Multi-task sequence-to-sequence learning을 이용한 한국어 형태소 분석과 구구조 구문 분석

황현선○, 이창기

강원대학교

{hhs4322, leeck}@kangwon.ac.kr

Korean morphological analysis and phrase structure parsing using multi-task sequence-to-sequence learning

Hyunsun Hwang ○, Changki Lee

Kangwon National University

요 약

한국어 형태소 분석 및 구구조 구문 분석은 한국어 자연어처리에서 난이도가 높은 작업들로서 최근에는 해당 문제들을 출력열 생성 문제로 바꾸어 sequence-to-sequence 모델을 이용한 end-to-end 방식의 접근법들이 연구되었다. 한국어 형태소 분석 및 구구조 구문 분석을 출력열 생성 문제로 바꿀 시 해당 출력 결과는 하나의 열로서 합쳐질 수가 있다. 본 논문에서는 sequence-to-sequence 모델을 이용하여 한국어 형태소 분석 및 구구조 구문 분석을 동시에 처리하는 모델을 제안한다. 실험 결과 한국어 형태소 분석과 구구조 구문 분석을 동시에 처리할 시 형태소 분석이 구구조 구문 분석에 영향을 주는 것을 확인 하였으며, 구구조 구문 분석 또한 형태소 분석에 영향을 주어 서로 영향을 줄 수 있음을 확인하였다.

주제어: 형태소 분석, 구구조 구문 분석, multi-task learning, sequence-to-sequence learning

1. 서론

형태소 분석은 한국어 자연어처리 중 하나로 형태소 분리, 품사 태깅, 원형 복원 등의 여러 단계를 거쳐 난이도가 높은 작업에 속한다. 구문 분석은 문장의 구조를 분석하는 방법으로 구구조 구문 분석과 의존 구문 분석이 사용된다. 그러나 한국어 특성상 구구조 구문 분석의 난이도가 높고 시간 복잡도가 $O(n^3)$으로 높아 한국어 자연어처리에서는 주로 의존 구문 분석이 사용되었다.

최근 기계학습 알고리즘 중 하나인 딥 러닝(Deep Learning)을 자연어처리에 적용하는 연구가 많이 진행되었다[1,2]. 그 중 sequence-to-sequence 모델은 입력열을 길이가 다른 출력열로 변환하는 모델로, end-to-end 방식의 신경망 구조를 사용한다. 이러한 방식의 sequence-to-sequence 모델은 복잡한 문제를 출력열 생성 문제로 바꾸어 기존의 복잡한 작업을 단순화 시키는 장점이 있다. Sequence-to-sequence 모델은 Neural Machine Translation(NMT) 모델에 처음 적용이 되어 기계번역 문제를 end-to-end 방식의 모델로 처리하였다[3,4]. 이후 다른 자연어처리 문제들에 적용이 되었는데, 특히 복잡한 한국어 형태소 분석과 구구조 구문 분석을 출력열 생성 문제로 바꾸어 end-to-end 방식의 접근을 시도한 연구들이 진행 되었다[5,6,7].

한국어 형태소 분석 및 구구조 구문 분석을 출력열 생성 문제로 바꾸었을 때, 이 두가지 문제는 하나의 열로서 표현이 가능하며 sequence-to-sequence 모델을 사용하여 동시에 분석이 가능하다. 본 논문에서는 sequence-to-sequence 모델을 이용하여 한국어 형태소 분석과 구구조 구문 분석을 동시에 처리하는 모델을 소개하며 이러한 모델의 단점을 설명하고 이를 극복하는 새로운 sequence-to-sequence 모델을 제안한다.

2. 관련 연구

한국어 형태소 분석은 형태소 분리, 품사 태깅, 원형 복원 등의 여러 단계를 거치며 특히 품사 태깅의 경우 CRFs나 Structural SVM등의 기계학습 알고리즘을 주로 사용하였다[8]. [5]에서는 이러한 여러 단계로 이루어진 한국어 형태소 분석을 sequence-to-sequence 모델을 이용하여 end-to-end 방식의 한국어 형태소 분석을 제안하였다. [6]에서는 sequence-to-sequence 모델에 입력열의 단어를 출력열에 복사하는 copying mechanism을 적용하여 학습데이터에서 상태적으로 적게 등장하는 고유 명사와 같은 단어들에 내성을 지닌 end-to-end 방식의 한국어 형태소 분석을 제안 하였다.

한국어 구구조 구문 분석은 주로 확률을 이용한 방법들이 연구되었으며[9], 최근에는 문장을 트리 구조 형태로 분석하는 구구조 구문 분석 결과를 구문 분석 태그가 포함된 괄호를 이용하여 하나의 열로서 표현하여 sequence-to-sequence 모델을 이용한 한국어 구구조 구문 분석을 시도한 연구가 진행되었다[7]. [7]에서는 sequence-to-sequence 모델의 성능을 높이기 위한 attention mechanism[10]과 input-feeding[11]의 기술을 적용하여 높은 한국어 구구조 구문 분석 성능을 보였다.

3. Multi-task sequence-to-sequence learning을 이용한 한국어 형태소 분석과 구구조 구문 분석

한국어 형태소 분석과 구구조 구문 분석은 하나의 출력열로서 표현될 수 있다. 본 논문에서는 sequence-to-sequence 모델을 이용하여 한국어 형태소 분석과 구구조 구문 분석을 동시에 처리하는 multi-task sequence-to-sequence 모델을 제안한다.

3.1 출력 결과를 합친 sequence-to-sequence learning

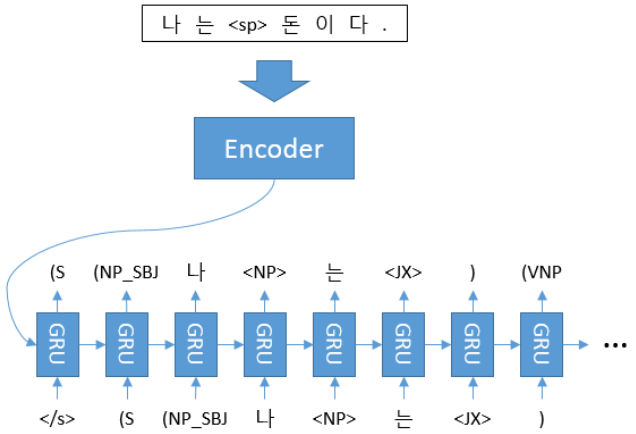

그림 1. 출력 결과를 합친 sequence-to-sequence 모델을 이용한 한국어 형태소 분석 및 구구조 구문 분석의 예시

[7]에서는 형태소 분석된 결과를 이용한 구구조 구문 분석을 시도하여 출력 결과의 대상 어절을 'XX'로 치환 하였다. 이는 이미 형태소 분석이 되어있다는 가정하에 구구조 구문 분석을 시도하기 때문이며, 이를 다시 형태소 분석 결과로 치환하면 sequence-to-sequence 모델을 이용하여 한국어 형태소 분석 및 구구조 구문 분석을 동시에 시도할 수 있다. 그림 1은 출력 결과를 합친 sequence-to-sequence 모델을 이용한 한국어 형태소 분석 및 구구조 구문 분석 예시 이다. Sequence-to-sequence 모델의 인코더 입력은 형태소 분석이 되지 않은 문장을 음절 단위로 넣게 되며 이때 어절을 구분하는 태그인 '<sp>'(띄어쓰기)태그를 같이 넣게 된다. 출력은 구문 분석 태그를 포함하여 형태소 분석된 결과를 [6]과 동일하게 음절 단위로 출력을 하게 된다. 이때 [6]과 마찬가지로 입력열의 단어가 출력열에도 등장 하게 됨으로 copying mechanism을 적용할 수 있다.

그러나 한국어 특성상 한국어 자연어처리는 형태소 분석이 상당히 중요하며, 이러한 모델은 구구조 구문 분석 시 형태소 분석 결과를 이용할 수 없다는 문제가 발생하게 된다.

3.2 Hidden state를 공유하는 Multi-task sequence-to-sequence learning

3.1절에서 설명되었듯이 한국어 구구조 구문 분석은 한국어 형태소 분석에 영향을 받기 때문에 출력 결과를 단순히 합친 sequence-to-sequence 모델로는 낮은 구구조 구문 분석 성능을 보이게 된다. 이에 따라 본 논문에서는 구구조 구문 분석이 형태소 분석 결과를 이용할 수 있도록 hidden state를 공유하는 multi-task sequence-to-sequence 모델을 제안한다.

기존의 multi-task learning은 하나의 hidden state에서 서로 다른 task의 결과를 출력하는 모델로서 하나의 신경망으로 서로 다른 문제를 동시에 해결 할 수 있다는 장점이 있다[12]. 그러나 한국어 형태소 분석과 구구조 구문 분석의 출력은 서로 다른 출력열의 형태로 길이가 다를 수 있다. 본 논문에서는 서로 다른 디코더를 이어 hidden state를 공유하는 multi-task sequence-to-sequence 모델을 설계하였다.

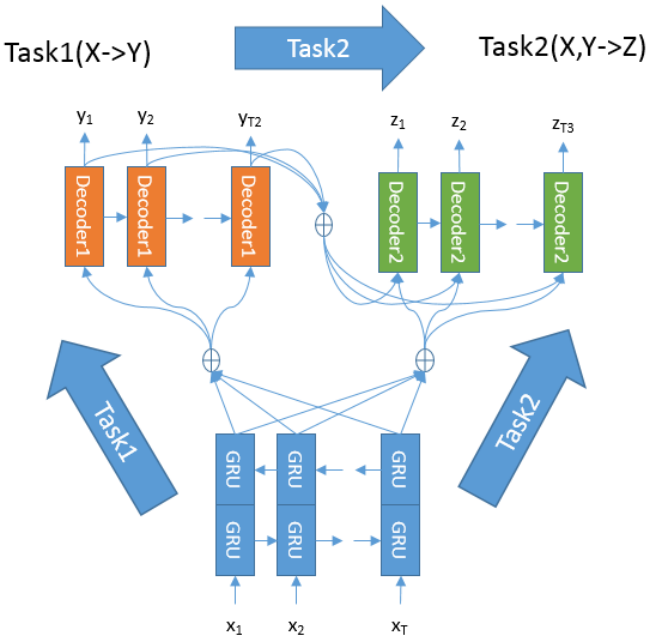

그림 2. Hidden state를 공유하는 multi-task sequence-to-sequence 모델

그림 2는 서로 다른 두 task의 디코더를 이어 hidden state를 공유하는 multi-task sequence-to-sequence 모델의 그림이다. $X(x_1,x_2,\cdots,x_T)$는 입력열이며, $Y(y_1,y_2,\cdots,y_{T2})$는 task1(형태소 분석)의 출력열이고, $Z(z_1,z_2,\cdots,z_{T3})$은 task2(구구조 구문 분석)의 출력열이다. Task1의 경우 입력열 X를 인코딩하여 출력열 Y를 디코딩하게된다. Task2의 경우 입력열 X를 인코딩한 hidden state들을 이용하여 출력열 Z를 디코딩 하나, 이때 task1의 디코더 hidden state들을 추가 정보로 보게 된다.

본 논문에서는 그림 2의 task1을 한국어 형태소 분석으로 설계하여 [6]과 동일한 attention mechanism, input-feeding, copying mechanism을 적용한 sequence-

to-sequence 모델로 설계 하였다. Task2는 한국어 구구조 구문 분석으로 설계하여 [7]과 동일한 attention mechanism, input-feeding을 적용한 sequence-to-sequence 모델을 사용하였으며, 추가적으로 task1의 정보를 사용하기 위해 다음과 같이 디코더를 재설계하였다.

$$ {}^{1}e_i^t = f_{ATT1}(E_{tgt}(z_{t-1}), h1_{t-1}, h2_{t-1}, {}^{1}h_i) $$

$$ {}^{1}a_i^t = \frac{{}^{1}e_i^t}{\sum_{ii=1}^{T} \exp({}^{1}e_{ii}^t)} $$

$$ {}^{1}c^t = \sum_{i=1}^{T} {}^{1}a_i^t \, {}^{1}h_i $$

$$ {}^{2}e_j^t = f_{ATT2}(E_{tgt}(z_{t-1}), h1_{t-1}, h2_{t-1}, {}^{2}h2_j) $$

$$ {}^{2}a_j^t = \frac{{}^{2}e_j^t}{\sum_{jj=1}^{T2} \exp({}^{2}e_{jj}^t)} $$

$$ {}^{2}c^t = \sum_{j=1}^{T2} {}^{2}a_j^t \, {}^{2}h2_j $$

$$ \mathrm{z} = \sigma(W_z E_{tgt}(z_{t-1}) + U1_z h1_{t-1} + U2_z h2_{t-1} + W_{zc1} \, {}^{1}c^t + W_{zc2} \, {}^{2}c^t + b_z) $$

$$ \mathrm{r} = \sigma(W_r E_{tgt}(z_{t-1}) + U1_r h1_{t-1} + U2_r h2_{t-1} + W_{rc1} \, {}^{1}c^t + W_{rc2} \, {}^{2}c^t + b_r) $$

$$ \mathrm{m} = \mathrm{f}(W_m E_{tgt}(z_{t-1}) + U2_m h2_{t-1} + U1_m(h1_{t-1} \odot \mathrm{r}) + W_{mc1} \, {}^{1}c^t + W_{mc2} \, {}^{2}c^t + b_m) $$

$$ h1_t = (1 - z) \odot h1_{t-1} + z \odot \mathrm{m} $$

$$ h2_t = f_2(W_{h2} h1_t + b_{h2}) $$

$$ z_t = argmax(softmax(W_{zh} h1_t + W_{zh2} h2_t + W_{zz} E_{tgt}(z_{t-1}) + W_{zc1} \, {}^{1}c^t + W_{zc2} \, {}^{2}c^t + b_z)) $$

$E_{tgt}(z_{t-1})$은 task2의 이전 시간의 디코딩 결과로 생성된 단어 z_{t-1}의 출력 언어 word embedding이며, $h1_{t-1}$과 $h2_{t-1}$은 task2 디코더의 이전 hidden state이다. f_{ATT1}는 task2의 디코딩 시간 t에서 입력열 X의 인코더 hidden state vector(${}^{1}h_i$)에 대한 attention weight를 결정하기 위한 신경망이다. 생성된 attention weight는 입력열 X의 인코더 hidden state vector들을 이용하여 task2의 디코딩 시간 t에서 입력열 X에 대한 context vector ${}^{1}c^t$를 생성한다. 마찬가지로 task1에 대한 정보를 task2에 적용 시키기 위해 task1의 디코더 hidden state vector(${}^{2}h2_j$)를 또 다른 신경망인 f_{ATT2}를 이용하여 task2의 디코딩 시간 t에서 task1의 디코더 hidden state vector(${}^{2}h2_j$)에 대한 attention weight를 결정하고 마찬가지로 context vector ${}^{2}c^t$를 생성한다. 이후 [7]과 동일한 변형된 GRU 디코더를 사용하며 위에서 입력열 X에 대한 context vector ${}^{1}c^t$외 추가적으로 task1에 대한 context vector ${}^{2}c^t$를 또 다른 가중치를 두어 추가정보로 넣었다.

4. 실험 및 결과

본 논문에서 제안한 multi-task sequence-to-sequence 모델의 성능을 평가하기 위해 [7]과 동일한 세종말뭉치의 구구조 구문 분석 데이터를 사용하였다. 모든 source, target word embedding은 200차원을 사용하였고 형태소 분석(task1)의 디코더 히든레이어의 크기는 1000, 구구조 구문 분석(task2)의 디코더 히든레이어의 크기는 500으로 설계 하였다. 추가적으로 해당 데이터에 대한 형태소 분석만의 성능과 형태소 분석이 되지 않은 문장을 입력으로 할 때의 구구조 구문 분석의 성능도 측정하였다.

표 1. 한국어 형태소 분석(Task1) 및 구구조 구문 분석 (Task2) 성능 평가 결과

모델	Task1 F1	Task2 F1
RNN-search + input-feeding + copying[6]	92.48 (baseline)	-
RNN-search + input-feeding[7](raw corpus)	-	81.78 (baseline)
RNN-search + input-feeding[7](정답 형태소 분석 이용)	-	89.03(+7.25)
Model 1	94.10(+1.62)	78.56(-3.22)
Model 2	**94.78(+2.30)**	**85.61(+3.83)**

표 1은 각각의 sequence-to-sequence 모델 별 한국어 형태소 분석과 구구조 구문 분석의 성능 평가 결과 이다. Task1은 형태소 분석을 나타내며, task2는 구구조 구문 분석을 나타낸다. 먼저 [6]에서 제안된 RNN-search + input-feeding + copying 모델로 형태소 분석만을 시도할 시 F1 92.48의 성능을 보여 해당 데이터가 [6]의 데이터보다 크기가 작고, 형태소 분석이 어려운 데이터임을 알 수 있다([6]의 학습데이터는 9만 문장, 본 논문에서 사용한 학습데이터는 3만9천 문장). 마찬가지로 [7]에서 제안된 RNN-search + input-feeding 모델로 형태소 분석이 되지 않은 문장을 입력으로 받았을 시 F1 81.78의 낮은 구구조 구문 분석 성능을 보여 정답 형태소 분석을 사용한 [7]의 F1 89.03의 성능과 비교해 한국어 구구조 구문 분석에서 형태소 분석이 중요함을 알 수 있다. 표 1에서 model 1은 3.1절에서 제안한 단순히 출력 결과를 합친 sequence-to-sequence 모델이며 model 2는 3.2절에서 제안한 hidden state를 공유하는 multi-task sequence-to-sequence 모델이다. 실험 결과 출력 결과를 합친 model 1의 경우 형태소 분석의 성능이 F1 94.10이었으나 구구조 구문 분석 성능은 F1 78.56으로 형태소 분석이 되지 않은 문장을 입력으로 받았을 시의 F1 81.78의 성능보다 낮게 나왔다. 그러나 구구조 구문 분석 시 형태소 분석(task1)의 정보를 보게 설계한 model 2의 경우 구구조 구문 분석의 성능이 F1 85.61으로 형태소 분석이 되지 않은 문장을 입력으로 받았을 때보다 높은 성능을 보였다. Model 1에서의 구구조 구문 분석은 형태소 분석 정보를 사용하지 못함과 동시에 sequence-to-sequence 모델의 디코더 출력이 구구조 구문 분석 태그뿐만 아니라 형태소 분석 결과도 출력을 해야 하기 때문에 구문 분석의 난이도가 올라 것으로 분석된다. Model 2에서의 구구조 구문 분석은 형태소 분석이 되지 않은 문장을 입력으로 받았을 시의 구구조 구문 분석의 성능보다 높게 나와 task1의 형태소 분석 정보를 효과적

으로 사용하였음을 확인할 수 있다. 또한 model 1과 model 2 모두 [6]의 모델을 이용하여 순수하게 형태소 분석만을 시도한 경우보다 높은 성능을 보여 한국어 구구조 구문 분석에 한국어 형태소 분석이 영향이 끼치는 것은 물론 한국어 형태소 분석에 한국어 구구조 구문 분석이 영향을 미칠 수 있음을 보여준다. 그림 3은 hidden state를 공유하는 multi-task sequence-to-sequence 모델의 구구조 구문 분석 attention weight 예시이다. 구구조 구문 분석 시 입력열에 대한 정보뿐만 아니라 형태소 분석의 디코더 hidden state에 대한 정보를 [7]과 유사하게 어절의 구구조 구문 분석 태그 생성시 해당하는 어절 정보를 보려고 하는 것을 볼 수 있다.

감사의 글

이 논문은 2016년도 정부(미래창조과학부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임 (No. NRF-2016R1C1B1014124)

5. 결론

본 논문에서는 한국어 형태소 분석 및 구구조 구문 분석을 동시에 시도하는 multi-task sequence-to-sequence 모델을 제안하였다. 실험 결과 한국어 구구조 구문 분석에 형태소 분석이 영향을 미치게 설계하여 성능이 향상되는 것을 확인하였으며, 한국어 형태소 분석에 구구조 구문 분석 또한 영향을 미칠 수 있음을 확인하였다. 향후 연구로 단일 task 데이터 및 대규모 raw corpus 활용 등을 이용한 성능 향상 방안을 모색할 예정이다.

참고문헌

[1] Collobert, Ronan, et al. "Natural language processing (almost) from scratch." Journal of Machine Learning Research, 12, 2011.

[2] 이창기, 김준석, 김정희, "딥 러닝을 이용한 한국어 의존 구문 분석", 제26회 한글 및 한국어 정보처리 학술대회, pp. 87-91, 2014.

[3] Sutskever, Ilya, Oriol Vinyals, and Quoc V. Le. "Sequence to sequence learning with neural networks." Advances in neural information processing systems, 2014.

[4] Cho, Kyunghyun, et al. "Learning phrase representations using RNN encoder-decoder for statistical machine translation." EMNLP 2014

[5] 이건일, 이의현, 이종혁. "Sequence-to-sequence 기반 한국어 형태소 분석 및 품사 태깅." 정보과학회논문지 44.1 (2017): 57-62.

[6] 황현선, 이창기. "Copying mechanism 을 이용한 Sequence-to-Sequence 모델기반 한국어 형태소 분석." 한국정보과학회 학술발표논문집 (2016): 443-445.

[7] 황현선, 이창기, Sequence-to-Sequence 모델을 이용한 한국어 구구조 구문 분석, HCLT 2016

[8] 이창기, “Structural SVM을 이용한 한국어 띄어쓰기 및 품사 태깅 결합 모델”, 정보과학회논문지 : 소프트웨어 및 응용, 40(12), pp826-832, 2013.

[9] 이공주, 김재훈, 김길창. "한국어 구구조 문법을 기반으로 하는 확률적 구문 분석." 한국정보과학회 1996 년도 가을 학술발표논문집, 제23권, 제2호(A), pp557-560, 1996.

[10] Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio. "Neural machine translation by jointly learning to align and translate." International Confernece on Learing Represnetations, 2015.

[11] Luong, Minh-Thang, Hieu Pham, and Christopher D. Manning. "Effective approaches to attention-based neural machine translation." EMNLP 2015

[12] 박천음, 이창기. "포인터 네트워크를 이용한 한국어 의존 구문 분석." 정보과학회논문지 44.8 (2017): 822-831.

그림 3. Hidden state를 공유하는 multi-task sequence-to-sequence 모델의 구구조 구문 분석 attention weight 예시