Stack Pointer Network를 이용한 한국어 형태소 분석

민진우, 나승훈, 신종훈 김영길

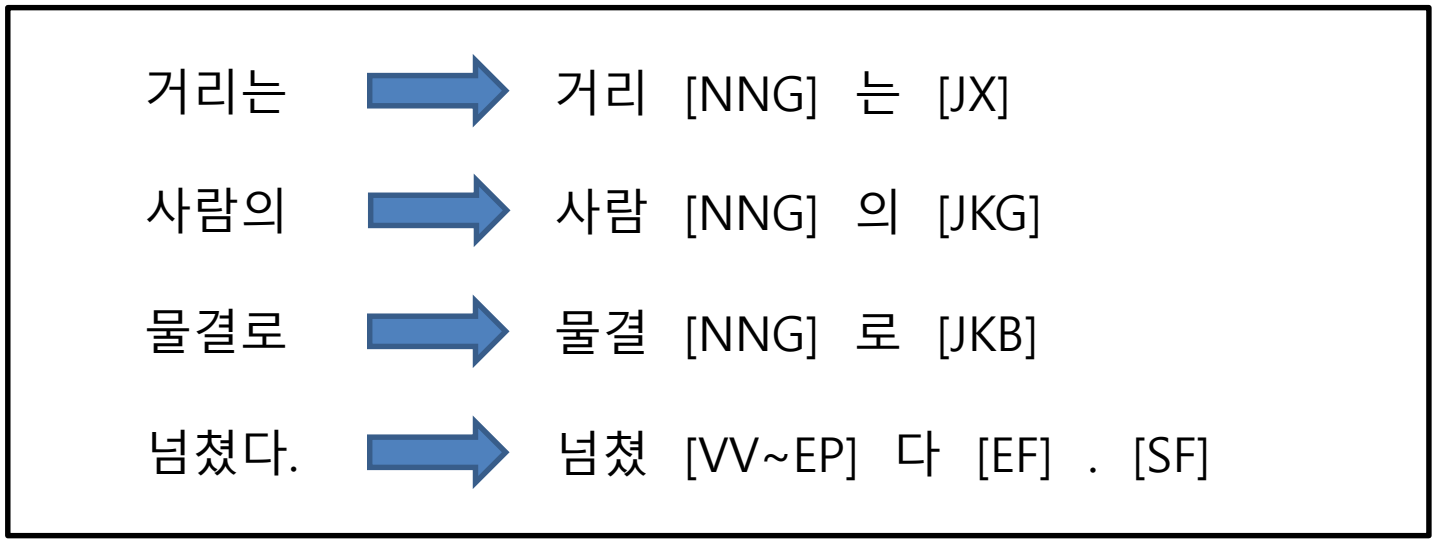
인지컴퓨팅 연구실
전북대학교

목차

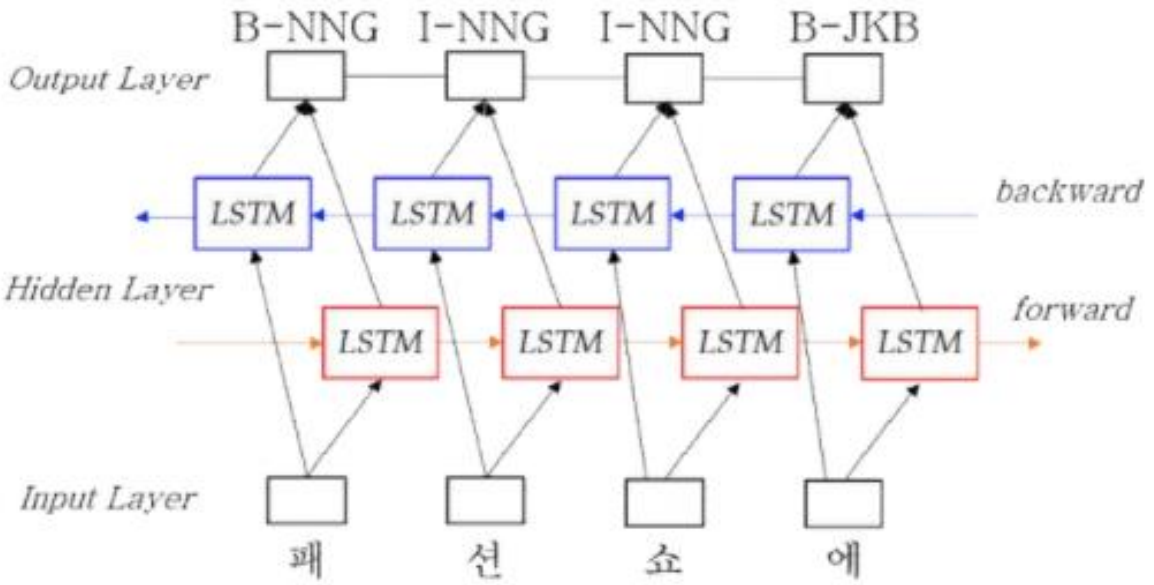
- 형태소 분석
- 관련연구
- Stack Pointer Network를 이용한 한국어 형태소 분석
- 실험결과

형태소 분석

- 정의(두 가지의 과정으로 구분)
  - 형태소 분석 : 문장 내의 어절을 뜻을 지니는 최소의 단위인 형태소로 분해하고 해당 형태소의 품사 후보를 생성
  - 품사 태깅 : 형태소의 품사 후보로부터 가장 적절한 품사를 결정하는 과정
  - 입력문
    - 예) 거리는 사람의 물결로 넘쳤다



품사 분포와 Bidirectional LSTM-CRFs를 이용한 음절단위 형태소 분석기(김혜민, HCLT '2016)



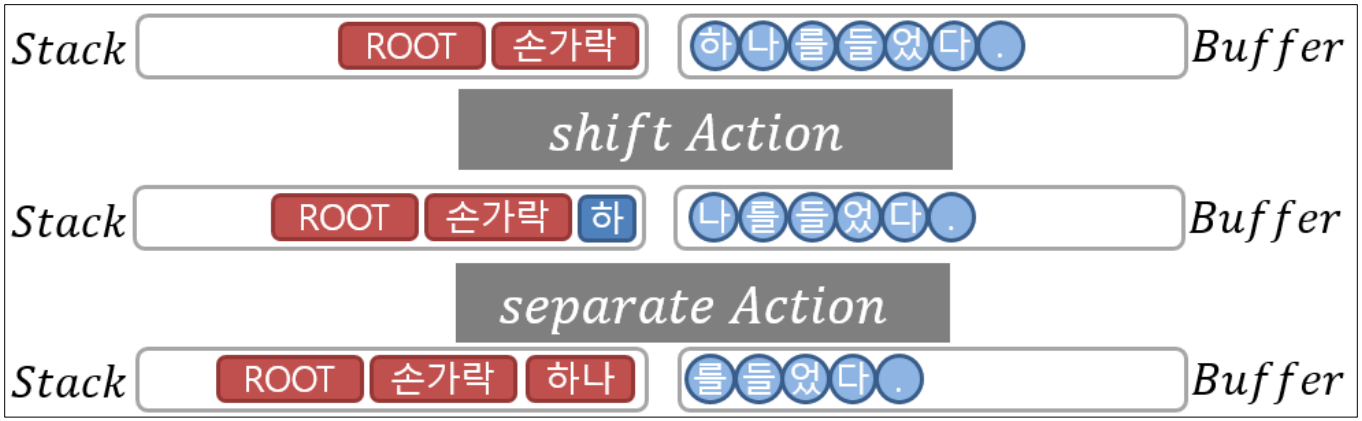
- Bi LSTM CRFs 형태소 분석
  - 음절 단위의 품사 태깅 방법
  - 순차 데이터를 모델링하는 양방향 LSTM에 출력 태그 간의 전이 확률을 얻는 CRF와 결합하는 방식
  - [B(Begin),I(Inside)] 등의 태그 등을 붙인 품사 태그를 결정하는 방식

동적 오라클을 이용한 뉴럴 전이기반 한국어 형태소 분석 및 품사 태깅(민진우, HCLT '2018)

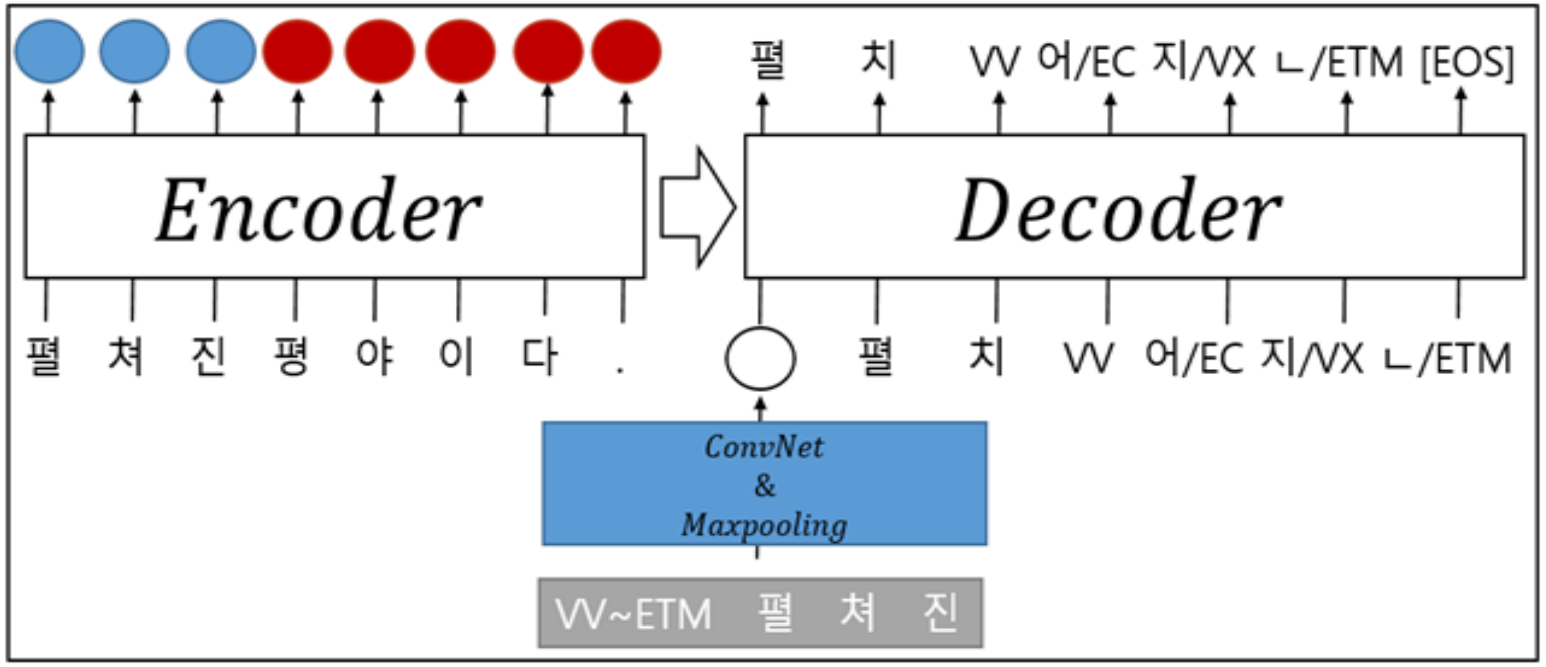
- 전이 기반 형태소 분석 모델
  - 두가지의 Action으로 구성
    - $separate$: 현재 음절을 형태소의 요소로 추가하는 Action
    - $shift$: 형태소의 끝 경계를 결정하고 품사를 결정하는 Action

S_t	B_t	Action	S_{t+1}	B_{t+1}
S	c, B	$Separate(t)$	$(t, c), S$	B
S	c, B	$shift$	c, S	B

  - 실행 예



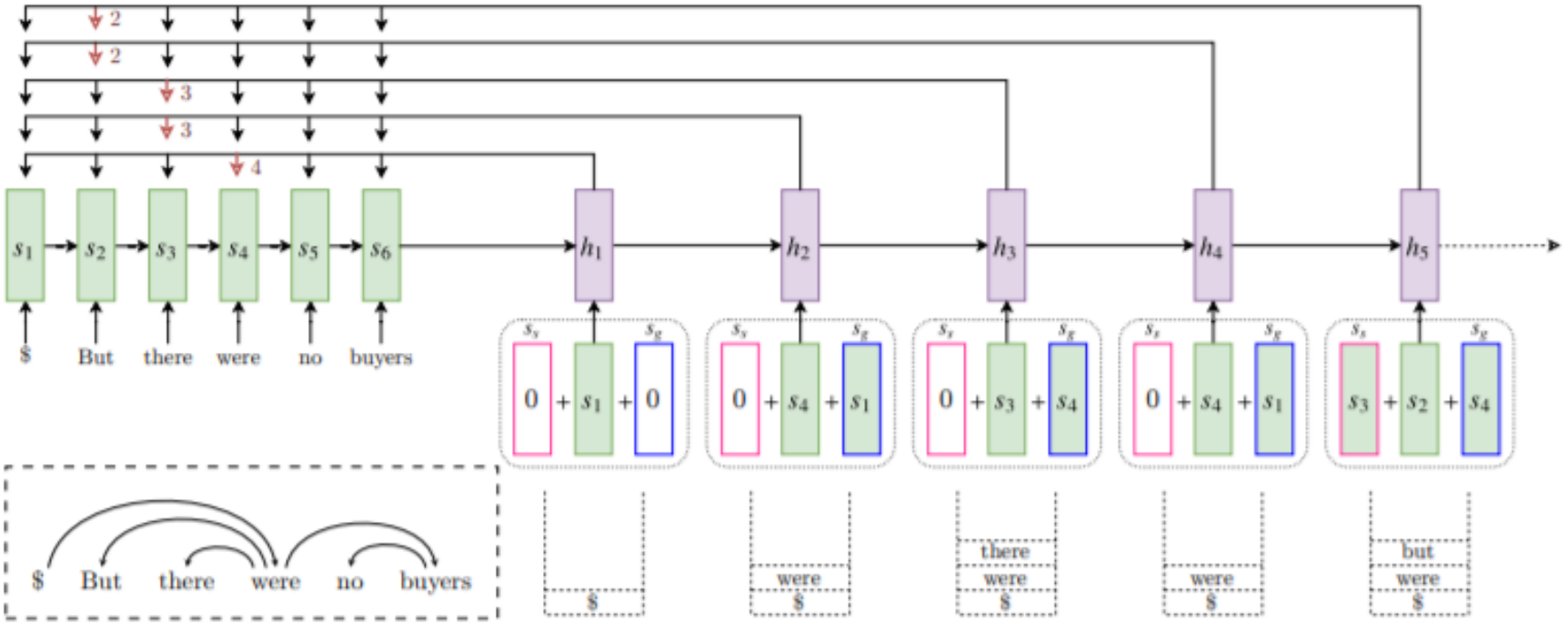
End-to-End 뉴럴 전이 기반 한국어 형태소 분석 (민진우, KCC '2019)



- End-to-End 전이 기반 형태소 분석 모델
  - 모델로 인식된 형태소가 복합 형태소인 경우 Sequence-to-Sequence 모델을 이용하여 단위 형태소로 분석하는 End-to-End 모델로의 확장
  - 내용 형태소는 미등록어 문제를 해결하기 위해서 음절을 디코딩 한 후 품사를 디코딩. 복합 기능 형태소는 미등록어 문제가 거의 발생하지 않아 "형태소/품사태그"의 결합 단위로 디코딩

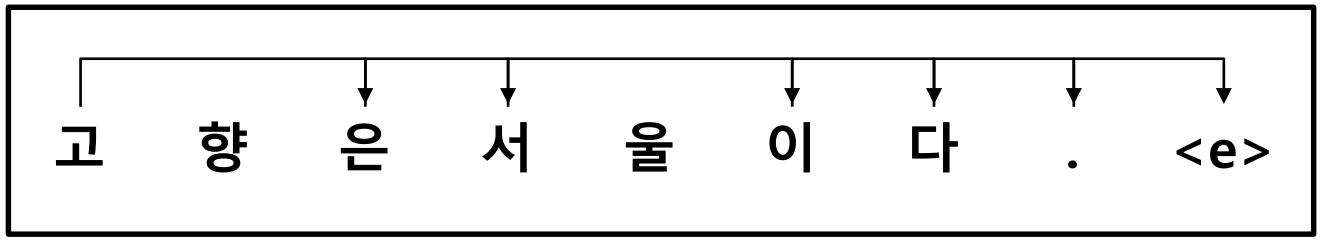
Stack-Pointer Networks for Dependency Parsing(X. ma, ACL '2018)

- 전이 기반 방식(Top-Down) 의존 파서
  - 포인터 네트워크를 확장하여 트리 구조를 반영하는 스택-포인터 네트워크를 제안
  - UAS : 95.87, LAS : 94.19 [PTB 데이터 셋]

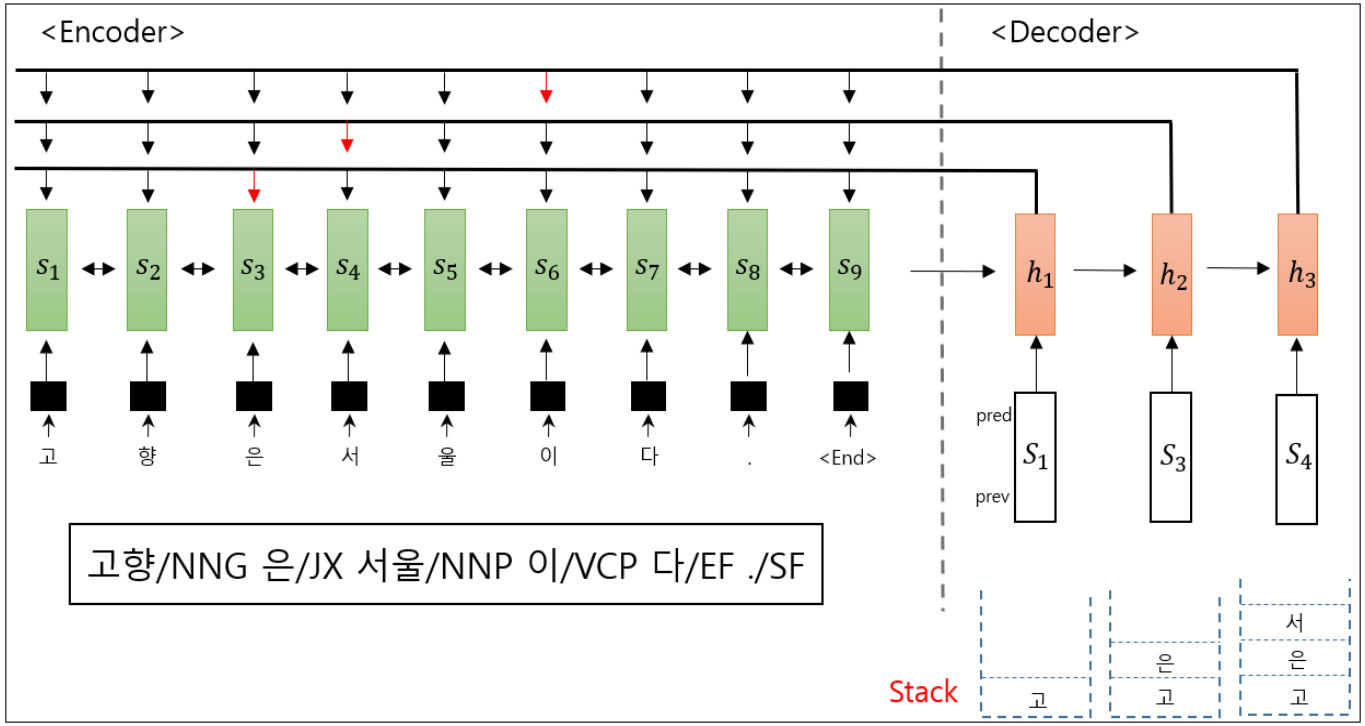


Stack Pointer Network를 이용한 한국어 형태소 분석

- 제안 아이디어
  - 기존 방식 : 음절에 대한 BI-태그를 부착하는 Bi-LSTM-CRF나 음절 단위에서 액션을 결정하여 형태소의 경계를 결정하는 전이 기반 모두 음절 단위의 결정
  - 제안 방식
    - Pointer Network를 이용하여 다음 형태소의 시작 음절을 Pointing 하는 방식



Stack Pointer Network를 이용한 한국어 형태소 분석



- 내부 Stack과 Pointer Network의 활용
  - 인코더에서 문장을 인코딩하고 디코더에서는 내부 Stack의 top 노드를 받아서 다음 형태소의 시작을 포인팅하며 형태소의 경계 결정
  - Stack은 pop연산이 없는 일종의 버퍼 역할 수행

Stack Pointer Network를 이용한 한국어 형태소 분석

- 인코더
  - 입력 임베딩
    - 문장의 끝을 알리는 “<End>”토큰을 포함한 음절 열을 임베딩
    - i 번째 입력 임베딩 벡터 $\boldsymbol{c}_i$와 해당 음절이 어절의 시작인지 아닌지를 나타내는지에 대한 띄어쓰기 임베딩 벡터 $\boldsymbol{p}_i$의 결합

$$\boldsymbol{x}_i = [\boldsymbol{c}_i; \boldsymbol{p}_i]$$

  - 인코딩
    - 입력열 $\{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n\}$을 다층의 LSTM을 통해 인코딩하여 은닉열 $\{\boldsymbol{s}_1, \cdots, \boldsymbol{s}_n\}$을 얻음
    - 얻어진 은닉열은 Source로 하여 디코더에서 어텐션을 수행하는 Key의 역할

$$\{\boldsymbol{s}_1, \cdots, \boldsymbol{s}_n\} = LSTM(\{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n\})$$

Stack Pointer Network를 이용한 한국어 형태소 분석

- 디코더
  - 두 단계의 과정을 거침
    1) 형태소의 경계 결정 단계
    2) 형태소의 품사 결정 단계
  - 형태소의 경계 결정
    - 디코더의 스텝 t에서는 현재 Stack의 top에 위치하는 음절 표상과 이전에 결정된 형태소의 품사의 임베딩을 결합하여 디코더의 입력인 스택(버퍼)의 top 노드 표상을 받음.
    - 디코더의 출력 와 은닉열 에 대해 Biaffine 어텐션 함수를 적용

$$e_i^t = (\boldsymbol{h}_t^T W \boldsymbol{s}_i + U^T \boldsymbol{h}_t + V^T \boldsymbol{s}_i + b)$$

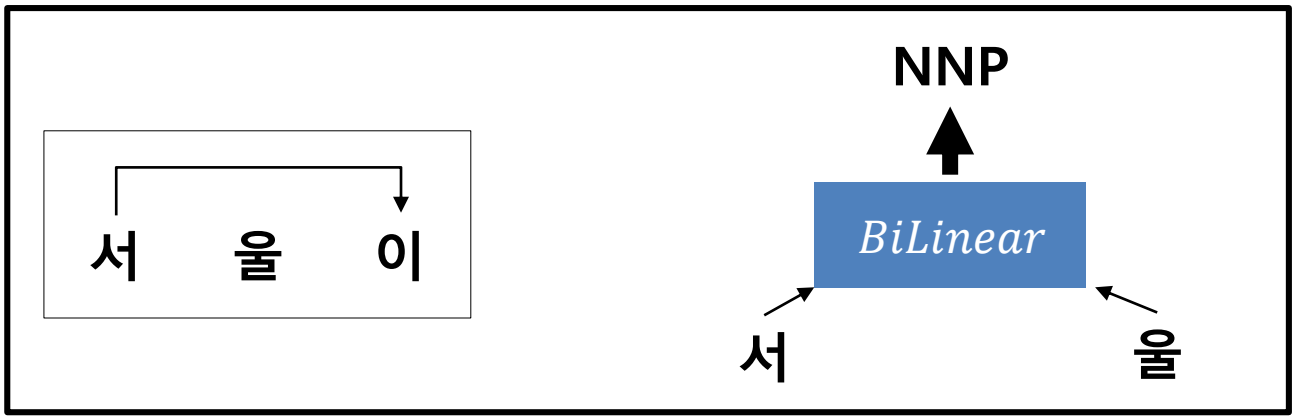
    - e_i^t는 t번째 디코딩 시점에 인코더의 i번째 음절에 대한 점수를 의미. 가장 점수가 높은 위치를 포인팅 한 후 스택에 넣고 다음 형태소의 시작을 포인팅
    - <End> 토큰을 포인팅하면 종료

Stack Pointer Network를 이용한 한국어 형태소 분석

- 디코더
  - 형태소의 품사 결정
    - 경계가 결정지어진 t시점에서의 형태소의 시작 음절과 끝 음절을 각각 s_s^t, s_e^t라 할 때 다음 수식과 같이 두 벡터간의 BiLinear 어텐션을 통해 형태소의 품사를 결정

$$e_j^t = s_s^t W s_e^t$$

    - e_j^t는 t번째 형태소에 대한 j번째 품사의 점수를 의미하며 가장 높은 점수를 갖는 품사를 선택하여 품사를 결정
    - Ex) “은”이 포인팅 되는 순간 “서”와 “울”은 형태소의 시작 s_s^t과 끝 s_e^t을 나타내며 “서울”이라는 형태소의 품사 NNP(고유명사)를 위의 결정



실험 세팅

- 데이터 셋
  - 세종 형태소 분석 말뭉치

	Train	Dev	Test
문장 수	197508	5000	50631
어절 수	2674563	97292	694523

- 파라미터

	Hyper Parameter	value
BERT	인코더 블록 개수	12
	은닉 차원수	768
	어텐션 헤드 수	12
	Optimizer	Adam
	학습률	$5e^{-5}$
LSTM	RNN 은닉 차원 수	512
	RNN Layers	5
	학습률	0.001
	드랍아웃	0.33

- 평가지표
  - 복합 형태소 단위 F1과 어절 정확도를 제시

실험 결과

- 형태소 분석 실험 결과

	형태소 F1	어절 정확도
CRF[3]	97.60%	96.14%
Phrase-Based CRF[4]	97.74%	96.35%
전이 기반	98.01%	96.78%
Bi-LSTM-CRF	98.03%	96.81%
스택 포인터 네트워크	**98.12%**	**96.92%**

- 결론 & 향후 연구
  - BERT를 사용함의 효과는 명확하나 현재 음절 단위의 전이 기반 모델에 비해 2% 낮은 성능을 보이고 있음

Q&A

감사합니다.