신조어 및 띄어쓰기 오류에 강인한 시퀀스-투-시퀀스 기반 한국어 형태소 분석기

Korean Morphological Analyzer for Neologism and Spacing Error based on Sequence-to-Sequence

저자 (Authors)
최병서, 이익훈, 이상구
Byeongseo Choe, Ig-hoon Lee, Sang-goo Lee

출처 (Source)
정보과학회논문지 47(1), 2020.1, 70-77(8 pages)
Journal of KIISE 47(1), 2020.1, 70-77(8 pages)

발행처 (Publisher)
한국정보과학회
The Korean Institute of Information Scientists and Engineers

URL
http://www.dbpia.co.kr/journal/articleDetail?nodeId=NODE09289740

APA Style
최병서, 이익훈, 이상구 (2020). 신조어 및 띄어쓰기 오류에 강인한 시퀀스-투-시퀀스 기반 한국어 형태소 분석기. 정보과학회논문지, 47(1), 70-77

ISSN 2383-630X(Print) / ISSN 2383-6296(Online)
Journal of KIISE, Vol. 47, No. 1, pp. 70-77, 2020. 1
https://doi.org/10.5626/JOK.2020.47.1.70

신조어 및 띄어쓰기 오류에 강인한 시퀀스-투-시퀀스 기반 한국어 형태소 분석기

(Korean Morphological Analyzer for Neologism and Spacing Error based on Sequence-to-Sequence)

최 병 서 † 이 익 훈 †† 이 상 구 †††
(Byeongseo Choe) (Ig-hoon Lee) (Sang-goo Lee)

요 약 한국어 커뮤니티 등에서 수집되는 인터넷 텍스트 데이터를 형태소 분석하기 위해서는, 띄어쓰기 오류가 있는 문장에서도 정확히 형태소 분석을 해내야 하고, 신조어 등의 사전 외 어휘 입력에 대한 원형복원 성능이 충분해야 한다. 그러나 기존 한국어 형태소분석기는 원형복원에 사전 또는 규칙 기반 알고리즘을 사용하는 경우가 많다. 본 논문에서는 시퀀스-투-시퀀스 모델을 기반으로 띄어쓰기 문제와 신조어 문제를 효과적으로 처리할 수 있는 한국어 형태소 분석기 모델을 제안한다. 본 모델은 사전을 사용하지 않고, 규칙 기반 전처리를 최소화한다. 일반적으로 사용하는 음절 외에도 음절 바이그램과 자소를 입력 자질로 같이 사용하며, 공백을 제거한 데이터를 학습 데이터로 같이 사용한다. 제안 모델은 세종 말뭉치를 이용한 실험에서 사전을 사용하지 않는 기존 형태소 분석기에 비해 뛰어난 성능이 나왔다. 띄어쓰기가 없는 데이터셋 및 인터넷에서 직접 수집한 데이터셋에 대해서도 높은 성능이 나오는 것을 확인하였다.

키워드: 형태소 분석, 품사 태깅, 시퀀스 투 시퀀스, 원형 복원, 인터넷 텍스트 데이터

Abstract In order to analyze Internet text data from Korean internet communities, it is necessary to accurately perform morphological analysis even in a sentence with a spacing error and adequate restoration of original form for an out-of-vocabulary input. However, the existing Korean morphological analyzer often uses dictionaries and complicate preprocessing for the restoration. In this paper, we propose a Korean morphological analyzer model which is based on the sequence-to-sequence model. The model can effectively handle the spacing problem and OOV problem. In addition, the model uses syllable bigram and grapheme as additional input features. The proposed model does not use a dictionary and minimizes rule-based preprocessing. The proposed model showed better performance than other morphological analyzers without a dictionary in the experiment for Sejong corpus. Also, better performance was evident for the dataset without space and sample dataset collected from Internet.

Keywords: morphological analysis, POS tagging, original form recovery, sequence-to-sequence, internet text data

· 본 연구는 문화체육관광부 및 한국콘텐츠진흥원의 2019년도 문화기술연구개발 지원사업으로 수행되었음(R2019050030)

† 학생회원 : 서울대학교 컴퓨터공학부 학생
bschoe@europa.snu.ac.kr
†† 종신회원 : 광주대학교 컴퓨터공학과 교수(Gwangju Univ.)
ihlee@gwangju.ac.kr
(Corresponding author임)
††† 종신회원 : 서울대학교 컴퓨터공학부 교수
sglee@europa.snu.ac.kr

논문접수 : 2019년 9월 24일
(Received 24 September 2019)
논문수정 : 2019년 11월 6일
(Revised 6 November 2019)
심사완료 : 2019년 11월 18일
(Accepted 18 November 2019)

정보과학회논문지 제47권 제1호(2020. 1)

1. 서 론

자연어 처리는 인간의 언어, 즉 자연어로 된 데이터를 컴퓨터로 분석하는 인공지능의 한 분야이다. 특히 소셜 네트워크 서비스(SNS), 웹 문서를 비롯한 인터넷 텍스트 데이터는 기하급수적으로 증가하고 있고, 이러한 빅 텍스트 데이터를 정보 검색, 번역, 챗봇, QA 시스템, 키워드 추출 등 다양한 분야에 활용하려는 시도가 이어지고 있다.

그런데 빅 텍스트 데이터, 특히 한국어 빅 텍스트 데이터를 활용하기 위해서는 형태소 분석이 필수적이다. 하지만 인터넷 텍스트는 정제된 텍스트와 달리 형태소 분석을 위해서는 추가적으로 고려해야 할 요 소가 존재한다. 인터넷 데이터의 문장은 일반적인 문장에 비해 문법 오류가 많은데, 특히 띄어쓰기가 잘못된 문장이 많다. 또한 인터넷 데이터에는 기존에 존재하지 않는 신조어 및 고유명사가 많이 나타난다. 따라서 인터넷 데이터를 분석하기 위한 형태소 분석기는 신조어 및 띄어쓰기 오류에 강건하여야 한다.

그러나 기존의 많은 형태소 분석기는 원형 복원 과정 등에서 사전을 이용한다. 이 경우 학습 데이터셋에서 입력 어절과 출력 형태소를 대응시켜 사전을 구축하는 전처리 과정이 필요하고, 학습 데이터셋에는 나타나지 않은 형태의 형태소 결합이 나타날 경우 원형 분석 사전을 이용해서 복원하는 것에는 한계가 있다. 또한 문맥에 따라 같은 표현형에서 다른 원형 복원이 이루어지는 경우 이를 분석하기 어렵다. 이로 인해 기존의 형태소 분석기는 신조어에 대한 분석 능력이 떨어지게 된다.

본 논문에서는 위의 어절 분리 문제와 OOV(out-of-vocabulary, 사전 외 어휘) 문제를 효과적으로 처리할 수 있게 함으로써 인터넷 텍스트 데이터 분석을 효과적으로 할 수 있는 형태소 분석기 모델을 제안한다. 본 논문에서 제안하는 모델은 시퀀스 투 시퀀스(sequence-to-sequence)[1]를 이용하는 모델로, 원형 복원 사전을 비롯한 언어 지식을 사용하거나 규칙 기반의 전처리 과정을 거치지 않고 입력 문장에서 End-to-end로 원형 복원까지 형태소 분석을 한다. 또한 입력 문장에 공백이 제대로 되어 있지 않은 경우에도 효과적으로 대응할 수 있도록 어절 경계에 의존하지 않는 방법으로 학습을 한다. 그리고 음절 바이그램(bigram)과 자소 임베딩을 추가 입력 자질로 사용하여 형태소 분석 성능을 높인다.

본 논문에서는 제안하는 모델의 형태소 분석 성능을 확인하기 위해 일반적으로 사용하는 정제된 한글 말뭉치인 세종 말뭉치에 대한 성능을 측정한다. 실험을 통해 본 모델이 기존 형태소 분석기 모델에 비해 경쟁력이 있고, 특히 사전을 사용하지 않는 모델들보다 성능이 높음을 확인한다. 또한 인터넷 데이터에서의 성능을 확인하기 위해 띄어쓰기가 제거된 데이터 및 인터넷에서 직접 수집한 텍스트 데이터에 대한 형태소 분석 성능을 기존 공개 형태소 분석기와 비교하여 본 모델이 띄어쓰기가 잘못된 입력에 대해서도 충분한 성능을 보장함을 확인한다. 그리고 학습 데이터에 없거나 적게 나타나는 형태소에 대한 분석 성능을 비교하여 본 모델이 고유명사나 인터넷 신조어를 잘 분석할 수 있음을 확인한다.

2. 관련 연구

형태소 분석을 위해서는 기본적으로 하나 이상의 형태소 조합으로 이루어진 어절 안에서 형태소를 나누는 작업이 필요하다. [3]과 같은 딥러닝을 이용한 연구들에서는 주로 음절 단위로 형태소 분할을 하며, 품사 태깅과 통합하여 진행되는 연구 또한 많다. 반면 [4,5]를 비롯한 전통적인 확률 기반 모델에서는 사전을 이용하여 음절 또는 자소 단위로 어절을 나눠 그 중 가능한 형태소 시퀀스 후보를 모두 고려하는 방식을 사용하였다.

전통적으로 형태소 분석기는 규칙과 은닉 마르코프 모델(Hidden Markov Model, HMM)에 기반한 형태소 분석기[4,5] 등의 방법이 주로 연구되었다. [6,7] 등은 CRF(Conditional random field)를 이용하여 형태소 분할과 품사 태깅을 한 뒤, 사전 및 은닉 마르코프 모델을 기반으로 원형 복원을 하였다.

최근에는 딥 러닝을 통한 한국어 형태소 분석기 연구 또한 활발하게 이루어지고 있다. 딥 러닝을 이용한 한국어 형태소 분석기는 음절 단위의 품사 태깅 방법론을 활용한 연속적 레이블링(Sequence labeling) 문제로 많이 연구되고 있다[3]. 이를 위해 순환 신경망(Recurrent neural net, RNN)을 이용하는 방식이 주로 연구되었다.

[3]은 BiLSTM-CRF을 사용하여 음절 단위 형태소 분할과 품사 복원을 동시에 진행 후 여러 사전을 순차적으로 사용하여 원형 복원을 하였고, N-gram이나 명사 사전 등을 다양한 요소를 입력으로 사용하여 높은 성능을 냈다. [2,8] 등은 BiLSTM 또는 BiLSTM-CRF 모델을 사용하여 원형 복원 사전을 사용하지 않는 형태소 분석기를 연구하였다. [8]은 자소 단위로 BiLSTM을 이용하여 형태소 원형을 먼저 복원한 뒤 BiLSTM-CRF를 이용하여 형태소 분할 및 품사 태깅을 하였다.

한편, 시퀀스 투 시퀀스(Sequence-to-sequence)[1]를 사용한 End-to-end 방식의 형태소 분석기 연구 역시 이루어지고 있다. 이 방식은 기본적으로 입출력 시퀀스의 길이 및 형태가 같을 필요가 없기 때문에 형태소 분할, 원형 복원 및 품사 태깅의 과정을 한 번에 처리할 수 있다. [9]는 시퀀스 투 시퀀스에 합성곱 요소(convolutional feature)를 사용하는 시도를 하였다. [10]은 기존

시퀀스 투 시퀀스 모델에 입력 추가 구조(input-feeding)과 복사방법론(copying mechanism)을 적용하여 성능을 비교하였다.

기존 한국어 형태소 분석기 모델의 원형 복원은 주로 말뭉치를 통해 구축한 원형 복원 사전을 통해 이루어져왔다[11,12]. 일반적으로 말뭉치에서 음절 간 대응 관계를 정렬(align)하여 원형 사전을 구축한다. 그리고 원형 복원 단계에서는 원형 복원이 필요한 형태소에 대해 사전을 통해 원형을 복원한다. 하지만 사전을 사용하는 원형 복원의 경우 학습 데이터셋에는 나타나지 않은 형태의 형태소 결합이 나타날 경우 원형 분석 사전을 이용해서 복원하는 것에는 한계가 있다. 또한 사전을 이용한 원형 복원은 문맥에 따른 중의성을 해소하는 것 역시 쉽지 않다.

[2,8] 등의 연구에서는 원형 복원을 품사 태깅과 같이 시퀀스 레이블링(Sequence Labeling) 문제로 접근하여 사전을 사용하지 않고 원형 복원을 하는 방법을 연구하였다. 하지만 시퀀스 레이블링에 기반한 방식은 사전을 사용한 방식과 마찬가지로 음절의 대응관계를 맞춰줘야 하는 정렬 전처리가 필요하다는 문제가 있다.

3. 모델 설명

3.1 모델 구조

본 논문의 모델은 기본적으로 음절 단위 시퀀스 투 시퀀스를 통해 형태소 분할과 품사 태깅, 원형 복원을 동시에 하는 End-to-end 방식 모델이다. 이 모델은 어휘사전 및 특수한 전처리가 필요 없고, 품사 태깅을 하지 않고도 키워드를 분리해낼 수 있다. 또한 기존 시퀀스 투 시퀀스 방식 연구와 달리 음절 바이그램과 자소 요소를 추가적으로 인코더의 입력으로 사용한다. 모델의 전체 구조는 그림 1과 같다.

3.2 입력 및 출력

본 모델에서는 한국어 음절을 신경망의 기본 입력 및 출력 단위로 사용하였다. 따라서 입력 문장을 음절 단위로 나누어 각 음절을 임베딩한 벡터를 신경망의 입력으로 넣었다. 띄어쓰기가 입력에 있을 경우 띄어쓰기 태그 <s>로, 음절 입력과 동등한 입력으로 처리하였다. 한글 외의 문자가 입력으로 들어올 경우 그 문자를 음절과 동등하게 학습하였다.

출력은 음절과 품사 태그로 구성된다. End-to-end를 위해, 디코더에서는 음절의 시퀀스와 품사 태그가 반복되어 출력된다. 음절이 연속적으로 출력되는 경우 이 음절들은 1개의 형태소로 보고, 그 형태소의 품사는 연속되는 출력 끝에 등장한 품사 태그가 태깅된다. 이렇게 되면 품사 태그가 형태소 사이의 구분자 역할을 겸한다. 출력에서는 어절 경계인 띄어쓰기는 출력하지 않고, 형태소 시퀀스만 출력하게 된다. 입력과 마찬가지로 한글 외의 문자는 각 문자를 음절과 동등하게 처리하여 학습하였다.

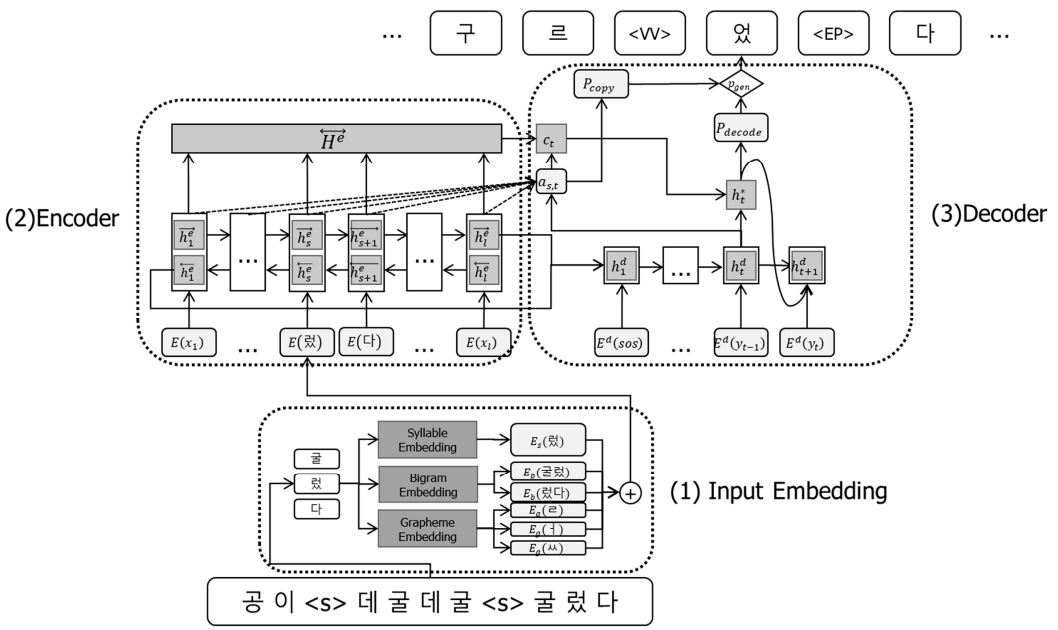

그림 1 형태소 분석기 모델 구조

Fig. 1 Architecture of Proposed Morphological Analyzer Model

또한 [9,10] 등의 시퀀스 투 시퀀스를 사용하는 기존 논문이 단순히 음절 임베딩만을 인코더로의 입력에 사용한 것과 달리, 더 많은 정보를 입력하기 위해 입력 문장을 임베딩할 때, 음절 임베딩 외에도 음절 바이그램과 자소를 같이 임베딩하여 사용하였다. 그림 1의 모델 구조 중, (1)이 신경망의 입력에 해당하는 부분이다.

세종 말뭉치의 형태소 분석 데이터셋의 형태소는 90% 이상이 1음절 또는 2음절로 된 형태소이다. 따라서 만약 음절 unigram과 음절 바이그램을 같이 사용한다면 입력 임베딩 자체에 대부분의 한국어 형태소 정보를 담을 수 있는 것이다. 따라서 음절 바이그램 임베딩을 사용함으로써 음절 임베딩만으로는 부족한 형태소 형성 및 구분에 관한 정보를 더욱 풍부하게 학습할 수 있다. 음절 바이그램 임베딩은 [3]에서 형태소 분석에 추가 요소로 활용한 바 있다.

본 모델에서는 그 음절의 직전 음절과의 바이그램, 그 음절의 다음 음절과의 바이그램 두 가지를 모두 임베딩하였다. 그리고 임베딩된 벡터는 음절 임베딩 벡터와 결합(concatenate)하여 사용하였다. 한글이 아닌 문자의 경우 영어, 숫자, 한자, 그 외로 분류하여 단순화하여 바이그램이 너무 많아지지 않도록 하였다. 또한 문장의 처음과 끝에는 <SOS>, <EOS> 토큰을 추가하여 부족한 글자 수를 보충하였다.

또한 음절을 구성하는 각 자소(Grapheme)의 임베딩을 추가로 학습하여 사용하였다. 자소를 입력 단위로 사용하면 음절 단위 데이터에 자주 나타나는 오타 및 음절 변형을 통한 신조어에 효과적일 것이라 판단하였다. [1]에서 음절 임베딩 없이 자소 임베딩을 통한 형태소 분석기를 연구한 바 있다. 본 모델에서는 인코더에 입력으로 들어가는 한글 음절을 초성, 중성, 종성으로 분해하여 각각 자소 임베딩 행렬을 사용하여 고정 차원의 벡터로 만들었다. 그리고 이 임베딩 벡터를 기존 음절 임베딩에 결합하여 사용하였다. 한글이 아닌 문자의 경우 영어, 숫자, 한자, 그 외로 분류하여 단순화하였다.

3.3 인코더-디코더 모델

형태소 분석을 실제로 수행하는 신경망으로는 시퀀스 투 시퀀스, 그 중에서도 주의 기반 인코더-디코더(attention-based encoder-decoder) 모델을 사용하였다. 그림 1의 모델 구조 중, (2)와 (3)이 각각 인코더와 디코더이다. 주의 기반 인코더-디코더는 임베딩된 문장을 LSTM과 같은 순환 신경망을 통해 인코딩하고, 이 인코딩된 고정 크기의 벡터와 인코더의 출력 시퀀스를 이용하여 원형 복원된 형태의 형태소를 음절 단위로 연속적으로 출력한다. 본 모델에서는 인코더와 디코더의 순환 신경망을 구성하는 기본 뉴런으로 LSTM을 두 층으로 쌓아서 사용하였다.

인코더는 양방향 구조를 적용하여, 문장 처음에서 시작하여 순차적으로 문장을 읽어들이는 LSTM과 문장 끝에서 시작하여 역순으로 문장을 읽어들이는 LSTM을 같이 사용하여 출력을 결합하였다.

디코더는 입력 추가 구조(input-feeding)가 적용된 주의 기반 디코더[13]을 기반으로 하였다. 주의 기반 모델은 순환 신경망의 출력과 인코더의 출력의 관계를 계산, 주의 분포(attention distribution) a_t를 이용하는 모델이다. 이를 통해 음절 단위 입력으로 인해 문장이 길어지더라도 정확한 분석이 가능하다.

또한 미등록 음절 처리를 위해 Pointer-generator Network[14] 모델을 적용하였다. 모델인 Pointer-generator Network는 디코더 출력에 인코더의 입력 중 일부가 그대로 복사되어 출력될 확률을 고려하는 모델이다. 일반적인 주의 기반 디코더의 출력 확률 분포 P_decode (y_t=w)에 더불어, 입력의 주의 분포 P_copy (y_t=w)를 같이 고려함으로써 출력되는 결과를 보정한다. 따라서 Pointer-generator Network는 OOV, 즉 기존에 학습되지 않은 어휘가 많이 나타나는 인터넷 데이터에 적합하다. 또한 Pointer-generator Network는 [10]에서는 미등록 음절을 위해 적용한 CopyNet과 달리 copy 확률 p_gen을 학습하여 복사와 생성의 균형을 유지한다.

4. 실험

4.1 실험 데이터

실험에는 21세기 세종계획 말뭉치, 통칭 세종 말뭉치를 사용하였다. 세종 말뭉치에는 형태소 분석 및 품사 태깅이 된 한국어 문장이 총 1,303,218 문장이 있으며, 본 논문에서는 이 말뭉치를 Train:Validate:Test의 비율이 85:5:10가 되도록 임의로 나누어 세종 Train Set, Valid Set, Test set을 각각 만들어 사용하였다.

또한, 세종 말뭉치에서 공백을 제거한 데이터셋을 같이 사용하였다. 모델 학습 시 공백을 제거한 문장과 공백을 제거하지 않은 문장을 1:1 비율로 섞어서 사용하여 띄어쓰기 오류에 대한 강건함을 높였고, Test Set도 공백을 제거하여 실험에서 사용하였다.

또한 인터넷 텍스트 데이터에서의 형태소 분석 성능을 확인하기 위해 인터넷 텍스트 샘플을 만들었다. 실제 인터넷 텍스트 표본을 만들기 위해, 웹 크롤러를 통해 DCinside(www.dcinside.com), 클리앙(www.clien.com), mlbpark(www.mlbpark.com), YouTube 댓글(www.youtube.com)에서 2019년 3월동안 한국어 문장을 임의로 500개 수집하였다. 그리고 이 500개의 문장에 대해 품사 태깅을 제외하고 형태소 원형 복원까지의 과정을 수행하였다. 단, 품사 태깅의 경우 모호성이 있을 수 있어 하지 않았다. 이를 통해 500쌍의 형태소 원형 복원

성능 평가 데이터셋(이하 SNU Internet Community Morpheme Dataset, SICMD)을 구축하였다.

4.2 학습 환경 및 변수

각 네트워크에서 입출력 음절 임베딩은 100차원, 음절 바이그램 임베딩은 50차원, 자소 임베딩은 10차원을 사용하였다. LSTM은 은닉 차원을 300차원인 2개의 레이어(layer)로 하였다. 학습 드롭아웃(dropout)값은 0.3으로 주었다. 학습 모델은 OpenNMT-py[15]를 기반으로 모델에 맞게 수정하여 사용하였다.

4.3 성능 평가 방법

성능 평가 기준으로는 일반적으로 형태소 분석 성능 평가에 사용하는 형태소 단위 F1-measure를 사용하였다. 또한, 품사 태깅 과정을 제외한 형태소 분할 및 원형 복원 과정만의 성능 역시 F1-measure를 통해 평가하였다.

또한 형태소 분석기의 띄어쓰기 오류에 대한 강건함을 확인하기 위해, 띄어쓰기를 제거한 테스트 입력에 대한 형태소 분석 성능을 기존 입력에 대한 성능과 비교하였다.

그리고, 신조어에 대한 강인함을 확인하기 위한 실험 역시 하였다. 학습 데이터에 잘 등장하지 않거나 아예 등장하지 않는 형태소에 대한 분석 성능을 확인하기 위해 세종 Train set에서 분석 결과로 나타나는 형태소를 등장 빈도에 따라 나누어 분류하였다. 그리고 각 모델의 세종 Test set에서 형태소 분석 결과를 평가할 때, Train set에서의 형태소 빈도에 따라 각각 F1-measure를 측정하였다.

4.4 실험 내용 및 결과

4.4.1 형태소 분석 성능 비교

본 논문에서 제안한 모델의 성능 평가를 위한 실험으로, 형태소 분석 결과의 형태소 단위 F1-measure를 비교하는 실험을 하였다.

성능 평가를 위한 데이터셋으로는 세종 말뭉치에서의 test 데이터셋과 SICMD를 이용하였다. 세종 말뭉치에 대해서는 품사를 제외하고 형태소의 원형 복원 형태만 비교하는 F1-measure와 품사 태깅까지 완료된 상태에서의 형태소 단위 F1-measure를 모두 확인하였다. SICMD에서는 품사를 제외하고 형태소의 원형 복원 형태만을 비교하는 F1-measure만 비교하였다.

성능 비교 대상은 동일한 테스트셋에서의 성능을 확인하기 위해 직접 실행할 수 있도록 학습된 모델이 공개된 형태소 분석기와 비교하였다. konlpy[16]에서 제공하는 형태소 분석기 중 kkma와 komoran을 사용하고, khaiii와 [8]을 추가로 비교 대상 모델로 사용하였다. 또한, [3,7,9,10,17]의 논문에 나온 형태소 분석 성능을 같이 비교하였다. 이 논문들은 역시 세종 말뭉치에서의 형태소 단위 F1-measure를 형태소 분석 성능으로 제시하고 있으나, 본 논문과는 다른 데이터셋 분할을 사용하였음을 표에 *로 표시하였다.

논문에서 제안한 모델은 음절 임베딩에 추가된 요소인 음절 바이그램 임베딩과 자소 임베딩의 성능을 확인하기 위해, 각 임베딩을 추가한 것과 추가하지 않은 모델을 비교하였다. 성능 비교 결과는 표 1과 같다.

우선 품사가 태깅된 세종 말뭉치에 대한 성능을 보면, 본 논문에서 제안한 세종 말뭉치의 성능이 기존 대부분의 연구의 성능보다 높음을 확인할 수 있다. 비록 세종 말뭉치에서 학습 데이터와 Test 데이터를 나누는 방식이 달라 직접적인 수치 비교는 힘들지만, [13]을 제외한 다른 형태소 분석기에 근접하거나 조금 더 높은 성능이 나타났다. 특히 품사를 포함한 세종 말뭉치 태깅 성능에

표 1 각 모델의 형태소 분석 성능 비교

Table 1 Comparison of morphological analysis performance of each model

	Model	Algorithm	Dictionary	Without POS Tag		Including POS Tag
				Sejong	SICMD	Sejong
Existing Model	Li et. al (2017)[9]	Seq2Seq	X	-	-	0.9715*
	Jung et. al (2018)[10]	Seq2Seq	X	-	-	0.9708*
	Lee (2013)[[17]	S-SVM	O	-	-	0.9803*
	Na and Kim (2018)[7]	CRF	O	-	-	0.9774*
	Kim and Choi (2018)[3]	BILSTM-CRF	O	-	-	**0.9877***
	kkma	HMM	O	0.8920	0.7979	0.8333
	komoran	HMM	O	0.9438	0.7990	0.8342
	khaiii	CNN	O	0.9544	0.7589	0.9421
	choi et. al (2016)[1]	BILSTM-CRF	X	0.9742	0.8008	0.9586
Proposed Model	Syllable Embedding(Baseline)	Seq2Seq	X	0.9814	0.8252	0.9723
	Syllable + Grapheme Embedding	Seq2Seq	X	0.9824	0.8321	0.9735
	Syllable + Bigram Embedding	Seq2Seq	X	0.9860	**0.8321**	0.9781
	Syllable + Grapheme + Bigram Embedding	Seq2Seq	X	**0.9868**	0.8317	0.9793

서 음절 바이그램 임베딩을 적용한 모델이 그렇지 않은 모델에 비해 0.005 이상의 F1-measure값 향상이 있었다. 특히, 원형 복원에 사전을 사용하지 않는 모델 중에서는 가장 높은 성능이 나타났다. 자소 임베딩은 성능 향상이 있었지만 음절 바이그램 임베딩에 비해서는 효과가 크게 없었다.

한편, 품사가 태깅되지 않은 세종 말뭉치에 대한 성능 역시 기존에 공개되어있는 형태소 분석기에 비해 크게 높은 성능을 가진 것으로 나타났다. 또한 SICMD에서의 형태소 분할 및 원형 복원 성능 역시 기존 형태소 분석기보다 높게 나타났다. 단, 이 데이터셋에 대해서는 자소 임베딩이 음절 바이그램 임베딩과 근접한 성능을 나타냈으나, 두 임베딩을 같이 사용한 경우에 대해서는 오히려 성능이 약간 감소하였다.

4.4.2 띄어쓰기를 제거한 데이터에서의 성능 확인

형태소 분석기가 띄어쓰기에 대해 얼마나 강건한지 확인하기 위해, 띄어쓰기를 제거할 경우 형태소 분석의 정확도가 얼마나 감소하는지에 대한 실험을 하였다.

기존의 비교 가능한 형태소 분석기와 논문에서 제안한 모델에 대해서, 세종 test 데이터셋에서 공백을 제거한 입력에 대한 형태소 단위 F1-measure를 비교하여 공백을 제거하기 전에 비해 얼마나 성능이 감소하는 지 확인하여 공개되어있는 모델, [8] 및 띄어쓰기 모듈을 포함한 모델인 [3,17]와 비교하였다. 실험 결과는 표 2에 정리하였다. 역시 데이터셋 분할이 다른 경우는 *로 표시하였다.

본 논문에서 제안한 모델은 품사 포함, 품사 미고려 두 가지 경우 모두에 대해 띄어쓰기가 없는 데이터셋에서의 성능 감소가 0.01 전후로 나타났다. 특히 자소 임베딩과 음절 바이그램 임베딩을 같이 사용하는 경우 띄어쓰기 제거에 따른 성능 감소가 최소로 나타났다. 반면 기존 형태소 분석기는 다수가 띄어쓰기가 없는 경우에 성능이 크게 감소하는 것을 확인할 수 있었다. 본 모델보다 성능이 높은 [3]은 본 논문에서 제안하는 모델과 달리 형태소 분할 및 원형 복원에서부터 사전 수록 여부를 입력 요소로 사용하는 모델이므로 직접 비교대상이 아니다.

4.4.3 형태소 등장 빈도에 따른 분석 성능 확인

형태소 분석기의 신조어 분석 능력을 확인하기 위해, 형태소의 등장 빈도별 분석 성능을 확인하는 실험을 하였다. 세종 말뭉치 중 Train set에서의 형태소를 등장 빈도에 따라 분류하여 세종 test set에서의 형태소 분석 F1-measure를 따로 측정하여 비교한다. 그 결과는 그림 2의 그래프와 같다.

기존 공개 형태소 분석기에 비해 모든 경우에서 높은 성능을 나타냈고, Train set에서 전혀 나타나지 않은 OOV 형태소에 대해서도 0.7~0.8의 F1-measure가 나타났다. 반면 기존 공개 형태소 분석기는 0.2~0.5의 F1-measure가 나타났다.

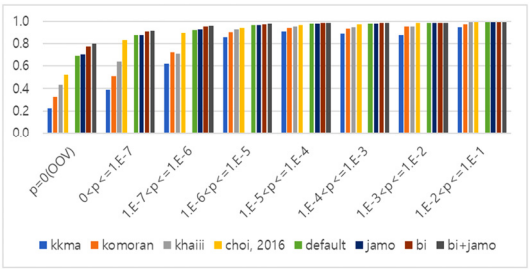

그림 2 형태소 등장 빈도에 따른 세종 테스트 셋에서의 분석 성능

Fig. 2 F1-Score according to Morpheme appearance frequency in Sejong Test Dataset

표 2 세종 말뭉치에서 공백을 제거한 입력에 대한 형태소 분석 성능 비교

Table 2 Comparison of morphological analysis performance for Sejong Dataset without space

	Model	Algorithm	Dictionary	Sejong-Without POS Tag (Diff from Performance at test data with space)	Sejong-Including POS Tag (Diff from Performance at test data with space)
Existing Model	Lee (2013)[17]	S-SVM	O		0.9699(-0.0104)*
	Kim and Choi (2018)[3]	BILSTM-CRF	O		0.9792(-0.0085)*
	Kkma	HMM	O	0.8696(-0.0224)	0.8141(-0.0191)
	Komoran	HMM	O	0.8275(-0.1163)	0.6841(-0.1501)
	Khaiii	CNN	O	0.5491(-0.4053)	0.4585(-0.4836)
	Choi, et. al (2016)[8]	BILSTM-CRF	X	0.6941(-0.2800)	0.6941(-0.3025)
Proposed Model	Syllable Embedding(Baseline)	Seq2Seq	X	0.9681(-0.0134)	0.9584(-0.0139)
	Syllable + Grapheme Embedding	Seq2Seq	X	0.9691(-0.0134)	0.9594(-0.0140)
	Syllable + Bigram Embedding	Seq2Seq	X	0.9765(-0.0095)	0.9677(-0.0104)
	Syllable + Grapheme + Bigram Embedding	Seq2Seq	X	0.9791(-0.0077)	**0.9710(-0.0083)**

전체적으로 등장 빈도가 낮은 형태소일수록 성능이 크게 차이나는 것을 확인할 수 있었고, OOV에서의 성능 차이가 가장 크게 나타나는 것을 확인할 수 있었다. 특히 음절 바이그램을 적용한 모델이 그렇지 않은 모델에 비해 OOV에서의 F1-measure가 0.1 가까이 증가하는 것을 확인할 수 있다.

4.4.4 SICMD에서 OOV 한글 형태소 분석 성능

인터넷에서 직접 수집한 데이터셋인 SICMD에서 세종 Train set에서 나타나지 않은 OOV 형태소 중 한글로 된 형태소를 골라 이에 대한 형태소 분석 성능을 확인하였다. 인터넷 데이터에는 형태소 분석이 애매한 한국어가 아닌 형태도 많이 나타났기 때문에 한글로 된 형태소에 대한 성능만 따로 확인하였다. 또한 이 실험에서는 F1-measure뿐만 아니라 정밀도(precision)와 재현율(recall)을 같이 확인하였다. 그 결과를 표 3에 정리하였다.

OOV 한글 형태소 분석 성능 확인 결과, 기존 모델에 비해 0.1 이상 높은 F1-measure가 나타났다. 특히 기존 모델에 비해 높은 것은 정밀도였다. 정밀도가 높다는 것은 false positive가 적다는 것으로, 오분석으로 실제로 존재하지 않는 형태소를 생성하는 경우가 적다는 것을 의미한다. 재현율의 경우에는 바이그램 임베딩을 사용하는 것으로 인해 조금 낮아졌는데, 이는 바이그램의 경우 기존 단어 형성 정보를 많이 담고 있기 때문에 기존 언어 상식과 다른 형태소가 새로 나타났을 경우 이를 제대로 분석해내지 못하는 경우가 있는 것으로 생각된다. 자소 임베딩의 경우 정밀도와 재현율이 모두 상승하는 효과가 있었고, 특히 바이그램 임베딩을 사용하지 않았을 때 더 큰 성능 향상이 있었다.

5. 결 론

본 논문에서는 한국어 인터넷 텍스트 데이터를 잘 분석하기 위한 한국어 형태소 분석기를 제안하였다. 인터넷 데이터의 특징인 띄어쓰기 문제와 OOV 문제를 해결하기 위해 시퀀스 투 시퀀스를 이용하고, 음절 바이그램 정보와 자소 정보를 같이 사용하여 형태소 분석을 하였다.

성능 평가를 통해 시퀀스 투 시퀀스를 이용하는 제안 기법이 사전이나 복잡한 전처리 없이도 충분히 경쟁력 있는 분석 정확도를 성취함을 보였다. 또한 음절 바이그램 정보와 자소 정보 임베딩을 이용하는 제안 기법이 형태소 분석 정확도를 높이는 것 역시 확인할 수 있었다. 특히 음절 바이그램 정보를 통한 성능 향상이 뚜렷하게 나타났다.

본 모델은 사전 정보 및 기존 언어 지식을 최소한으로 사용하면서도 다른 모델과 비교해 성능 경쟁력이 있는 모델으로서의 의미가 있다. 또한 정제된 데이터가 아니라 어절 구분이 제대로 안 되어있거나 새로운 단어가 많이 출연하는 데이터에서도 충분한 성능을 가지기 때문에 활용도가 높을 것으로 기대한다.

향후 연구로는 제안한 연구결과를 바탕으로 음절 바이그램 임베딩이나 자소 임베딩의 단순 결합이 아닌 더 효율적으로 적용하는 연구나 인터넷 데이터의 초성체나 오타 등 다른 특징을 더 잘 고려하는 형태소 분석기 연구가 가능할 것으로 보인다.

References

[1] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," *Advances in neural information processing systems*, pp. 3104-3112, 2014.

[2] A. Matteson, C. Lee, Y. B. Kim, and H. Lim, "Rich Character-Level Information for Korean Morphological Analysis and Part-of-Speech Tagging," *arXiv preprint arXiv:* 1806.10771, 2018.

[3] 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 KIISE*, Vol. 45 No. 8, pp. 792-800, 2018.

[4] S. Lee, J. Tsujii, and H.-C. Rim, "Hidden Markov model-based Korean part-of-speech tagging considering high agglutinativity, word-spacing, and lexical correlativity," *Proc. of the 38th Annual Meeting on Association for Computational Linguistics*, Association

표 3 SICMD에서 OOV 한글 형태소 분석 성능

Table 3 OOV Hangeul morpheme analysis performance in SICMD

	Model	precision	recall	F1-measure
Existing Model	kkma	0.4000	0.2261	0.2889
	komoran	0.3949	0.2191	0.2818
	khaiii	0.2820	0.4594	0.3495
	Choi et. al (2016)[8]	0.3280	0.5088	0.3989
Proposed Model	Syllable Embedding(Baseline)	0.4717	0.5300	0.4992
	Syllable + Grapheme Embedding	0.5113	**0.5618**	0.5354
	Syllable + Bigram Embedding	0.5983	0.4947	0.5416
	Syllable + Grapheme + Bigram Embedding	**0.6026**	0.4982	**0.5455**

for Computational Linguistics, pp. 384-391, 2000.
[5] S. Lee, H. Lim, and H.-C. Rim, "Two-Level Part-of-Speech Tagging for Korean Text Using Hidden Markov Model," *Proc. of the 6th Annual Conference on Human and Cognitive Language Technology*, pp. 305-312, 1994.
[6] S.-H. Na, "Conditional random fields for korean morpheme segmentation and pos tagging," *ACM Transactions on Asian and Low-Resource Language Information Processing*, Vol. 14.3, No. 10, 2015.
[7] S.-H. Na, and Y.-K. Kim, "Phrase-based statistical model for korean morpheme segmentation and POS tagging," *IEICE Transactions on Information and Systems*, Vol. 101, No. 2, pp. 512-522, 2018.
[8] J. Choi, J. Youn, and S. Lee, "A grapheme-level approach for constructing a Korean morphological analyzer without linguistic knowledge," *2016 IEEE International Conference on Big Data (Big Data)*, IEEE, 2016.
[9] 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 KIISE*, Vol. 44, No. 1, pp. 57-62, 2017.
[10] S. Jung, C. Lee, and H. Hwang, "End-to-End Korean Part-of-Speech Tagging Using Copying Mechanism," *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*, Vol. 17.3, No. 19, 2018.
[11] J.-C. Shin, and C.-Y. Ock, "A Korean Morphological Analyzer using a Pre-analyzed Partial Word-phrase Dictionary," *Journal of KISS : Software and Applications*, Vol. 39, No. 5, pp. 415-424, 2012.
[12] K. Shim, "Morpheme Restoration for Syllable-based Korean POS Tagging," *Journal of KISS : Software and Applications*, Vol. 40, No. 3, pp. 182-189, 2013.
[13] M.-T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," *arXiv preprint arXiv:* 1508.04025, 2015.
[14] A. See, P. J. Liu, and C. D. Manning, "Get to the point: Summarization with pointer-generator networks," *arXiv preprint arXiv:* 1704.04368, 2017.
[15] G. Klein, Y. Kim, Y. Deng, J. Senellart, and A. M. Rush, "Opennmt: Open-source toolkit for neural machine translation," *arXiv preprint arXiv:* 1701.02810. 2017.
[16] E. L. Park, and S. Cho, "KoNLPy: Korean natural language processing in Python," *Proceedings of the 26th Annual Conference on Human & Cognitive Language Technology*, Vol. 6, 2014.
[17] C. Lee, "Joint Models for Korean Word Spacing and POS Tagging using Structural SVM," *Journal of KISS : Software and Applications*, Vol. 40, No. 12, pp. 826-832, 2013.

최 병 서

2017년 서울대학교 컴퓨터공학부 학사
2019년 서울대학교 컴퓨터공학부 석사. 관심분야는 자연어 처리, 빅데이터, 데이터베이스, 머신러닝

이 익 훈

1996년 서울시립대학교 전산통계학과 학사
1998년 서울시립대학교 전산통계학과 석사
2005년 서울대학교 컴퓨터공학과 박사
2013년~2017년 NEXT Institute 교수
2017년~현재 광주대학교 컴퓨터공학과 조교수. 관심분야는 데이터베이스, 빅데이터, 자연어 처리

이 상 구

1985년 서울대학교 계산통계학과 학사
1987년 노스웨스턴대학교 컴퓨터과학 석사
1990년 노스웨스턴 대학교 컴퓨터과학 박사. 1999년~2001년 미국 조지타운대학교 방문교수. 1992년~현재 서울대학교 컴퓨터공학부 교수. 관심분야는 데이터베이스, Semantic 기술, e-Business 기술