Input-feeding RNN Search 모델과 CopyNet을 이용한 한국어 의미역 결정

배장성○, 이창기
강원대학교
jseffort88@gmail.com, leeck@kangwon.ac.kr

Korean Semantic Role Labeling using Input-feeding RNN Search Model with CopyNet

Jangseong Bae○, Changki Lee
Kangwon National University

요 약

본 논문에서는 한국어 의미역 결정을 순차열 분류 문제(Sequence Labeling Problem)가 아닌 순차열 변환 문제(Sequence-to-Sequence Learning)로 접근하였고, 구문 분석 단계와 자질 설계가 필요 없는 End-to-end 방식으로 연구를 진행하였다. 음절 단위의 RNN Search 모델을 사용하여 음절 단위로 입력된 문장을 의미역이 달린 어절들로 변환하였다. 또한 순차열 변환 문제의 성능을 높이기 위해 연구된 인풋-피딩(Input-feeding) 기술과 카피넷(CopyNet) 기술을 한국어 의미역 결정에 적용하였다. 실험 결과, Korean PropBank 데이터에서 79.42%의 레이블 단위 f1-score, 71.58%의 어절 단위 f1-score를 보였다.

주제어: 의미역 결정, RNN Search, Input-feeding, CopyNet

1. 서론

의미역 결정(Semantic Role Labeling)은 문장의 각 서술어의 의미와 그 논항(Argument)들의 의미역을 결정하여 "누가, 무엇을, 어떻게, 왜" 등의 의미 관계를 찾아내는 자연어처리의 한 단계이며 관계추출, 질의응답과 같은 다양한 자연어처리 응용의 성능 향상을 위한 입력으로 사용될 수 있다.

최근 자연어처리 연구에는 이미지처리, 음성인식 연구에서 큰 성공을 거둔 딥러닝(Deep-learning)을 자연어처리에 적용하는 연구가 많이 이루어지고 있다[1-11]. 그 중 순환 신경망(Recurrent Neural Network)을 이용한 Sequence-to-Sequence 모델은 임의 길이의 입력 순차열을 다른 종류의 순차열로 변환하는 확률모델로 기계번역 연구에 주로 사용되고 있다[2-8].

본 논문에서는 기존 한국어 의미역 결정 연구와 [11,12,13] 달리 한국어 의미역 결정 문제를 순차열 분류 문제(Sequence Labeling Problem)가 아닌 순차열 변환 문제(Sequence-to-Sequence Learning)로 접근하였다. 이를 위해 순차열 변환에 적합한 RNN Search 모델을 한국어 의미역 결정에 적용하였고, 순차열 변환 문제에 사용되어 성능 향상이 있다고 알려진 인풋-피딩(Input-feeding) 기술과 카피넷(CopyNet) 기술을 적용하였다[6,7]. 실험 결과, Korean PropBank[14] 데이터에서 79.42%의 레이블 단위 f1-score, 71.58%의 어절 단위 f1-score를 보였다.

본 논문의 구성은 다음과 같다. 제 2장에서는 관련 연구를 소개하고 제 3장에서는 본 논문에서 제안한 한국어 의미역 결정 시스템을 기술한다. 제 4장에서는 실험 및 결과를 분석하고, 제 5장에서는 결론에 대해 기술한다.

2. 관련 연구

의미역 결정 연구는 크게 격틀사전에 기반을 둔 방법과 말뭉치에 기반을 둔 방법으로 나눌 수 있다. 격틀사전에 기반을 둔 방법은 서술어와 논항에 대한 문법 관계를 기술한 격틀(Frame)과 논항들의 정보를 기술한 선택제약(Selectional Restriction)등을 이용하여 의미역을 결정하는 방법이다. 말뭉치에 기반을 둔 방법은 의미역이 태깅된 말뭉치를 구축하고, 구축된 말뭉치를 이용하여 기계학습 방법으로 의미역을 결정하는 방법이다. 말뭉치를 이용하는 방법은 격틀사전에 기반을 둔 방법에 비해 적용률이 높은 장점이 있다[15]. 최근 한국어 의미역 결정 연구에는 Korean PropBank, 엑소브레인 언어분석 말뭉치[16]와 같은 공개된 말뭉치를 이용한 연구가 주로 이루어지고 있다[11,12,13].

[11,12,13]은 한국어 의미역 결정을 순차열 분류 문제로 접근하였고, 기계학습 방법을 이용하여 연구를 진행하였다. [12]의 연구는 Structural-SVM(Support Vector Machine)을 이용해 여러 자질조합을 이용하여 의미역 결정의 성능을 높였으나 자질 조합 설계에 많은 시간이 소요되는 단점이 있다. [11]의 연구는 양방향 Long Short-Term Memory(LSTM) 순환 신경망을 사용하여 멀리 떨어져있는 어절의 정보를 볼 수 있게 되어, 의미역 결정 성능에 큰 영향을 주는 구문 분석 정보의 사용 없이

도 좋은 성능을 얻을 수 있었다. [13]의 연구는 의미 정보를 기계학습의 자질로 채택하는 방법을 제시하였다. [13]의 연구는 의미역의 부가격 필터링 및 유의어 사전, 개체명인식 정보, 격틀사전 확장 등을 통해 높은 성능을 보였으나, 전체 시스템의 크기가 커지고 개체명 인식기 등을 추가적으로 사용해야 하는 단점이 있다.

본 연구는 한국어 의미역 결정을 순차열 변환 문제로 접근하였고, 순차열 변환 문제를 다루는 기계번역 분야의 [5]의 연구와 이를 구문분석 분야에 적용한 [9]의 연구, 형태소 분석에 적용한 [10]의 연구를 바탕으로 하고 있다. [6,7]의 연구는 순차열 변환 문제에 사용되어 성능 향상을 보인 기술로 [6]의 연구는 네트워크의 수직, 수평적 깊이를 밸런스 있게 만들어주는 역할을 한다. [7]의 연구는 출력 결과의 Out-of-Vocabulary(OOV) 문제를 해결 하고, 고유명사의 출력 확률이 작아지는 문제점을 해결할 수 있게 한다. 본 연구는 [6,7]의 기술들을 한국어 의미역 결정에 적용한다.

3. 한국어 의미역 결정 시스템

3.1 입력 및 출력 설계

본 논문에서는 한국어 의미역 결정을 순차열 변환 문제로 접근하여 그림 1과 같이 입력과 출력을 설계하였다.

입력은 한국어 음절, 형태소 그리고 동사를 나타내는 태그, 공백 음절을 나타내는 태그로 이루어져있다. 동사를 나타내는 태그는 <PRED>로 시작하여 </PRED>로 끝나며, <SP>는 하나의 공백을 나타내어 어절의 띄어쓰기 정보를 알 수 있다. <PRED> 태그는 문장의 현재 동사를 나타내기 위해 사용한다. 즉, 2개 이상의 동사로 이루어진 문장을 표시하시 위해서는 동사의 개수만큼 문장이 반복된다.

본 논문의 출력은 어텐션의 가중치(Weight)를 통해 어절의 위치 정보를 찾을 수 있다는 가정을 전제로 하며 다음과 같이 설계하였다. 출력은 한국어 음절 및 동사 태그, 동사의 의미 번호, 그리고 의미역 태그로 구성되며, 의미역 태그가 부여되지 않은(None label) 어절은 출력에서 제외 한다.

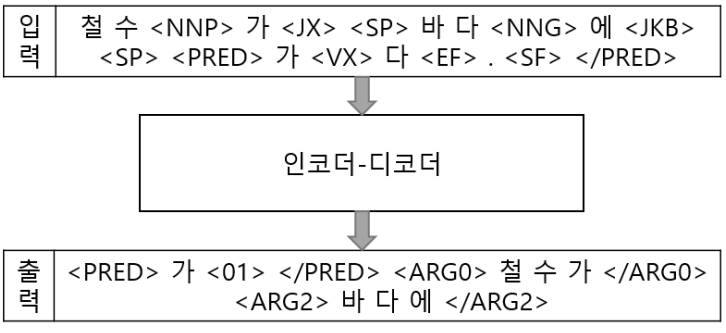

그림 1. 한국어 의미역 결정 시스템의 입력 및 출력

3.2 Input-feeding RNN Search 모델

RNN 인코더-디코더(Encoder-Decoder) 모델은 순환 신경망을 이용하여 P(y|x)를 최대화 하는 모델로(x:입력 열, y: 출력 결과 열), 그림 2는 RNN 인코더-디코더 모델을 나타낸다.

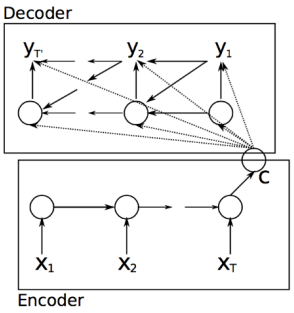

그림 2. RNN 인코더-디코더 모델

인코더는 입력 열을 실수로 이루어진 벡터 표현 c(Context Vector)로 인코딩한다. 디코더는 이로부터 P(y|x)를 최대화하는 출력 열을 생성한다. 본 논문은 인코더에 Long Term Dependency를 학습할 수 있는 순환 신경망의 일종인 Gated Recurrent Unit(GRU)[3]을 사용한다. RNN 인코더-디코더 모델은 입력 열을 길이에 상관없이 고정된 차원의 단일 벡터로 인코딩하는데, 이로 인해 입력 열이 길어질 경우 성능이 하락하는 문제점이 있다. 또한 출력 열의 변환이 잘못 되었을 때 원인을 분석하기 어려운 단점이 있다.

[7]에서는 어텐션 메카니즘(Attention Mechanism)을 인코더와 디코더 사이에 두어 이러한 문제점들을 해결하는 RNN Search 모델을 소개하였다. 그림 3은 Input-feeding RNN Search 모델을 나타낸다.

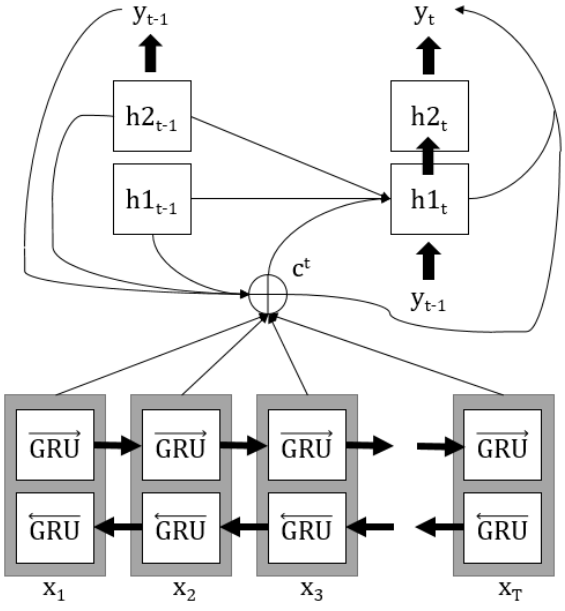

그림 3. Input-feeding RNN Search 모델

인코더는 Bidirectional(양방향) GRU를 사용하여 입력 열을 실수로 이루어진 벡터 표현 c로 인코딩하며 그 식은 아래와 같다. 또한 Forward(정방향)와 Backward(역방향)의 GRU는 가중치를 공유하지 않는다.

$$\overrightarrow{h_t} = GRU_{Forward}(E_{src}(x_t), \overrightarrow{h}_{t-1})$$

$$\overleftarrow{h_t} = GRU_{Backward}(E_{src}(x_t), \overleftarrow{h}_{t+1})$$

$$\overleftrightarrow{h_t} = [\overrightarrow{h_t}; \overleftarrow{h_t}]$$

인풋-피딩을 적용한 RNN Search 모델의 수식은 다음과 같다.

$$e_i^t = f_{ATT}(E_{tgt}(y_{t-1}), h1_{t-1}, h2_{t-1}, \overleftrightarrow{h_i})$$

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

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

$$z = \sigma(W_z E_{tgt}(y_{t-1}) + U1_z h1_{t-1} + U2_z h2_{t-1} + W_{zc}c^t + b_z)$$

$$r = \sigma(W_r E_{tgt}(y_{t-1}) + U1_r h1_{t-1} + U2_r h2_{t-1} + W_{rc}c^t + b_r)$$

$$m = f(W_m E_{tgt}(y_{t-1}) + U2_m h2_{t-1} + U1_m(h1_{t-1} \odot r) + W_{mc}c^t + b_m)$$

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

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

$$y_t = argmax(softmax(W_{yh}h1_t + W_{yh2}h2_t + W_{yy}E_{tgt}(y_{t-1}) + W_{yc}c^t + b_y))$$

$E_{tgt}(y_{t-1})$은 이전 시간의 디코딩 결과로 생성된 출력인 y_{t-1}의 출력 워드임베딩이다. W, U1, U2는 가중치 행렬이며, b는 Bias이고, σ는 Sigmoid 함수이다. 또한 f는 Tanh 함수이고, ⊙는 Element-wise product 이다. f_{ATT}은 어텐션 가중치를 결정하기 위한 Feed-Forward Neural Network(FFNN)이며, 이전의 출력 정보 $E_{tgt}(y_{t-1})$, 이전 히든레이어의 상태 $h1_{t-1}$/$h2_{t-1}$, 인코딩된 히든레이어의 상태를 입력으로 받는다. 새로 생성된 어텐션 가중치는 인코딩 된 히든레이어의 상태에서 t시간의 새로운 Local Context Vector c^t를 생성한다. 디코더는 두 개의 히든레이어를 사용하며 첫 번째 층은 어텐션 메카니즘과 인풋-피딩을 적용한 GRU이며, 비선형변환 함수로는 Tanh를 사용하였다. 두 번째 층은 ReLU를 비선형변환 함수로 사용하는 FFNN이다. 출력에서는 Softmax 함수를 사용하여 각 출력의 생성 확률을 만들어 낸다. 이중 값이 가장 큰 출력을 해당 디코딩 시간의 결과로 생성한다.

인풋-피딩은 그림 3의 t-1 시간의 두 번째 히든레이어의 상태 $h2_{t-1}$를 디코딩 시 추가 입력 정보로 넣어 t-1 시간에 결정된 어텐션 가중치 정보를 현재 시간에 활용하여 네트워크의 수직, 수평적 깊이를 균형 있게 만들어 주는 역할을 한다.

3.3 Input-feeding RNN Search with CopyNet 모델

기존 RNN Search 모델은 출력 결과를 생성할 때 고유명사나, UNK(Unknown Word)로 인한 Out-of-Vocabulary(OOV) 문제점과 고유명사의 출력 확률이 작아지는 문제가 있다. [7]은 입력 정보를 복사할지, 디코더에 의해 생성되는 출력 정보를 사용할 지를 정할 수 있게끔 시스템을 설계하여 이러한 문제점을 해결하였다. 본 논문에서도 위와 같은 문제를 해결하기 위해 디코더의 출력 결과 생성을 그림4와 같이 확장하였고, 수식은 아래와 같다.

$$e_{copy_j^t} = f_{ATT_copy}(E_{tgt}(y_{t-1}), h1_{t-1}, h2_{t-1}, \overleftrightarrow{h_i})$$

$$s_t = W_{yh}h1_t + W_{yh2}h2_t + W_{yy}E_{tgt}(y_{t-1}) + W_{yc}c^t + b_y$$

$$P(y_t|x) = \begin{cases} \frac{1}{Z}\left(\exp(s_t) + \sum_{j:x_j=y_t}\exp\left(e_{copy_j^t}\right)\right), & y_t \in X \\ \frac{1}{Z}\exp(s_t), & otherwise \end{cases}$$

e_{copy}는 카피넷을 위한 어텐션 값 이며, s_t는 입력이 현재 디코더의 결과로 출력될 값을 의미한다. Z는 정규화 상수를 나타낸다. 마지막으로 $P(y_t|x)$는 각 입력의 출력 값과 카피 어텐션 값에 따라 출력 사전에 있는 어휘와 입력열에 등장하는 어휘 중 더 높은 확률을 갖는 결과를 출력하게 된다.

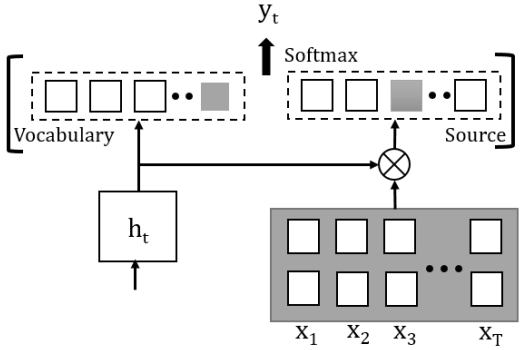

그림 4. 카피넷이 적용된 디코더 출력

3.4 디코딩

디코딩은 디코더에서 출력한 음절 별 확률을 이용한 Beam search를 사용하였다. 본 논문에서 사용된 Beam size는 5, 10, 20 이며 그 중 성능이 가장 좋은 10을 사용하였다.

4.실험 및 결과

사용된 데이터는 Korean PropBank 말뭉치 중 Newswire 말뭉치이며, 학습데이터에 19302문장, 평가데이터에 3778 문장을 사용하였다. 실험에 사용된 Input-feeding RNN Search with CopyNet 시스템은 Theano[17]를 이용하여 자체적으로 구현하였으며, 학습은 RMSProp를 사용하였다. 입력 및 출력 언어 모두 200차원의 워드임베딩(word embedding)을 사용하였고, 히든레이어의 유닛 수는 500을 사용하였다. 또한 과적합 문제를 해결하기 위해 Dropout[18]을 사용하였으며, 그 값은 0.25로 모든 레이어에 동일하게 적용하였다. 모든 실험은 Intel(R)

Core(TM) i7-4790K CPU @ 4.00GHz, 32GB RAM, Nvidia GeForce GTX 980 Ti, Ubuntu 14.04 환경에서 수행되었다.

본 논문에서 제시하고 있는 성능은 논항 인식 및 분류(AIC: Argument Identification & Classification)에 해당하며 성능 지표로는 정확도와 재현율의 조화평균인 f1-score를 사용한다. 또한 어절 단위, 레이블 단위의 성능으로 나누어 평가한다. 어절 단위 성능은 음절로 출력된 결과들이 정확히 하나의 어절과 일치하고, 의미역 태그 또한 일치할 때의 성능이며, 레이블 단위 성능은 음절의 맞고 틀림 여부에 상관없이 레이블을 맞추었을 때의 성능이다.

표 1. 한국어 의미역 결정 성능 평가 결과(AIC)

모델	어절 단위	레이블 단위
RNN Search 모델 + 인풋-피딩	65.29	75.48
RNN Search 모델 + 인풋-피딩 + 카피넷	71.58	79.42
Structural-SVM[11]	76.96	-
Bidirecitonal LSTM-CRFs[12]	78.17	-

정답

<PRED> 모 르 <01> </PRED> <ARG1> 제 의 하 ㄹ 지 </ARG1>
<PRED> 밝 히 <01> </PRED> <ARG1> 모 르 ㄴ 다 고 </ARG1>
<ARG0> 관 계 자 가 </ARG0> <ARGM-TMP> 1 일 </ARGM-TMP>
<PRED> 맡 <01> </PRED> <ARG1> 인 수 문 제 를 </ARG1>
<AUX> 있 는 </AUX> <ARG0> 르 푸 는 </ARG0>
<PRED> 제 의 <01> </PRED> <ARG1> 인 수 를 </ARG1>
<ARGM-MNR> 공 식 </ARGM-MNR>
<PRED> 위 하 <01> </PRED> <ARG1> 제 의 하 기 </ARG1>
<ARG0> 사 업 계 획 서 를 </ARG0>
<PRED> 있 <01> </PRED> <ARG1> 르 노 가 </ARG1> <ARG2> 마 무 리 중 에 </ARG2>

시스템이 출력한 결과

<PRED> 모 르 <01> </PRED> <ARG1> 제 의 하 ㄹ 지 </ARG1>
<PRED> 밝 히 <01> </PRED> <ARG1> 모 르 ㄴ 다 고 </ARG1>
<ARG0> 관 계 자 가 </ARG0> <ARGM-TMP> 1 일 </ARGM-TMP>
<PRED> 맡 <01> </PRED> <ARG1> 인 수 문 제 를 </ARG1>
<AUX> 있 는 </AUX> <ARG0> 르 푸 는 </ARG0>
<PRED> 제 의 <01> </PRED> <ARG1> 인 수 를 </ARG1>
<ARGM-MNR> 공 식 </ARGM-MNR>
<PRED> 위 하 <01> </PRED> <ARG1> 제 의 하 기 </ARG1>
<ARG0> 사 업 계 획 서 를 </ARG0>
<PRED> 있 <01> </PRED> <ARG1> **제 의** 가 </ARG1> <ARG2> 마 무 리 중 에 </ARG2>

그림 5. Input-feeding RNN Search with CopyNet 시스템의 출력 예시

표 1은 한국어 의미역 결정 성능 평가 결과를 나타낸다. 본 논문의 입력, 출력 설계 방식에 맞는 비교 가능한 의미역 결정 연구가 없어, 동일한 데이터를 사용한 연구를 [11,12] 비교 대상으로 한다. RNN Search 모델, 인풋-피딩, 카피넷을 사용한 모델의 성능이 어절 단위 71.58%, 레이블 단위 79.42%의 성능을 보였다. 어절 단위 성능에서 기존 연구보다 낮은 성능을 보였으나, 레이블 단위 성능은 더 앞선 것을 볼 수 있었다. 어절 단위의 성능은 시스템이 생성하는 의미역 태그 및 모든 음절이 고려되기 때문에, 이미 알고 있는 어절을 분류 문제로 해결한 [11,12]의 성능보다 필연적으로 낮다고 생각된다. 그림 5는 본 시스템의 출력 결과이며, 의미역 태그는 맞았으나 음절이 틀려 오류로 측정된 결과가 나타남을 알 수 있다. 또한, 카피넷을 적용한 모델이 그렇지 않은 모델보다 어절 단위 성능에서 6.29% 더 높은 성능을 보였다. 이를 통해 카피넷이 순차열 변환 문제에서 OOV문제와 고유명사의 출력 확률 저하 문제를 해결하여 성능 향상에 크게 도움이 됨을 알 수 있다.

5. 결론

본 논문의 취지는 기존 Sequence-to-Sequence 모델을 이용하여 한국어 의미역 결정 연구 진행의 가능성을 탐색하는데 있다. 따라서 한국어 의미역 결정 문제를 순차열 변환 문제로 보고 접근하였고, 이 문제를 해결하기 위해 Sequence-to-Sequence 모델을 한국어 의미역 결정에 맞게 설계 하였다. 또한 카피넷을 적용하여 OOV 문제와 출력 열의 고유명사 출력 확률 저하 문제를 해결하여 순차열 변환 문제에 성능 향상이 있음을 보였다.

향후 연구로는 Sequence-to-Sequence 모델을 한국어 의미역 결정에 맞게 구조 및 기능을 조정하고, 다른 자연어처리 분야에 적용하는 방법을 연구할 계획이다.

감사의 글

이 논문은 2016년도 정부(미래창조과학부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임. (No.R0101-16-0062, (엑소브레인-1세부) 휴먼 지식증강 서비스를 위한 지능진화형 WiseQA 플랫폼 기술 개발)

참고문헌

[1] Ronan Collobert, et al, “Natural Language Processing (almost) from scratch”, The Journal of Maching Learing Research, Vol. 12, pp. 2493-2537, 2011.

[2] Sutskever, Ilya, Oriol Vinyals, and Quoc V. Le, "Sequence to sequence learning with neural networks", Advances in neural information processing systems, pp. 3104-3112, 2014.

[3] Kyunghyun Cho, et al, "On the properties of neural machine translation: Encoder-decoder approaches", Eighth Workshop on Syntax,

Semantics and Structure in Statistical Tranlation, pp. 103-111, 2014.
[4] Dzmitry Bahdanau, et al, "Neural machine translation by jointly learning to align and translate", International Confernece on Learing Represnetations, 2015.
[5] Chung Junyoung, et al, "A character-level decoder without explicit segmentation for neural machine translation", arXiv preprint arXiv:1603.06147 (2016).
[6] Luong Minh-Thang, et al, "Effective approaches to attention-based neural machine translation." arXiv preprint arXiv:1508.04025 (2015).
[7] Gu Jiatao, et al, "Incorporating copying mechanism in sequence-to-sequence learning." arXiv preprint arXiv:1603.06393 (2016).
[8] 이창기, et al, "문자 단위의 Neural Machine Translation", 한글 및 한국어 정보처리학술대회, pp.115-118, 2015.
[9] Vinyals Oriol, et al, "Grammar as a foreign language", Advances in Neural Information Processing Systems, pp. 2755-2763, 2015.
[10] 인건일, et al, “Sequence-to-Sequence 모델을 이용한 한국어 형태소 분석 및 품사 태깅”, 한국컴퓨터종합학술대회 논문집, pp. 693-695, 2015.
[11] 배장성, 이창기, “Bidirectional LSTM CRF를 이용한 End-to-end 한국어 의미역 결정”, 한국정보과학회 학술발표논문집, pp. 566-588, 2015.
[12] 이창기, 임수종, 김현기, Structural SVM 기반의 한국어 의미역 결정, 정보과학회논문지 제 42권 제 2호, pp. 220-226, 2015.
[13] 임수종, 김현기, “의미 정보를 이용한 한국어 의미역 인식 연구”, 한글 및 한국어 정보처리학술대회, pp 18-22, 2015.
[14] Martha Palmer, et al, Korean PropBank [Online]. Available: http://catalog.ldc.upenn.edu/LDC2006T03.
[15] 김병수, 외 4인, 부트스트랩핑 알고리즘을 이용한 한국어 격조사의 의미역 결정, 한국정보과학회논문지 제 33권, 제 1호, 2006.
[16] Exobrain corpus[Online]. Available: https://astc.etri.re.kr/.
[17] Bastien, Frédéric, et al. "Theano: new features and speed improvements." arXiv preprint arXiv:1211.5590 (2012).
[18] Geoferrey Hinton, et al, “Improving neural networks by preventing co-adaptation of feature detectors”,CoRR, abs/1207.0580, 2012.