# Stack-Pointer Network를 이용한 한국어 의존 구문 분석

차다은○, 이동엽, 임희석

고려대학교
chadaeun57@gmail.com, judelee93@korea.ac.kr, limhseok@korea.ac.kr

# Stack-Pointer Network for Korean Dependency Parsing

Da-Eun Cha○, Dong-Yub Lee, Heui-Seok Lim
Korea University

**요 약**

의존 구문 분석은 자연어 문장에 포함된 단어들 간의 의존 관계를 분석하는 과제로 다양한 자연어 이해 과제에 요구되는 핵심 기술 중 하나이다. 본 연구에서는 단어와 문자 자질을 적용한 기존 Stack-Pointer Network의 인코더의 입력 단어 표상을 확장하여, 한국어를 비롯한 형태적으로 복잡한 언어(morphologically rich language)에 적합하도록 음절-태그 단위, 형태소 단위, 형태소 품사 정보 자질을 보강한 의존 구문 분석 모델을 제안한다. 실험 결과 제안하는 모델은 의존 구조로 변환된 세종 구문 분석 말뭉치에서 UAS 90.58%, LAS 88.35%의 성능을, 2018 국어 정보 처리 시스템 경진 대회 평가 데이터에서 UAS 84.69%, LAS 82.02%의 성능을 보였다. 더불어 제안하는 모델은 포함된 문장의 전체 길이가 긴 의존 관계, 의존소와 지배소의 거리가 먼 의존 관계, 의존소를 구성하는 형태소의 개수가 많은 의존 관계에서 기존 Stack-Pointer Network보다 향상된 성능을 보였다.

주제어: Stack-Pointer Network, 의존 구문 분석, 자질 보강

## 1. 서론

의존 구문 분석은 자연어 문장에 포함된 단어들 간의 의존 관계를 분석하는 과제로, 각 단어의 지배소(head) 및 의존소(modifier)를 파악하고 해당 단어와 지배소 · 의존소와의 의존 관계를 분석하는 것을 목표로 한다. 의존 구문 분석은 다양한 자연어 이해 과제에 요구되는 핵심 기술 중 하나로 [1], [2]와 같이 다양한 의존 구문 분석 방법들이 제안되어 왔으며, 이들은 크게 전이 기반 방법과 그래프 기반 방법으로 나뉜다.

기존 한국어 의존 구문 분석 연구에서는 주로 전이 기반 방법이 적용되어 왔다. 전이 기반 방법은 의존 구문 분석 트리를 구축하기 위한 파싱 전이 액션 선택 단계의 수가 문장의 길이에 선형적이라는 장점을 가진다. 그러나 지역적 정보에 의존하여 각 파싱 전이 액션을 결정하므로 문장 내에서 거리가 먼 단어들 간의 의존 관계를 분석하기 어려우며, 전체 문장의 의존 구문 분석 트리가 한 단계의 잘못된 파싱 전이 액션 선택에 취약하다는 한계를 갖는다. 이러한 한계를 극복하기 위해 Stack-LSTM 기반 모델[3], Pointer Network 기반 모델[4]과 같이 파싱 전이 액션 선택 과정에서 문장 전체의 정보를 반영하는 전이 기반 한국어 의존 구문 분석 모델들이 제안되어 왔다.

Stack-Pointer Network[5]는 Pointer Network[6]를 기반으로 하여 현재까지 구축된 부분 트리의 구조를 반영하는 내부 스택 정보를 적용한 전이 기반 의존 구문 분석 모델로, 파싱 전이 액션 결정 과정에서 전체 문장의 정보와 현재까지 구축된 부분 트리의 구조를 고려할 수 있다.

본 연구에서는 단어의 임베딩 벡터와 함께 문자 기반 단어 표상을 적용한 기존 Stack-Pointer Network의 인코더의 단어 표상에 [7]에서 제안된 음절-태그 단위와 형태소 단위 및 [8]에서 제안된 형태소 품사 정보 자질을 함께 활용하는 자질 보강을 통해 한국어를 비롯한 형태적으로 복잡한 언어(morphologically rich language)의 의존 구문 분석에 적합하도록 확장한 모델을 제안한다.

실험에는 [9]의 방법에 따라 의존 구조로 변환된 세종 구문 분석 말뭉치[10]를 이용하였으며, 실험 결과 제안하는 모델은 세종 구문 분석 말뭉치에서 UAS 90.58%와 LAS 88.35%의 성능을, 2018 국어 정보 처리 시스템 경진 대회 평가 데이터에서 UAS 84.69%, LAS 82.02%의 성능을 보였다. 더불어 제안하는 모델은 포함된 문장의 전체 길이가 긴 의존 관계, 의존소와 지배소의 거리가 먼 의존 관계, 의존소를 구성하는 형태소의 개수가 많은 의존 관계에서 기존 Stack-Pointer Network보다 향상된 성능을 보였다.

## 2. 제안 모델

본 연구에서는 Stack-Pointer Network을 단어 표상에서의 자질 보강을 통해 확장한 한국어 의존 구문 분석 모델을 제안한다. 제안하는 모델의 전체 구조는 그림 1과 같다.

Stack-Pointer Network는 Pointer Network에 내부 스택 정보를 적용한 의존 구문 분석 신경망 모델이다. Pointer Network는 RNN encoder-decoder 모델을 기반으로 어텐션 메커니즘을 적용하여, 출력 클래스가 입력열의 길이에 따라 달라지는 문제에서 출력열의 조건부 확률을 학습하는 신경망이다. Stack-Pointer Network는 인코더(encoder), 디코더(decoder), 파서(parser)로 구성되며, 문장 내의 각 문장이 현재 입력의 의존소일 조건부 확률을 Pointer Network를 기반으로 학습함으로써 top-down

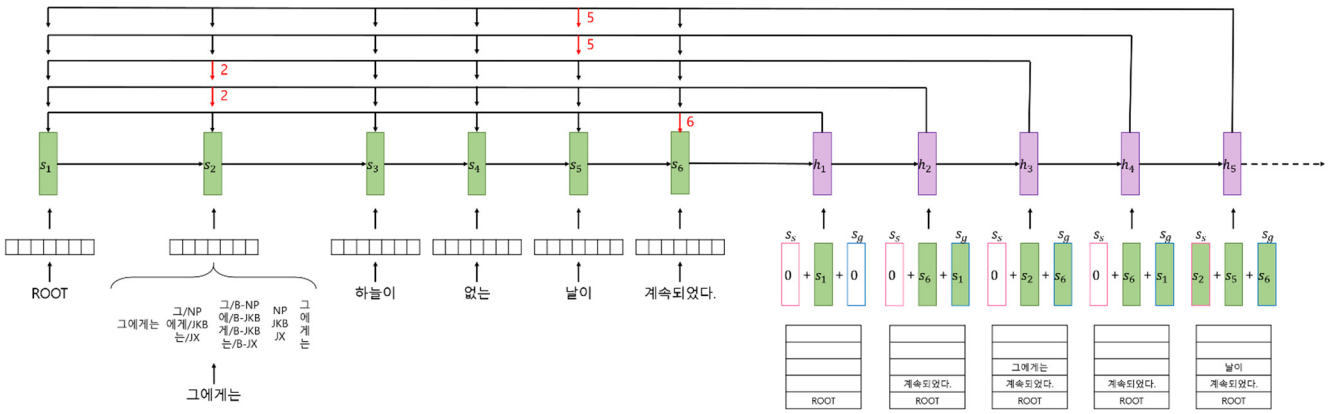


**그림 1 제안하는 모델의 전체 구조**

방식으로 의존 분석 트리를 구축한다.

제안하는 모델은 기존 Stack-Pointer Network의 인코더에서의 단어 표상을 한국어 등의 형태적으로 복잡한 언어에 적합하도록 확장하였다. 기존 Stack-Pointer Network 에서는 문자들의 임베딩 벡터로부터 CNN을 통해 얻은 문자 기반 단어 표상을 단어의 임베딩 벡터와 결합하여 단어 표상으로 활용한다. 제안하는 모델은 이와 더불어 [7]에서 제안된 음절-태그 단위와 형태소 단위, 미등록 형태소의 처리를 위해 [8]에서 제안된 형태소 품사 정보 자질을 추가하여 이들을 기반으로 한 단어 표상을 기존의 단어 표상과 함께 결합하여 활용하였다.

## 2.1 Stack-Pointer Network

Stack-Pointer Network의 인코더(encoder)는 양방향 LSTM-CNN 구조로, 전체 문장에 포함된 단어들의 연속으로부터 각 단어를 인코더 은닉 상태(encoder hidden state) $s_i$로 인코딩한다. 이에 따라 각 단어의 인코더 은닉 상태 $s_i$는 해당 단어의 정보와 함께 전체 문장의 정보를 반영한다. 제안하는 모델에서는 기존 Stack-Pointer Network의 단어 표상에서 활용한 단어의 임베딩 벡터와 CNN을 통해 인코딩된 문자 자질과 더불어, CNN을 통해 인코딩된 음절-태그 단위, 형태소 단위, 형태소 품사 정보의 자질들을 함께 활용한 단어 표상을 적용한다. 단어 표상에서의 자질 보강은 2.2에서 보다 구체적으로 후술한다.

디코더(decoder)는 단방향 LSTM 구조로, 내부 스택의 최상위 요소에 대응하는 입력 $\beta_t$로부터 $t$번째 파싱 전이 액션 선택을 위한 디코더 은닉 상태 $h_t$를 생성한다. 디코더의 입력 $\beta_t$로 내부 스택의 최상위 요소의 인코더 은닉 상태 $s_h$와 더불어 해당 요소의 지배소의 다른 의존소인 *형제 요소*의 인코더 은닉 상태 $s_s$, 해당 요소의 지배소의 지배소인 *조부모 요소*의 인코더 은닉 상태 $s_g$의 합을 사용하여 현재까지 구축된 의존 분석 부분 트리의 정보를 반영한다.

$$\beta_t = s_h + s_g + s_s$$

파서(parser)는 내부 스택의 최상위 요소의 인코더 은닉 상태 $s_i$와 현재 파싱 전이 액션 선택 단계의 디코더 은닉 상태 $h_t$로부터 [11]에서 제안된 Biaffine Attention Mechanism을 적용하여 내부 스택 최상위 요소의 의존소를 선택하고 이 의존소와의 관계를 예측함으로써 파싱 전이 액션을 결정하고 내부 스택을 업데이트한다. Biaffine Attention Mechanism 적용 이전에 $s_i$와 $h_t$를 MLP 레이어를 통해 저차원으로 변환하며 모델의 과적합을 방지한다.

$t$번째 파싱 전이 액션 선택 단계에서 내부 스택의 최상위 요소의 의존소로 $i$번째 단어를 선택할 점수 $e_i^t$는 다음과 같이 계산한다.

$$e_i^t = h_t^T W_e s_i + U_e^T h_t + V_e^T s_i + b_e$$

이 때 $W_e$는 bi-linear 가중치 행렬, $U_e$와 $V_e$는 각각 디코더 은닉 상태와 인코더 은닉 상태에 대한 가중치 행렬, $b_e$는 편향이며 이들은 학습되는 파라미터들이다. 선택한 의존소는 스택의 최상위 요소로 추가되어 다음 파싱 전이 액션 선택에서 처리되고, 선택한 의존소가 내부 스택의 최상위 요소 자기자신일 경우 내부 스택에서 해당 요소를 제거한다.

마찬가지로 $t$번째 파싱 전이 액션 선택 단계에서 지배소와의 의존 관계가 $i$일 점수 $l_i^t$는 다음과 같이 지배소의 인코더 은닉 상태 $s_h$와 의존소의 인코더 은닉 상태 $s_m$으로부터 계산한다. 이 때 $W_l$는 bi-linear 가중치 행렬, $U_l$와 $V_l$는 각각 디코더 은닉 상태와 인코더 은닉 상태에 대한 가중치 행렬, $b_l$는 편향이며 이들은 학습되는 파라미터들이다.

$$l_i^t = s_h^T W_l s_m + U_l^T s_h + V_l^T s_m + b_l$$

## 2.2 단어 표상에서의 자질 보강

제안하는 모델은 Stack-Pointer Network의 인코더에 입력되는 단어 표상에서의 자질 보강을 통하여 Stack-Pointer Network를 한국어 등의 형태적으로 복잡한 언어에 적합하도록 확장한 모델로, 단어의 임베딩과 더불어 문자, 음절-태그, 형태소, 형태소 품사 정보의 자질들을 기반으로 한 단어 표상을 결합하여 얻은 단어 표상을 적

용한다.

기존 Stack-Pointer Network에서는 단어에 포함된 문자들의 임베딩 벡터들로부터 CNN을 통해 문자 기반 단어 표상을 생성하고 이를 단어의 임베딩 벡터와 결합한 단어 표상을 인코더의 입력으로 사용한다. 이러한 단어 표상은 단어 임베딩과 더불어 문자 단위의 정보를 반영하므로 한국어와 같이 형태적으로 복잡한 언어의 처리가 가능하지만, 그럼에도 문자 단위에서 애매성이 발생한다는 한계를 갖는다.

[7]에서 제안된 음절-태그 단위와 형태소 단위는 각각 음절과 형태소의 표층 정보와 품사 태그로 구성된 정보이다. 이러한 자질을 활용함으로써 문자 단위보다 의미적으로 유의미한 단어의 하위 단위인 형태소 단위의 정보를 적용할 수 있으며, 품사 태그를 함께 사용함으로써 표층 정보만을 활용했을 때의 애매성을 줄이고 의존 구문 분석에서 품사 정보를 반영할 수 있다. 더불어 [8]에서 제안된 바와 같이 형태소 품사 정보를 단어 표상에 활용함으로써 미등록 형태소에 대한 처리가 가능하다. 이에 따라 제안하는 모델은 단어의 임베딩에 형태소, 음절-태그, 형태소 품사 정보, 문자 자질들의 임베딩을 각각 convolutional layer와 max pooling layer를 거친 결과 생성한 단어 표상을 결합한 단어 표상을 적용한다. 그림 2는 제안하는 모델의 단어 표상 생성 방식을 나타낸 것이다.

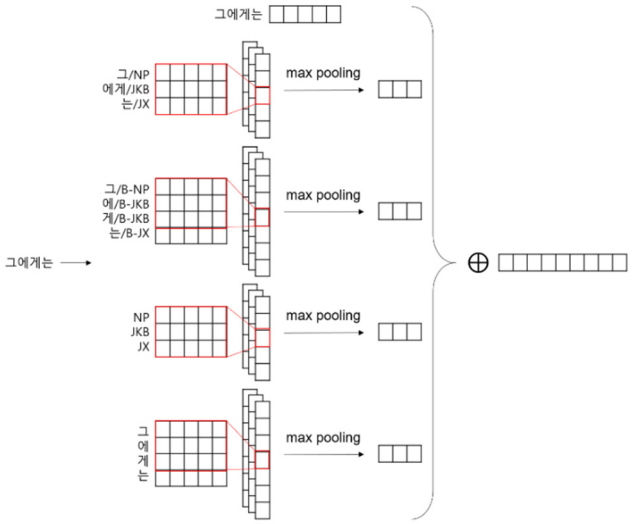


**그림 2 제안하는 모델의 단어 표상 생성 방식**

본 연구에서는 Glove [12]를 사용하여 음절-태그 임베딩과 형태소 임베딩을 학습하였다. 세종 현대문어 형태 분석 말뭉치[10]와 한국어 위키피디아에서 수집된 문장들의 형태 분석 결과를 결합하여 약 1,300만개 문장으로 구성된 학습 코퍼스를 구축하였으며, 이에 Glove를 적용하여 약 16,000개의 음절-태그와 20만개의 형태소에 대한 100차원 임베딩 벡터를 학습하였다.

## 3. 실험

본 연구에서는 [9]의 방법에 따라 세종 구문 분석 말뭉치[10]를 의존 구문 분석 말뭉치로 변환한 데이터를 실험에 사용하였으며, 해당 데이터는 약 5만개 문장을 포함한 학습 데이터와 약 5,000개 문장을 포함한 평가 데이터로 구성된다. 실험에 사용한 hyper-parameter들을 표 1에 제시하였다.

**표 1 실험에 사용된 hyper-parameter들**

| | Hyper-parameter | Value |
|---|---|---|
| **Training Data** | word vocab size | 215316 |
| | morpheme vocab size | 42712 |
| | syllable-tag vocab size | 11495 |
| | morpheme POS vocab size | 46 |
| | character vocab size | 3599 |
| **Glove** | morpheme window size | 15 |
| | morpheme dimension | 100 |
| | syllable-tag window size | 15 |
| | syllable-tag dimension | 100 |
| **Encoder** | CNN filter size | 3 |
| | CNN number of filters | 100 |
| | CNN dropout | 0.33 |
| | LSTM state size | 256 |
| | LSTM layers | 3 |
| | LSTM dropout | 0.33 |
| **Decoder** | state size | 512 |
| | layers | 1 |
| | dropout | 0.33 |
| **Parser** | arc MLP size | 512 |
| | label MLP size | 128 |
| | training epoch | 24 |
| | initial learning rate | 0.001 |
| | decay rate | 0.75 |
| | word dimension | 300 |
| | morpheme POS dimension | 50 |
| | character dimension | 50 |

표 2는 제안하는 모델에서 인코더 입력 단어 표상에서의 자질 선택에 따른 세종 구문 분석 말뭉치 대상 성능 비교 결과이다.

**표 2 자질 선택에 따른 성능 비교 결과**

| 모델 | UAS(%) | LAS(%) |
|---|---|---|
| 단어 + 문자 (기존 Stack-Pointer Network) | 87.48 | 83.83 |
| 단어 + 형태소 | 86.65 | 85.53 |
| 단어 + 음절-태그 | 88.78 | 86.12 |
| 단어 + 형태소 + 음절-태그 | 89.87 | 87.52 |
| 단어 + 형태소 + 음절-태그 + 형태소 품사 정보 | 90.17 | 87.84 |
| 단어 + 형태소 + 음절-태그 + 문자 | 90.45 | 88.10 |
| **단어 + 형태소 + 음절-태그 + 형태소 품사 정보 + 문자** | **90.58** | **88.35** |

실험 결과 제안하는 모델은 UAS 90.58%, LAS 88.35%의 성능을 보였으며, 기존 Stack-Pointer Network보다 UAS는 3.1%, LAS는 4.52% 향상된 성능을 보였다.

그림 3은 기존 Stack-Pointer Network와 제안하는 모델의 문장 전체 길이, 의존소와 지배소의 거리, 의존소를 구성하는 형태소의 수에 따른 성능 차이를 나타낸 것이다.

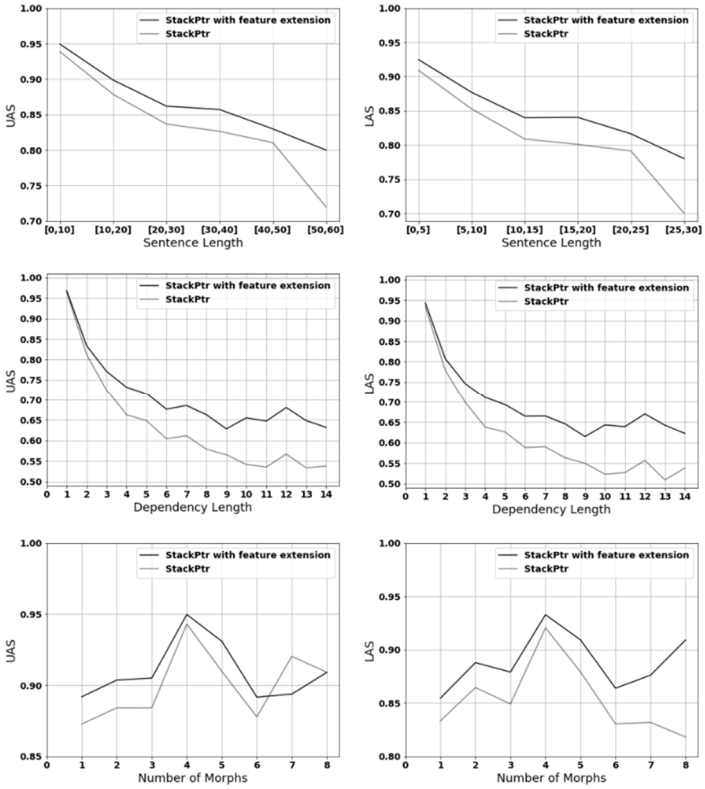


**그림 3 기존 Stack-Pointer Network와 제안하는 모델의 상세 성능 비교 결과**

제안하는 모델과 기존 Stack-Pointer Network를 비교한 결과 문장의 길이가 긴 의존 관계와 의존소와 지배소의 거리가 먼 의존 관계에서의 성능이 향상되었다. 의존소에 포함된 형태소의 개수가 7개 이상으로 매우 많은 경우에는 UAS가 감소하였으나 이 경우에도 기존 모델에 비하여 의존 관계의 유형을 보다 정확하게 분석하여 LAS가 향상되었다.

표 3은 제안하는 모델의 2018 국어 정보 처리 시스템 경진 대회 평가 데이터 기준 성능 평가 결과를 나타낸 것이다.

**표 3 평가 데이터 기준 성능 평가 결과**

| 모델 | UAS(%) | LAS(%) |
|---|---|---|
| 자질 보강 Stack-Pointer Network | 84.69 | 82.02 |

## 4. 결론

본 연구에서는 단어 임베딩에 문자 자질로부터 CNN을 통해 얻은 문자 기반 단어 표상을 결합하여 활용하는 기존 Stack-Pointer Network의 단어 표상에서 음절-태그 단위, 형태소 단위, 형태소 품사 정보 자질을 함께 활용하는 자질 보강을 통해 한국어를 비롯한 형태적으로 복잡한 언어의 의존 구문 분석에 적합하게 확장한 모델을 제안하였다. 제안하는 모델은 Stack-Pointer Network를 기반으로 하여 의존 구문 분석에 문장 전체의 정보 및 현재까지 구축된 의존 분석 부분 트리의 정보를 반영할 수 있으며, 자질 보강을 통해 단어 표상에서의 애매성을 줄이고 품사 정보를 활용하였다. 실험 결과 제안하는 모델은 세종 구문 분석 말뭉치를 의존 분석 말뭉치로 변환한 데이터에서 UAS 90.58%, LAS 88.35%의 성능을 보였으며, 2018 국어 정보 처리 시스템 경진 대회 평가 데이터에서 UAS 84.69%, LAS 82.02%의 성능을 보였다. 또한 문장 전체 길이가 긴 의존 관계, 의존소와 지배소의 거리가 먼 의존 관계, 의존소를 구성하는 형태소의 수가 많은 의존 관계에 대하여 기존 Stack-Pointer Network보다 높은 성능을 보였다.

## 참고문헌

[1] C. Dyer, M. Ballesteros, W. Ling, A. Matthews and N.A. Smith, Transition-Based Dependency Parsing with Stack Long Short-Term Memory, ACL, 2015

[2] T. Dozat and C.D. Manining, Deep biaffine attention for neural dependency parsing, TACL, 2016

[3] 나승훈, 신종훈, 김강일, “순환 컨트롤러를 이용한 Stack LSTM기반 한국어 의존 파싱”, 한국정보과학회 2016년 동계학술대회 논문집, pp 446-448, 2016

[4] 박천음, 이창기. “포인터 네트워크를 이용한 한국어 의존 구문 분석”, 정보과학회논문지, 제44권 제8호 pp 822-831, 2017

[5] X. Ma, Z. Hu, N. Peng, G. Neubig, E.Hovy, Stack-Pointer Networks for Dependency Parsing, ACL, 2018

[6] O. Vinyals, M. Fortunato and N. Jaitly, Pointer networks, NIPS, 2015

[7] 나승훈, 신종훈, 김강일, “Stack LSTM 기반 한국어 의존 파싱을 위한 음절과 형태소의 결합 단어 표상 방법”, 한글 및 한국어 정보처리 학술대회, pp 9-13, 2016

[8] 나승훈, 이건일, 신종훈, 김강일, “Deep Biaffine Attention을 이용한 한국어 의존 파싱”, 한국정보과학회 한국컴퓨터종합학술대회 논문집, pp. 584-586, 2017

[9] 이창기, 김준석, 김정희, “딥 러닝을 이용한 한국어 의존 구문 분석”, 한글 및 한국어 정보처리 학술대회, pp 87-91, 2014

[10] 국립국어원, 21세기 세종 계획, 2012

[11] T. Luong, H. Pham, C.D. Manning, Effective approaches to attention-based neural machine translation. Proceedings of EMLNP, pp. 1412-1421, 2015

[12] J. Pennington, R. Socher, C.D. Manning, “Glove: Global Vectors for Word Representation”, EMNLP, 2014