# 음절 임베딩과 양방향 LSTM-CRF를 이용한 한국어 문장 자동 띄어쓰기

이현영○, 강승식
국민대학교 컴퓨터공학과
le32146@gmail.com, sskang@kookmin.ac.kr

# Bi-LSTM-CRF and Syllable Embedding for Automatic Spacing of Korean Sentences

Hyun-Young Lee○, Seung-Shik Kang
Dept. of Computer Science, Kookmin University

**요 약**

본 논문에서는 음절 임베딩과 양방향 LSTM-CRF 모델을 이용한 한국어 문장 자동 띄어쓰기 시스템을 제안한다. 문장에 대한 자질 벡터 표현을 위해 문장을 구성하는 음절을 Unigram 및 Bigram으로 나누어 각 음절을 연속적인 벡터 공간에 표현하고, 양방향 LSTM을 이용하여 현재 자질에 양방향 자질들과 의존성을 부여한 새로운 자질 벡터를 생성한다. 이 새로운 자질 벡터는 전방향 신경망과 선형체인(Linear-Chain) CRF를 이용하여 최적의 띄어쓰기 태그 열을 예측하고, 생성된 띄어쓰기 태그를 기반으로 문장 자동 띄어쓰기를 수행하였다. 문장 13,500개와 277,718개 어절로 이루어진 학습 데이터 집합과 문장 1,500개와 31,107개 어절로 이루어진 테스트 집합의 학습 및 평가 결과는 97.337%의 음절 띄어쓰기 태그 분류 정확도를 보였다.

주제어: 자동 띄어쓰기, Bi-LSTM, 음절 임베딩, 선형체인 CRF

## 1. 서론

한국어 자동 띄어쓰기는 부분적으로 잘못된 문장과 띄어쓰기가 전혀 되어 있지 않은 문장을 대상으로 어절 단위 경계에 공백을 삽입하는 문제로 볼 수 있다[3]. 부분적으로 잘못 되어 있는 경우는 맞춤법 교정, 복합명사 분해 등 2-3 어절에 걸친 띄어쓰기 오류를 고려하는 형태로 이는 공백을 완전히 제거한 후 띄어쓰기가 없는 문장으로 변환 후 띄어쓰기를 적용하는 형태로 해결할 수 있다[1,2,3].

자동 띄어쓰기 연구는 규칙 기반 방식과 말뭉치 기반의 확률 및 통계적인 방법으로 나누어진다. 규칙 기반 방식은 사전과 규칙을 정하고, 형태소 분석 결과 등을 이용한다[1,4]. 이는 언어 자원을 보고 규칙을 정해야하는 만큼 언어학적 지식이 필요하고, 새로운 규칙이 발견될 때마다 추가해야하는 등 많은 시간과 노력이 필요하다. 이와 더불어, 형태소 분석을 사용하는 추가적인 시간도 필요하다.

확률 및 통계적인 방법은 대용량의 말뭉치로부터 n-gram 빈도수의 확률 및 통계 정보를 이용하여 어절의 경계에 공백을 삽입하는 형태로 띄어쓰기를 한다[2,5]. 빈도수 기반의 방식 외에 기계 학습 방법은 입력된 문장의 각 음절을 띄어쓰기 태그 클래스로 분류하는 형태로 Hidden Markov Model(HMM), Conditional Random Field(CRF), Structural SVM 모델 등을 이용하여 자동 띄어쓰기 태그 열 부착(sequence labeling) 방식으로 해결한다[3,6,7].

분류 문제에서 우수한 성능을 보여주는 딥러닝은 자연어 처리 분야에서도 적용되어 활발히 연구가 진행되고 있다[8,9,10]. [10]에서는 단방향 LSTM의 변형인 단방향 GRU와 CRF를 결합한 모델을 한국어 자동 띄어쓰기 문제에 적용하였다.

본 논문에서는 띄어쓰기가 전혀 적용되지 않은 문장을 입력으로 입력문장의 각 음절에 해당하는 띄어쓰기 태그 클래스(B 또는 I)로 분류하는 방법으로 단방향 LSTM-CRF 모델보다 태그 열 부착에서 우수한 성능을 보여주는 양방향LSTM-CRF 모델을 한국어 자동 띄어쓰기 문제에 적용하였다[9].

## 2. 음절 임베딩과 양방향 LSTM-CRF를 이용한 한국어 문장 자동 띄어쓰기

한국어 자동 띄어쓰기를 태그 열 부착 문제로 정의하고 띄어쓰기가 전혀 적용되지 않은 입력 문장의 각 음절을 띄어쓰기 태그 클래스(B 또는 I)로 분류하는 방식으로 자동 띄어쓰기 문제를 해결한다. 각 음절 분류가 끝난 후, B 태그 음절 앞에는 공백을 삽입하는 형태로 자동 띄어쓰기를 하였다. 띄어쓰기 태그 클래스 B(beginning)는 어절의 첫음절을 의미하고, I(inside)는 어절의 중간 혹은 마지막 음절을 나타낸다.

### 2.1. 음절 Unigram 및 Bigram 임베딩

자연어처리의 딥러닝 모델은 단어, 문장 등의 텍스트를 연속적인 벡터 공간에 표현하는 벡터 공간모델을 사

용하기 때문에 임베딩 방식은 해당 자연어처리 문제의 성능 향상에 영향을 끼치는 중요한 요소 중 하나로 word2vec, GloVe, FastText 등 다양한 임베딩 방식이 존재한다[11, 12, 13, 14]. [11, 12]는 언어의 문법적 특성보다는 구조적인 특성, [13]은 어절의 빈도수 정보를 가중치로, [14]는 [11, 12]방식에 어절의 음절 n-gram 정보도 함께 사용하여 단어 임베딩을 한다. 영어에서는 이러한 다양한 임베딩 접근 방식을 활용하여 문장 분류, 감정 분석 등에 활용하고 있다[15]. 이러한 임베딩 방식의 기본 임베딩 단위는 단어이므로 모든 단어를 연속적인 벡터 공간에 표현하기에는 어려움 있다. 또한, 학습되지 않은 단어 벡터를 처리해야하는 문제도 발생한다.

한국어는 음절 단위 조합으로 단어를 생성하고, 자주 사용되는 음절의 수는 한정되는 만큼 본 논문에서는 모든 단어를 벡터로 표현하는 것보다는 음절을 연속적인 벡터 공간에 표현하는 방식의 음절 임베딩(syllable embedding)을 사용하였다. 그림 1은 음절 임베딩을 위한 말뭉치 문장의 음절 Unigram 및 Bigram 사전을 구성하는 방법을 나타낸다.

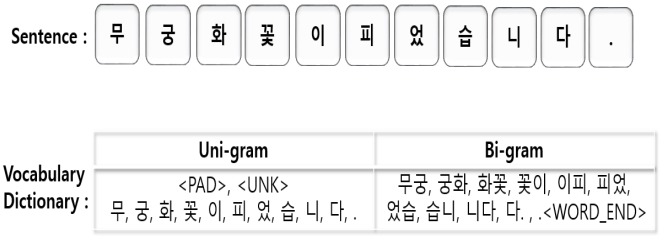


그림 1. 말뭉치 기반의 Unigram과 Bigram

### 2.2. 양방향 LSTM-CRF

딥러닝에서 문장은 순차적으로 나열된 단어, 음절 등을 하나의 자질로 볼 수 있다. 이러한 순차적인 자질 정보를 가진 데이터 분류에서 널리 사용되는 LSTM은 현재 입력 자질에 과거 자질과의 의존성을 부여한다. 하지만 양방향 LSTM은 과거뿐만 아니라 미래의 자질 정보도 함께 사용하는 특징이 있다[9].

선형체인 CRF는 태그 열을 예측하는데 있어서 두 가지 정보를 사용한다. 현재 입력 데이터의 태그 클래스의 점수와 이웃하는 태그들과의 최적의 태그 열을 구성하기 위한 전이 점수를 사용한다[9]. 예를 들어, “무궁화꽃이피었습니다.”라는 문장에 정답 태그 열은 “BIIBIBIIIII”이다. 이때, 선형체인 CRF는 “무궁화꽃이피었습니다.”라는 문장의 각 음절의 태그 클래스가 B인지 I 에 대한 지역적(local) 점수와 최적의 이웃하는 태그 열을 예측하는 전역적(global) 점수를 가지고 log likelihood를 계산하여 최적의 태그 열을 예측한다. 즉, 양방향 LSTM-CRF는 양방향 LSTM으로 각 음절에 양방향 정보와 결합된 자질 정보를 생성하고, 이 자질 정보로 선형체인 CRF는 입력 열에 대한 최적의 태그 열을 예측한다.

제안하는 모델은 그림 2와 같이 입력 문장을 음절 Unigram과 Bigram으로 연속적인 벡터 공간에 표현한 음절 벡터와 양방향 LSTM를 이용하여 음절 벡터를 새로운 자질 정보로 인코딩하고 전방향 신경망(feedforward neural network)을 이용하여 생성된 지역적 태그 점수와 선형체인 CRF를 이용하여 태그 열 부착을 수행하였다. 이때, 전방향 신경망은 각 음절에 태그 클래스 점수를 계산하기 위해 비선형 함수를 사용하지 않은 출력층 한 개만을 사용하였다.

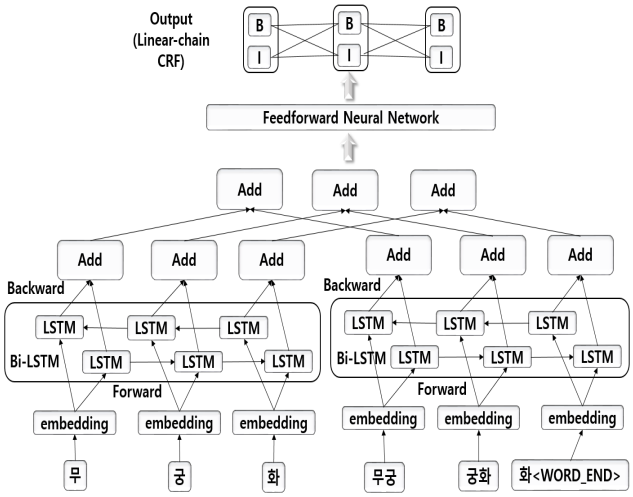


그림 2. 음절 Unigram과 Bigram을 이용한 양방향 LSTM-CRF

## 3. 실험 및 결과

자동 띄어쓰기 실험을 위한 말뭉치 데이터는 “차세정 언어처리 경진대회 2018”의 자동 띄어쓰기 태스크에서 제공하는 말뭉치를 사용했다. 말뭉치 크기는 15,000개 문장으로 308,825개 단어, 980,908개 음절로 구성되어 있다. 자동 띄어쓰기 학습 및 평가를 위해 15,000개 문장을 13,500개의 학습 문장, 1,500개 테스트 문장으로 구성하여 학습 및 평가를 수행하였다.

표 1. 자동 띄어쓰기 학습 및 테스트 데이터

| | Train | Test | Total |
|---|---|---|---|
| 문장 수 | 13,500 | 1,500 | 15,000 |
| 단어 (중복 단어 포함) | 277,718 | 31,107 | 308,825 |
| 음절 (중복 음절 포함) | 882,134 | 98,774 | 980,908 |

양방향 LSTM-CRF 모델은 텐서플로우[1]로 구현하였다. 표 2는 학습 및 평가를 위해서 음절 임베딩 종류, 양방향 LSTM의 전방향 셀과 후방향 셀의 출력 연산 종류, 임베딩 크기, LSTM 셀 유닛 크기 등을 다양하게 구성한 모델 종류를 나타내고, 각 모델들은 확률적 경사 하강법(stochastic gradient descent)으로 학습하였다.

각 모델의 성능 평가를 위해 공백 재현율(spacing recall), 띄어쓰기 태그 정확도(syllable accuracy), 어절 재현율(word recall), 어절 정확도(word

1) Tensorflow, Available: https://www.tensorflow.org/

precision), F1 score를 사용한다. 어절 재현율과 어절 정확도에서의 어절 기준은 공백으로 한다.

표 2. 모델 종류

| Model | Syllable Embedding | Bi-LSTM's Output Operation | Embedding and LSTM cell unit size |
|---|---|---|---|
| 1 | Unigram | Add | 250 |
| 2 | | | 300 |
| 3 | Unigram + Bigram | Add | 250 |
| 4 | | | 300 |

*Epoch: 5, 10, 15, 20, 25, 30, 35, 40
*Learning rate: 0.001
* Batch size: 1

자동 띄어쓰기 실험 결과는 표 3과 같다. 음절 Unigram 벡터만 사용한 자동 띄어쓰기의 음절 정확도보다 음절 Unigram과 음절 Bigram을 함께 사용한 경우가 97.337%의 성능을 보였다. 또한 어절 정확도, 어절 재현율, 공백 재현율, F1 Score에서도 음절 Unigram만 사용한 경우보다 음절 Unigram과 음절 Bigram을 함께 사용한 경우에 어절 정확도는 90.032%로, 어절 재현율은 89.526%, 공백 재현율 95.25%, F1 Score 89.779%의 성능을 보였다.

표 3. 자동 띄어쓰기 성능 평가 (단위: %)

| Model | Spacing Recall | Syllable Accuracy | Word Recall | Word Precision | Word F1 Score |
|---|---|---|---|---|---|
| 1 | 93.80 | 96.346 | 86.356 | 86.518 | 86.437 |
| 2 | 92.77 | 96.178 | 84.906 | 86.319 | 85.607 |
| 3 | **95.25** | 97.332 | **89.526** | **90.032** | **89.779** |
| 4 | 95.19 | **97.337** | 89.410 | 90.030 | 89.719 |

## 5. 결론

띄어쓰기가 전혀 적용되지 않은 한국어 문장의 자동 띄어쓰기 문제를 태그 열 부착 문제로 보고 문장의 각 음절을 자동 띄어쓰기 태그(B 또는 I)로 분류하기 위해 양방향 LSTM-CRF와 음절 임베딩을 이용하는 모델을 제안하였다. 성능 평가 결과로는 음절 벡터 사용 및 음절 Unigram 벡터만 사용하는 모델보다 음절 Unigram 벡터와 음절 Bigram 벡터를 함께 사용한 모델이 97.337%의 띄어쓰기 태그 분류 정확도를 보였다.

## 감사의 글

본 연구는 2017년도 정부(미래창조과학부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임 (No.NRF-2017M3C4A7068186).

## 참고문헌

[1] Kye Sung Kim, Hyun Ju Lee and Sang Jo Lee, "Three-Stage Word-Spacing System for Continuous Syllable Sentence in Korea," Journal of KISS(B): Software and Applications, Vol. 25, No. 12, pp.1838~1844, 1998.

[2] Kwangseob Shim, "Automatic Word Spacing Using Raw Corpus and a Morphological Analyzer," Journal of KIISE, Vol.42, No.1, pp.68~75, 2015.

[3] Kwangseob Shim, "Automatic Word Spacing based on Conditional Random Fields," KOREAN JOURNAL OF COGNITIVE SCIENCE, Vol.22, No.2, pp.217~233, 2011.

[4] Seung-Shik Kang, "Eojeol-Block Bidirectional Algorithm for Automatic Word Spacing of Hangul Sentences," Journal of KISS : Software and Applications, Vol.27, No.4, pp.441~447, 2000.

[5] Seung-Shik Kang, "Automatic Correction of Word-spacing Errors using by Syllable Bigram," Speech Sciences, Vol.8, No.2, pp.83~90, 2001.

[6] Lee, Do-Gil, Hae-Chang Rim, and Dongsuk Yook. "Automatic word spacing using probabilistic models based on character n-grams." IEEE Intelligent Systems, Vol.22, pp.28~35, 2007.

[7] Changki Lee, "Joint Models for Korean Word Spacing and POS Tagging using Structural SVM", 한국정보과학회 학술발표논문집, pp.604~606, 2013.

[8] Young, Tom, et al. "Recent trends in deep learning based natural language processing." arXiv preprint arXiv:1708.02709. 2017.

[9] Huang, Zhiheng, Wei Xu, and Kai Yu. "Bidirectional LSTM-CRF models for sequence tagging." arXiv preprint arXiv:1508.01991. 2015.

[10] Hyunsun Hwang and Changki Lee, "Automatic Korean Word Spacing using Deep Learning," 한국정보과학회 학술발표논문집, pp.738~740, 2016.

[11] Mikolov, Tomas, et al. "Distributed representations of words and phrases and their compositionality." In: Advances in neural information processing systems, pp.3111~3119, 2013.

[12] Mikolov, Tomas, et al. "Efficient estimation of word representations in vector space." arXiv preprint arXiv:1301.3781, 2013.

[13] Pennington, Jeffrey, Richard Socher, and Christopher Manning. "Glove: Global vectors for word representation." Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), pp.1532-1543, 2014.

[14] Bojanowski, Piotr, et al. "Enriching word vectors with subword information." arXiv preprint arXiv:1607.04606. 2016.

[15] Kim, Yoon. "Convolutional neural networks for sentence classification." arXiv preprint arXiv:1408.5882 (2014).