**KIPS Trans. Softw. and Data Eng.**
Vol.11, No.4 pp.169~178
ISSN: 2287-5905 (Print), ISSN: 2734-0503 (Online)


https://doi.org/10.3745/KTSDE.2022.11.4.169

# Korean Morphological Analysis Method Based on BERT-Fused Transformer Model

Changjae Lee† · Dongyul Ra††

## ABSTRACT

Morphemes are most primitive units in a language that lose their original meaning when segmented into smaller parts. In Korean, a sentence is a sequence of eojeols (words) separated by spaces. Each eojeol comprises one or more morphemes. Korean morphological analysis (KMA) is to divide eojeols in a given Korean sentence into morpheme units. It also includes assigning appropriate part-of-speech(POS) tags to the resulting morphemes. KMA is one of the most important tasks in Korean natural language processing (NLP). Improving the performance of KMA is closely related to increasing performance of Korean NLP tasks. Recent research on KMA has begun to adopt the approach of machine translation (MT) models. MT is to convert a sequence (sentence) of units of one domain into a sequence (sentence) of units of another domain. Neural machine translation (NMT) stands for the approaches of MT that exploit neural network models. From a perspective of MT, KMA is to transform an input sequence of units belonging to the eojeol domain into a sequence of units in the morpheme domain. In this paper, we propose a deep learning model for KMA. The backbone of our model is based on the BERT-fused model which was shown to achieve high performance on NMT. The BERT-fused model utilizes Transformer, a representative model employed by NMT, and BERT which is a language representation model that has enabled a significant advance in NLP. The experimental results show that our model achieves 98.24 F1-Score.

Keywords : Natural Language Processing, Morphological Analysis, Transfer Learning, Transformer, BERT-fused Model

# BERT-Fused Transformer 모델에 기반한 한국어 형태소 분석 기법

이 창 재† · 나 동 열††

## 요 약

형태소는 더 이상 분리하면 본래의 의미를 잃어버리는 말의 최소 단위이다. 한국어에서 문장은 공백으로 구분되는 어절(단어)의 조합이다. 형태소 분석은 어절 단위의 문장을 입력 받아서 문맥 정보를 활용하여 형태소 단위로 나누고 각 형태소에 적절한 품사 기호를 부착한 결과를 생성하는 것이다. 한국어 자연어 처리에서 형태소 분석은 가장 핵심적인 태스크다. 형태소 분석의 성능 향상은 한국어 자연어 처리 태스크의 성능 향상에 직결된다. 최근 형태소 분석은 주로 기계 번역 관점에서 연구가 진행되고 있다. 기계 번역은 신경망 모델 등으로 어느 한 도메인의 시퀀스(문장)를 다른 도메인의 시퀀스(문장)로 바꾸는 것이다. 형태소 분석을 기계 번역 관점에서 보면 어절 도메인에 속하는 입력 시퀀스를 형태소 도메인 시퀀스로 변환하는 것이다. 본 논문은 한국어 형태소 분석을 위한 딥러닝 모델을 제안한다. 본 연구에서 사용하는 모델은 기계 번역에서 높은 성능을 기록한 BERT-fused 모델을 기반으로 한다. BERT-fused 모델은 기계 번역에서 대표적인 Transformer 모델과 자연어 처리 분야에 획기적인 성능 향상을 이룬 언어모델인 BERT를 활용한다. 실험 결과 형태소 단위 F1-Score 98.24의 성능을 얻을 수 있었다.

키워드 : 자연어처리, 형태소분석, 전이학습, Transformer, BERT-fused 모델

## 1. 서 론

한국어에서 문장은 공백으로 구분되는 어절 또는 단어들의 조합이다. 각 어절은 공백 없이 형태소로 결합되어 있다[1]. 형태소란 더 이상 분리하면 의미를 잃어버리는 말의 최소 단위이다. 형태소 분석은 어절 단위의 문장을 입력 받아서 문맥 정보를 활용하여 형태소 단위로 분리하고 각 형태소에 적절한 품사 기호를 부착한 결과를 생성하는 것이다. 형태소 분석은 두 가지 단계로 분리된다. 첫 번째, 어절에 대하여 형태소 후보들로 구성된 열을 생성한다. 두 번째, 형태소 분석에서 나온 후보들 중 올바른 후보를 선택한다. 형태소 분석기는 형태소와 품사 정보가 있는 사전 등을 참고하여 각각의 형태소에 품사를 태깅한다. 그러므로 형태소 열을 선택하는 작업은

† 비 회 원 : 연세대학교 소프트웨어학부 연구원
†† 종신회원 : 연세대학교 소프트웨어학부 교수
Manuscript Received : August 4, 2021
First Revision : September 24, 2021
Accepted : October 17, 2021
* Corresponding Author : Dongyul Ra(dyra@yonsei.ac.kr)

자동적으로 형태소의 품사도 결정한다[1]. 한국어는 형태소적 중의성을 가지며 어순이 자유로워서 같은 어절이라도 문장 내에서 쓰임에 따라 다른 뜻을 지닐 수 있기 때문에 형태소 분석 결과가 다를 수 있다. 따라서 문맥에 맞는 형태소와 품사를 할당하여야 하므로, 형태소 분석 및 품사 태깅은 복잡한 문제이다.

본 논문은 형태소 분석 기법에 대한 것이다. 형태소 분석은 한국어 자연어 처리에서 필수적으로 이루어지고, 형태소 분석의 성능 향상은 한국어 자연어 처리 전체의 성능 향상으로 연결된다. 이러한 중요성 때문에 형태소 분석 성능을 높이려는 연구가 많이 진행되고 있다. 기존 연구들은 형태소 후보들을 생성하고, 올바른 후보를 선택하는 방법으로 접근하였다. 하지만 딥러닝의 부상과 그에 따른 기계 번역의 높은 성능 향상에 주목하여 형태소 분석을 기계 번역 기법으로 시도하는 연구들이 나타나고 있다. 기계 번역은 신경망 모델 등으로 어느 한 도메인의 시퀀스를 다른 도메인의 시퀀스로 바꾸는 것이다. 한국어 문장을 영어 문장으로 번역하는 것이 그 예이다. 형태소 분석 문제에 기계 번역 기법을 도입하는 배경은 형태소 분석을 어절 도메인에 속하는 시퀀스를 대하여 형태소 도메인의 시퀀스로 변환하는 문제로 바라보는 것이다.

우리는 높은 성능의 형태소 분석 모델을 구현하기 위하여 형태소 분석에 기계 번역 관점을 적용하여 기계 번역 태스크에서 우수한 성능을 기록한 BERT-fused 모델[8]을 이용한다. BERT-fused 모델은 기계 번역에서 대표적인 Transformer 모델[6]과 자연어 처리의 전반적인 태스크에서 높은 성능 향상을 이룬 언어표현 딥러닝 모델 BERT[7]을 모두 활용한 모델이다. 또한 본 논문에서는 문장 길이의 제한 없이 분석 대상 어절이 속한 문장의 문맥을 최대로 활용하여 어절별로 형태소 분석을 수행하는 방법을 제안한다.

제안하는 모델의 성능 평가를 위하여 세종 말뭉치[1]로부터 데이터를 구축하여, 모델을 훈련, 검증 및 평가하고 다른 연구들의 형태소 분석 모델과 성능을 비교한다. 실험 결과 본 연구에서 제안하는 모델은 형태소 단위 F1-Score 98.24의 높은 성능을 보여주었다.

## 2. 관련 연구

### 2.1 기계 번역 모델

Recurrent Neural Network (RNN)의 등장으로 기계 번역 태스크에서 RNN Encoder-Decoder 구조를 가진 신경망 모델 Seq2Seq(Sequence-to-Sequence)[2-4]가 주로 사용되었다. Encoder는 입력 받은 시퀀스를 고정된 길이의 벡터로 압축하여 Decoder로 전달한다. 압축 표현된 벡터를 받은 Decoder는 매 시간마다 하나의 원소 혹은 토큰을 출력한다. 이 구조는 Encoder가 입력 시퀀스를 정해진 길이의 벡터로 압축하기 때문에 Decoder가 사용할 수 있는 입력 시퀀스 전체에 대한 표현력이 줄어드는 문제점이 있다. 이러한 문제점을 해결하기 위하여, Decoder에 Attention 기법을 도입한 Seq2Seq[5] 모델이 나왔다. Attention 기반의 Seq2Seq 모델은 Encoder가 입력 시퀀스를 처리할 때 토큰마다 나온 벡터를 모두 연결(concatenate)하여 Decoder에 넘겨준다. 이 벡터를 넘겨받은 Decoder는 매 시간마다 입력 토큰 벡터별로 Attention 가중치를 계산하여, 출력 토큰을 정할 때 사용한다. 이러한 방법에도 불구하고, 연산 병렬성이 떨어지고, 입출력 시퀀스 간 장거리 종속성을 학습하기 어려운 점을 가진 RNN의 단점으로 인해 Seq2Seq 모델은 한계에 직면하였다. 이러한 한계를 극복하고자, Encoder와 Decoder의 RNN을 Attention으로 대체한 Transformer 모델[6]이 등장하였다. Transformer 모델은 Encoder와 Decoder가 층별로 구성되어 있고, 각 층마다 Attention 연산과 Feed-Forward 연산이 수행된다.

### 2.2 BERT

BERT(Bidirectional Encoder Representations from Transformers)는 Transformer의 Encoder 부분과 같은 구조를 가진 층들이 수직적으로 쌓인 언어 표현 모델이다[7]. Self-Attention을 통하여 양방향 문맥을 학습한다. BERT는 전이 학습에 이용되는 모델로 먼저 레이블(label)이 없는 데이터를 이용하여 Pre-training을 수행한다. 그 다음, Pre-training 결과 가중치로 초기화한 BERT가 포함된 모델을 특정 자연어 처리 태스크를 위한 정답 레이블 데이터를 이용하여 Fine-tuning한다. BERT를 Pre-training 하는 데 사용되는 방법은 Masked Language Modeling, Next Sentence Prediction 두 가지가 있다.

### 2.3 한국어 형태소 분석

한국어 형태소 분석은 한국어 자연어 처리에서 차지하는 중요성 때문에 많은 연구가 이루어져 왔다[1,14-20].

딥러닝 기술의 부상 이전까지는 기존 기계 학습 방법 등을 사용하며, 형태소 분리, 품사 태깅, 그리고 원형 복원 등의 단계를 두었다. 이러한 방식은 이전 단계의 오류가 다음 단계를 거치면서 누적되거나 모델 전체의 성능이 단계 별 최소 성능에 영향을 크게 받는 문제가 있었다. 이 문제를 해결하고자 End-to-End 방식의 딥러닝 기반 형태소 분석 모델이 등장하였다.

딥러닝의 부상 이후 형태소 분석에 딥러닝 기반의 모델을 주로 사용하게 되었다. 그 중에서도 Encoder-Decoder 구조로 이루어진 Sequence-to-Sequence 모델[14,15,17,18,20] 및 Transformer 모델[19] 그리고 언어표현 딥러닝 모델 BERT[19,20] 등을 활용한 형태소 분석 연구가 활발하게 진행되고 있다.

1) https://ithub.korean.go.kr/

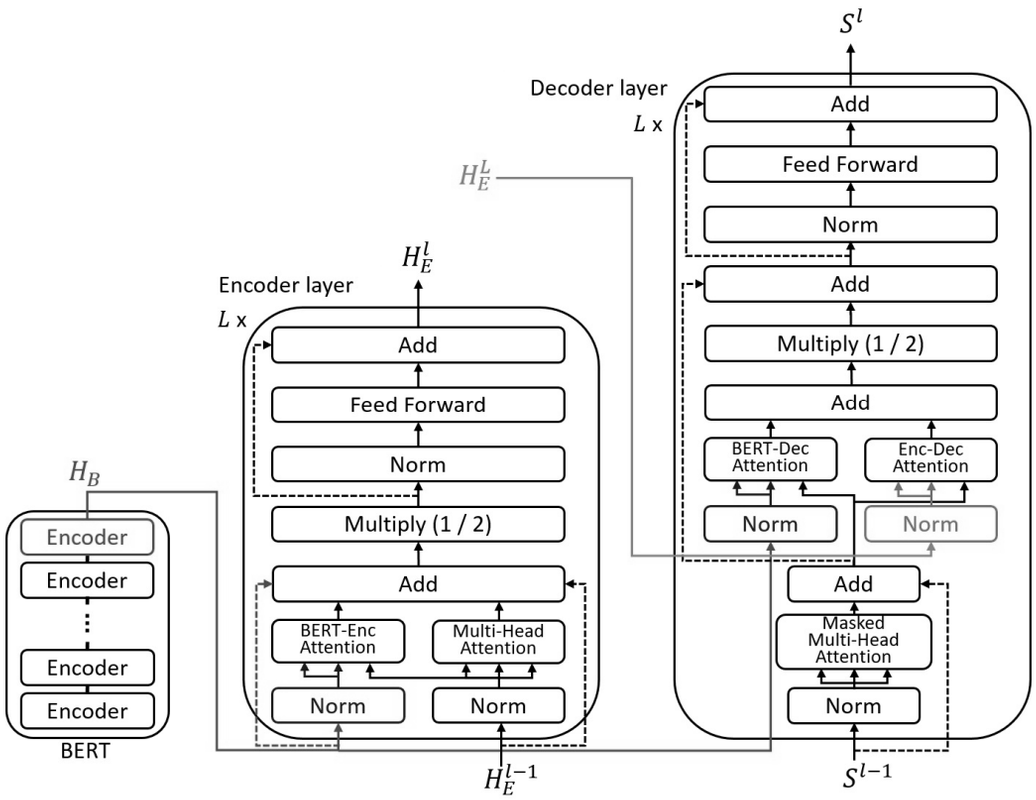



Fig. 1. Architecture of Korean Morphological Analysis Model

## 3. 한국어 형태소 분석 딥러닝 모델

본 논문에서는 BERT-fused 모델 기반의 한국어 형태소 분석 딥러닝 모델을 제안한다. Pre-trained BERT로는 ETRI(한국전자통신연구원)에서 제공하는 KorBERT 어절 기반 모델을 사용한다[2]. Wordpiece 사전은 KorBERT에서 제공되는 사전으로 하였다. Encoder 입력에 이용되는 토큰화 사전은 KorBERT 어절 기반 모델의 사전으로, 총 토큰 개수는 30,797개이다. Decoder 입력에 활용되는 토큰화 사전은 KorBERT 형태소 기반 모델의 사전으로, 이 사전에 있는 토큰은 총 30,349개이다. 기존 모델들은 훈련이 끝난 이후 문장 길이가 최대 길이를 넘어갈 경우, 형태소 분석이 불가하거나 성능이 매우 떨어졌다. 그러나 본 모델은 문장의 길이에 관계없이 형태소 분석이 가능하다는 장점이 있다.

### 3.1 BERT-fused 모델

BERT-fused 모델은 Transformer 모델과 BERT를 모두 활용한 모델이다[8]. IWSLT'14 De->En 기계 번역 태스크에서 State-Of-The-Art를 기록하였다. 기계 번역 모델은 각 쌍을 이루는 대규모의 데이터로 새롭게 Pre-training을 하기 때문에 자원 소모가 크다. 기계 번역에도 BERT 등의 Pre-trained 모델을 활용하여 자원 소모를 줄이면서 성능을 향상시키고자 하는 방법들이 연구되었다.

BERT-fused 모델은 Pre-trained BERT의 출력을 기계 번역 모델(Transformer 모델)의 입력으로 사용한다. 입력 문장의 문맥 표현인 BERT의 출력을 Transformer의 Encoder와 Decoder의 각 층의 cross-attention 작업의 입력으로 활용된다. 이것은 기존 Transformer 모델의 Encoder, Decoder에서 수행하는 attention 작업에 더하여 BERT-fused 모델이 추가적으로 수행하는 attention 작업이다(Fig. 1의 BERT-Enc, BERT-Dec Attention).

### 3.2 한국어 형태소 분석 모델 구조

3.1에서 소개한 BERT-fused 모델을 기반으로 한 한국어 형태소 분석 모델을 구축한다. Encoder와 Decoder는 각 $L=6$ 층으로 구성하였다. BERT, Encoder, Decoder 의 각 층의 입출력 벡터의 차원인 $d_m$은 pre-trained KorBERT 출력 차원수와 동일한 768로 하였다. 각 층의 Intermediate 벡터의 차원은 $d_{ff} = d_m \times 4 = 3,072$이다. Multi-head Attention의 head 수 $h=12$이고, key와 value의 차원 수인 $d_k$, $d_v$는 각자 64로 하였다. Attention 모듈의 연산 결과를 정규화는 PostNorm 방식이 아닌 Attention 모듈의 입력을 먼저 정규화 하는 PreNorm을 사용하였다[9,10]. 우리의 한국어 형태소 분석 모델 구조는 Fig. 1과 같다.

2) http://aiopen.etri.re.kr/

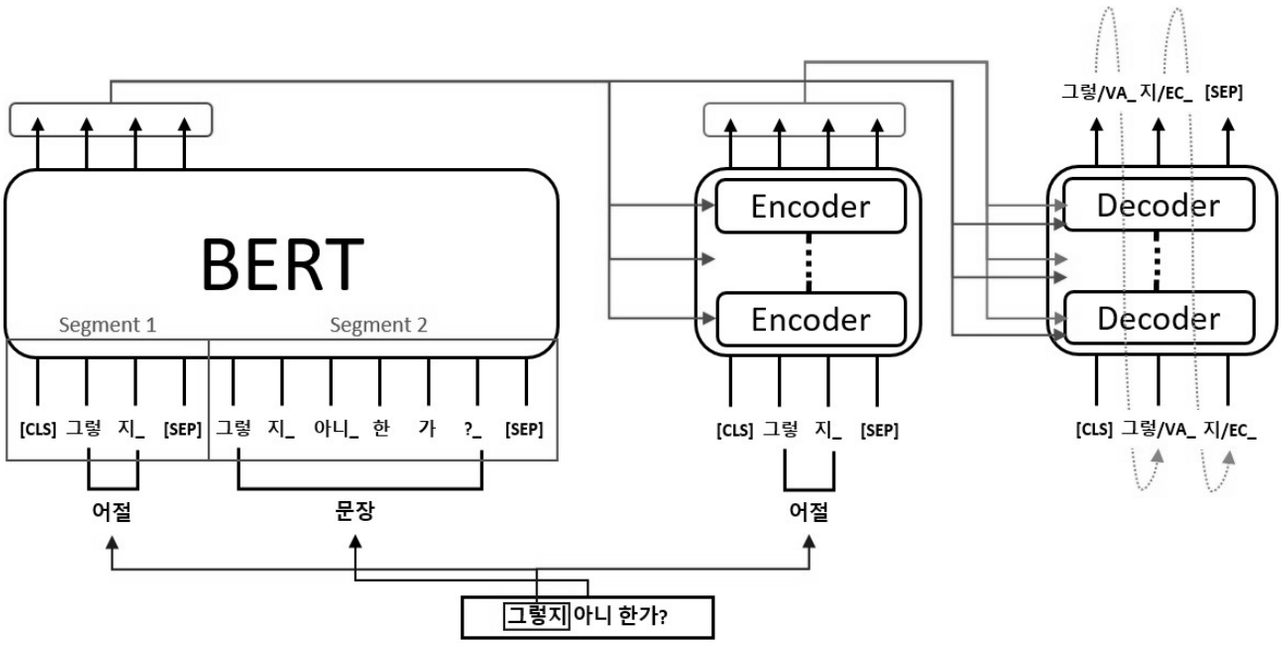



Fig. 2. Example of Korean Morphological Analysis (Prediction)

### 3.3 한국어 형태소 분석 알고리즘

형태소 분석은 한 문장의 어절 리스트를 형태소 리스트로 변환하는 것이 목표이다. $\boldsymbol{X}$를 어절 도메인, $\boldsymbol{Y}$를 형태소 도메인이라 하자. 한국어 형태소 분석 모델에서 어절과 형태소는 각자 도메인의 토큰(wordpiece)들의 시퀀스(sequence)로 나타내어진다. 즉 우리 모델의 처리 단위는 토큰이다. 문장의 한 어절에 대한 토큰 시퀀스를 $\boldsymbol{x}\in\boldsymbol{X}$라 하자. 이 어절은 1개 이상의 형태소들로 나누어진다. 즉 형태소 리스트로 변환된다. 이 형태소 리스트에 대한 토큰 시퀀스를 $\boldsymbol{y}\in\boldsymbol{Y}$라 하자. $\boldsymbol{l_x}$와 $\boldsymbol{l_y}$는 $\boldsymbol{x}$와 $\boldsymbol{y}$의 길이(토큰 수)를 나타낸다. $\boldsymbol{x}$와 $\boldsymbol{y}$의 $i$번째 원소를 각각 $x_i$, $y_i$라 하자. $T$는 Decoder의 최대 입•출력 길이를 나타낸다고 하자. 우리의 모델은 $\boldsymbol{x}$를 입력으로 받아 $\boldsymbol{y}$를 출력하는 것을 목표로 한다. Fig. 2는 분석 대상 어절과 이를 포함하는 문장을 BERT와 Transformer 모델에게 입력하는 방법과 분석 과정을 보여 주고 있다. 그리고 Fig. 3는 우리 모델의 한국어 형태소 분석 알고리즘을 나타낸다.

형태소 분석 모델이 문맥 정보 없이 어절만 이용할 경우 어휘적 중의성 해결이 어렵다. 예를 들어, “문학상수상자로 선정되었다” 문장에서 복합명사가 포함된 “문학상수상자로” 어절은 문맥에 대한 정보가 없다면 “문학/NNG”, “상수/NNG”, “상자/NNG”, “로/JKB”와 같이 오분석될 수 있다. 그러나 우리 모델은 BERT에게 같이 제공되는 문장의 문맥을 중의성 해소에 이용하여 “문학상/NNG”+“수상자/NNG”+“로/JKB” 로 올바르게 분석한다.

문장의 토큰 시퀀스가 BERT의 입력 제한 길이보다 길면, 분석 대상 어절을 기준으로 앞뒤 어절을 최대한 포함시킨 문장의 일부분을 사용하도록 하였다. Fig. 4는 BERT의 최대 입력 길이 보다 긴 토큰 시퀀스를 가진 문장에서 분석 대상 어절의 위치마다 BERT에 입력되는 문장부분을 보여준다. 분석할 어절의 앞뒤로 가능한한 긴 문장부분을 BERT에게 제공한다. 어절의 토큰 시퀀스는 Encoder의 입력으로도 주어진다. 어절의 토큰 시퀀스 부분에 대한 BERT의 출력은 Encoder 및 Decoder의 attention 작업에 이용된다. 즉, BERT의 모든 출력이 Encoder와 Decoder에서 사용되는 것이 아니라 문장부분을 제외한 분석 대상 어절에 해당하는 출력 부분만 사용된다. Encoder의 출력은 Decoder의 attention 작업의 입력으로 제공된다. Decoder는 한 어절에 대한 분석 결과인 형태소열에 대한 토큰 시퀀스를 생성한다. 전체 문장 내의 각 어절에 대해서 이와 같은 형태소 분석 작업을 반복 한다.

Fig. 3에서 BERT(•)는 입력 어절에 대한 문맥 표현 벡터열을 출력하는 BERT 연산을 나타낸다. 그 결과인 $H_B$는 ($\boldsymbol{l_x}$, $d_m$) 차원의 행렬이다. 가독성을 위하여 입력 어절을 포함하는 문장의 입력 표기를 생략하였다.

**enc_attn(Q,K,V)**는 Encoder의 Attention 연산을 나타내는데, **Q**, **K**, 그리고 **V**는 각각 query, key, value를 의미한다. Encoder는 2가지 attention 연산을 이용하여 입력 토큰들에 대한 인코딩을 수행한다(Fig 3의 단계 2~7). 먼저 $\textbf{enc_attn}_{\boldsymbol{S}}$는 Encoder의 self-attention이다. $\textbf{enc_attn}_{\boldsymbol{B}}$는 Encoder에서의 BERT 출력에 대한 cross-attention을 나타낸다. Encoder의 작업 결과는 ($\boldsymbol{l_x}$, $d_m$) 차원의 행렬이다. 단계 8 이후는 Decoder의 동작을 나타낸다. $\textbf{dec_attn}_{\boldsymbol{S}}$는 Decoder의 Masked self-attention을 나타낸다. $\textbf{dec_attn}_{\boldsymbol{B}}$는 Decoder에서의 BERT출력에 대한 cross-attention, $\textbf{dec_attn}_{\boldsymbol{E}}$는 Decoder에서의 encoder 출력에 대한 cross-attention을 나타

1) BERT가 입력 어절 $x \in X$를 $H_B = \text{BERT}(x)$로 encode한다.
2) *FOR EACH* Encoder의 $l \in [L]$ 층마다:
3) $ENC\_S^l = \text{enc\_attn}_S\left(\text{Norm}(H_E^{l-1}), \text{Norm}(H_E^{l-1}), \text{Norm}(H_E^{l-1})\right)$
4) $ENC\_B^l = \text{enc\_attn}_B\left(\text{Norm}(H_E^{l-1}), \text{Norm}(H_B), \text{Norm}(H_B)\right)$
5) $ENC\_I^l = \frac{1}{2}(H_E^{l-1} + ENC\_S^l + H_B + ENC\_B^l)$
6) $H_E^l = ENC\_I^l + \text{FFN}(\text{Norm}(ENC\_I^l))$
7) *END EACH*
8) *FOR EACH* $1 \le t \le T-1$:
9) *FOR EACH* Decoder의 $l \in [L]$ 층마다:
10) $DEC\_S^l = S_{<t+1}^{l-1} + \text{dec\_attn}_S\left(\text{Norm}(S_{<t+1}^{l-1}), \text{Norm}(S_{<t+1}^{l-1}), \text{Norm}(S_{<t+1}^{l-1})\right)$
11) $DEC\_E^l = \text{dec\_attn}_E\left(DEC\_S^l, \text{Norm}(H_E^L), \text{Norm}(H_E^L)\right)$
12) $DEC\_B^l = \text{dec\_attn}_B\left(DEC\_S^l, \text{Norm}(H_B), \text{Norm}(H_B)\right)$
13) $DEC\_I^l = DEC\_S^l + \frac{1}{2}(DEC\_E^l + DEC\_B^l)$
14) $S_{<t+1}^l = DEC\_I^l + \text{FFN}\left(\text{Norm}(DEC\_I^l)\right)$
15) *END EACH*
16) $S_{<t+1}^L$중 $s_t^L$의 Linear transformation과 softmax 연산 결과 값을 통하여 $t$번째 예측 형태소 토큰 $\hat{y}_t$를 얻는다.
17) *IF* $\hat{y}_t$가 종료 토큰이면:
18) *GO TO* 21
19) *END IF*
20) *END EACH*
21) 예측 형태소 토큰 열 $\hat{y}$ 출력 후 알고리즘 종료

Fig. 3. Korean Morphological Analysis Model (Prediction)

낸다. Feed-Forward 연산은 FFN(•), Normalization 연산은 Norm(•)으로 표기한다.

$H_E^0$는 Encoder에서 층 0에 대한 1 ~ $l_x$ 의 모든 시간대에서 Encoder로의 입력으로서 이는 시퀀스 $x$의 토큰들의 임베딩 벡터로 구성한다. $H_E^l$는 Encoder의 $l$번째 층의 출력을 나타낸다. $H_E^L$는 Encoder의 마지막 층의 출력이다.

시간 t에서 디코더의 층 $l$의 출력을 $s_t^l$로 나타내자. $S_{<t}^l = (s_1^l, \cdots, s_{t-1}^l)$는 처음 시간부터 시간 $t$ 이전까지 Decoder 층 $l$이 출력한 벡터 시퀀스이다. 이는 Decoder의 Masked self-attention에 이용된다(단계 10). 여기서 $s_t^0$는 시간 $t$에서 Decoder로의 (즉 층 1로의) 입력을 나타낸다. 매 시간 $t$ 마다 최종층의 출력 $s_t^L$에 linear feed-forward 연산과 softmax 연산을 적용하여 최대 확률을 가지는 토큰 $\hat{y}_t$을 출력 토큰으로 결정한다(단계 16). 각 시간 $t$ 마다 Decoder는 바로 이전 시간의 출력 토큰 $\widehat{y_{t-1}}$의 임베딩 벡터를 입력 $s_t^0$의 값으로 받는다. 단, $s_1^0$은 형태소 토큰 시퀀스의 시작을 알리는 특수 토큰 [CLS]의 임베딩 벡터를 받는다. 단계 17~18과 같이 Decoder가 결정한 출력 토큰이 종료 스페셜 토큰과 같으면 Decoder의 작업을 종료한다.

### 3.4 훈련 방법

한국어 형태소 분석 모델을 훈련시킬 때는 정답을 제공하는 학습데이터를 이용한다. 분석 대상 어절의 토큰들은 BERT와 Encoder의 입력으로 넣는다. 훈련에서는 Decoder의 입력으로 (바로 앞 시간에 디코더가 예측한 출력 토큰 대신에) 앞 시간에 나와야 할 정답 토큰을 준다.

훈련 방법은 다음과 같다. 먼저 BERT의 파라미터를 고정하고(frozen BERT) Transformer의 BERT-Encoder Attention, BERT-Decoder Attention을 포함한 Encoder와 Decoder를 훈련시켰다(Transformer + frozen BERT). 이 때, Learning Rate $lrate$를 Equation (1)과 같이 적용하였다.

$$lrate = d_{model}^{-0.5} \bullet \min(step\_num^{-0.5}, step\_num \bullet warmup\_steps^{-1.5}) \quad (1)$$

시간 $t$에서 Decoder가 예측한 토큰들에 대한 확률과 해당 시간에서 출력되어야 할 정답 토큰 정보를 이용하여 훈련에 필요한 loss를 결정한다. 우리 모델의 Decoder는 매 시간마다 하나의 토큰을 결정하여 출력하여야 하므로 (즉 분류 작업), Cross-Entropy를 loss로 사용한다[Equation (2)]. $N$을 형태소 도메인의 토큰 사전의 크기라 하자. 시간 $t$ 마다 one-hot 벡터 $y_t$를 준비한다. 시간 $t$ 에서 정답 토큰의 레이블 인덱스가 $i$라면 ($1 \le i \le N$), $y_t$의 원소 $y_{t,i} = 1$로 하고, 나머지 다른 원소들 $y_{t,i'} = 0$ 로 한다(for all $i' \neq i$). $\widehat{y_{t,i}}$는 시간 $t$에서 레이블 인덱스 $i$의 정답 토큰에 대한 Decoder의 예측 확률이라 하자. 그렇다면 한 어절에 대한 훈련예제를 $\Omega$라 하면 이에 대한 loss는 Equation (2)와 같다.

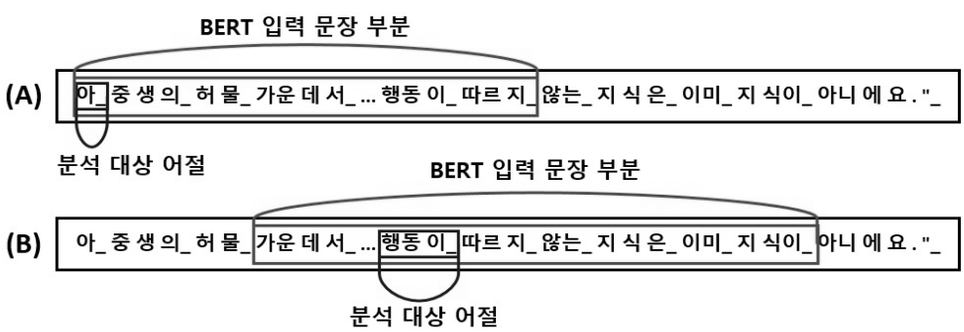



Fig. 4. Example of Input to BERT of Korean Morphological Analysis Model

Table 1. Parameter Values on Training with frozen BERT

| Parameters | Values |
|---|---|
| Optimizer | Adam |
| Beta1/Beta2/Epsilon | 0.9/0.997/1e-9 |
| Warmup steps | 16,000 |
| Dropout rate | 0.2 |
| Label smoothing | 0.1 |

Table 2. Parameter Values on Fine-tuning with BERT

| Parameters | Values |
|---|---|
| Optimizer | Adam |
| Learning rate | 2e-5 |
| Beta1/Beta2/Epsilon | 0.9/0.990/1e-7 |
| Dropout rate | 0.2 |
| Label smoothing | 0.1 |

$$J(\Omega) = -\sum_{t=1}^{T}\sum_{i=1}^{N}[y_{t,i}\log(\widehat{y_{t,i}})] \quad (2)$$

추가적으로, 모델의 일반화 성능을 높이기 위하여 Label Smoothing을 적용하였다. 이때 $y_{t,i}$대신에 Equation (3)의 $y_{t,i}^{LS}$를 사용한다. $\alpha$는 Label Smoothing 계수로 0과 1사이의 값을 갖는다.

$$y_{t,i}^{LS} = y_{t,i}(1-\alpha) + \alpha/N \quad (3)$$

검증은 1 Epoch 훈련이 끝날 때마다 하였으며, 조기 종료 방법을 활용하여 가장 높은 검증 F1-Score를 보이는 가중치를 얻었다.

이 가중치로 Transformer 모델을 초기화 시키고 BERT와 함께 Fine-tuning 하였다(Transformer + BERT fine-tuning). Fine-tuning을 수행할 때는 Transformer 모델과 BERT 모두 동일한 Learning Rate로 가중치 업데이트를 진행하였다. Fine-tuning 수행 과정에서도 조기 종료 방법을 통하여, 최고 F1-Score를 기록한 가중치를 얻어서 최종 평가 때 사용한다. Table 1과 Table 2는 훈련에 적용한 hyper-parameter 정보이다.

### 3.5 Beam Search

Beam Search는 Decoder가 출력 토큰 시퀀스를 구하는데 사용된다. 한국어 형태소 분석 모델은 입력 어절의 토큰 시퀀스 $\boldsymbol{x}$에 대하여, 추론을 통하여 $P(\boldsymbol{y}|\boldsymbol{x})$를 최대로 하는 형태소 토큰 시퀀스 $\boldsymbol{y}$를 구하려 한다. 그러나 가능한 모든 $\boldsymbol{y}$의 수는 매 시간마다 토큰 사전 크기의 배수로 증가한다. 따라서 매 시간마다 가능한 $\boldsymbol{y}$를 모두 고려하는 것은 자원 소모가 매우 크다. 이 문제의 효율적 처리를 위해 Beam Search가 도입 되었다[11-13].

Table 3. Number of Sentences/Eojeols/Morphemes

| | Number of Sentences | Number of Eojeols | Number of Morphemes |
|---|---|---|---|
| Train | 660,449 | 7,892,595 | 18,015,019 |
| Validation | 82,557 | 988,278 | 2,256,135 |
| Test | 82,557 | 987,016 | 2,253,038 |

Beam Search는 매 시간마다 가장 높은 가능성을 가지는 Beam size $k$개의 시퀀스를 유지한다. 각 시퀀스는 다음 시간에 $N$개의 시퀀스로 확장되므로 총 $k \times N$개가 생기는데 이 중에서 가장 높은 확률을 가지는 $k$개만을 선택한다. 반면 Greedy Search는 매 시간 $\boldsymbol{t}$마다 가장 높은 확률을 가진 토큰을 선택하므로 항상 1개의 시퀀스만 고려한다. Beam Search는 $k$개 시퀀스를 유지하므로 Greedy Search보다는 최적에 더 가까운 결과를 산출할 수 있다.

## 4. 실험 및 오류 분석

### 4.1 데이터

국립국어원에서 제공하는 세종말뭉치에서 현대문어 - 형태분석 말뭉치 자료를 사용하였다. 오류가 있는 문장들을 제외하고 실험에 사용된 총 문장의 수는 825,563개이다. 무작위로 섞은 다음 훈련, 검증, 그리고 평가를 위해 문장 수를 기준으로 하여 8:1:1 비율로 나누었다.

### 4.2 성능 평가 방법

성능 평가 척도는 형태소 단위 F1-Score를 적용하였다. 시스템이 출력한 형태소 열에서 실제 정답과 일치하는 비율은 Precision이다[Equation(4)]. 실제 정답에서 시스템이 출력한 형태소와 일치한 형태소 비율이 Recall이다[Equation (5)]. F1-Score는 Precision과 Recall을 조화 평균한 값이다[Equation (6)].

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative} \quad (4)$$

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive} \quad (5)$$

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (6)$$

본 모델은 어절 단위로 형태소 분석을 진행하기 때문에, 어절 경계를 계산할 필요 없이 어절 마다 정확한 성능 평가가 가능하다. 성능 평가를 위하여 모델의 출력 형태소 열과 실제 정답 형태소 열을 비교한다. 이 때, 공백 단위로 분리하여 형태소 별로 비교한다. 정답 형태소 열을 기준으로 처음 정답 형태소와 모델 출력 형태소들을 차례대로 비교하며 정답과 일치한 형태소가 있으면, True Positive를 하나 증가시킨다. 모델 출력에서 일치한 형태소의 다음 것부터 정답 형태소의 다음 것과 비교한다.

### 4.3 훈련 및 검증 성능

Table 4는 Transformer + frozen BERT 훈련 Epoch 마다 Loss 값과 검증 형태소 단위 F1-Score를 측정한 결과이다. Transformer + frozen BERT 훈련 Epoch 9 일 때, F1-Score가 가장 높으므로 Transformer + BERT fine-tuning의 대상으로 선택하였다.

Table 5는 Transformer + BERT fine-tuning Epoch 마다 Loss 값과 검증 형태소 단위 F1-Score를 측정한 결과이다. Transformer + BERT fine-tuning Epoch 5 일 때, F1-Score가 가장 높으므로 평가 모델 가중치의 대상으로 선택하였다.

본 모델의 평가에 적용할 Beam Search의 Beam size $k$를 정하기 위해서, 검증 F1-Score가 가장 높은 Transformer + BERT fine-tuning Epoch 5에서 Greedy Search 형태소 분석 결과가 틀린 어절을 가려내었다. 그리고 Greedy Search에서 틀린 어절에 대한 Beam Search 형태소 분석 성능을 $k$별로 비교하고 가장 높은 성능을 보인 $k=3$를 평가 때 Beam Search의 Beam size $k$로 선택하였다. Table 6은 그 결과를 나타낸다.

### 4.4 오류 분석

Table 7은 본 모델의 형태소 분석 결과 중 오류가 발생한 경우를 보여준다. Output 항목은 Transformer + BERT fine-tuning 훈련 Epoch 5의 결과 가중치를 보유한 모델의 형태소 분석 결과이다.

오류 1은 "소꿉장난"에서 "꿉" 음절이 "헴" 음절로 변경되는 현상을 보여준다. 오류 2는 입력 단어 "트렌드"로부터 "트랙"으로 두 음절이 변경된 단어가 출력되었다. 오류 3에서는 "퀭한" 어절의 정답 형태소 중 "퀭" 대신 의미가 다른 "드래곤" 형태소가 출력되었다. 이는 자주 등장하지 않은 단어("퀭")가

Table 4. Transformer+frozen BERT Loss/Validation F1-Score

| Epoch | Loss | Validation F1-Score |
|---|---|---|
| 1 | 1.9304 | 95.52 |
| 2 | 1.4669 | 96.74 |
| 3 | 1.4344 | 97.10 |
| 4 | 1.4217 | 97.28 |
| 5 | 1.4145 | 97.35 |
| 6 | 1.4095 | 97.48 |
| 7 | 1.4057 | 97.49 |
| 8 | 1.4026 | 97.57 |
| 9 | 1.4001 | **97.61** |
| 10 | 1.3981 | 97.58 |

Table 5. Transformer+BERT Loss/Validation F1-Score

| Epoch | Loss | Validation F1-Score |
|---|---|---|
| 1 | 1.4034 | 98.10 |
| 2 | 1.3970 | 98.15 |
| 3 | 1.3943 | 98.19 |
| 4 | 1.3925 | 98.21 |
| 5 | 1.3911 | **98.21** |
| 6 | 1.3900 | 98.20 |

Table 6. Validation F1-Score on Each Method of Search

| Search | Validation F1-Score |
|---|---|
| Greedy Search | 55.250 |
| Beam Search ($k=2$) | 56.016 |
| Beam Search ($k=3$) | **56.024** |
| Beam Search ($k=4$) | 56.022 |

Table 7. Examples of Morphological Analysis Errors

| | | |
|---|---|---|
| 1 | Input | **소꿉장난하는** 기분이겠다. |
| | Target | 소꿉장난/NNG 하/XSV 는/ETM |
| | Output | **소헴장난/NNG** 하/XSV 는/ETM |
| 2 | Input | {메가 **트렌드}(Mega** Trend)의 저자 ... |
| | Target | 트렌드/NNG }/SS (/SS Mega/SL |
| | Output | **트랙/NNG** }/SS (/SS Mega/SL |
| 3 | Input | **퀭한** 눈에 글썽거리는 눈물을 발견하는 순간 ... |
| | Target | 퀭/XR 하/XSA ㄴ/ETM |
| | Output | **드래곤/NNG** 하/XSA ㄴ/ETM |
| 4 | Input | ... 땅파먹고 사느라 배운 건 별로 **없다만** 그래두 ... |
| | Target | 없/VA 다만/EC |
| | Output | **바다/NNG 이/VCP 니/EC** |
| 5 | Input | ... 셜리 맥클레인이 **엘리베이트** 안내원으로 나온다. |
| | Target | 엘리베이트/NNG |
| | Output | **엘리베이터/NNG** |

전혀 다른 단어(“드래곤”)로 변경되는 오류에 해당한다. 오류 4는 단어가 의미적으로 또는 연어(collocation)적으로 관련 단어로 교체(“땅”→“바다”)되어 발생하는 오류로 “없다만”이라는 어절에 대한 형태소 분석 결과가 모두 틀렸다. 오류 5에서 입력 어절 “엘리베이트”에 대한 모델의 출력 형태소 “엘리베이터/NNG”는 Target과 동일하지 않으나, 비표준어 “엘리베이트”가 표준어 “엘리베이터”로 철자 오류 교정이 되었음을 볼 수 있다. 이러한 오류들은 Sequence-to- Sequence 모델에서 주로 발견되는 현상으로 본 모델의 한계점을 보여준다.

### 4.5 평가 결과 및 타 연구의 분석 모델과의 비교

Table 8은 기존의 연구와 본 연구의 모델 별로 사용된 훈련/검증/평가 데이터를 나타낸다. Table 8의 모든 데이터는 세종 말뭉치의 전부 또는 일부를 사용했다.

다른 한국어 형태소 분석 시스템들도 세종 말뭉치로부터 데이터를 만들어 사용하였으나, 사용 데이터 범위나 훈련/검증/평가 비율이 다른 점 등의 이유로 평가 데이터가 일치하지 않아 성능 면에서 서로를 정확히 비교하는 것은 어렵다.

Table 9는 기존 연구와 본 연구의 형태소 단위 F1-Score 평가 결과이다.

[14]는 Sequence-to-Sequence 모델에 Out-of-vocabulary 문제와 고유명사의 출력 확률이 작아지는 문제를 해결하기 위하여 복사 매커니즘을 도입하였다. [15]는 한국어 형태소 분석과 품사 태깅 방법을 End-to-End 방식으로 통합하여 접근하였다. Sequence-to-Sequence 주의 기반 Encoder-Decoder 구조로 GRU를 사용하였다. 음절 단위로 입력 받아서 Beam search로 Decoding 하는 형태소 분석을 수행한다. [16]은 Encoder-Decoder의 구조가 아닌 양방향 RNN Network 기반의 모델이다. 양방향 LSTM Network 위에 Conditional Random Field를 사용하였다. 음절 단위로 품사를 태그하고 사전을 이용하여 원형 복원한다. [17]은 Sequence-to-Sequence 모델을 기반으로 하며, 음절 단위 입력을 받아서 형태소 분할, 품사 태깅과 원형 복원을 동시에 수행한다. 미등록 음절 처리를 위해 Decoder는 Pointer-generator Network를 사용하였다. 신조어와 띄어쓰기가 없는 데이터에 대하여 장점을 가진다. [18]은 양방향 다층 LSTM의 Encoder와 스택 포인터 네트워크로 구성된 Decoder를 사용한다. 지금까지 소개한 모델들의 성능은 우리 모델의 성능보다 낮다.

Table 8. Datasets for Each Model

| Models | Train/Validation/Test Sentences |
|---|---|
| Hwang(2016)[14] | 88,225/1,000/9,185 |
| Li(2017)[15] | 90%/1%/10% of 100K |
| Kim(2018)[16] | 640K/160K/- |
| Choe(2020)[17] | 85%/5%/10% of 1,303,218 |
| Min(2020)[18] | 202,508/5,000/52,781 |
| Choi(2020)[19] | 90K/1K/10K |
| Youn(2020)[20] | 90%/-/10% of 660K |
| Proposed model | 660,449/82,557/82,557 |

Table 9. Test F1-Score of Each Model

| Models | Test F1-Score |
|---|---|
| Hwang(2016)[14] | 97.08 |
| Li(2017)[15] | 97.15 |
| Kim(2018)[16] | 97.92 |
| Choe(2020)[17] | 97.93 |
| Min(2020)[18] | 98.12 |
| Choi(2020)[19] | 98.27 |
| Youn(2020)[20] | 98.27 |
| Proposed model(Greedy Search) | 98.23 |
| Proposed model(Beam Search) | 98.24 |

[19]는 한국어 형태소 분석에 Transformer를 사용한 첫 시도이다. Encoder에는 KorBERT를 사용하였다. Decoder에 Attention과 복사 매커니즘을 도입하였다. 형태소 도메인의 토큰 시퀀스 출력으로부터 형태소 간 경계는 파악 가능하나 어절 경계를 알 수 없기 때문에, 어절 경계를 구분하는 이진 분류기를 추가하였다. 평가 성능은 F1-Score 98.27로 우리 모델의 성능보다 약간 높다. 그러나 BERT 입력의 최대 길이를 제한하여 어절 길이가 100 이하인 문장들만 훈련 및 평가에 사용하였다. 따라서 문장 길이가 최대 길이보다 길면 처리하지 못하는 문제점이 있다.

[20]은 Seuqence-to-Sequence와 BERT를 파이프라인 형태로 연결한 모델을 사용한다. 형태소 분석 과정을 형태소 원형 복원과 형태소 분리(품사 태깅 포함)의 두 단계로 나누어 처리 하였다. 이 모델 역시 Seuqence-to-Sequence 최대 길이와 BERT의 최대 길이 제한으로 인해, 아주 긴 문장의 처리가 어렵다.

우리의 모델은 형태소 단위 F1-Score 98.24를 달성하여 현재까지 최고 수치를 기록한 [19]와 [20]의 성능과 거의 비슷하다. 실제로는 훈련/평가에 사용한 데이터가 일치하지 않아 성능 수치에 의한 정확한 비교는 어렵다. 이들 연구와 달리 우리 모델은 어절 별로 형태소 분석을 수행하므로 문장의 길이에 영향을 받지 않는 장점이 있다.

우리 모델의 형태소 분석 속도를 측정한 결과 3.16문장/초, 37.87어절/초로, 이는 딥러닝이 적용되지 않은 상용 형태소 분석 시스템보다 상당히 느린 편이다. 딥러닝 모델을 사용한 다른 형태소 분석 연구들은 모두 속도를 공개하지 않았

다. 따라서 다른 딥러닝 기반 형태소 분석 모델과 본 모델과의 속도 비교는 불가능하다. 우리의 모델은 BERT와 Transformer 모델을 같이 활용하여 높은 성능을 달성할 수 있었으나, 규모가 큰 딥러닝 모델의 특성상 방대한 계산량으로 인한 속도 저하가 하나의 단점으로 지적될 수 있다. 이는 향후 해결해야 할 문제점이다.

## 5. 결 론

본 연구에서는 한국어 형태소 분석을 위한 딥러닝 모델을 제안하였다. 형태소 분석은 한국어 자연어 처리 작업(task)의 성능에 큰 영향을 주기 때문에 그 중요성이 크다.

최근 대부분의 한국어 형태소 분석 연구는 Sequence-to-Sequence 딥러닝 모델을 기반으로 한다. 이들과 같은 맥락에서 우리는 기계 번역에서 최고 성능을 이룩한 BERT-fused 모델을 한국어 형태소 분석에 도입하였다. BERT-fused 모델은 기계 번역에 사용되는 Transformer와 자연어 처리의 전반적인 태스크에서 성능 도약을 이루어낸 BERT 언어표현 모델을 활용한 모델이다. 우리는 ETRI가 제공하는 한국어 Pre-trained BERT인 KorBERT 어절단위 모델을 사용하였다.

우리의 형태소 분석 모델은 형태소 단위 F1-Score 98.24의 평가 성능을 달성하였다. 이는 현재까지의 형태소분석 연구에서 알려진 가장 높은 성능과 비슷한 수준이다. 하지만 다른 시스템들에 비해서 우리의 모델은 문장 길이의 제한을 받지 않는 장점이 있다. 우리는 본 연구에서 한국어 형태소 분석을 기계 번역과 유사한 문제로 보아 기계 번역과 동일한 딥러닝 모델을 사용할 수 있음을 보였다. 우리의 현재 모델을 기반으로 하여 앞으로 더 높은 성능의 모델을 개발하고자 한다.

## References


[1] D. Ra, M. Cho, and Y. Kim, "Enhancing a Korean part-of-speech tagger based on a maximum entropy model," *Journal of the Korean Data Analysis Society*, Vol.9, No.4, pp.1623-1638, 2007.

[2] K. Cho, et al., "Learning phrase representations using RNN Encoder-decoder for statistical machine translation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp.1724-1734, 2014.

[3] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in Neural Information Processing Systems*, pp.3104-3112, 2014.

[4] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *Proceedings of the International Conference on Learning Representations*, San Diego, California, 2015.

[5] T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp.1412-1421, 2015.

[6] A. Vaswani, et al., "Attention is all you need," in *Advances in Neural Information Processing Systems*, pp.6000-6010, 2017.

[7] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of NAACL-HLT,* Minneapolis, Minnesota, pp.4171-4186, 2019.

[8] J. Zhu, et al., "Incorporating BERT into neural machine translation," in *Proceedings of the International Conference on Learning Representations*, Addis Ababa, Ethiopia, 2020.

[9] Q. Wang, et al., "Learning Deep Transformer Models for Machine Translation," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Florence, Italy, pp.1810-1822, 2019.

[10] T. Nguyen and J. Salazar, "Transformers without tears: Improving the normalization of self-attention," in *Proceedings of the 16th International Workshop on Spoken Language Translation*, 2019.

[11] A. Graves, "Sequence transduction with recurrent neural networks," in *Proceedings of the 29th International Conference on Machine Learning Workshop on Representation Learning*, Edinburgh, Scotland, 2012.

[12] M. Freitag and Y. Al-Onaizan, "Beam search strategies for neural machine translation," in *Proceedings of the First Workshop on Neural Machine Translation*, Vancouver, Canada, pp.56-60, 2017.

[13] E. Battenberg, et al., "Exploring neural transducers for end-to-end speech recognition," in *Proceedings of 2017 IEEE Automatic Speech Recognition and Understanding Workshop*, Okinawa, Japan, pp.206-213, 2017.

[14] H. S. Hwang and C. K. Lee, "Korean morphological analysis using sequence-to-sequence learning with copying mechanism," in *Proceedings of the Korea Computer Congress 2016*, pp.443-445, 2016. (in Korean)

[15] J. Li, E. H. Lee, and J.-H. Lee, "Sequence-to-sequence based morphological analysis and part-of-speech tagging for Korean language with convolutional features," *Journal of Korean Institute of Information Scientists and Engineers*, Vol.44, No.1, pp.57-62, 2017. (in Korean)

[16] S.-W. Kim and S.-P. Choi, "Research on joint models for Korean word spacing and POS (Part-Of-Speech) tagging based on bidirectional LSTM-CRF," *Journal of Korean Institute of Information Scientists and Engineers*, Vol.45, No.8, pp.792-800, 2018. (in Korean)

[17] B. Choe, I.-h. Lee, and S.-g. Lee, "Korean morphological analyzer for neologism and spacing error based on sequence-to-sequence," *Journal of Korean Institute of Information Scientists and Engineers*, Vol.47, No.1, pp.70-77, 2020. (in Korean)

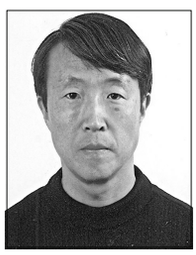
[18] J. Min, S.-H. Na, J.-H. Shin, and Y.-K. Kim, "Stack pointer network for Korean morphological analysis," in *Proceedings of the Korea Computer Congress 2020*, pp.371-373, 2020. (in Korean)

[19] Y. Choi and K. J. Lee, "Performance analysis of Korean morphological analyzer based on transformer and BERT," *Journal of Korean Institute of Information Scientists and Engineers*, Vol.47, No.8, pp.730-741, 2020. (in Korean)

[20] J. Y. Youn and J. S. Lee, "A pipeline model for Korean morphological analysis and part-of-speech tagging using sequence-to-sequence and BERT-LSTM," in *Proceedings of the 32nd Annual Conference on Human & Cognitive Language Technology*, pp.414-417, 2020. (in Korean)

**이 창 재**

https://orcid.org/0000-0003-4556-438X
e-mail : cjlee7128@yonsei.ac.kr
2019년 연세대학교 컴퓨터공학과(학사)
2021년 연세대학교 전산학과(석사)
2021년 ~ 현 재 연세대학교 소프트웨어학부 연구원
관심분야 : 자연어처리, 인공지능, 최적화



**나 동 열**

https://orcid.org/0000-0003-1449-4614
e-mail : dyra@yonsei.ac.kr
1978년 서울대학교 전자공학과(학사)
1980년 한국과학기술원 전산학과(석사)
1989년 미시간주립대학교 전산학과(박사)
1980년~1990년 한국전자통신연구원 선임연구원
1991년 ~ 현 재 연세대학교 소프트웨어학부 교수
관심분야 : 자연어처리, 정보검색, 인공지능