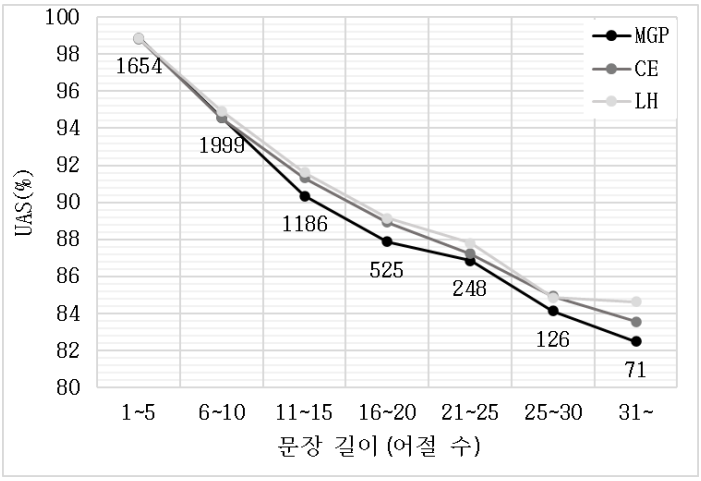

그림 4. 문장 길이에 따른 손실 함수 별 UAS. 각 항목 아래의 숫자는 해당 길이를 가지는 문장의 수를 의미한다.

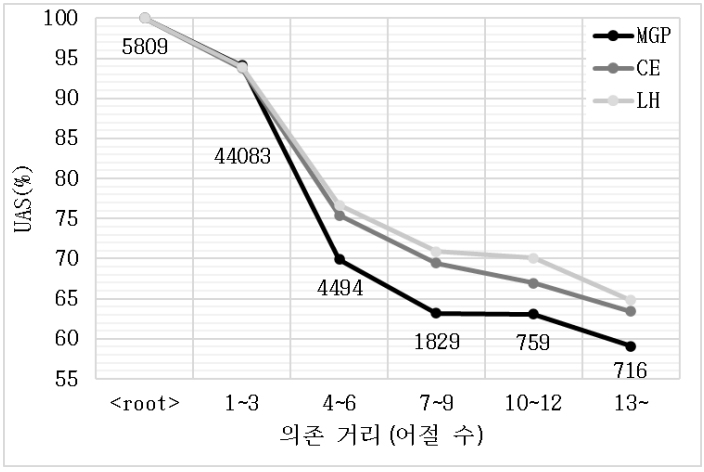

그림 5. 의존 거리에 따른 손실 함수 별 UAS. 각 항목 아래의 숫자는 해당 의존 거리를 가지는 의존 관계의 수를 의미한다.

원거리 예측 정답률 향상은 명사구(NP) 의존 관계 예측의 정답률 향상과도 밀접한 관련이 있다. 명사구 의존 관계는 일반적으로 다른 의존 관계에 비해 예측이 어렵고, 예측에 실패하면 평균적으로 더 많은 오류 전파를 유발한다[14]는 점에서 중요하다. 세종 의존 구문 말뭉치는 명사구 의존 관계의 의존 거리가 평균보다 멀어 명사구의 예측이 어렵다.

[표 2]는 세종 의존 구문 말뭉치에서 빈도가 높은 명

[1] Hyperparameter Tuning 과정 이후에도 한국어 Left-To-Right 모델을 구현한 선행 연구[5]의 성능을 완벽하게 재현하지 못했다. 연구에서 사용한 형태소 기반 ETRI KorBERT 대신 자체 Wordpiece BERT[9]를 사용한 등의 차이가 원인일 것으로 보인다.

사구 관련 의존 관계 라벨 3가지(NP_SBJ(주어), NP_AJT(부사어), NP_OBJ(목적어))에 대하여 각각의 의존 거리와 손실 함수 별 UAS 정답률을 비교한 것이다. 검토한 모든 명사구 의존 관계 라벨에서 LH > CE > MGP의 순서로 정답률이 높았다. 특히 의존 거리가 평균을 상회하는 주어와 부사어에서는 MGP와 LH 사이에 약 1.6%p의 성능 차이가 발생하였다.

표 2. 명사구 의존 관계에 대한 말뭉치 통계 및 손실 함수 별 UAS(%) 성능.

		전체	주어	부사어	목적어
총 수		57690	7095	6560	5144
총 비율%		100.0	12.30	11.37	8.92
평균 거리		1.98	3.23	2.85	1.38
UAS%	MGP	90.92	87.96	84.88	92.94
	CE	91.45	89.03	86.04	93.08
	LH	**91.78**	**89.57**	**86.48**	**93.49**

5. 결론

본 연구에서는 한국어 의존 구문 분석을 위한 Left-To-Right 모델을 중심으로, 학습에 사용되는 손실 함수가 최종 의존 구문 분석의 성능 및 결과에 유의미한 영향을 줌을 확인하였다.

MGP, CE, LH 세 가지의 손실 함수를 적용한 결과, LH 손실 함수를 사용해 학습시킨 모델이 가장 뛰어난 성능을 보였다. 특히 예측이 어려운 원거리 의존 관계와 명사구 의존 관계에 대한 정답률이 유의미하게 향상됨을 확인하였다.

대부분의 의존 구문 분석 연구에서 성능을 높이기 위한 방법으로 효율적인 입력 표상과 학습 모델을 제안한다. 그러나 본 연구의 결과는 같은 모델, 같은 입력 표상을 사용하고 손실 함수만 교체한 경우에도 성능에 유의미한 변화가 나타난다는 점을 시사한다. 따라서 의존 구문 분석의 성능 향상을 위해서 손실 함수의 선택에도 유의할 것을 제언한다.

향후 연구를 통해 손실 함수가 학습 과정에 미치는 영향을 확인하고, 한국어 의존 구문 분석에서 각 손실 함수가 가지는 의미를 통계적으로 분석하려고 한다. 또한, 기존 손실 함수들을 보완하여 의존 구문 분석 모델의 학습에 보다 적합한 손실 함수를 제안하고자 한다.

참고문헌

[1] Ma *et al.*, “Stack-Pointer Networks for Dependency Parsing”, Proceedings for ACL-2018, 2018.

[2] Dozat and Manning, “Deep Biaffine Attention for Neural Dependency Parsing”, Proceedings for ICLR 2017, 2017.

[3] Fernández-González and Gómez-Rodríguez, “Left-to-Right Dependency Parsing with Pointer Networks”, Proceedings for NAACL 2019, 2019.

[4] Zhang, Ma and Hovy, “An Empirical Investigation of Structured Output Modeling for Graph-Based Neural Dependency Parsing”, Proceedings for ACL 2019, 2019.

[5] 한장훈 외, “순차적 구문 분석 방법을 이용한 포인터 네트워크 기반의 한국어 의존 구문 분석기”, 제31회 한글 및 한국어 정보처리 학술대회 논문집, pp.533-536, 2019.

[6] 최맹식, 김학수, 정석원, “CRFs를 이용한 의존구조 분석 및 의존 관계명 부착”, 정보과학회논문지, 제41권, 제4호, pp. 302-308, 2014.

[7] Janocha and Czarnecki, “On Loss Functions for Deep Neural Networks in Classification”, Presented at TFML 2017, 2017.

[8] Devlin, Chang, Lee and Toutanova, “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”, arXiv preprint arXiv:1810. 04805, 2018.

[9] 소찬호 외, “대화 시스템의 개체 생략 복원을 위한 유효 발화문 인식”, 제31회 한글 및 한국어 정보처리 학술대회 논문집, pp. 54-59, 2019.

[10] Pennington, Socher and Manning, “Glove: Global Vectors for Word Representation”, Proceedings for EMNLP 2014, 2014.

[12] Kim, Jernite, Sontag and Rush, “Character-Aware Neural Language Models”, AAAI 2016, 2016.

[13] 임준호 외, “의존 구문분석을 위한 한국어 의존관계 가이드라인 및 엑소브레인 언어분석 말뭉치”, 제27회 한글 및 한국어 정보처리 학술대회 논문집, 2015.

[14] Ng and Curran, “Identifying Cascading Errors using Constraints in Dependency Parsing”, Proceedings for ACL 2015, 2015.