트랜스포머 기반 한국어 형태소 원형복원 및 분리

신형진○, 박정연, 이재성
충북대학교
{shinhj, parkjeongyeon, jasonlee}@cbnu.ac.kr

Korean Morpheme Restoration and Segmentation based on Transformer

Hyeong Jin Shin○, Jeongyeon Park, Jae Sung Lee
Chungbuk National University

요 약

최근 한국어 언어 모델이나 단어 벡터 생성 등에서는 효과적인 토큰을 만들기 위해 품사 태그 없이 형태소 열만을 사용하고 있다. 본 논문에서는 입력 문장에 대해 품사 태그열 생성없이 형태소 열만을 직접 출력하는 효율적인 모델을 제안한다. 특히, 자연어처리에서 적합한 트랜스포머를 활용하기 위해, 입력 음절과 원형 복원된 형태소 조각이 1:1로 대응되는 새로운 형태소 태깅 방법을 제안한다. 세종 품사 부착 말뭉치를 대상으로 평가해 본 결과 공개 배포되어 있는 기존 형태소 분석 모델들보다 형태소 단위 F1 기준으로 약 7%에서 14% 포인트 높은 성능을 보였다.

주제어: 형태소 원형복원, 형태소 분석, 트랜스포머, 언어모델

1. 서론

형태소 분석 및 태깅은 주어진 입력 문장 혹은 어절을 형태소 열 및 대응되는 품사 태그열로 분석하여 출력한다. 특히, 응용에 따라 품사 태그열은 사용치 않고 형태소 열만을 사용하는 경우가 있는데, 최근에는 단어 벡터나 언어모델 개발 시 형태소 경계를 고려한 토큰을 생성하는데 활용되고 있다[1, 2]

본 논문에서는 입력 문장에 대해 품사 태그열 생성없이 형태소 열을 직접 출력하는 효율적인 모델을 제안하고 실험한다. 출력되는 형태소 열은 형태소 원형복원 과정과 형태소 분리 과정을 결합하여 만들어진다. 특히, 자연어처리에서 높은 성능을 보이고 있는 트랜스포머를 활용하기 위해, 입력 음절과 형태소 조각이 1:1로 대응되는 새로운 형태소 태깅 방법을 제안한다. 이 태깅 방법을 적용하면 기존의 형태소 품사부착 말뭉치에서 자동으로 본 연구 모델의 학습용 데이터를 구축할 수 있다.

세종 품사부착 말뭉치를 이용하여 실험하고, 현재 공개되어 있는 형태소 분석기들과 비교해 본 결과, 기존 모델보다, 형태소 단위 F1 기준으로 약 7% 포인트 높은 성능을 보였다.

2. 음절 단위 형태소 태깅

형태소 분석은 일반적으로 형태소 원형복원, 형태소 분리, 형태소 품사 태깅의 3단계가 서로 복합적으로 결합되어 처리된다[3], 원형복원은 표층형의 어절을 심층형(원형)의 어절로 변화하는 것이고, 형태소 분리는 원형의 어절에서 각 형태소들을 구분해내는 것이다. 예를 들어 “불렀다”의 원형복원은 “부르었다”가 되고, 여기에 다시 형태소 분리를 적용하면 “부르+었+다”가 된다(그림 1.a).

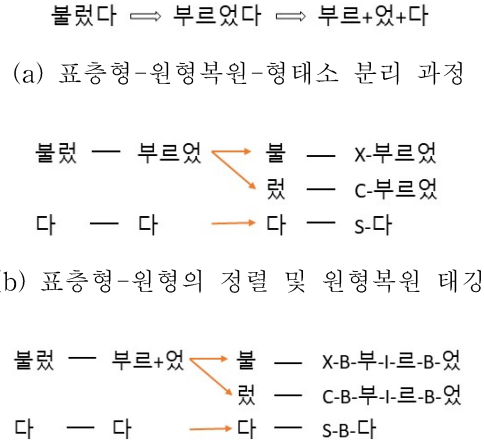

그림 1. 음절 단위 형태소 태깅 예

음절 단위 형태소 복원은 표층형과 원형의 정렬을 이용한다. 정렬은 두 문자열을 대응시켜 최대한 같은 문자가 우선 대응이 되도록 하고, 대응이 되지 않는 부분 문자열을 나중에 대응되도록 정렬한다. 이를 입력 음절 단위로 분리한 후, 표층형과 원형이 2개 이상의 음절로 대응 되는 경우는 X 태그로, 한 음절로 대응되는 경우는 S 태그로 표시한다. 표층형이 여러 음절이어서 분리되는 경우(예: 불렀)는 첫 음절은 X 태그로 표시하고, 나머지 음절들은 C 태그로 표시하되 C 뒤의 내용은 X 태그의 내용과 같게 한다. C태그는 입력 음절에 대응하기 위한 것으로 생성 시에는 C태그는 무시된다(그림 1.b).

음절 단위 형태소 분리는 형태소 복원에서 형태소 경계 처리를 더 추가한 것이다. 즉, 형태소 경계를 BI태그

로 추가한다(B는 형태소 시작, I는 형태소 중간). 예를 들어 "부르+었"은 "B-부-I-르-B-었"으로 표시되며, 이를 X, C, S 태그와 결합하여 표시한 것이 그림1.c와 같다.

표 1은 세종 품사부착 말뭉치[4]를 기준으로 표층형과 원형의 음절 대응수를 나타낸 것으로 1:1부터 2:3까지 다양하다. 실제 조사 결과, 비교적 단순한 대응이 대부분이며, 복잡한 대응(1:4, 2:3 등)은 그 비율이 적어 출력 어휘 수에 크게 영향을 주지 않았다. 아래 표 1에서, 1:1a는 표층형과 원형의 음절이 같은 경우이며, 1:1b는 음절의 변화가 발생하는 경우이다.

표 1. 표층형과 원형의 음절수 대응 예시와 빈도
(굵은 표시 음절 대응)

대응		표층형	원형	빈도(단위: 천)
1:1	a	**새**+가	**새**+가	30,778
	b	**말**+라	**마**+라	87
1:2		**갔**+다	**가았**+다	1,512
1:3		**뭐**+가	**무엇이**+가	13
1:4		**뭔**+일	**무엇이ㄴ**+일	0.2
2:1		**고우**+니	**곱**+니	0.5
2:2		새+**로운**	새+**롭ㄴ**	40
2:3		**바쁜**+지	**바쁘ㄴ**+지	17

3. 형태소 태깅 트랜스포머 모델

2장에서 제안한 음절 단위 형태소 태깅을 이용하여 형태소 복원 및 분리 모델을 만들 수 있다. 본 논문에서는 그림 2와 같은 모델을 사용한다. 즉, 각 음절을 트랜스포머 인코더의 입력으로 받아 인코딩하고, 그 결과를 Bi-LSTM의 입력으로 받아 다시 형태소 태깅을 출력한다. 출력 결과는 후처리를 통해 태그에 따라 형태소 열로 출력한다.

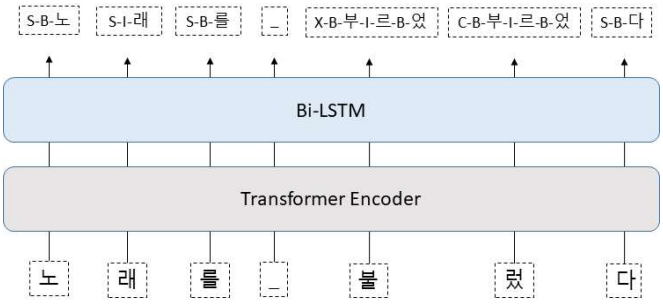

그림 2. 트랜스포머 기반 형태소 원형복원 및 분리 모델

원형복원은 한글 음절에 대해서만 발생하며, 한자나 영어, 일본어 및 특수문자에는 이루어지지 않는다. 따라서, 한글을 제외한 문자는 copy mechanism[5]을 적용하여 잘못된 변형이 일어나지 않도록 했다. 예를 들어, "淸"(청)과 같은 한자는 "S-B-淸"과 같은 형태의 출력으로 복사하였다.

4. 실험 및 평가

데이터 및 모델 파라미터

실험데이터는 세종 품사부착 말뭉치[4]를 사용하였으며, 표층형과 원형이 불일치하거나 입력 오류가 있는 것을 수정하여 사용하였다. 사용한 데이터는 약 83만개의 문장으로 이를 8:1:1로 나누어 학습셋, 개발셋, 평가셋으로 사용하였다. 모델의 파라미터는 아래 표 2와 같다.

표 2. 모델 파라미터

항목	설정값
학습률	0.00001
배치크기	64
드롭아웃	0.3
임베딩크기	768
히든크기	768
레이어 수	transformer: 2 Bi-LSTM: 2

평가방법

평가는 음절 단위, 형태소 단위, 어절 단위로 이루어졌다. 음절 단위 평가의 경우, 띄어쓰기를 제외한 모든 음절을 정확도(accuracy)로 평가했다. 평가 수식은 아래 식 (1)과 같고, 예측된 음절이 위치까지 일치한 경우를 맞은 것으로 하였다.

$$accuracy = \frac{\text{예측이 맞은 음절 (태그 포함)}}{\text{문서에 나타난 모든 음절}} \quad (1)$$

형태소 단위 평가와 어절 단위 평가는 precision, recall을 이용한 F1 score로 평가하였다. 예측된 출력 열 O와 정답 열 G에 대한 수식은 아래 식 (2), (3), (4)와 같다.

$$Precision = \frac{\max(len(G), len(O)) - ED(G,O)}{len(O)} \quad (2)$$

$$Recall = \frac{\max(len(G), len(O)) - ED(G,O)}{len(G)} \quad (3)$$

$$F1\ score = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (4)$$

식 (2), (3)에서 예측 출력 O의 길이는 형태소(또는 어절) 분리에 따라 정답 G의 길이와 달라질 수 있으므로, 이를 반영하여 맞은 형태소(또는 어절) 수를 계산하였다. 즉, 맞은 형태소(또는 어절) 수는 예측 열과 정답 열 중, 길이가 긴 값에서 편집거리(edit distance)를 뺀 것으로 하였다. 어절 단위 평가는 분리된 형태소들을 분리기호를 포함한 상태로 어절 단위로 묶어, 형태소 분리까지 일치한 경우에만 맞은 것으로 했다.

실험결과

형태소 원형복원 및 분리를 각 단위별로 평가한 결과는 아래 표 3과 같다.

표 3. 평가 결과 (%)

평가 단위	pre.	recall	f1	acc.
음절 단위				99.29
형태소 단위	98.70	98.59	98.64	
어절 단위	98.05	98.05	98.05	

음절 단위 정확도 평가에서 99.29의 성능을 기록했다. 출력 오류의 종류에는 'BI' 태그 오류가 가장 많았고, 그 다음으로는 받침 분리 오류 등의 순서였다. 아래 표4는 테스트셋에서 나타난 약 25,000개의 오류 음절 중 200개의 표본을 추출하여 분석한 표이다.

표 4. 오류 분석표

	개수	비율(%)
BI 태그 오류	159	79.5
받침 분리 오류	38	19.0
기타	3	1.5

'BI' 태그 오류는 '무명'이라는 어절에 대해 'S-B-무', 'S-I-명'으로 주석되지 않고, 'S-B-무', 'S-B-명'으로 예측되어 형태소가 분리되는 경우 등을 의미한다. 받침 분리 오류는 '키스신'이라는 어절에서 '신'은 scene을 의미하며 명사로써 독립적인 형태소인데, '시+ㄴ'과 같이 'ㄴ'이 어미나 조사로 분석되어 잘못 분리된 경우 등을 의미한다.

아래 표 5는 기존 연구들과 원형복원의 성능을 비교한 것이다. 비교 결과 다른 연구에 비해 우수한 결과를 보였다. (여기에서 어절 단위 평가는 B, I 태그를 없애고 복원된 형태소들을 이어붙여 평가한 것이므로, 태그 정보까지 비교한 표 3의 음절 단위 평가보다 성능이 높아졌다)

표 5. 형태소 원형복원 성능 비교 (%, F1 score)

원형복원 모델	어절 단위
심광섭 [6]	96.47
윤준영 등 [7]	99.26
본 연구 모델	99.42

기존의 형태소 분석기와 비교하기 위해, 공개 배포되고 있는 대표적인 형태소 분석 모델인 Mecab-ko와 Khaiii의 성능을 비교하였다. 여기서 Mecab-ko 는 일본어를 기본으로 개발된 Mecab을 한국어에 적용한 것으로 형태소의 격자구조와 CRF를 이용하여 구현한 것이다. 그러나, 일본어에는 없고 한국어에만 있는 원형복원 기능이 빠져있어 성능에 차이가 있다. 이를 보완하기 위해 후처리로 원형복원을 처리한 것이 Mecab-ko(후처리)이다 [9]. Khaiii는 딥러닝 모델기반의 형태소 분석기로 CNN 기반으로 만들어졌다. 실험은 공개된 API와 소스를 이용해 직접 수행하였다. 즉, 본 연구에서 사용한 평가 데이터를 수행하여 비교하였다. 비교 결과, 아래 표 6에서 나타난 것과 같이, 형태소 단위 F1 score를 평가한 결과, Mecab-ko와는 무려 14% 포인트 이상 차이가 나고, Mecab-ko (후처리)와 Khaii와는 두개 모두보다 약 7% 포인트 차이로 본 연구 모델이 더 높은 성능을 보여주었다.

표 6. 형태소 분석 도구의 형태소 원형복원 및 분리 성능 비교 (%, F1 score)

형태소 분석 도구	형태소 단위
Mecab-ko [8]	84.20
Mecab-ko (후처리) [9]	91.24
Khaiii [10]	91.17
본 연구 모델	98.64

5. 결론

트랜스포머 인코더 모델은 자연어처리에 매우 효율적이나, 입력과 출력의 수가 같아야 사용할 수 있다. 본 논문에서는 음절 단위 형태소 태깅 방법을 제안하고, 이를 이용하여 트랜스포머 기반의 한국어 형태소 원형복원 및 분리 모델을 제안하였다.

세종 품사부착 말뭉치를 이용하여 실험해 본 결과, 공개 소프트웨어인 Mecab-ko와 Khaiii에 비해 형태소 단위 F1 기준으로 약 7%포인트부터 14%포인트까지 더 우수한 성능을 보였다.

본 연구의 결과는 한국어에서 정확한 형태소 단위 처리가 필요한 응용 프로그램들, 특히 최근에 많은 연구가 진행되고 있는 형태소 경계를 고려한 언어모델 토큰화 연구 등에 활용될 수 있을 것이다.

감사의 글

이 논문은 2021년도 정부(교육부)의 재원으로 한국연구재단 기초연구사업의 지원을 받아 수행된 연구임(No. 2021R1I1A3059545).

참고문헌

[1] K. Park, J. Lee, S. Jang, and D. Jung, "An Empirical Study of Tokenization Strategies for Various Korean NLP Tasks," In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pp. 133-142, 2020.

[2] B. Kim, HS. Kim, SW. Lee, G. Lee, D. Kwak, D. H. Jeon, S. Park, S. Kim, S. Kim, D. Seo, H. Lee, M. Jeong, S. Lee, M. Kim, S. H. Ko, S. Kim, T. Park, J. Kim, S. Kang, NH. Ryu, K. M. Yoo, M. Chang, S. Suh, S. In, J. Park, K. Kim, H. Kim, J. Jeong, Y. G. Yeo, D. Ham, D. Park, M. Y. Lee, J. Kang, I. Kang, JW. Ha, W. Park, N. Sung, "What Changes Can Large-scale Language Models Bring? Intensive Study on HyperCLOVA: Billions-scale Korean Generative Pretrained Transformers," *In Proceedings of the 2021 Conference on Emperical Methods in Natural Language Processing*, pp. 3405-3424, 2021.

[3] 이재성, “한국어 형태소 분석을 위한 3단계 확률 모델,” *정보과학회논문지: 소프트웨어 및 응용, Vol.38, No.5*, pp. 257-268, 2011.
[4] 국립국어원, 21세기 세종계획 최종 성과물 (2011년 12월 수정판), 2011
[5] Thang Luong, Ilya Sutskever, Quoc Le, Oriol Vinyals, Wojciech Zaremba, “Addressing the Rare Word Problem in Neural Machine Translation,” *In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, Vol. 1, pp. 11-19, 2015.
[6] 심광섭, “음절 단위의 한국어 품사 태깅에서 원형 복원,” *정보과학회논문지*, Vol. 40, No. 3, pp. 182-189, 2013.
[7] 윤준영, 이재성, “Sequence-to-Sequence와 BERT-LSTM을 활용한 한국어 형태소 분석 및 품사 태깅 파이프라인 모델,” 제32회 한글 및 한국어 정보처리 학술대회 논문집(2020), pp. 414-417, 2020.
[8] T. Kudo, “Mecab: Yet another part-of-speech and morphological anlyzer,” Last modified February 18, 2013, taku910.github.io/mecab/
[9] 전태희, “한국어 텍스트의 토큰화 방법에 관한 언어학적 연구-fastText 단어 임베딩을 이용하여-,” *언어사실과 관점*, Vol. 55, pp. 309-354, 2022.
[10] KaKao Khaiii(Kakao Hangul Analyzer III), https://github.com/kakao/khaiii