BERT를 이용한 한국어 의존 구문 분석

박천음*°, 이창기*, 임준호**, 김현기**
강원대학교 컴퓨터과학과*, 한국전자통신연구원**
{parkce, leeck}@kangwon.ac.kr, {joonho.lim, hkk}@etri.re.kr

Korean Dependency Parsing with BERT

Cheoneum Park*, Changki Lee*, Jun-Ho Lim**, Hyunki Kim**
Kangwon National University*, Electronics and Telecommunications Research Institute**

요 약

최근 BERT는 다양한 자연어처리 태스크에 적용되어 많은 성능 향상을 보이고 있다. BERT는 양방향성을 가진 트랜스포머(transformer)를 기반으로 언어 모델을 사전 학습하고 자연어처리 태스크를 위하여 출력층(layer)을 추가한 후에 fine-tuning한다. 의존구문분석은 문장 구조를 의존소(modifier)와 지배소(head) 간의 의존 관계로 표현한 자연어처리 태스크 중 하나이며, 어텐션(attention)을 이용한 모델들이 높은 성능을 보이고 있다. 본 논문에서는 한국어 의존구문분석을 해결하기 위하여 BERT를 이용한 어텐션 기반 의존 구문 분석 모델을 제안한다. 본 논문에서 사용하는 BERT는 한국어 특성을 반영하기 위하여 형태소 분석을 수행하고, OOV (Out Of Vocabulary) 문제를 해결하기 위하여 BPE (Byte Pair Encoding)를 적용한 대용량 코퍼스로 언어 모델을 학습하였다. 실험 결과, 본 논문에서 제안한 모델이 기존 한국어 의존 구문 분석 연구들 보다 좋은 (세종 코퍼스) UAS 94.06%, LAS 92.00%, (SPMRL) UAS 93.86%, LAS 93.30%의 성능을 보였다.

1. 서 론

최근 자연어처리 분야에서 높은 성능을 보이고 있는 BERT (Bidirectional Encoder Representations from Transformers) [1]는 대용량 코퍼스를 언어 모델로 학습한 모델이다. 사전 학습된 BERT 모델은 다양한 자연어처리 태스크를 위하여 출력 층(layer)을 추가한 후에 fine-tuning 하는 방법으로 사용된다. BERT는 양방향성을 가진 트랜스포머(transformer) [2]를 기반으로 하여 네트워크의 모든 레이어에서 전체 문맥 정보를 확인하여 언어 모델을 학습한다. 언어 모델 학습을 위하여 문장 내에서 임의의 단어에 대하여 마스킹(masking)하고 이를 예측하는 masked language modeling (masked LM)과 다음 문장 예측 기법을 적용한다.

구문 분석은 문장의 구조적, 의미적 중의성을 해결하기 위하여 문장 성분 사이의 관계를 분석하고 구조화하는 자연어처리 태스크 중 하나이며, 구구조 구문 분석(Phrase structure parsing)과 의존 구문 분석(Dependency parsing) 등이 있다. 한국어는 어순이 자유롭고 문장 성분의 생략이 빈번하기 때문에 문장구조를 지배소(head)와 의존소(modifier)로 이루어진 의존 관계로 표현하는 의존 구문 분석[3]이 주로 연구되었다. 의존 구문 분석은 의미 분석(상호참조해결, 의미역 결정, 개체명 인식 등)과 정보 추출, 온톨로지 확장, 질의응답, 문서요약 등에 응용될 수 있다.

의존 구문 분석을 해결하기 위하여 어텐션(attention) [4]을 이용한 딥 러닝 모델이 주로 사용되었다[5-8]. 그 중 [7]에서 제안된 biaffine attention을 이용한 모델은 입력된 문장에 대하여 의존소와 지배소의 문장 구조와 의존 관계의 스코어(score)를 biaffine 연산으로 계산하며, 좀더 간결한 방법으로 bilinear 연산을 사용할 수 있다. 본 논문에서는 한국어 의존구문분석을 위하여 한국어 대용량 코퍼스로 사전 학습한 BERT 모델 위에 LSTM RNN과 어텐션 층을 추가한 의존 파싱 모델을 제안한다.

2. BERT를 이용한 한국어 사전 학습

BERT는 여러 층(하나의 층은 트랜스포머의 블록 (block) L임)이 쌓인 양방향성 트랜스포머 인코더로 구성된다. [1]에서 BERT의 히든 레이어 차원 수를 H로, self-attention의 헤드(head) 수를 A라 정의하고, BERT-base 모델의 하이퍼 파라미터를 $L=12, H=768, A=12$로 구성하고, BERT-large 모델은 $L=24, H=1024, A=16$으로 구성한다. 본 논문에서는 BERT-base에 대하여 언어 모델을 학습한다. 사전 학습을 위하여 웹에서 수집한 뉴스 및 위키피디아 데이터를 사용하며, 조사를 사용하는 한국어 특성에 따라 입력된 모든 단어에 대하여 자동 형태소 분석을 수행한다. 언어 모델을 학습할 때 입력 형태소들을 subword (byte pair encoding, BPE) [9]로 토큰화(tokenization)하며, 단어 사전은 30,349개의 BPE 토큰들로 구성된다.

BERT는 트랜스포머의 인코더 부분을 기반으로 하며, 트랜스포머의 인코더는 멀티 헤드 어텐션(multi-head attention)과 FFNN (feed-forward neural network)으로 구성된다. 멀티 헤드 어텐션은 각 헤드마다 scaled dot-product attention으로 매트릭스 Q, K, V에 대한 어텐션 스코어를 계산하고, 모든 헤드의 어텐션을 연결(concat)한 것이며, 아래 식과 같다.

$$\mathrm{MultiHead}(Q,K,V)=\mathrm{concat}(head_1,\dots,head_h)W^O \quad (1)$$

$$head_i=\mathrm{Attention}(QW_i^Q, KW_i^K, VW_i^V) \quad (2)$$

$$\mathrm{Attention}(Q_i,K_i,V_i)=\mathrm{softmax}(Q_iK_i^T/\sqrt{d_k})V_i \quad (3)$$

트랜스포머 인코더에서 사용하는 멀티 헤드 어텐션은 셀프 어텐션(self-attention)과 같으며 자기 자신에 대한 구조 정보를 파악하여 어텐션 스코어를 계산한다. BERT는 트랜스포머의 인코더 부분을 사용하므로 Q,K,V는 같은 인코딩을 갖게 되며, 히든 스테이트(hidden state)의 차원 수는 h로 나뉘고, 각각 가중치 W_i^Q, W_i^K, W_i^V를 곱한다. 입력된 Q_i,K_i,V_i의 어텐션 스코어는 scaled dot-product attention을 이용하여 계산하며, 계산된 어텐션

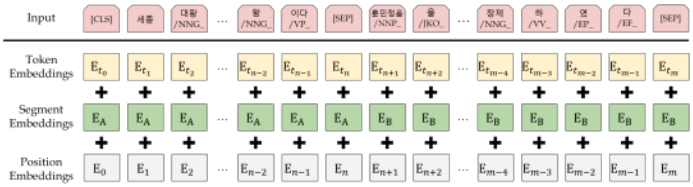

그림 1. BERT 입력 표현 예제

스코어는 $head_i$가 된다. h개의 $head_i$를 모두 연결하고 가중치 W^O를 곱한 값이 멀티 헤드 어텐션의 히든 스테이트가 된다. 이 후, 다음 레이어에서 position-wise FFNN인 $\max(0, xW_1 + b_1)W_2 + b_2$연산을 수행하여 트랜스포머 블록의 출력을 만든다.

본 논문에서의 BERT의 입력은 [그림 1]의 입력(Input)과 같이 입력 형태소에 BPE를 적용하여 토큰화 한 것이며, 각 형태소의 마지막 토큰에는 "_"을 붙여 형태소 단위를 구분한다. 입력된 각 토큰들은 토큰 임베딩(token embedding)을 적용하여 벡터화한다. 모든 입력열의 첫 번째 토큰으로는 [CLS]가 주어지며, 분류 문제를 해결할 때 해당 토큰의 벡터를 이용한다. BERT는 두 개의 열(sequence)를 연결하여 입력 받는데, 이때 두 입력 열을 구분하는 토큰으로 [SEP]를 사용하며, 각 열의 마지막 위치에 삽입한다. 첫 번째 [SEP] 토큰까지는 0으로 그 이후 [SEP] 토큰까지는 1 값으로 마스크를 만들고 세그먼트 임베딩(segment embedding)을 적용한다. 입력열의 위치 정보를 적용하기 위하여 토큰 길이 최대 512까지의 포지션 임베딩(position embedding)을 만들며, 앞서 언급한 각 임베딩들을 토큰 별로 모두 더하여 BERT의 입력 벡터로 사용한다.

[1]과 마찬가지로 본 논문에서도 masked LM과 다음 문장 예측 기법을 이용하여 한국어 코퍼스에 대한 사전 학습을 수행한다. 일반적인 언어 모델링은 한 방향으로만 학습이 가능지만, BERT의 경우에는 양방향성을 가지는 트랜스포머 인코더를 이용하기 때문에 입력 토큰의 일부를 임의로 마스킹하고, 마스킹 된 토큰을 예측하는 방법을 사용한다. 마스킹된 토큰의 히든 스테이트는 문맥 정보를 확인하여 단어 사전에 대한 softmax가 수행된다. 마스킹은 전체 토큰의 15%에 대하여 수행되지만, 그 중 80%만 [MASK] 토큰으로 변경하고, 10%는 임의의 토큰으로 바꾸며, 나머지 10%는 변경하지 않고 원래의 토큰을 유지한다. 다음 문장 예측은 입력된 문장 쌍이 실제 연결된 문장인지 아닌지를 학습하는 방법이다.

3. BERT 사전 학습을 이용한 어텐션 기반 의존 구문 분석 모델

어텐션 기반 의존 구문 분석 모델은 포인터 네트워크를 이용한 방법[5, 8, 10]이나 biaffine 어텐션을 이용한 모델[6, 7, 11] 등이 있다. 본 논문에서는 [그림 2]와 같이 사전 학습 된 BERT를 기반으로 하여 [7]과 같이 biaffine 어텐션과 bilinear 어텐션을 이용한 모델로 의존 구문 분석을 수행한다. 학습 데이터는 입력열 $X = \{x_1, x_2, \ldots, x_n\}$와 입력 자질 $F = \{f_1, f_2, \ldots, f_n\}$, 의존 구조 결과 $Y = \{y_1, y_2, \ldots, y_n\}$, 의존 관계 결과 $Z = \{z_1, z_2, \ldots, z_n\}$로 구성된다. 입력된 형태소는 BPE로 토큰화하여 토큰 임베딩을 얻고, 세그먼트 임베딩과 포지션 임베딩을 더하여 BERT의 입력 표현 E_i를 만든다. 그 후, 트랜스포머를 거쳐 사전 학습 된 BERT로부터 단어 표현 T_i를 얻고, 이를 기반으로 bidirectional LSTM (bi-LSTM) [12]을 수행하여 히든 스테이트 $\mathbf{r}_i$를 만든다. 이때 BERT 단어 표현 T_i와

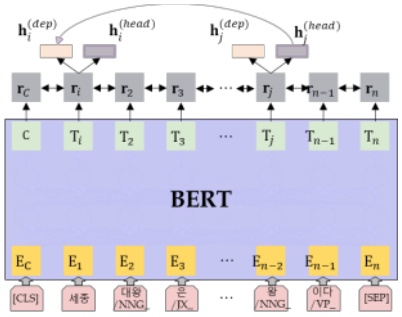

그림 2. BERT를 이용한 deep biaffine attention 모델

어절 범위 자질 임베딩, 형태소 범위 자질 임베딩을 함께 연결하여 bi-LSTM입력으로 넘긴다. 어절 범위 자질은 어절 단위로 표현되는 의존 구문 분석의 어절 범위 특성을 학습하기 위함이고, 형태소 범위 자질은 형태소 분석 결과의 형태소 범위 특성을 반영하기 위함이다. [7]에 따라 의존 파싱을 수행하기 위하여 각 i번째 토큰에 대하여 다음과 같이 비선형연산을 적용하여 히든 스테이트 $\mathbf{h}_i^{(dep)} = \text{elu}\left(\text{FFNN}^{(dep)}(\mathbf{r}_i)\right)$, $\mathbf{h}_i^{(head)} = \text{elu}\left(\text{FFNN}^{(head)}(\mathbf{r}_i)\right)$를 만든다. 이 때 사용되는 활성 함수(activation function)는 elu [13]이다. 본 논문에서 의존 구조 결과를 출력하기 위하여 어텐션을 수행할 때 bilinear 어텐션과 biaffine 어텐션 각각을 적용하며, 식은 다음과 같다.

$$s_{j,i}^{(arc)} = \begin{cases} \mathbf{h}^{\mathrm{T}}{}_i^{(dep)}\mathbf{U}\mathbf{h}_j^{(head)}, & bilinear \\ \mathbf{h}^{\mathrm{T}}{}_i^{(dep)}\mathbf{U}\mathbf{h}_j^{(head)} + \mathbf{w}^{\mathrm{T}}\mathbf{h}_j^{(head)}, & biaffine \end{cases} \quad (6)$$

위 식에서 bilinear는 $\mathbf{h}_j^{(head)}$에서 $\mathbf{h}_i^{(dep)}$에 대한 행렬곱으로 어텐션 스코어를 계산하고, biaffine은 bilinear에 bias 항을 더 추가하여 어텐션 스코어를 계산한다. 계산된 어텐션 스코어 $s_{j,i}^{(arc)}$는 softmax가 적용되어 의존 구조 결과를 출력한다.

Bilinear 어텐션 스코어를 계산할 때 의존 관계를 결정하기 위하여 $s^{(rel)} = softmax(\mathbf{w}\mathbf{r}_i) \in \mathbb{R}^{n \times l}$과 같은 연산을 수행한다. 여기서 l은 의존 관계 레이블 사전 크기이다. Biaffine 어텐션을 수행한 경우에는 식 (4, 5)와 같이 bi-LSTM 히든 스테이트 $\mathbf{r}_i$로부터 비선형연산을 적용하여 $\mathbf{h}_i^{(rel_dep)}$와 $\mathbf{h}_i^{(rel_head)}$를 만든다. 그 후에 의존 구조에서 결정된 head의 위치 y_i의 히든 스테이트에 대하여 식 (6)의 biaffine 어텐션 $\mathbf{h}^{\mathrm{T}}{}_{y_i}^{(rel_dep)}\mathbf{U}\mathbf{h}_j^{(rel_head)} + \mathbf{w}^{\mathrm{T}}\mathbf{h}_j^{(rel_head)}$으로 어텐션 스코어를 계산하고, softmax를 적용하여 의존 관계를 결정한다.

4. 실험

본 논문에서는 ETRI에서 사전 학습한 BERT 모델을 이용하였다. BERT를 사전 학습하기 위하여 사용한 데이터는 웹에서 수집한 뉴스 및 위키피디아 데이터이며, 총 23.5 기가바이트이다. 형태소 분석을 위하여 AIOpen [14]에서 공개한 ETRI 언어 분석기를 사용하였다. 사전 학습을 위한 BERT 하이퍼 파라미터는 다음과 같다. 구글에서 공개한 BERT-base (트랜스포머 블록 수: 12, 히든 레이어 차원 수: 768, 어텐션 헤드 수: 12) 옵션을 따르며, 각 히든 레이어의 활성 함수는 gelu [15]를 사용하

고, 히든 레이어의 드랍아웃(dropout)은 0.1, 언어 모델 학습을 위한 최대 문장 길이는 512로 설정하였다.

의존 구문 분석의 학습을 위하여 의존 구조로 변환된 세종 데이터 셋[16, 17]과 SPMRL'14 공개 데이터 셋[6, 18]을 사용하였다. 세종 데이터 셋은 총 59,659 문장으로, 90%인 53,842 문장을 학습에 사용하고, 10%인 5,817 문장을 평가에 사용하였다. SPMRL의 학습 셋은 23,010 문장, 개발 셋은 2,066 문장, 평가 셋은 2,287 문장으로 구성된다. 평가 척도는 Unlabeled Attachment Score(UAS)와 Labeled Attachment Score(LAS)를 사용하였다.

BERT를 이용한 의존 파싱 모델의 RNN 타입은 LSTM을 이용하며, LSTM의 드랍아웃은 0.1, LSTM의 히든 레이어 스택 수는 1, LSTM의 히든 레이어 차원 수는 BERT의 히든 레이어 차원 수와 같은 768로 설정하였다. 자질 표현의 차원 수는 1600, 의존 구문 분석을 위한 어텐션 레이어의 차원 수는 400으로 설정하였다. 학습율(learning rate)은 사전 학습 된 BERT를 fine-tuning 하기 때문에 5e-5로 설정하였고, 학습 알고리즘은 Adam [19]을 사용하며 Adam 가중치 감소(weight decay)는 1e-02로 설정하였다.

[표 1]은 세종 코퍼스에 대하여 본 논문에서 제안한 BERT 기반 attention 모델의 성능과 기존 한국어 의존 구문 분석 연구[5, 6, 10, 11, 16]들의 성능을 비교한 것이다. 본 논문에서 제안한 모델이 기존 연구 성능인 UAS 90.37~92.85% 보다 높은 성능을 보였다. 본 논문에서 사용한 biaffine 어텐션 방법이 UAS 94.06%, LAS 92.00%로 가장 좋은 성능을 보였고, bilinear 어텐션 방법이 UAS 93.85%, LAS 91.78%로 두번째로 좋은 성능을 보였다.

표 1. 세종 코퍼스 의존 구문 분석 성능 비교(자동 분석 형태소 이용)

Dependency parsing	UAS	LAS
이창기[16] with MI	90.37	88.17
나승훈[6]: deep biaffine attention	91.78	89.76
박천음[5]: 포인터 네트워크	92.16	89.88
안휘진[10]: deep biaffine + 스택 포인터 네트워크	92.17	90.08
박성식[11]: ELMo + 멀티헤드 어텐션	92.85	90.65
BERT + LSTM deep bilinear	93.85	91.78
BERT + LSTM deep biaffine	**94.06**	**92.00**

[표 2]는 SPMRL 데이터 셋에 대하여 본 논문에서 제안한 BERT 기반 attention 모델의 성능과 기존 연구들의 성능 비교를 보인다. 실험 결과, 본 논문에서 제안한 모델 중 biaffine 어텐션 방법이 UAS 93.86%, LAS 93.30%, bilinear 어텐션 방법이 UAS 93.87%, LAS 93.06%의 성능으로 기존 연구들에 비하여 좋은 성능을 보였다.

표 2. SPMRL 데이터 셋 의존 구문 분석 성능 비교(자동 분석 형태소 이용)

Dependency parsing	UAS	LAS
SPMRL' 14 Best [20]	89.10	87.27
나승훈[6]: deep biaffine attention	90.85	89.31
민진우[21]: deep biaffine attention + dual	91.07	N/A
BERT + LSTM deep bilinear	**93.87**	93.06
BERT + LSTM deep biaffine	93.86	**93.30**

5. 결론

본 논문에서는 한국어 대용량 코퍼스를 BERT로 사전 학습하고, 이를 어텐션 기반의 의존 구문 분석 모델에 적용할 것을 제안하였다. 교착어인 한국어 특성(조사 사용 등)을 잘 반영한 사전 학습을 위하여 형태소 분석을 수행하고 BPE를 적용하였으며, 어절 단위로 분석이 수행되는 의존 구문 분석의 특성과 BPE로 나눠진 입력 단어에 형태소 범위 특성을 반영하기 위하여 어절 범위 자질, 형태소 범위 자질을 추가하였다. 실험 결과, 세종 코퍼스에서 biaffine 모델은 UAS 93.87%, LAS 91.85%, bilinear 모델은 UAS 94.06%, LAS 92.00%, 그리고 SPMRL 데이터 셋에서 biaffine 모델은 UAS 93.86%, LAS 93.30%, bilinear 모델은 UAS 93.87%, LAS 93.06%로 기존 연구들에 비하여 더 좋은 성능을 보였다.

감사의 글

이 논문은 2019년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임 (No.2013-2-00131, 휴먼 지식증강 서비스를 위한 지능진화형 Wise QA 플랫폼 기술 개발)

이 논문은 2017년 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(NRF-2017M3C4A7068188)

참고문헌

[1] J. Devlin, et al. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv preprint arXiv:1810.04805, 2018.

[2] A. Vaswani, et al. Attention Is All You Need. *Neural Information Processing Systems (NIPS)*, pp. 5998-6008, 2017.

[3] D. Hays. Dependency theory: a formalism and some observations. *Language*, pp. 511-525, 1964

[4] D. Bahdanau, et al. Neural machine translation by jointly learning to align and translate. *Proc. of ICLR'15*, arXiv:1409.0473, 2015.

[5] 박천음, et al. 멀티 레이어 포인터 네트워크를 이용한 한국어 의존 구문 분석, *HCLT*, pp. 92-95, 2017.

[6] 나승훈, et al. Deep Biaffine Attention을 이용한 한국어 의존 파싱, *KCC*, pp. 584-586, 2017.

[7] T. Dozat and C. D. Manning. Deep biaffine attention for neural dependency parsing. arXiv preprint arXiv:1611.01734, 2016.

[8] X. Ma, et al. Stack-Pointer Networks for Dependency Parsing. *In Proc. of ACL*, pp. 1403-1414, 2018

[9] R. Sennrich, et al. Neural Machine Translation of Rare Words with Subword Units. *In Proc. of ACL*, pp.1715-1725, 2016.

[10] 안휘진, et al. Deep Bi-affine Network와 스택 포인터 네트워크를 이용한 한국어 의존 구문 분석 시스템. *HCLT*, pp. 689-691, 2018.

[11] 박성식, et al. ELMo와 멀티헤드 어텐션을 이용한 한국어 의존 구문 분석. *HCLT*, pp. 8-12, 2018.

[12] S. Hochreiter and J. Schmidhuber. Long Short-term Memory. Neural Computation, pp. 1735-1780, 1997.

[13] D. Clevert, et al. Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs). arXiv preprint arXiv:1511.07289v5, 2016.

[14] http://aiopen.etri.re.kr/

[15] D. Hendrycks and K. Gimpel. Gaussian Error Linear Units (GELUs). arXiv preprint arXiv:1606.08415v3, 2018.

[16] 이창기, et al. 딥 러닝을 이용한 한국어 의존 구문 분석. HCLT, pp. 87-91, 2014.

[17] 국립국어원. 21세기세종계획. 2012.

[18] D. Seddah, et al. Introducing the SPMRL 2014 Shared Task on Parsing Morphologically-Rich Languages, SPMRL-SANCL 2014, 2014.

[19] D.P. Kingma and J.L. Ba. ADAM: A METHOD FOR STOCHASTIC OPTIMIZATION. arXiv preprint arXiv:1412.6980v9, 2015.

[20] X. Zheng. Incremental Graph-based Neural Dependency Parsing. *In Proc. of EMNLP*, 2017.

[21] 민진우, et al. Dual Decomposition을 이용한 뉴럴 그래프 기반 한국어 의존 파싱. KCC 2018, pp. 643-645, 2018.