# 음절과 형태소 정보를 이용한 한국어 문장 띄어쓰기 교정 모델*)

최정명[1,O], 오병두[1], 허탁성[1], 정영석[2], 김유섭[2]

한림대학교 융합소프트웨어학과[1]
한림대학교 소프트웨어융합대학[2]
{ jeong5905, iambd822, gjxkrtjd221, dnfkdi1995}@gmail.com, yskim@hallym.ac.kr

# Korean sentence spacing correction model using syllable and morpheme information

Jeong-Myeong Choi [1,O], Byoung-Doo Oh[1], Tak-Sung Heo[1], Yeong-Seok Jeong[2], Yu-Seop Kim[2]
Hallym University, Department of Convergence Software [1]
Hallym University, College of Software [2]

**요 약**

한국어에서 문장의 가독성이나 맥락 파악을 위해 띄어쓰기는 매우 중요하다. 또한 자연 언어 처리를 할 때 띄어쓰기 오류가 있는 문장을 사용하면 문장의 구조가 달라지기 때문에 성능에 영향을 미칠 수 있다. 기존 연구에서는 N-gram 기반 통계적인 방법과 형태소 분석기를 이용하여 띄어쓰기 교정을 해왔다. 최근 들어 심층 신경망을 활용하는 많은 띄어쓰기 교정 연구가 진행되고 있다. 기존 심층 신경망을 이용한 연구에서는 문장을 음절 단위 또는 형태소 단위로 처리하여 교정 모델을 만들었다. 본 연구에서는 음절과 형태소 단위 모두 모델의 입력으로 사용하여 두 정보를 결합하여 띄어쓰기 교정 문제를 해결하고자 한다. 모델은 문장의 음절과 형태소 시퀀스에서 지역적 정보를 학습할 수 있는 Convolutional Neural Network와 순서정보를 정방향, 후방향으로 학습할 수 있는 Bidirectional Long Short-Term Memory 구조를 사용한다. 모델의 성능은 음절의 정확도와 어절의 정밀도, 어절의 재현율, 어절의 F1 score를 사용해 평가하였다. 제안한 모델의 성능 평가 결과 어절의 F1 score가 96.06%로 우수한 성능을 냈다.

주제어: 띄어쓰기 교정, 다중 필터 1D-CNN, Bidirectional LSTM

## 1. 서론

최근 블로그, SNS, 메신저, 공개된 데이터 등 인터넷에서 존재하는 대량의 텍스트 데이터를 활용하여 Chat Bot, QA System, 감성 분석 등 다양한 자연 언어 처리 연구를 진행한다. 하지만 이런 대량의 데이터에는 띄어쓰기 오류가 다수 존재한다. 자연 언어 처리의 다양한 작업에서 띄어쓰기 오류가 있는 문장을 사용하게 되면 성능 결과에 영향을 미칠 수 있다. 따라서 자연 언어 처리 작업의 전처리 과정으로 띄어쓰기 오류를 교정하여 올바른 문장을 생성하는 것이 필요하다. 또한 발화자의 음성 정보를 텍스트로 변환하는 기술인 speech-to-text에서 발화자는 발화시 문장의 띄어쓰기 부분에서 의도적으로 공백 부분에서 휴지를 가지며 말하지 않기 때문에 text로 변환한 결과에서 띄어쓰기 오류가 존재할 수 있다. 그렇기 때문에 텍스트 생성 같은 작업에서도 후처리로 띄어쓰기 교정이 필요하다.[1][2]

본 연구에서는 띄어쓰기 교정 모델을 구성하기 위해 텍스트에서 추출할 수 있는 정보인 음절과 형태소를 모델의 입력 값으로 사용한다. 각각의 음절과 형태소는 다중 필터 1D-CNN 과 bi-LSTM을 통해 지역적 정보와 순서정보를 추출한 뒤 음절과 형태소 정보를 결합한다. 마지막 층에서는 띄어쓰기 여부를 출력한다.

이후 본 연구는 다음과 같이 구성된다. 2장은 띄어쓰기 교정과 관련된 연구에 대해 설명한다. 3장은 본 연구에서 제안하는 띄어쓰기 교정 모델에 대해 설명한다. 4장은 실험에 사용되는 데이터와 실험 내용 및 실험 결과에 대해 설명하고, 5장은 본 연구의 결론과 향후 연구 계획에 대해 설명한다.

## 2. 관련 연구

한국어 띄어쓰기 교정 연구는 규칙 기반, 통계 기반, 확률 기반 방식에서 심층 신경망 방식으로 발전하고 있다. [3]에서는 띄어쓰기 알고리즘을 prepare, forward, backward, heuristic 4단계로 구성하여 띄어쓰기 오류를 교정한다. [4]에서는 자동 띄어쓰기 방법으로 공백 삽입 여부를 결정하기 위해 3개의 확률 값을 두고 확률 값에 음절의 bi-gram에 따른 가중치를 적용한 합을 구하여 임계치 이상인 경우 공백을 삽입하는 방법을 사용하여 띄

*) 이 논문은 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임 (No. 2019R1A2C2006010)

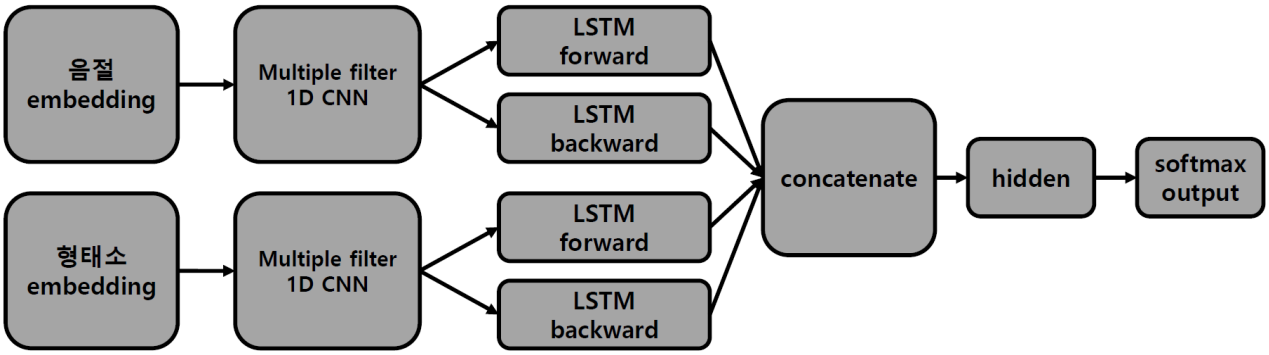


그림 1. 다중 필터 1D-CNN과 bi-LSTM 결합한 모델 구조

어쓰기 교정을 하였다. [5]에서는 띄어쓰기가 전혀 되어 있지 않은 한국어 문장에 띄어쓰기 여부를 나타내는 태그를 부착하는 레이블링 문제로 보고 CRF를 이용하여 자동 띄어쓰기를 시도하였다. [6]은 Encoder와 Decoder가 각각 bi-LSTM으로 구성된 모델과 형태소 분석을 통해 한국어 자동 띄어쓰기 방법을 제안한다. [7]에서는 bi-LSTM과 CRF를 결합한 구조의 모델에 음절 단위의 자질을 통해 띄어쓰기 교정하는 방법을 제안한다. [8]에서는 음절 단위의 자질을 사용하여 BERT를 LSTM, CRF 등과 결합한 모델의 띄어쓰기 교정 성능을 비교한다. [9]는 음절 단위의 n-gram 임베딩 사전을 만들어 GRU-CRF 구조의 모델로 학습시켜 띄어쓰기 교정을 한다. [10]은 LSTM 기반의 sequence to sequence 모델의 Decoder에 어텐션 기법을 적용하여 자동 띄어쓰기 문제를 해결하였다. [11]은 음절 단위의 자질을 사용하여 종단 간 심층 신경망으로 띄어쓰기 교정 시스템을 구성하였다.

## 3. 띄어쓰기 교정 모델

그림 1은 본 연구에서 사용하는 띄어쓰기를 교정을 위한 다중 필터 1D-CNN 과 bi-LSTM을 결합한 인공 신경망 모델 구조를 나타낸 그림이다. 음절과 형태소 단위의 정보들은 독립적인 1D-CNN과 bi-LSTM을 통해 훈련되고 bi-LSTM의 출력 값을 통해 음절과 형태소 정보가 결합된다. 이 결합된 정보는 hidden layer에서 추가적으로 훈련한 뒤 softmax layer를 통해 띄어쓰기 여부를 출력한다.

모델의 입력 값으로 문장을 음절과 형태소 단위로 정수 인코딩한 시퀀스 값이 들어간다. 정수 시퀀스 데이터는 각 독립적인 임베딩 레이어를 통해 벡터의 시퀀스가 되어 $[x_1, x_2, \dots x_{L-1}, x_L]$ 형태가 된다.

$$c_i = f(w \cdot x_{i:i+h-1} + b) \quad (1)$$

$$c^h = [c_1, c_2, \dots, c_{n-h+1}] \quad (2)$$

벡터 시퀀스는 다양한 필터 크기를 가지는 다중 필터 1D-CNN에 입력된다. 수식(1)과 (2)를 통해 convolution 연산된 1D-CNN의 출력 값은 concatenate layer를 통해 결합되어 $[c^a, c^b, \dots c^k]$ 형태가 된다.$x$는 벡터 시퀀스를 의미하고 $h$는 필터 크기, $w$는 가중치 벡터, $b$는 bias 벡터, $f$는 non-linear함수를 의미한다.[12]

다중 필터 1D-CNN의 출력이 결합된 값은 LSTM layer에 입력된다. 이 때 정방향과 후방향의 순서정보를 학습하면 문장의 정보를 구체화할 수 있는 장점이 있기 때문에 bi-LSTM 구조를 사용한다. 독립적인 네트워크를 통해 훈련되던 음절과 형태소 단위의 정보는 bi-LSTM의 출력 값들을 concatenate layer를 통해 $[h_s^f, h_s^b, h_m^f, h_m^b]$형태로 결합된다.

결합된 값들은 hidden layer를 거쳐 최종적으로 softmax layer를 통해 띄어쓰기 여부를 나타내는 $[p_1, p_2, p_3]$ 형태로 확률 값을 출력한다. 세 개의 출력 값은 spacing, no spacing, padding을 의미한다.

## 4. 실험 및 결과

### 4.1 데이터

본 논문에서는 띄어쓰기 교정 모델을 학습시키기 위해 국립국어원 언어정보 나눔터에서 제공하는 세종 말뭉치와 직접 수집한 뉴스 기사 데이터를 사용한다. 세종 말뭉치와 뉴스 기사 데이터는 문법과 띄어쓰기 규칙의 완성도가 높기 때문에 띄어쓰기 교정 모델을 학습시키기 위한 데이터로 적합하다.

지도 학습을 하기 위해 띄어쓰기에 대한 label tag를 생성했다. label tag는 음절의 다음이 띄어쓰기가 아니면 <NS>, 띄어쓰기면 <S>로 하였고 문장을 같은 길이로 맞추어 주기 위한 패딩은 <P>로 하였다.

수집된 문장은 총 1300만 문장이다. 이 중 모델의 파라미터 튜닝과 훈련 시 과적합을 방지하기 위해 수집된 문장의 1/5을 무작위로 선택하여 검증 데이터로 사용하였다. 그리고 모델의 최종 성능을 측정하기 위해 수집된 세종 말뭉치 중 1000개를 무작위로 선택하여 시험 데이터로 사용하였다.

### 4.2 실험 환경

학습에 사용된 모델의 파라미터는 다음과 같다. 시퀀스의 최대길이는 100으로 설정하였고, 음절과 형태소 시

퀀스가 입력되는 임베딩 층의 차원은 각각 128, 64로 설정하였다. 다중 필터 1D-CNN은 필터 크기 2,3,4,5를 가지는 4개의 1D-CNN을 사용하였고 각 CNN의 뉴런 개수는 64로 설정하였다. bi-LSTM에서 정방향과 후방향 LSTM의 뉴런 개수는 128로 설정하였고 dropout은 0.5로 하였다. 1D-CNN의 활성화 함수는 ELU로 하였고, LSTM의 활성화 함수는 tanh로 설정했다. 모델의 최적화 함수로 adam을 사용하였고 learning rate는 초기값 1e-3부터 시작하여 1e-6까지 점점 줄어들도록 polynomial decay방법을 사용했다.

본 연구는 한국어 문장에서 형태소 정보를 추출하기 위해 공개 라이센스 형태소 분석기인 Mecab-Ko[1]를 사용했다. Mecab-Ko는 대량의 데이터를 분석할 때 분석 시간이 빠를 뿐 아니라 분석 품질 또한 우수하다.[2]

### 4.3 평가 지표

수식 (3)~(6)는 모델이 예측한 띄어쓰기에 대한 평가 지표를 나타낸다. 수식 (3)은 문장에서 <NS>, <S>, <P> 태그가 잘 분류되었는지를 나타낸다. 수식 (4)~(6)은 띄어쓰기가 없는 문장에 모델이 예측한 태그를 적용시켜 만들어진 문장의 어절이 올바르게 완성되었는지를 나타내는 평가지표이다. 수식(4)는 예측된 태그를 적용시킨 문장에서 올바르게 분류된 어절 수에서 정답 문장의 어절 수를 나눈 값으로 재현율을 의미한다. 수식(5)는 예측된 태그를 적용시킨 문장 중 올바르게 분류된 어절 수에서 예측된 태그를 적용시킨 문장의 전체 어절수를 나눈 값으로 정밀도를 의미한다. 수식(6)은 어절의 재현율과 정밀도를 조화평균한 값이다.

$$Accuracy = \frac{predicted\ correct\ tags}{atual\ entire\ tags} * 100 \quad (3)$$

$$Recall_{어절} = \frac{predicted\ correct\ 어절}{actual\ entire\ 어절} * 100 \quad (4)$$

$$Precision_{어절} = \frac{predicted\ correct\ 어절}{predicted\ entire\ 어절} * 100 \quad (5)$$

$$F1\ score_{어절} = 2 * \frac{Recall_{어절} * Precision_{어절}}{Recall_{어절} + Precision_{어절}} \quad (6)$$

### 4.4 실험 결과

표 1은 모델의 입력데이터로 형태소만 사용했을 경우와 음절만 사용했을 경우, 형태소와 음절 모두 사용했을 경우의 성능을 나타낸다. 수집한 데이터의 일부에도 띄어쓰기 오류가 존재하기 때문에 성능 평가를 위한 시험 데이터 1000개는 띄어쓰기가 정확히 되었는지 재검토하여 구축한 것을 사용하였다. 음절과 형태소 각각 단일 입력데이터로 사용한 경우는 F1 score가 92.98%, 94.43%로 음절과 형태소 모두 입력 데이터로 사용하여 훈련된 모델의 평가 결과인 96.06% 보다 낮은 성능을 보였다. 음절과 형태소가 모델의 입력 데이터로 들어가 독립적으로 다중 필터 1D-CNN과 bi-LSTM을 통해 학습되고 네트워크 후반부에 정보가 결합된 것이 성능 향상에 도움이 됐다.

표 1. 제안하는 모델의 입력 데이터에 따른 성능 변화

| 입력 데이터 | 태그 | 어절 | | |
|---|---|---|---|---|
| | Accuracy | Precision | Recall | F1 score |
| **형태소** | 98.29 | 93.00 | 92.96 | 92.98 |
| **음절** | 98.70 | 95.11 | 93.75 | 94.43 |
| **음절&형태소** | 99.13 | 96.54 | 95.59 | **96.06** |

표 2. 기존 연구된 띄어쓰기 교정 모델과 성능 비교

| 모델 | 태그 | 어절 | | |
|---|---|---|---|---|
| | Accuracy | Precision | Recall | F1 score |
| [3](2015) | 98.06 | 92.27 | 94.15 | 93.20 |
| [9](2016) | 98.32 | 92.68 | 91.96 | 92.32 |
| [10](2018) | - | 93.72 | 94.27 | 93.99 |
| [11](2019) | 98.53 | 95.06 | 93.46 | 95.06 |
| Proposed model | 99.13 | 96.54 | 95.59 | **96.06** |

표 2는 기존에 연구된 띄어쓰기 교정 모델과 제안하는 모델의 성능을 비교한 표이다. 이전 연구에서 훈련에 사용되는 데이터들은 세종 말뭉치, ETRI 말뭉치 등으로 수집된 데이터 종류와 전처리 방법에 따라 데이터셋의 차이가 존재한다. 성능 측정을 위한 시험 데이터로는 이전 연구에서도 공통적으로 사용하는 세종 말뭉치를 사용한다.

[3]은 한 음절 씩 이동하며 어절 사전을 탐색하는 방식의 4단계 알고리즘을 통해 띄어쓰기 교정을 하고 어절 단위 F1- score 성능 평가한 결과 93.20%를 보여주었다. [9]는 GRU와 CRF 모델을 사용해 unigram, bigram, trigram의 조합과 명사 사전을 사용해 띄어쓰기 교정한 결과 어절단위 F1 score 93.32% 성능을 냈다. [10]는 음절 단위 데이터와 LSTM 기반 모델을 사용해 Encoder와 Decoder를 사용하였고, Decoder에는 어텐션 메커니즘 기법을 적용해 띄어쓰기를 교정하였다. 그 결과 어절 단위 F1 score 93.99%의 성능을 냈다. [11]은 음절 단위 데이터와 양방향 LSTM Encoder를 통해 자질 벡터를 생성하고 선형체인 CRF를 사용하는 모델을 구성하여 성능을 평가한 결과 어절 단위 F1 score가 95.06%으로 우수한 성능을 냈다.

본 연구에서는 음절 정보와 형태소 정보 두 개의 입력 데이터를 사용하였다. 각 음절과 형태소 시퀀스는 독립

[1] https://bitbucket.org/eunjeon/mecab-ko

[2] https://iostream.tistory.com/144

적인 다중 필터 1D-CNN과 bi-LSTM 구조의 모델을 학습하고 bi-LSTM의 출력을 통해 음절과 형태소 정보를 결합하여 띄어쓰기를 교정하였다. 그 결과 기존 연구에서 우수한 성능을 보였던 [11] 보다 F1 score 1% 향상된 성능을 보여주었다.

## 5. 결론

본 연구에서 우리는 음절과 형태소 정보를 다중 필터 1D-CNN과 bi-LSTM을 결합한 구조에 입력하여 띄어쓰기를 교정하는 모델을 제안했다. 모델의 성능은 세종 말뭉치를 통해 평가한 결과 입력 데이터가 형태소 < 음절 < 음절&형태소 순으로 높은 성능을 냈다. 음절이나 형태소의 단일 정보만 사용해 띄어쓰기 교정을 수행했을 때는 기존 연구들과 큰 차이가 없는 성능을 보였지만 음절과 형태소 두 정보를 결합하는 모델로 구성하여 띄어쓰기 교정을 수행했을 때는 어절 단위 F1 score 96.06%의 우수한 성능 결과를 냈다.

향후 연구로는 한국어를 사용하는 다양한 자연 언어 처리 연구에서 띄어쓰기 교정 모델을 적용시켜 교정 전과 후에 얼만큼 성능 변화가 있는지 비교 분석할 예정이다.

## 참고문헌


[1] 노희경, 이강희, “구글, 네이버, 다음 카카오 API 활용앱의 표준어 및 방언 음성인식 기초 성능평가”, *예술인문사회 융합 멀티미디어논문지*, 제7권, 제12호, pp. 819-829, 2017.

[2] 임동희, 강승식, 장두성, “음성 인식 후처리를 위한 띄어쓰기 오류의 교정”, *한국정보과학회 학술발표논문집*, 제33권, 제1호, pp. 25-27, 2006.

[3] 심광섭, “말뭉치와 형태소 분석기를 활용한 한국어 자동 띄어쓰기”, *정보과학회논문지*, 제42권, 제1호, pp.68-75, 2015.

[4] 강승식, “음절 바이그램 단순화 기법에 의한 한국어 자동 띄어쓰기 시스템의 성능 개선”, *한국정보과학회 언어공학연구회 학술발표논문집*, pp. 227-231, 2003.

[5] 심광섭, “CRF를 이용한 한국어 자동 띄어쓰기”, *한국인지과학회*, 제22권, 제2호, pp. 217-233 2011.

[6] 장재나, 왕규봉, 강상우, “양방향 LSTM을 이용한 한국어 자동 띄어쓰기, *한국정보과학회 학술발표논문집*, pp. 2052-2053, 2018.

[7] 김선우, 최성필, “Bidirectional LSTM-CRF 기반의 음절 단위 한국어 품사 태깅 및 띄어쓰기 통합 모델 연구”, *정보과학회논문지*, 제45권, 제8호, pp. 792-800, 2018.

[8] 황태욱, 정상근, “BERT를 이용한 한국어 자동 띄어쓰기”, *한국정보과학회 학술발표논문집* pp. 374-376, 2019.

[9] 황현선, 이창기, “딥러닝을 이용한 한국어 자동 띄어쓰기”, *한국정보과학회 학술발표논문집*, pp. 738-740, 2016.

[10] 이태석, 강승식, “LSTM 기반의 sequence-to-sequence 모델을 이용한 한글 자동 띄어쓰기”, *스마트미디어저널*, 제7권, 제4호, pp.17-23, 2018.

[11] 이현영, 강승식, “종단 간 심층 신경망을 이용한 한국어 문장 자동 띄어쓰기”, *정보처리학회논문지*, 제8권, 제11호, pp. 441-448, 2019.

[12] KIM Yoon, “Convolutional neural networks for sentence classification. arXiv preprint arXiv:1408.5882, 2014.