딥러닝을 이용한 전이 기반 한국어 형태소 분석 및 품사 태깅

민진우[○†], 나승훈[†], 김영길[††]

전북대학교[†], ETRI[††]
Jinwoomin4488@gmail.com, nash@jbnu.ac.kr, kimyk@etri.re.kr

A Transition based Joint Model for Korean Morpheme Segmentation and POS Tagging Using Deep Learning

Jin-Woo Min[○†], Seung-Hoon Na[†], Young-Kil Kim[††]
Chonbuk National University[†], ETRI[††]

요 약

한국어 형태소 분석은 많은 자연어 처리 분야에서 핵심적인 역할을 수행하고 있기 때문에 형태소를 분류하고 형태소에 맞는 알맞은 품사를 결정하는 것은 매우 중요하다. 형태소의 품사를 태깅하는 대표적인 방법은 크게 음절 단위 형태소 분석과 단어 단위 형태소 분석의 두 가지로 나눌 수 있다. 본 논문에서는 의존 파싱 분야에서 널리 활용되고 있는 전이 기반 방식을 적용하여 전이 기반 단어 단위 한국어 형태소 분석 모델을 제안하고 해당 모델을 한국어 형태소 분석 데이터인 세종 품사 부착 말뭉치 셋에 적용하여 F1 97.77 %로 기존의 성능을 더욱 향상시켰다.

주제어: 딥러닝, 형태소 분석, 품사 태깅, 전이 기반

1. 서론

형태소 분석은 많은 자연어 처리 분야에서 핵심적인 수행하고 있다. 한국어 형태소 분석은 일반적으로 형태소 분석과 품사 태깅의 두 가지의 과정으로 구분하며 형태소 분석은 문장 내의 어절을 뜻을 지니는 최소의 단위인 형태소로 분해하고 해당 형태소의 품사 후보를 생성하는 작업이고 품사 태깅은 위의 품사 후보로부터 가장 적절한 품사를 결정하는 것이다[1].

형태소 분석은 크게 음절 기반 형태소 분석과 단어 기반 형태소 분석으로 나뉠 수 있으며 음절 기반 형태소 분석은 입력된 문장을 음절 단위로 나누고 순차 레이블링 문제로 보고 [B(Begin), I(inside)] 혹은 [B, I, E(End), S(Single)]가 태그가 포함된 품사태그를 결정하는 방식이다. 반면, 형태소 기반 방식은 분할된 형태소에 직접 바로 태그를 부여하는 방식이다[9]. 한국어 형태소 분석에 대한 연구는 CRF(Conditional Random Fields), SVM(Support Vector Machine)[2,4]등 기존의 기계학습 방법이 주를 이루었으나 최근 들어 한국어 형태소 분석에서도 다양한 자연언어 처리에서 각광받고 있는 RNN 계열의 딥러닝 모델들[5-7,9]을 적용하는 연구가 많이 진행되고 있다.

의존 파싱 문제에서 널리 연구되고 있는 전이 기반 방식[8]은 입력에 대한 버퍼와 스택의 상태로부터 자질벡터들을 얻어 결합한 후 딥러닝 신경망을 통해 해당 전이 액션을 결정하는 방식이다.

본 논문에서는 의존 파싱에서 널리 활용되고 있는 전이 기반 방식을 한국어 형태소에 맞는 액션을 정의하고 액션에 의해 형태소를 분할하고 품사를 부여하는 형태소 기반 방식으로 딥러닝 모델에 적용하여 세종 품사 부착 말뭉치 셋에서 F1 97.77%으로 기존 모델보다 높은 성능을 보였다.

2. 관련 연구

한국어 품사 태깅에 대한 다양한 연구가 진행되었다. 음절 기반 한국어 형태소 분석은 주로 순차 레이블링 기반으로 연구가 진행되었는데 이러한 순차 레이블링의 기계학습 모델은 CRF, SVM 모델 등이 있다. [4]에서는 Structural SVM를 활용하여 한국어 띄어쓰기 및 품가 태깅 결합 모델을 제안하였다. [2]에서는 CRF에 기반한 형태소 분석 모델을 제안하였으며 1) 형태소 분할 단계, 2) 품사 태깅 단계, 3) 복합 형태소 분할 및 태깅 단계의 세 단계로 품사 태깅을 진행한다.

딥러닝을 이용한 품사 태깅 연구도 진행되었는데 [5]에서는 음절 기반으로 형태소 분석을 진행하였으며 품사 태그의 빈도수를 계산 한 후 softmax로 수치화하여 벡터의 값으로 활용하고 품사 태깅, 개체명 인식 등 순차 레이블링 문제에서 우수한 성능을 보이는 Bi-LSTM CRF 모델을 적용하였다.

Sequence-to-Sequence 모델은 임의 길이의 한 종류의 시퀀스를 다른 한 종류의 시퀀스로 변환하는 딥러닝 모델로 기계번역 분야에서 탁월한 성능을 보여주고 있다. [6]에서는 입력문장을 해당 형태소와 품사 태그로 번역하는 모델로 보고 Sequence-to-Sequence 모델을 한국어 형태소 분석 및 품사 태깅 문제에 적용하는 연구가 진행되었다. [7]에서는 Sequence-to-Sequence 모델

표 1. 전이 액션 별 스택 및 버퍼 정보의 갱신 과정

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

표 2. 형태소 분석 전이 액션의 실행 예

Action	S	B	$Tagging$
$init$	[]	[내,<SP>,고,향,은,<SP>,서,울,이,다,.]	…
$Split(MM)$	[내]	[<SP>,고,향,은,<SP>,서,울,이,다,.]	내/MM
$Merge$	[내]	[고,향,은,<SP>,서,울,이,다,.]	…
$Split(NNG)$	[내,고]	[향,은,<SP>,서,울,이,다,.]	고향/NNG
$Merge$	[내,고]	[은,<SP>,서,울,이,다,.]	…
$Split(JX)$	[내,고,은]	[<SP>,서,울,이,다,.]	은/JX
$Merge$	[내,고,은]	[서,울,이,다,.]	…
$Split(NNP)$	[내,고,은,서]	[울,이,다,.]	서울/NNP
$Merge$	[내,고,은,서]	[이,다,.]	…
$Split(VCP)$	[내,고,은,서,이]	[다,.]	이/VCP
$Split(EF)$	[내,고,은,서,다]	[.]	다/EF
$Split(SF)$	[내,고,은,서,다,.]	[]	./SF

을 확장하여 입력 열의 단어들이 출력 열에도 등장하는 경우 해당 단어들을 복사하는 Copying Mechanism을 활용하는 연구도 진행되었다.

[9]에서는 중국어 형태소 분할 문제를 전이 기반 방식으로 적용하여 현재 음절을 현재 스택이 가리키는 형태소에 부착하는 액션, 현재 스택의 형태소를 결정짓고 버퍼가 가리키는 음절을 스택으로 이동하여 새로운 형태소의 시작 음절로 하는 2가지 액션으로 적용하여 중국어 형태소 분할 문제에서 기존의 성능을 향상시켰다.

본 논문에서는 [9]의 전이 기반 형태소 분할 문제를 한국어 형태소 분석 및 품사 태깅 문제로 확장하여 적용하여 기존의 한국어 형태소 분석의 성능을 향상시킬 수 있음을 보인다.

3. 전이 기반 한국어 형태소 분석 모델

본 논문에서 사용한 모델의 구조는 [9]의 전이 기반 형태소 분할 문제를 한국어 형태소 분석 및 품사 태깅 문제로 확장하였으며 형태소 분할 및 태깅을 전이 액션으로 하고 이를 딥러닝을 통해 결정하는 모델이다.

3.1. 형태소 분할 및 품사 태깅을 위한 전이 액션

본 논문에서는 [9]에서의 전이 액션을 한국어 형태소 분석 및 품사 태깅 문제에 알맞게 확장하였다. 형태소 분할 및 품사 태깅을 위한 액션은 Split Action, Merge Action 2가지이고 역할은 다음과 같다.

• *Split* Action : 현재 스택에 존재하고 있는 형태소의 끝 경계를 결정 짓고 현재 버퍼가 가리키고 있는 음절을 스택에 PUSH한 다음 품사태그를 부여한 후 해당 형태소의 시작으로 하는 액션

• Merge Action : 현재 스택의 top이 가리키고 있는 형태소에 현재 버퍼가 가리키고 있는 음절을 해당 형태소의 구성요소로 추가하는 액션. 현재 버퍼가 Focus를 다음 음절로 이동하는 동작만을 수행

Split Action, Merge Action의 두 전이 액션 별 스택 및 버퍼 정보의 갱신 과정은 다음 표 1로 설명한다. 전이 액션을 위한 버퍼와 스택은 B, S로 표기하고 기호 c, t는 각각 음절과 품사 태그로 정의한다. Split Action이 이루어지면 버퍼에 있던 음절 c가 스택으로 PUSH되고 음절(형태소의 시작)에 품사 t가 부여됨을 알 수 있다. 다음으로 Merge Action이 실행되면 형태소에 해당 음절을 추가하는 액션으로 실제로 하는 동작은 버퍼의 Focus를 다음 음절로 이동하는 역할만을 하게 된다.

표 2는 형태소 분석에 대한 전이 액션이 이루어지는 과정을 보여준다. 표 2의 초기 상태의 버퍼를 보면 입력은 공백을 포함한 한국어 원문의 음절 열이며 *Split* Action 이 수행 되면 해당 형태소의 시작 음절의 위치에 품사 태그를 부여하는 것을 볼 수 있다. 형태소의 경계가 결정지어지는 것은 스택에 새로운 형태소의 시작 음절이 Push되는 시점이다. 예를 들어, 스택의 TOP에 형태소 "고향"이 있을 때 "의"라는 음절이 스택에 Push되면 형태소의 경계가 결정된다. 또한, 공백 음절 역시 Merge Action 액션이 수행되지만 실제 형태소에는 포함되지 않는다.

3.2. 버퍼의 입력 표상

버퍼의 입력 표상은 임력열 $\boldsymbol{x} = \{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n\}$로부터 LSTM을 통해 얻어지게 되는데 음절 임베딩 벡터 $\boldsymbol{x}_t$를 얻기 위하여 입력 문장의 t번째 음절을 c_t라 하고 해당 음절을 기준으로 자질을 추출하여 $\boldsymbol{x}_t$를 구성한다. 추출 자질 유형은 다음 표 3에 제시하며 임베딩을 얻는 과정은 수식 (1)로 표현한다.

표 3. 입력으로 사용되는 자질 유형

음절 자질	Explanation
Unigram	c_{t-1}, c_t, c_{t+1}
Bigram	$c_t c_{t+1}$
Trigram	$c_t c_{t+1} c_{t+2}$

$$\boldsymbol{x}_i = lookup([uni(i); bi(i); tri(i)]) \quad (1)$$

위의 수식에서 uni, bi, tri함수는 각각 표 3에 대응되어 n-gram 자질을 추출하는 함수이고 $lookup$함수는 임베딩 Lookup Table을 보고 해당 임베딩 벡터를 얻어내는 함수이다. 버퍼의 입력 표상은 다음 식 (2)과 같이 계산할 수 있다.

$$\{\boldsymbol{h}_1, \cdots, \boldsymbol{h}_n\} = LSTM(\{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n\}) \quad (2)$$

식 (2)에서 보듯이 입력 열 $\boldsymbol{x}$을 LSTM을 통해 언어낸 은닉 열 $\boldsymbol{h} = \{\boldsymbol{h}_1, \cdots, \boldsymbol{h}_n\}$을 버퍼의 입력 표상으로 사용하게 된다. 버퍼의 입력 표상을 얻기 위해 사용된 LSTM 신경망의 유닛은 512개가 사용되었고 각 자질의 차원은 32차원으로 입력 임베딩은 160차원이다.

3.3. 모델의 구조

본 논문에서 제안한 전이 기반 형태소 분석 모델의 구조는 다음 그림 1와 같다.

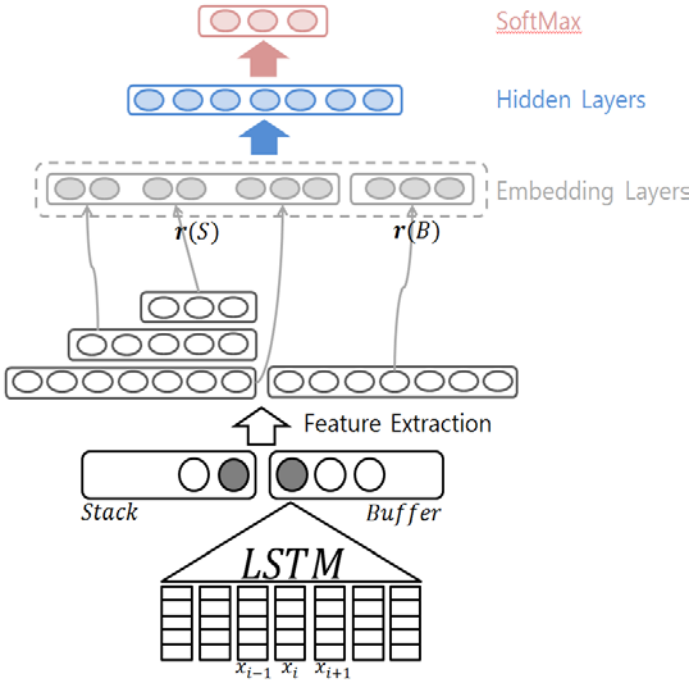

그림 1. 전이 기반 형태소 분석 모델의 구조

그림 1에서 보듯이 먼저 음절 단위로 LSTM을 통해 얻어진 은닉벡터들이 버퍼 B에 채워지게 된다. 전이 액션을 결정하기 위한 버퍼와 B, S의 해당 상태 벡터들을 각각 $\mathbf{r}(B)$, $\mathbf{r}(S)$라고 하자. S_0, S_1을 버퍼 혹은 스택의 TOP, 2번째 TOP 노드로 표현하고 해당 노드의 입력 음절에 대한 위치를 얻어내는 함수는 $cpos$이며 각 상태 백터는 다음 식 (3),(4)를 통해 얻어진다[1].

$$\mathbf{r}(B) = \boldsymbol{h}_{cpos(B_0)} \quad (3)$$

$$\mathbf{r}(S) = [\boldsymbol{h}_{cpos(S_0)}; lookup(tag(S_0)); lookup(mtag(S_1))] \quad (4)$$

식 (3)에서 보듯이 버퍼의 TOP 노드의 상태 백터 $\mathbf{r}(B)$은 단순히 입력 열 $\boldsymbol{x}$로부터 LSTM을 통해 얻어진 은닉열 $\boldsymbol{h}$에서 TOP 노드에 해당하는 위치의 은닉 벡터의 값을 취하게 된다. 식 (3)가 수행되는 과정을 표 2로 예를 들면 처음 $Merge$ Action이 수행되는 3번 째 줄 버퍼 B의 TOP노드는 "고"를 나타내고 문장 내 음절 열에서 위 음절은 세 번째에 위치하고 있으므로 LSTM을 통해 얻어진 은닉열 $\boldsymbol{h}$에서 세 번째 은닉 상태인 $\boldsymbol{h}_3$를 취하게 되는 것이다.

$\mathbf{r}(S)$는 식 (4)로 계산되며 은닉벡터를 취하는 것에 더하여 a노드에 해당하는 품사를 얻어내는 함수 tag(a)와 형태소 표층형과 해당 품사태그의 결합 정보를 얻어내는 함수인 mtag(a)를 각각 적용하여 얻어낸 후 각각 $lookup$함수를 통해 변환된 임베딩 벡터들과 결합하여 상태 벡터를 얻는다. 만약, 노드 a의 형태소가 "고향" 이면 tag(a), mtag(a)는 각각 "NNG", "고향/NNG"를 반환한다. mtag(a)가 수행 되는 노드는 3.1절에서 설명한 바와 같이 새로운 형태소가 스택에 PUSH되어 이전 형태소의 경계가 결정되는 시점이므로 S_1와 같이 스택의 2번째 TOP 형태소와 태그에 대해 적용된다.

추가로 mtag(a)에 의해 얻어져 적용되는 Lookup Table에 대해서는 [형태소-품사] 단위로 Glove 알고리즘을 적용하여 얻은 약 70만개의 형태소에 대한 100차원의 사전 학습한(pretrained) 임베딩 벡터를 사용하였다. 위에서 정의한 상태벡터 $\mathbf{r}(B)$, $\mathbf{r}(S)$을 연결하여 품사 태거 상태 표상 $\boldsymbol{p}_t$를 얻는다.

$$\boldsymbol{p}_t = [\mathbf{r}(B); \mathbf{r}(S)] \quad (5)$$

식 (5)을 통해 얻어지게 되는 품사 태거 상태 표상 집합을 $\boldsymbol{p} = \{\boldsymbol{p}_1, \cdots, \boldsymbol{p}_t\}$로 표현하며 식 (6)의 LSTM 신경망의 입력으로 하여 LSTM을 통해 인코딩 된 후 linear classifier를 사용하여 다음 전이액션으로의 점수 $\boldsymbol{score}_t$를 계산한다. 식 (6)의 $LSTM_t$함수는 LSTM을 통해 얻어진 은닉열로부터 t번째 은닉 벡터를 취하도록 한다.

$$\boldsymbol{score}_t = \boldsymbol{W} \cdot (LSTM_t(\{\boldsymbol{p}_1, \cdots, \boldsymbol{p}_t\})) + \boldsymbol{b} \quad (6)$$

얻어진 $\boldsymbol{score}_t$는 출력 층으로 연결되어 softmax를 통해 전

[1] 엄밀하게 정의하면 각 노드는 음절을 포함한 튜플의 형태로 존재하여 튜플의 음절을 얻어내는 함수인 char를 이용하여 $\boldsymbol{h}_{cpos(char(B_0))}$가 정확한 수식이지만 편의상 위의 형태를 사용한다. 식 (3),(4) 동일.

이 확률 중에 최대가 액션을 다음 전이 액션으로 하여 형태소의 분할 및 품사를 결정한다. 제안 모델의 장점은 이전 전이 액션으로 결정된 스택 내의 형태소와 해당 품사의 정보를 다음 전이 액션을 결정하기 위한 자질로 사용할 수 있어 추가적인 성능향상을 바라볼 수 있다.

4. 실험

4.1. 실험 셋팅

본 논문에서 제안한 모델을 평가하기 위해 [2]와 동일한 집합인 세종 품사 부착 말뭉치 약 25만 문장 중 75%를 학습 셋, 5%를 검증 셋 그리고 나머지 20%를 평가 셋으로 하여 본 모델을 학습하였고 모델의 학습률은 0.1로 설정하였고 모든 히든 레이어의 Dropout 비율은 0.8로 설정하였다.

4.2. 실험 결과

성능 비교를 위해 본 모델에 대한 비교 베이스 라인 모델으로는 순차 레이블링 문제에서 높은 성능을 보여주는 CRF 모델과 Bi-LSTM CRF 모델을 사용하였다. CRF에 대해서는 2가지 표기법을 사용하였다. 하나는 [B,I] 표기법이고 추가적인 표기법은 [B,I,E,S]표기법으로 각각 CRF, CRF(BIES)로 구분하며 [B,I,E,S] 표기법에 대한 설명은 다음과 같다.

- S: 형태소가 단일 음절일 경우 품사 태그
- B: 형태소의 시작 음절의 품사 태그
- E: 형태소의 마지막 음절의 품사 태그
- I: 형태소의 시작과 끝을 제외한 나머지 음절의 품사 태그

표 4. 모델 별 형태소 분석 성능

	F1(morph)
CRF * [3]	97.61%
CRF(BIES) *	97.75%
Bi-LSTM CRF *	96.96%
SVM [4]	98.03%
Seq2Seq [6]	97.15%
Copying Mechanism [7]	97.08%
전이기반 *	**97.77%**

(*는 평가셋이 동일)

표 4는 모델 별 형태소 분석 성능을 F1-measure로 보여 주고 있다. 동일 평가 셋에서 평가한 베이스 라인 모델인 CRF, Bi-LSTM CRF에 비해 제안한 전이 기반 형태소 분석 모델의 성능이 추가적인 성능 향상을 가져왔음을 확인 할 수 있다.

5. 결론

본 논문에서는 전이 기반 방식을 한국어 형태소 분석 문제에 알맞게 액션을 정의한 후 제안 모델 적용하여 기존의 방식에 비해 추가적인 성능 향상을 가져왔다. 차후 전이 기반 방식을 한국어 품사 태깅 및 의존 파싱 통합 모델에 대한 연구로 확장할 예정이다. 이에 나아가 전이 기반 방식을 개체명 인식 및 의미역 결정 문제에도 적용할 예정이다.

감사의 글

본 연구는 미래창조과학부 및 정보통신기술진흥센터의 정보통신·방송 연구개발사업의 일환으로 수행하였음. [R7119-16-1001, 지식증강형 실시간 동시통역 원천기술 개발]

참고문헌

[1] 이충희, 임준호, 임수종, 김현기. "기분석사전과 기계학습 방법을 결합한 음절 단위 한국어 품사 태깅." 정보과학회논문지 43.3 (2016): 362-369.

[2] 나승훈, 양성일, 김창현, 권오욱, 김영길. "CRF 에 기반한 한국어 형태소 분할 및 품사 태깅." HCLT 2012.

[3] Seung-Hoon Na . Conditional Random Fields for Korean Morpheme Segmentation and POS Tagging. ACM Transactions on Asian and Low-Resource Language Information Processing , 14(3), 2015

[4] 이창기. "Structural SVM 을 이용한 한국어 띄어쓰기 및 품사 태깅 결합 모델." 정보과학회논문지: 소프트웨어 및 응용 40.12 (2013): 826-832.

[5] 김혜민, 윤정민, 안재현, 배경만, 고영중. 품사 분포와 Bidirectional LSTM-CRFs를 이용한 음절 단위 형태소 분석기, HCL 2016

[6] 이건일, 이의현, 이종혁. "Sequence-to-sequence 모델을 이용한 한국어 형태소 분석 및 품사 태깅." 한국정보과학회 학술발표논문집 (2016)

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

[8] Dyer Chris, M. Ballesteros, W. Ling, A. Matthews, N. A. Smith, "Transition-based dependency parsing with stack long short-term memory." arXiv preprint arXiv:1505.08075 (2015).

[9] Zhang, Meishan, Yue Zhang, and Guohong Fu. "Transition-Based Neural Word Segmentation." ACL (1). 2016.