Second-Order TreeCRF를 이용한 한국어 의존 파싱

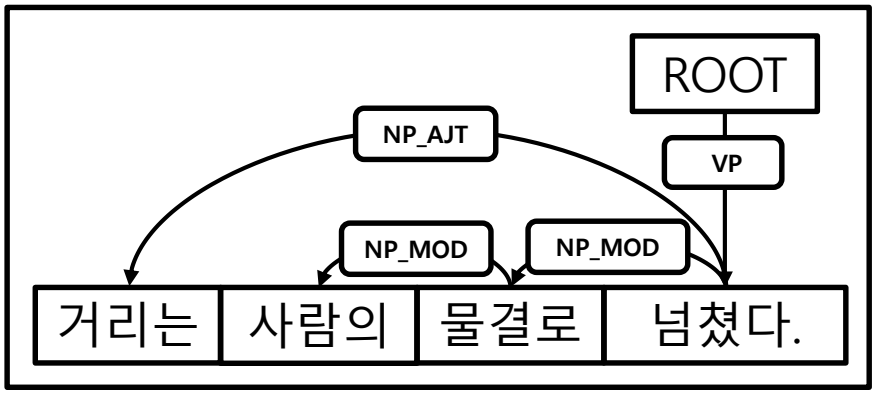
민진우[O1], 나승훈[1], 신종훈[2], 김영길[2]
[1]전북대학교 인지컴퓨팅연구실, [2]ETRI

목차

- 서론
- 관련연구
- Biaffine 어텐션 기반 한국어 의존 파서
- Second-Order TreeCRF를 이용한 한국어 의존 파싱
- 실험결과

구문 분석

문장의 구조를 결정하는 구문 분석(파싱)의 한 갈래로 단어의 Head(지배소)와 Modifier(의존소)의 관계에 따라 문장의 구조를 결정



– Paradigm

<전이 기반 방식>

버퍼와 스택으로부터 자질을 추출한 후 모델을 통해 다음 전이 액션을 결정하고 결정된 액션에 따라 버퍼와 스택의 상태를 갱신하면서 트리를 생성해 나가는 방식

<그래프 기반 방식>

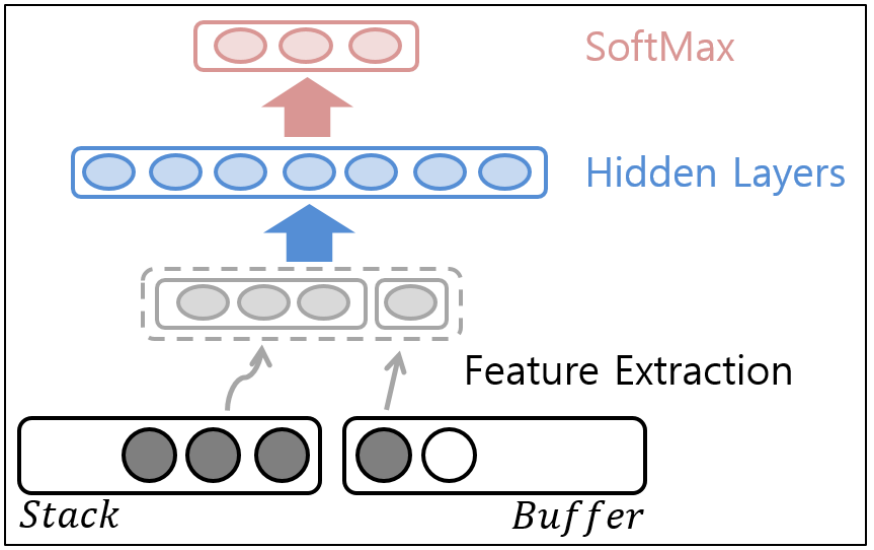
문장 내의 가능한 모든 단어 쌍의 의존관계를 찾아서 점수화하여 가장 높은 점수를 갖는 의존 트리를 찾는 방식

지역적 탐색 모델

전역적 탐색 모델

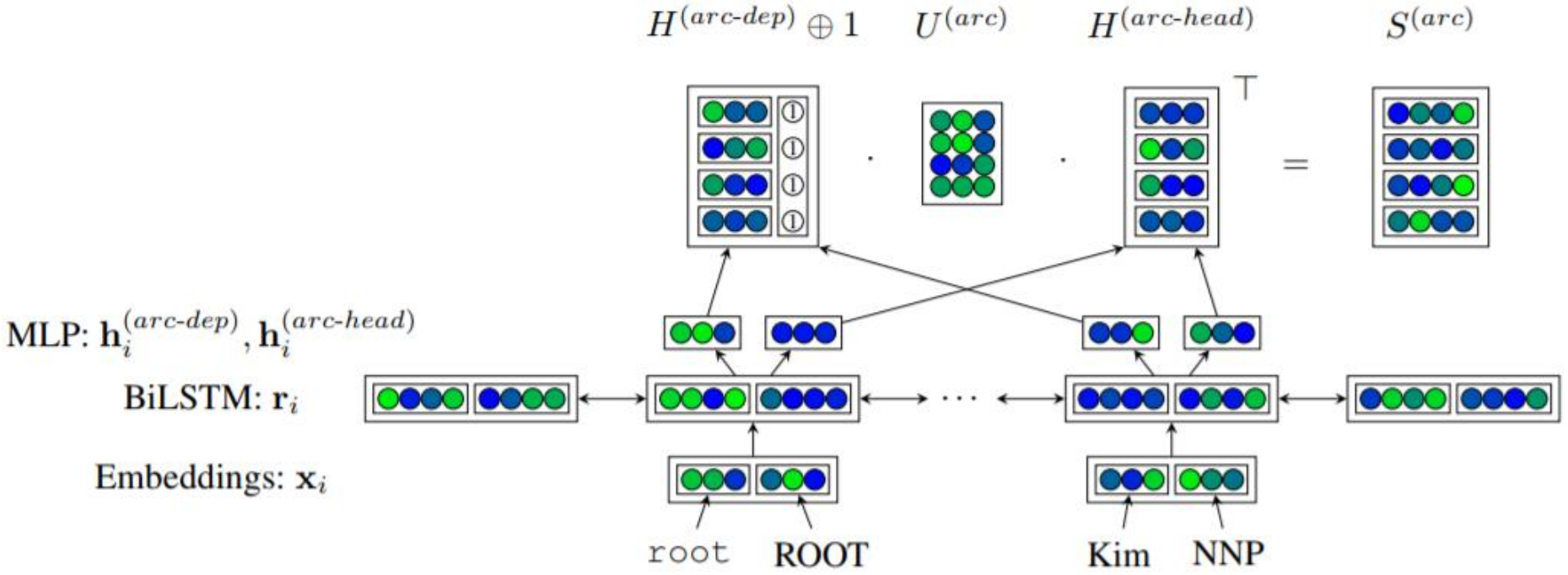
전이기반 파서

Action	S_t	S_{t+1}
$SHIFT$	$(\sigma, b_0 \mid \beta, T)$	$(\sigma \mid b_0, \beta, T)$
$LEFT(l)$	$(\sigma \mid s_1 \mid s_0, b_0 \mid \beta, T)$	$(\sigma \mid s_1, b_0 \mid \beta, T \cup \{b_0, s_0, l\})$
$Right(l)$	$(\sigma \mid s_1 \mid s_0, \beta, T)$	$(\sigma \mid s_1, \beta, T \cup \{s_1, s_0, l\})$



- 전이 시스템
  - Arc-Standard : [Shift, Left_Arc, Right_Arc]
    - Stack의 Top Element 사이에서 의존성을 결정
  - Arc-Eager: [Shift, Reduce, Left_Arc, Right_Arc]
    - Stack과 Buffer의 Top element 사이에서 의존성 결정
  - Arc-Hybrid
    - Arc-Standard와 Arc-Eager의 혼합의 형태로 LeftArc는 Arc-Eager 방식, Right-Arc는 Arc-Standard 방식을 사용

Deep Biaffine Attention for Neural Dependency Parsing (Dozat, T. ICLR '2017)



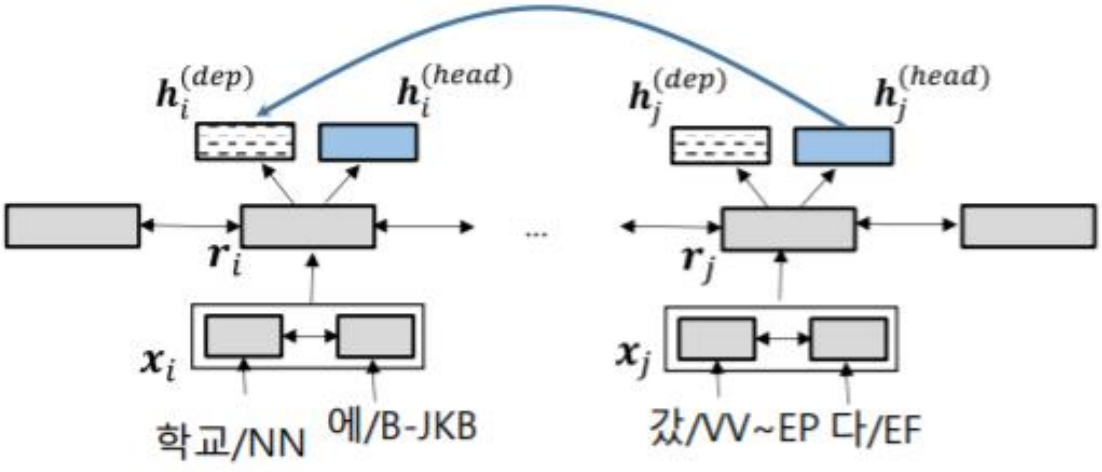
$$S_i^{(arc)} = h_i^{(arc-dep)} U h^{(arc-head)} + w^T h^{(arc-head)}$$

- 그래프 기반 방식
- Biaffine Attention을 이용하여 모든 단어 쌍에 대한 점수를 계산

Deep Biaffine Attention을 이용한 한국어 의존 파싱 (나승훈, KCC '2017)

Biaffine attention

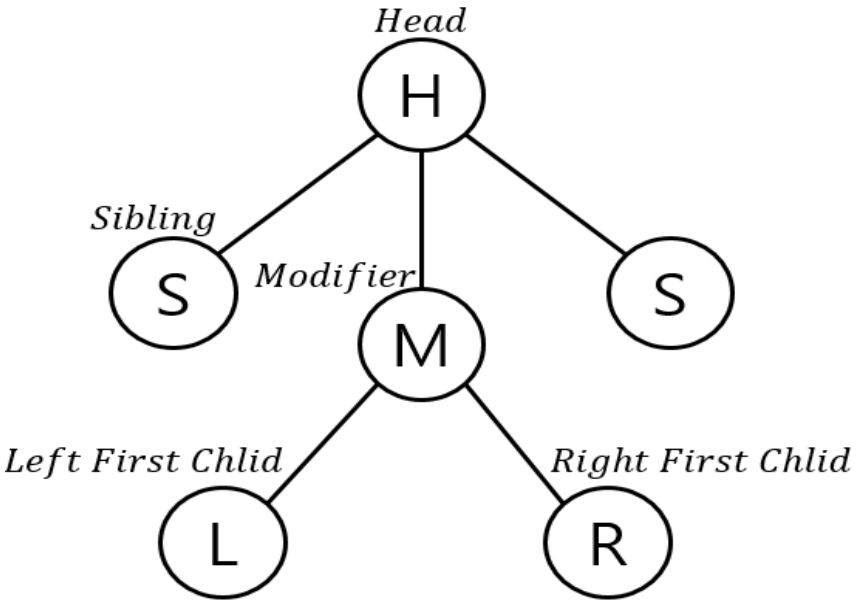
$$s(j,i) = \boldsymbol{h}_i^{T\,(dep)} \boldsymbol{U} \boldsymbol{h}_j^{(head)} + \boldsymbol{w}^T \boldsymbol{h}_j^{(head)}$$



- $\boldsymbol{x}_i$: Word representation using LSTM-based composition
- $\boldsymbol{r}_i$: $\mathrm{BiLSTM}_i(\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n)$
- $\boldsymbol{h}_i^{(dep)} = \mathrm{MLP}^{(dep)}(\boldsymbol{r}_i)$
- $\boldsymbol{h}_i^{(head)} = \mathrm{MLP}^{(head)}(\boldsymbol{r}_i)$

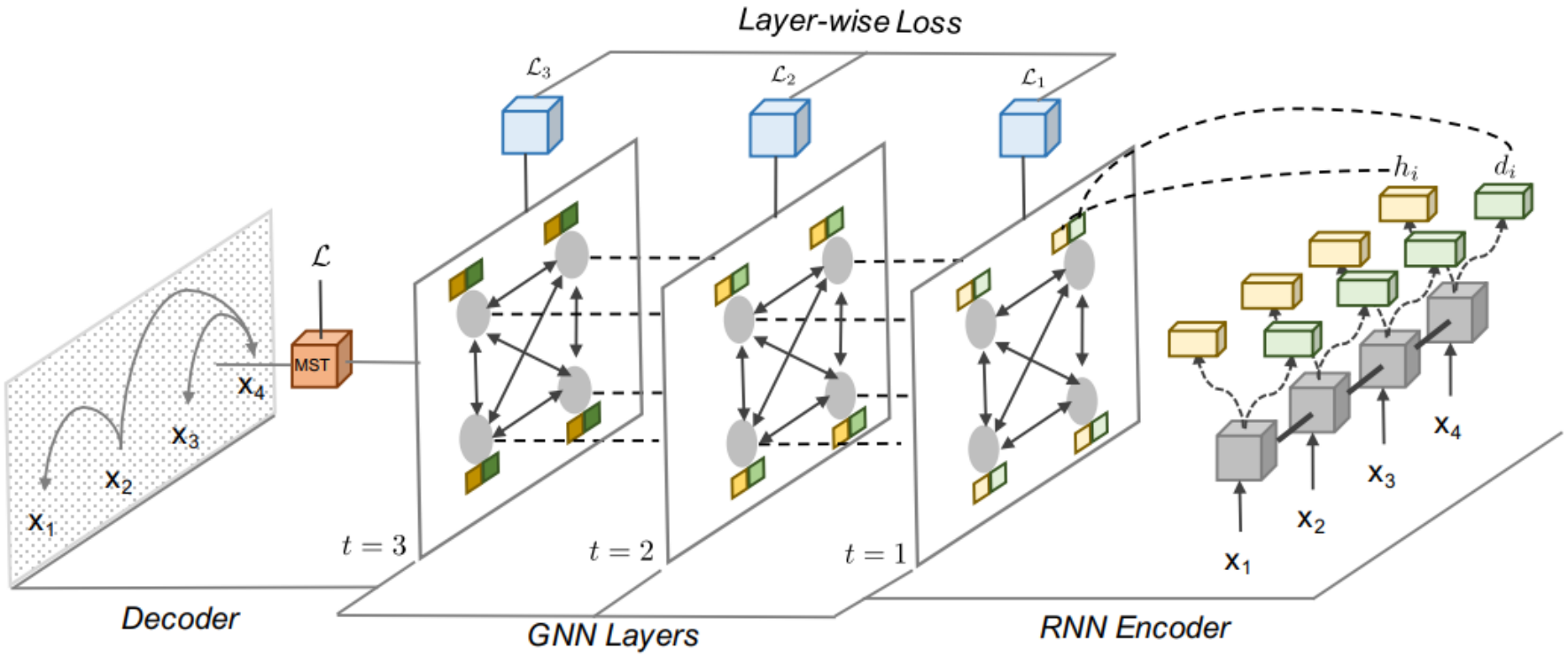
- Deep Biaffine Attention Model을 한국어와 같이 형태학적으로 복잡한 언어에 맞게 확장한 모델
- 단어(어절)을 구성하는 형태소 열을 LSTM으로 합성하여 단어 표상을 구성

2nd Order Sibling 모델



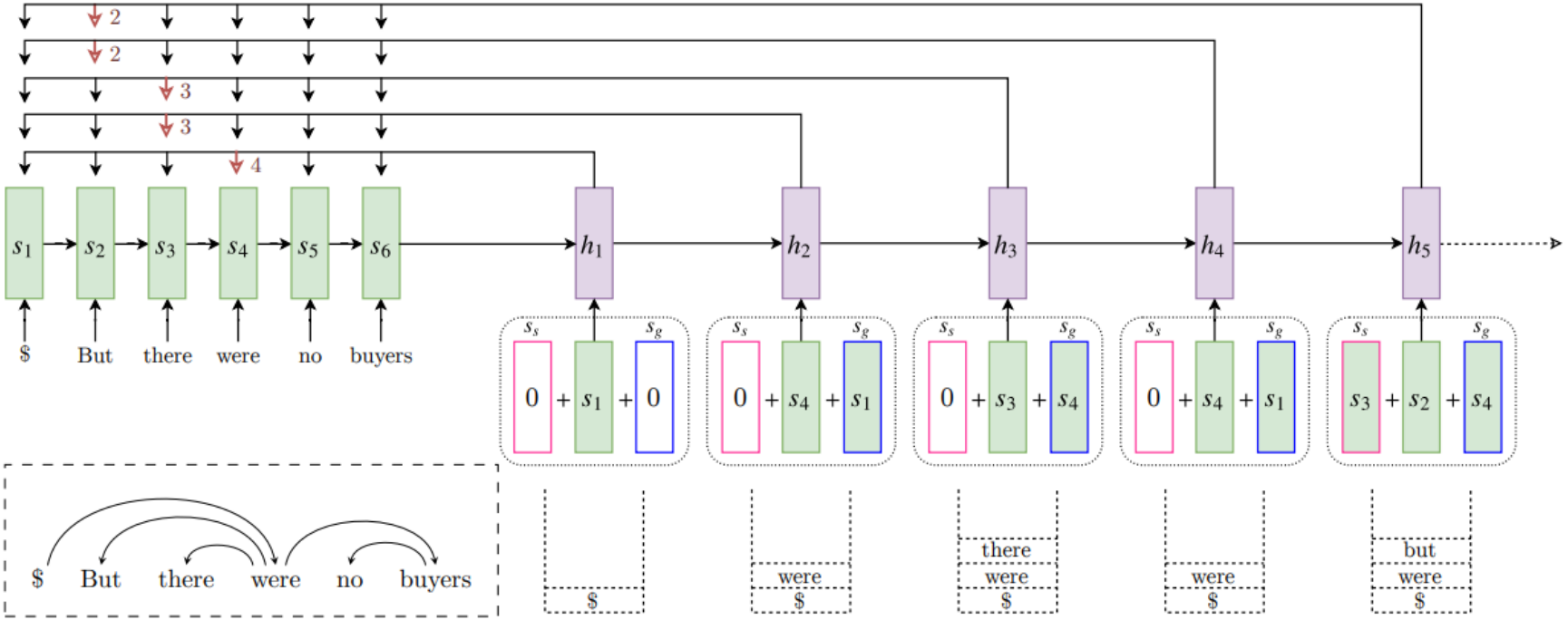
- First-order model: $f(h, m)$
  - 주어진 m(modifier)에 대해서 h(head)를 선택하는 모델
- Second-order Sibling model: $g(h, m, s)$
  - 주어진 h(head)에 대해서 left (and right) children을 n-gram sibling model로 선택하는 모델

Graph-based Dependency Parsing with Graph Neural Networks(Tao Ji, ACL '19)



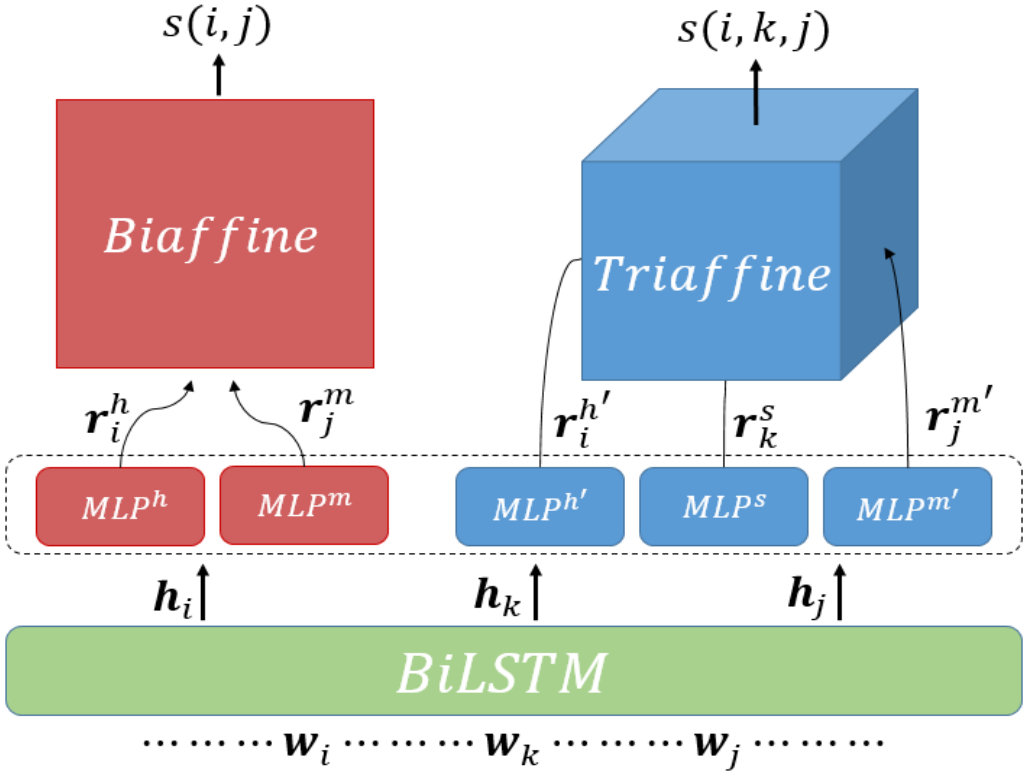
- Deep Biaffine Attention모델에 대해 일종의 그래프로 보고 지배소에 대한 표상과 의존소의 표상을 Graph Neural Network를 통해 High-Order Feature 자질을 학습한 후 최종 층에서 Biaffine을 수행하는 모델

Stack-Pointer Networks for Dependency Parsing (Xuezhe Ma, ACL '18)



- Root에서 시작하여 스택의 노드에 대해 자식을 선택하는 Top-Down 방식의 전이 기반 모델
- 디코더의 입력으로 선택된 노드에 대한 표상 뿐 아니라 sibling, grand-parent 노드의 정보까지 더하여 성능 향상

Second-Order TreeCRF를 이용한 한국어 의존 파싱



- 같은 지배소를 공유하는 형제(sibling)에 대한 점수를 얻고 이를 전체 트리에 대한 구조적 손실을 계산하는 Second-Order TreeCRF(Yu Zhang, ACL '20)를 한국어 의존파싱에 적용

Second-Order TreeCRF를 이용한 한국어 의존 파싱

- 인코더
  - 입력 표상
    - 형태소 단위로 LSTM을 통 합성한 단어 표상 : $\boldsymbol{w}_i$
    - 어절의 시작 형태소와 끝 형태소의 태그를 결합한 어절 태그 : $\boldsymbol{t}_i$
    - BERT 기반 인코더를 통해 얻어진 어절의 마지막 위치의 토큰 : $\boldsymbol{b}_i$

$$\boldsymbol{x}_i = [\boldsymbol{w}_i; \boldsymbol{t}_i; \boldsymbol{b}_i;]$$

  - Bi-LSTM Encoder
    - 입력열 $\{x_1, \cdots x_n\}$을 다층의 Bi-LSTM을 통한 인코딩

$$\{\boldsymbol{z}_1, \cdots \boldsymbol{z}_n\} = \mathrm{BiLSTM}(\{\boldsymbol{x}_1, \cdots \boldsymbol{x}_n\})$$

Second-Order TreeCRF를 이용한 한국어 의존 파싱

- Biaffine Parser
  - 지배소, 의존소에 대한 표상

$$\boldsymbol{h}_i = MLP^h(\boldsymbol{z}_i), \boldsymbol{d}_i = MLP^d(\boldsymbol{z}_i)$$

  - Biaffine Attention
    - $s(i,j)$는 j번째 단어(의존소)에 대해 지배소 단어 i의 점수

$$s(i,j) = \begin{bmatrix} \boldsymbol{d}_j \\ 1 \end{bmatrix}^T W^{biaffine} \boldsymbol{h}_i$$

- 손실 함수
  - Local token-wise 손실 함수의 적용
    - 전체 트리 구조에 대한 손실이 아닌 문장 내의 각 토큰에 대해 독립적으로 정답 지배소에 대한 확률을 최대화하는 cross-entropy 손실 함수 적용

$$L(i,j) = -\log \frac{e^{s(i,j)}}{\sum_{0 \le k \le n} e^{s(k,j)}}$$

Second-Order TreeCRF를 이용한 한국어 의존 파싱

- Second-Order TreeCRF
  - 지배소, 의존소를 포함한 형제에 대한 표상

$$\boldsymbol{h}_i' = MLP^{h'}(\boldsymbol{z}_i), \boldsymbol{d}_i' = MLP^{d'}(\boldsymbol{z}_i), \boldsymbol{s}_i = MLP^{s}(\boldsymbol{z}_i)$$

  - Triaffine Attention 함수 적용
    - 의존소 j와 지배소 i가 주어졌을 때 i와 같은 지배소를 공유하는 인접한 형제 노드 k에 대한 점수

$$s(i,k,j) = \begin{bmatrix} \boldsymbol{s}_k \\ 1 \end{bmatrix}^T \boldsymbol{h}_i' W^{biaffine} \begin{bmatrix} \boldsymbol{d}_j' \\ 1 \end{bmatrix}$$

  - 손실 함수
    - 정답 트리 $s(x,y)$에 대한 조건부 확률

$$p(y|x) = \frac{e^{s(x,y)}}{z(x) \equiv \sum_{y' \in Y(x)} e^{s(x,y')}}$$

    - 입력 문장에 대해 정답 트리의 조건부 확률 을 최대화 하는 손실 함수 적용

$$L(x,y) = -\log p(y|x) = -s(x,y) + \log Z(x)$$

Second-Order TreeCRF를 이용한 한국어 의존 파싱

- Second-Order TreeCRF
  - 정규화 항 $z(x)$를 계산하는 Inside 알고리즘 적용

Algorithm 1 Second-order Inside Algorithm.

define: $I, S, C \in \mathbb{R}^{n \times n \times B}$ ▷ B is #sents in a batch
initialize: $C_{i,i} = \log e^0 = 0, 0 \leq i \leq n$
for $w = 1$ **to** n **do** ▷ span width
  Batchify: $0 \leq i; j = i + w \leq n$
  $I_{i,j} = \log \left(e^{C_{i,i}+C_{j,i+1}} + \sum_{i<r<j} e^{I_{i,r}+S_{r,j}+s(i,r,j)} \right) + \text{s(i, j)}$
  $S_{i,j} = \log \sum_{i \leq r < j} e^{C_{i,r}+C_{j,r+1}}$
  $C_{i,j} = \log \sum_{i < r \leq j} e^{I_{i,r}+C_{r,j}}$
end for ▷ refer to Figure 3
return $C_{0,n} \equiv \log Z$

실험 세팅

- 데이터 셋

	Train	Dev	Test
세종 구문 분석 데이터 셋	52832	1000	5817
SPMRL'14 한국어 공개 데이터 셋	23010	2066	2187

- 실험 세팅
  - CRF를 이용한 자동 형태소 분석 결과(F1: 97.60%) 이용
  - 사전 학습 임베딩 임베딩
    - Glove 알고리즘을 통해 "형태소/태그" 단위로 사전 학습한 100차원의 형태소 임베딩 사용
  - BERT 모델
    - 자체 학습한 BERT Large모델 사용.

실험 세팅

- 하이퍼 파라미터

	Hyper Parameter	value
BERT	인코더 블록 개수	12
	은닉 차원수	768
	어텐션 헤드 수	12
	최대 문장 길이	512
	활성함수	gelu
	Optimizer	Adam
	학습률	$1.5e^{-5}$
Model	Optimizer	Adam
	학습률	0.001
	Dropout rate	0.33
Encoder	형태소 임베딩	100
	어절 태그 임베딩	50
	BERT 임베딩	768
	Char RNN 은닉 차원 수	512
	RNN 은닉 차원 수	512
	RNN Layers	3
Transition	자질 차원 수	64
Graph	Arc MLP	512
	Label MLP	128

실험 결과

– 세종 구문 분석 데이터 셋에서의 실험 결과

	UAS	LAS
Stack LSTM	90.11%	87.70%
Stack LSTM + 컨트롤러	90.83%	88.49%
Bert + ELMo + ML biaffine	93.06%	90.07%
Bert + Biaffine	94.06%	92.00%
(Roberta Large) Biaffine 어텐션	94.31%	92.42%
(Roberta Large) 1st Order Tree CRF	**94.47%**	**92.57%**
(Roberta Large) 2nd Order Tree CRF	94.34%	92.40%

실험 결과

– SPMRL'14 한국어 공개 데이터 셋에서의 실험 결과

	UAS	LAS
Stack LSTM	89.10%	87.34%
Stack LSTM + 컨트롤러	91.84%	91.29%
Bert + ELMo + ML biaffine	93.34%	92.85%
Bert + Biaffine	93.87%	93.06%
(Roberta Large) Biaffine 어텐션	94.64%	94.12%
(Roberta Large) 1st Order Tree CRF	**94.84%**	**94.40%**
(Roberta Large) 2nd Order Tree CRF	94.53%	94.04%

결론

- 결론
  - 베이스 라인 모델인 Biaffine 어텐션 모델 대비 1nd Oder TreeCRF 세종 셋에서 UAS : 0.16%, LAS : 0.15% 그리고 SPMRI'14 셋에서 UAS : 0.20%, LAS : 0.28% 성능 향상
  - 영문과 달리 형제 노드에 대한 정보를 반영하는 2nd Oder TreeCRF 모델은 성능 향상에 저하를 주어 어떠한 영향을 미치는 지에 대한 연구 필요

Q&A

감사합니다.