Transformer를 이용한 한국어 Head-Tail 품사 태거

김정민○, 서현재, 강승식
국민대학교, 컴퓨터공학과
kimjm@kookmin.ac.kr, hjworld32@gmail.com, sskang@kookmin.ac.kr

Korean Head-Tail POS-Tagger by using Transformer

Jung-Min Kim○, Hyun-Jae Suh, Seung-Shik Kang
Kookmin University, Dept. of Computer Science

요 약

한국어의 품사 태깅 문제는 입력 어절의 형태소 분석 후보들로부터 통계적으로 적절한 품사 태그를 가지는 후보들을 찾는 방식으로 해결하여 왔다. 어절을 형태소 단위로 분리하고 품사를 부착하는 기존의 방식은 품사태그 정보를 딥러닝 feature로 사용할 때 문장의 의미를 이해하는데 복잡도를 증가시키는 요인이 된다. 본 연구에서는 품사 태깅 문제를 단순화 하여 한 어절을 Head와 Tail이라는 두 가지 유형의 형태소 토큰으로 분리하여 Head와 Tail에 대해 품사를 부착한다. Head-Tail 품사 태깅 방법을 Sequence-to-Sequence 문제로 정의하여 Transformer를 이용한 Head-Tail 품사 태거를 설계하고 구현하였다. 학습데이터로는 KCC150 말뭉치의 품사 태깅 말뭉치 중에서 788만 문장을 사용하고, 실험 데이터로는 10만 문장을 사용하였다. 실험 결과로 토큰 정확도는 99.75%, 태그 정확도는 99.39%, 토큰-태그 정확도는 99.31%로 나타났다.

주제어: POS-Tagging, Head-Tail, Head-Tail 품사 태깅

1. 서론

딥러닝 기반 자연어처리 시스템에서는 Feature로 형태소 분석과 품사 태깅이 매우 중요한 요소가 된다. 이러한 형태소분석은 형태소 단위로 분리하고 불규칙 형태소의 원형복원과 품사 태깅 과정이 존재한다[1,2]. 한국어의 품사 태깅 문제는 문장 내에서 형태소 토큰 단위로 원형을 복원하여 그에 적절한 형태소 후보와 품사들을 결정하는 문제이다[3,4].

문서 임베딩, 기계 이해, 감성분석, 문서 분류 등 딥러닝 언어처리에서는 형태소 단위를 무시하고 단순히 subword 단위의 토큰열로 문서 벡터를 구성하는 방식을 취하고 있다. 형태소 분석 기반의 품사 태깅 방식은 '아름다운'을 '아름답' + 'ㄴ'과 같이 자소 단위로 원형 복원 방식으로 처리하고 있다. 이러한 방식은 의미적으로 중요하지 않은 부분까지 원형을 복원하므로 딥러닝 언어처리 문제를 해결하는데 불필요한 정보(조사, 어미 등)가 세부적으로 포함되어 오히려 문제를 어렵게 만들 수 있다.

예를 들어, 문장 "오리콤은 이에 모든 조직과 프로세스를 바꾸기로 했다." 라는 문장이 있을 때 형태소 분석결과는 "오리콤/NNP+은/JX 이/NP+에/JKB 모든/MM 조직/NNG+과/JC 프로세스/NNG+를/JKO 바꾸/VV+기/ETN+로/JKB 하/VV+었/EP+다/EF ./SF"라는 결과를 얻게 된다. 여기서 "바꾸기로" 와 "했다" 의 경우에 "바꾸기로" 라는 어절은 "바꾸 기 로"에서 "바꾸" 라는 동사와 "기 로" 라는 어미와 조사로 세개의 형태소 단위로 토큰화 되었고, "했다" 라는 어절의 경우 "하 었 다" 에서 "하" 이라는 동사와 "었 다" 라는 세개의 형태소 단위로 토큰화된 것을 볼 수 있다. 여기서 "바꾸"와 "하" 라는 형태소는 각 어절에서 문법적으로 중요한 의미를 가지는 개별적인 뜻을 가진 명사, 형용사, 용언 같은 어휘형태소로 볼 수 있다. 반면에 "바꾸"와 "하" 뒤에 오는 형태소는 조사나 어미 등 어휘형태소를 보조하거나, 격을 부여하거나, 동사나 형용사를 명사로 바꾸는 등 문법적 의미가 있는 문법형태소로 딥러닝 학습에 Feature 정보로 전달하기에는 상대적으로 중요한 의미를 가지지 않는다. 문법형태소와 어휘형태소의 원형까지 세부적인 단위로 토큰화하게 되면 딥러닝 모델이 문장내의 정보를 복잡하게 만들어 자연어처리 문제를 해결하고자 할 때 문장내의 의미를 파악하는데 오히려 방해가 될 수 있다.

본 논문에서는 문장 내에서의 형태소 단위를 원형을 복원하지 않고 어휘형태소를 음절 단위의 Head 토큰, 문법형태소 부분을 음절 단위의 Tail 토큰으로 단순화하는 방식으로 Head-Tail에 대해 품사 태깅을 수행한다. Head-Tail 품사 태깅 문제를 Sequence-to-Sequence 문제로 보고 Sequence-to-Sequence에서 보편적으로 사용되는 Transformer를 이용한 Head-Tail 품사 태거를 제안한다.[5,6,7]

2. 관련 연구

2.1. Transformer

Transformer는 인코더(Encoder)와 디코더(Decoder)로 구성되며, 기존의 LSTM을 이용한 Seq2Seq(with attention)에서 LSTM을 사용하지 않고 Attention만을 이용하여 Sequence-to-Sequence 문제를 해결한 모델이다. 그림 1은 Transformer의 구성도로, Transformer는 Scaled-dot-product-attention과 Multi-head Attention으로 구성되며, 수식은 아래 식(1)과 같다.

$$Attention(Q,K,V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \quad (1)$$

Q,K,V라는 벡터가 주어졌을 때 각 Q에 대해 모든 K를

내적 하여 문장 내의 Q라는 단어와 모든 K라는 단어의 연관성을 계산한 후 하이퍼파라미터 $\sqrt{d_k}$로 나누어 준 뒤 softmax를 취한 후 V를 곱한다. 이때 일반 dot-product-attention을 $\sqrt{d_k}$로 Scaling 하므로 Scaled-dot-product-attention 이라고 한다.

Multi-head-attention은 Scaled-dot-product-attention을 각 레이어마다 계산한 후 Concat 하여 사용한다. 수식으로 표현하면 아래 식(2)와 같다.

$$MltiHead(Q,K,V) = Concat(head_1,...,head_h)W^O \\ where\ head_i = Attention(QW_i^Q, KW_i^K, VW_i^V) \quad (2)$$

또한 입력으로 받은 워드 임베딩 벡터에 대해 Positional Encoding이라는 작업을 하게 되는데 이는 Transformer가 LSTM처럼 순차적인 데이터를 처리하는 것이 아니라서 Transformer에게 각 단어의 위치 정보를 제공하기 위함이다.

본 논문에서는 Sequence-to-Sequence 문제에서 높은 성능을 보여주는 Transformer를 사용하여 Head-Tail 품사 태깅 문제를 해결한다[8].

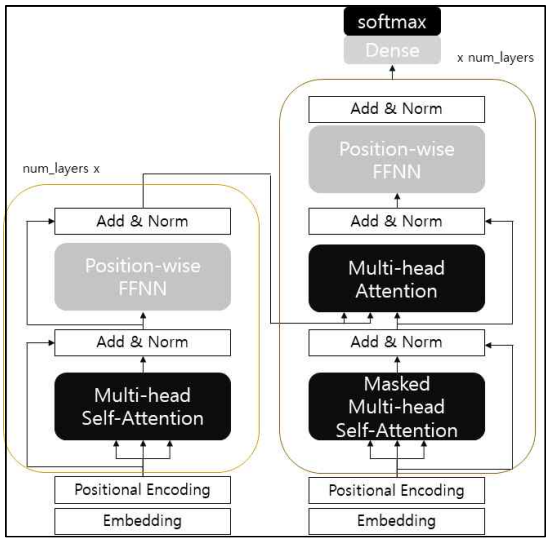

그림 1 Transformer 모델 구성도

3. 실험

3.1. Tokenizer

입력데이터의 OOV 문제를 해결하기 위해 음절단위 토큰 중 자주 등장하는 음절 단위 토큰 집합은 하나의 토큰으로 간주하고 이에 해당하는 토큰단위로 분리하는 Sentence-piece subword tokenizer에서 모든 서브워드 토큰에서 Loss가 가장 적게 증가 시키는 토큰들을 제거해나가는 방식의 unigram 모드를 사용하였다[9].

3.2. Head-Tail의 정의

한국어는 개별적인 의미를 가지는 어휘형태소와 문법적인 기능을 보조하는 문법형태소로 구성된다. Head-Tail 토큰화 기법은 하나의 어절을 Head와 Tail이라는 두 개의 토큰으로만 분리한다.

어절에서 Head 토큰은 어휘형태소 부분이고 Tail은 문법형태소 부분으로 구성된다. Head와 Tail은 불규칙 용언이나 축약, 탈락에 의해 음절 경계에서 형태소가 변형되는 경우에 형태소의 원형을 복원하지 않고 음절단위로 분할한다. 예를 들어, '갔다'의 경우에 '갔'을 Head로 하고, '다'를 Tail로 분할하여 Tail은 어절에서 Head 토큰을 제외한 문법형태소 부분이다. 또한, Head-Tail 토큰에서 복합어(복합명사, 복합 문법형태소 등)는 분해하지 않는다.

3.3. 품사태깅 데이터셋

학습 데이터셋은 KCC150[1] 원시말뭉치에 대한 품사 태깅 결과물인 KCC150 형태분석 말뭉치[2]를 사용한다. 말뭉치 자체는 형태소의 원형을 복원한 형태소 말뭉치이다. 따라서 말뭉치에서 원문 어절이 있을 때 형태소 분석된 데이터에서 형태소의 제일 처음 등장하는 음절길이만큼 원문의 앞부분을 잘라 Head 토큰으로 하고, 원문에서 Head를 제외한 나머지 부분을 Tail로 한다. Tail 부분의 Tag는 "_"로 구분하여 병합하여 하나의 태그로 간주한다.

태그는 "/" 문자로 구분하며 Head와 Tail 사이에 '+' 문자를 넣어 Head와 Tail을 구분하여 주었다. 예를 들어 "통합보건교육은 이 대학만의 특화된 프로그램이다."라는 문장의 형태소 분석과 Head-Tail 토큰 분리 예는 표1과 같다.

표 1. Head-Tail 토큰 분리

	어절	형태소 분석결과	Head-Tail 토큰 분리
1	통합보건교육은	통합보건교육/NNG + 은/JX	통합보건교육/NNG + 은/JX
2	이	이/MM	이/MM
3	대학만의	대학/NNG + 만/JX + 의/JKG	대학/NNG + 만의/JX_JKG
4	특화된	특화되/VV + ㄴ/ETM	특화된/VV_ETM
5	프로그램이다	프로그램/NNG + 이/VCP + 다/EF	프로그램/NNG + 이다/VCP_EF
6	.	./SF	./SF

표1에서 "대학만의", "프로그램이다"의 경우에 형태소 분석결과는 (대학, 만, 의),(프로그램, 이, 다)와 같이 문법형태소 부분을 각 형태소 단위로 분리하였는데, "만, 의, 이, 다" 라는 토큰은 "대학, 프로그램" 어휘형태소에 대한 문법적인 기능어로서 상대적으로 덜 중요한 토큰으로 볼 수 있다. "특화된"의 경우 "특화되,ㄴ" 로 자소 단위까지 토큰화가 되었는데 이러한 자소 단위 토큰 'ㄴ'은 feature로 사용시 불필요한 정보가 될 수 있다[10]. 따라서 토큰 분리 문제를 Head와 Tail이라는

1) http://nlp.kookmin.ac.kr/kcc
2) https://github.com/bufsnlp2030/BUFS-JBNUCorpus2020

두 가지 단위로 단순화시켰고 기능어 부분을 Tail 토큰 하나로 통합하였다. 이와 같은 방법으로 " 대 학/NNG(Head)+만의/JK_JKG(Tail), 특화된/VV_ETM(Head), 프로그램/NNG(Head)+이다/VCP_EF(Tail)" 라는 음절 단위 토큰으로 분리하고 태깅하였다. "전해진다, 했다, 전달된, 배포할" 등에 대해서도 동일한 방법으로 Head-Tail로 분리하였다.

표 2. 불규칙 변형 어절의 Head-Tail 토큰 분리

어절	형태소 분석결과	Head-Tail 토큰화
전해진다	전하/VV+어/EC+지/VX+ㄴ다/EF	전 해 / V V + 진 다/EC_VX_EF
했다	하/VV+었/EP+다/EF	했/VV+다/EP_EF
전달된	전달되+ㄴ	전달된/VV_ETM
배포할	배 포 하 / V V + ㄹ/ETM	배포할/VV_ETM

Head-Tail로 토큰화시 어휘의 개수가 비약적으로 증가하는 문제가 발생할 것으로 보이지만, 원형 복원과정에서 음절을 자소 단위로 분해하는 형태론적 변형은 불규칙 어간의 변형, 탈락, 축약 등에 의해 발생한다. 형태론적 변형이 발생하는 용언은 10,114개로 각 용언마다 원형을 복원하지 않음으로 인하여 추가되는 어휘 개수는 1만여 개이다[11]. 이 개수는 영어의 품사태깅에서 명사의 복수형과 동사의 과거/과거분사/현재진행형 등을 별개의 토큰을 간주함으로 인하여 추가되는 어휘 개수에 비해 적은 편이다.

3.4. 데이터셋 구성

전체 8,875,652 문장 중에서 하이퍼파라미터에서 지정한 토큰사이즈를 넘어서는 문장들을 제외하고 7,882,000 문장의 Head-Tail 문장을 학습데이터로 사용하였고 테스트데이터 또한 지정한 토큰사이즈를 넘어서는 문장들을 제외하고 100,000 문장을 사용하였다.

4. 품사 태깅 모델

4.1. 모델의 구성

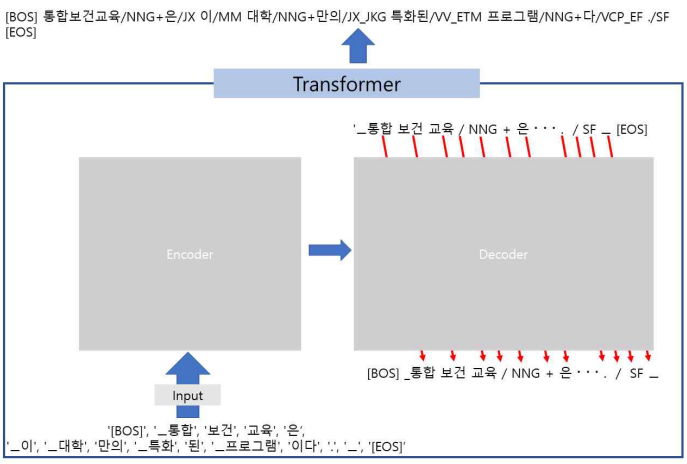

그림 2 모델 학습 구성도

입력은 일반 문장을 Subword Tokenizer로 분리한 토큰을 입력으로 한다. 출력 또한 Head-Tail로 태깅된 문장을 Subword Tokenizer로 분리한 토큰을 출력으로 하여 Transformer에 학습을 한다.

입력으로 일반문장을 Transformer의 인코더로 입력으로 하고 인코더로 추출한 문맥정보를 디코더로 전달한다. 이러한 문맥정보와 함께 디코더의 입력으로 문장의 시작을 의미하는 "[BOS]" 토큰을 전달하고 예측 값을 다시 디코더의 입력으로 하여 예측 결과가 태깅 결과의 끝을 알리는 "[EOS]" 토큰이 나타날 때까지 Autoregressive Model로 학습한다.

4.2. 하이퍼파라미터

학습 시 Batch는 500으로 하였으며, Epoch는 6, Subword Token은 토큰 개수가 200개를 넘지 않는 문장을 사용하여 실험하였다. Loss function은 Categorical Crossentropy, Optimizer는 Adam Optimizer를 사용하였다.

표 3. 하이퍼파라미터

Batch Size	500
Epoch	6
Input,Output Token Size	200
Loss function	Categorical Crossentropy
Optimizer	Adam Optimizer

4.3. 실험 및 평가

표 4. 실험 결과

구분	Accuracy
토큰-Tag 정확도	99.31%
토큰 정확도	99.75%
Tag 정확도	99.39%

트랜스포머를 이용하여 Head-Tail을 학습시키고 100,000 문장의 Test Dataset으로 검증을 수행하였다. 형태소 분리 문제를 원형복원이 아닌 단순한 음절분리 문제로 단순화 시켜 태깅을 수행한 결과로 Head-Tail 토큰 분리 정확도는 99.75%, Tag 부착 정확도는 99.39%로 나타났으며, 토큰 분리 및 Tag 부착 정확도는 99.31%, 로 평가되었다.

5. 결론 및 향후 연구

한국어의 어절을 어휘형태소 부분인 Head와 문법형태소 부분인 Tail로만 분리하는 Head-Tail 형태소 분리 방식에 따라 Head-Tail 품사 태깅 문제를 Sequence-to-Sequence 문제로 정의하고, 대표적인 Sequence-to-Sequence 모델인 Transformer에 적용하여 Head-Tail 품사 태거를 구현하였다. 토큰의 분리 단위를 음절 단위로 단순화하였기 때문에 Transformer가 토큰분리를 쉽게 예

측할 수 있었고 높은 정확도의 품사 태깅 성능을 보여주었다. 향후 Head-Tail을 딥러닝 모델의 Feature로 적용하여 일반 품사 태깅과 비교실험을 통해 딥러닝 모델에 어떠한 영향을 미치는지 그 성능을 평가하고자 한다.

또한 딥러닝 모델의 입력으로 Head-Tail을 사용할시, 토큰화 과정 중 복합명사, 복합어 미분해로 인한 토큰 개수가 증가하고 모델의 파라미터 수가 급격히 증가하는 문제가 발생한다. 이러한 토큰 개수 증가 문제는 Head-Tail 토큰화 이후에 복합명사 분해 과정을 추가하여 해결할 예정이다.

Acknowledgement

이 논문은 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(NRF-2021R1F1A1061433).

참고문헌

[1] 강승식, "음절 특성을 이용한 한국어 불규칙 활용 어절의 형태소 분석 방법", 제5회 한글 및 한국어 정보처리 학술발표 논문집, pp.385-394, 1993

[2] 강승식, "다층 형태론과 한국어 형태소 분석 모델", 제6회 한글 및 한국어정보처리 학술대회, pp.140-145, 1994

[3] 나승훈, "딥 러닝에 기반한 한국어 품사 태깅", 2014 한국정보과학회 제41회 정기총회 및 동계학술발표회, pp.426-428, 2014

[4] 민진우, "BERT를 이용한 전이 기반 한국어 형태소 분석 및 품사 태깅", 한국정보과학회 2019 한국소프트웨어종합학술대회 논문집, pp.401-403, 2019

[5] 이건일, "Sequence-to-sequence 모델을 이용한 한국어 형태소 분석 및 품사 태깅", 한국정보과학회 2016년 한국컴퓨터종합학술대회 논문집, pp.693-695, 2016

[6] 이건일, "Sequence-to-sequence 기반 한국어 형태소 분석 및 품사 태깅", 정보과학회논문지, 제44권, 제1호, pp.57-62, 2017

[7] 윤준영, "한국어 형태소 분석 및 품사 태깅을 위한 딥 러닝 기반 2단계 파이프라인 모델", 정보과학회 논문지, 제48권, 제4호, pp.444-452, 2021

[8] A. Vaswani, N.Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.-N. Gomez, L. Kaiser, I. Polosukhin, "Attention is all you need," Advances in Neural Information Processing Systems, pp. 5998-6008, 2017

[9] 송현제, "서브워드 토큰화와 한국어 형태소 분석기", 정보과학회지, 제39권, 제4호, pp.15-20, 2021

[10] 이현영, "대용량 말뭉치를 이용한 한국어 Head-Tail 토큰화",제12회 융합·스마트미디어 시스템 워크샵, pp.25-28, 2021

[11] 강승식, "한국어 형태소 분석과 정보검색", 홍릉과학출판사, 2002