CNN과 LSTM을 이용한 한국어 형태소 분석 및 품사 결정의 정확도 향상

이용호○, 노정빈, 송민표, 신연순
동국대학교 컴퓨터공학과
yh1483@gmail.com, shwjdqls@gmail.com, 12songmp@gmail.com, ysshin@dongguk.edu

An Improvement The Accuracy of POS Tagging for Korean Using CNN-LSTM

Yongho Lee○, Jeongbin Noh, Minpyo Song, Younsoon Shin
Department of Computer Science and Engineering, Dongguk University

요 약

자연어 처리(Natural Language Processing)는 인공지능 분야에서 주목받는 분야로 정보검색, 챗봇(chatbot) 등에 주요하게 사용되고 있다. 그 중 형태소 분석 및 품사 결정(Part-of-Speech Tagging)은 자연어 처리의 첫 단계로 문장을 최소 단위로 쪼개는 역할을 한다. 하지만 불규칙적인 문법 요소와 모호성을 가지는 한국어 특성 상, 기존 형태소 분석 방법으로 높은 정확도를 기대하기는 어렵다. 본 논문에서는 기존 단어 임베딩 방식을 개선하고, 특징 벡터를 사용하여 CNN(Convolutional Neural Network)과 LSTM(Long Short-Term Memory) 모델에 학습시키는 한국어 형태소 분석기를 제안한다.

1. 서 론

자연어 처리[1]는 컴퓨터를 이용하여 인간의 언어 현상을 모사할 수 있도록 구현하는 인공지능의 주요 분야이다. 이는 정보 검색, QA 시스템, 문서 자동 분류, 신문기사 클러스터링, 챗봇 등에서 다양한 응용이 이루어지고 있다.

형태소 분석이란 자연어 처리의 첫 단계로 대상 어절을 최소의 의미 단위인 형태소로 분석하는 것을 의미한다. 본 단계에서 미등록어, 오탈자, 띄어쓰기 오류 등과 같은 OOV(Out of Vocabulary)에 의한 형태소 분석의 오류와 중의성, 신조어 처리 등의 문제를 보인다. 이와 같은 문제는 형태소 분석을 수행하는 것에 있어 치명적인 약점으로 볼 수 있다. 그 이유로 한국어는 종종 불분명한 띄어쓰기나 다양한 복합 유형 등에 따라 의미의 통합이나 분해가 상이한 양상을 보인다는 점과 이러한 형태소를 정확하게 분석하는 것이 매우 어렵다는 점에 있다.

본 논문에서는 형태소 분석의 정확도를 높이기 위하여 CNN-LSTM 모델을 이용한다. 또한, 단어들의 특징과 시간 정보를 활용하여 OOV가 포함된 문장을 정확하게 판단하는 형태소 분석기를 연구하였다. 본 과정에서 자연어 처리에서 전반적으로 사용되는 단어 임베딩(word embedding)의 방식을 개선함으로써 정확도 향상에 초점을 맞췄다.

2. 관련 연구

현재 형태소 분석기에 사용되는 학습 모델은 HMM(Hidden Markov Model), MEMM(Maximum Entropy Markov Model), CNN, LSTM 등이 있다. 제시한 모델들은 지도 학습(supervised learning)을 통해 형태소 분석된 데이터를 이용하여 머신러닝(machine learning) 모델을 학습한다.

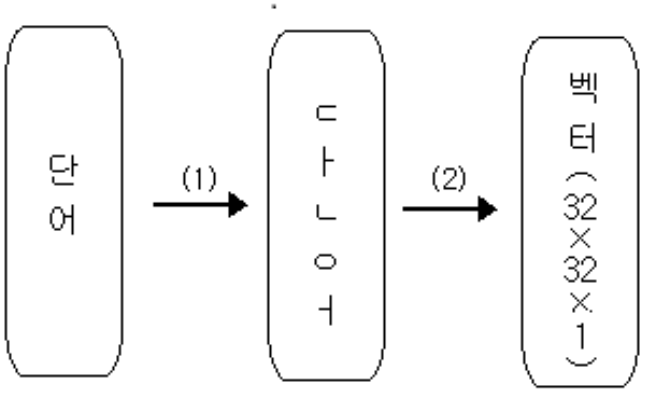
2.1 HMM / MEMM

HMM과 MEMM[2]은 통계적 마르코프 모델로, 시스템을 은닉된 상태와 관찰 가능한 결과로 분류한다. 한 시점에서 오직 하나의 상태와 출력을 가질 수 있으며 시점의 진행에 따라 상태를 전이한다. 그리하여 특정한 출력이 도출될 확률을 계산하거나, 특정한 출력에 대한 최적의 상태 전이 과정을 유추한다.

이 모델들은 단어장에 있는 단어에 대한 처리[3]는 정확하지만 한국어의 특성 상 불규칙적인 문법요소 때문에 OOV에 대한 형태소 분석은 정확하지 않다. 예를 들어, “왜 여기 있음?” 이라는 문장에서 “있음”을 이미 명사로 학습하였다면 이 문장에서도 명사로 인식할 것이다.

2.2 LSTM

LSTM[4]은 순차적으로 들어오는 데이터에 대해 학습한 정보를 오랫동안 유지할 수 있는 모델로, 입력 데이터를 시계열 데이터와 유사하게 처리하여 문장의 문맥적인 요



[그림 1] 단어 임베딩 과정

소까지 고려할 수 있다. 위의 과정에서 스트링 객체를 사용할 수 없기 때문에 학습 전에 단어 임베딩을 사용하여 스트링을 벡터로 변환하는 과정이 필요하다.

문맥적인 요소까지 포함하여 처리하기 때문에, 학습한 문장과 유사한 형태의 문장에 대해서도 정확한 형태소 분석이 가능하다. 예를 들어, “왜 여기에 있어?”라는 문장을 학습한 경우, 문맥적인 의미까지 학습하기 때문에 “왜 여기 있음?”과 같이 유사한 문맥을 가진 문장도 유사하게 결과가 출력된다. 하지만 문맥적인 정보를 가지고 있지 않은 오·탈자의 형태소 분석은 다소 부족하다.

3. 제안 모델

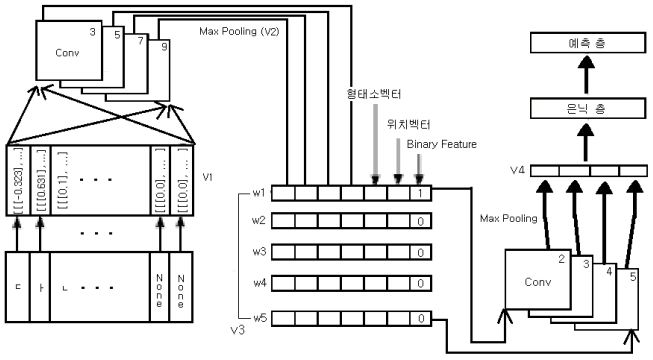
CNN-LSTM은 CNN과 LSTM 두 모델을 결합하여 사용한다. CNN에서 입력 값을 컨볼루션 망(convolution layer)을 이용하여 이미지의 특징을 추출하고, 추출된 데이터를 이용하여 LSTM에서 시계열 데이터와 유사하게 처리하여 문맥적인 요소까지 고려할 수 있도록 한다.

3.1 전처리 과정

본 모델을 학습시키기 위해서는 단어의 자소를 분리하고, 이를 벡터화하는 전처리 과정이 필요하다.[5] 본 논문에서는 문장 단위로 형태소 분석이 되어있는 텍스트 파일을 사용하여, 학습 모델에서 사용할 단어 벡터와 형태소 벡터를 구한다.

[그림 1]은 전처리 과정인 단어의 벡터화 과정을 보여준다. (1)에서 단어의 자소를 분리하는데 한국어의 경우 유니코드를 이용하여 각각의 자소를 초성, 중성, 종성으로 분류하고, 한국어가 아닌 경우에는 결과 값에 그대로 포함한다. 예를 들어, 문장에 ‘단어’라는 단어가 있다면 이를 ‘ㄷㅏㄴㅇㅓ’로 분리한다. ‘start지점’이라는 단어의 자소를 분리하면, ‘start’는 한국어가 아니므로 결과 값에 그대로 포함되고, ‘지점’은 자소가 분리되어 ‘ㅈㅣㅈㅓㅁ’이라는 결과를 얻는다. 따라서 최종 결과물은 ‘startㅈㅣㅈㅓㅁ’이 된다.

(2)에서는 위와 같이 자소가 분리된 단어를 Word2Vec(Word to Vector) 방법으로 학습 모델에서 사용할 수 있도록 32×32×1의 벡터로 변환한다. 각각의 자소를 순서대로 학습하고 생성된 자소 벡터들을 결합함으로써 단어 벡터를 생성한다. 자소의 개수가 32개가 넘는 단어는 단어의 32-gram들의 합을 단어의 벡터로 사용한다.



[그림 2] 학습 모델 과정

3.2 학습 모델

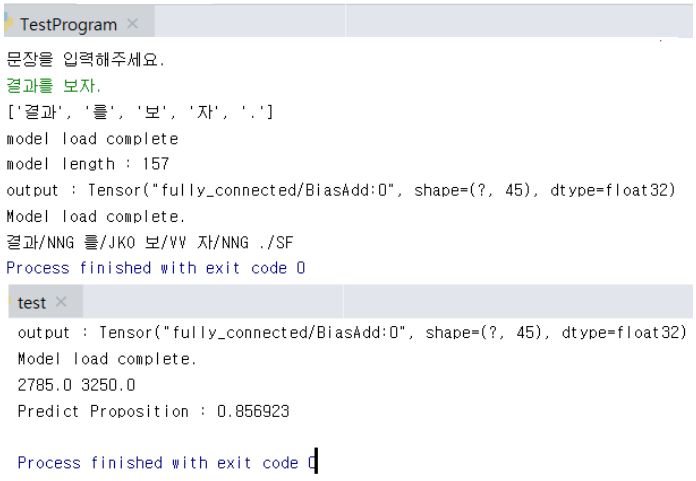
[그림 2]는 학습 모델의 구성이다. LSTM에서 5개의 고정된 크기의 벡터를 입력 데이터로 사용하기 때문에, 문장에서 단어의 개수를 5개로 제한한다. 단어 벡터인 v1이 필터 크기가 3, 5, 7, 9인 컨볼루션 망을 통과하면서 특징 벡터인 v2를 얻는다. 이를 통해 얻은 v2와 단어의 이전 형태소 정보를 갖는 형태소 벡터, 문장에서의 위치를 나타내는 위치 벡터, 그리고 단어의 타겟(target) 정보를 갖는 이진 특성값(binary feature)을 결합하여 벡터의 집합 v3를 만든다.

v3는 단어 구성 요소의 특징을 가지며 필터 크기가 2, 3, 4, 5인 컨볼루션 망을 이용하여 문맥적인 요소를 갖는 특징 벡터인 v4를 얻는다. 최종 단계로 v4를 결합하여 LSTM의 입력 데이터로 사용한다. LSTM을 통해서 나온 예측을 은닉 층의 입력 데이터로 재활용하여 이 단어가 이전 단어의 형태소가 나왔을 때의 예측 값을 통해 타겟의 태그에 대한 확률 벡터를 얻을 수 있다.[6] 각 단어의 결과 벡터 값과 세종말뭉치를 통해 얻은 그 단어의 품사에 대한 원핫벡터(one-hot vector)를 통하여 학습방향을 정한다.

4. 실 험

4.1 조 건

문법에 맞는 문장, 문법에 맞지 않는 문장, 오·탈자가 포함된 문장의 가공되지 않는 데이터(raw data)를 입력데이터로 사용하여 HMM, LSTM, CNN-LSTM의 정확도를 비교한다. 문법에 맞지 않는 문장은 문법에 맞는 문장의 문맥 순서를 변경하여 만들었고, 오·탈자가 포함된 문장은 문법에 맞는 문장을 추가, 중복, 삭제, 교차하여 만들었다. 예를 들어, 문법에 맞는 문장이 “나는 학교에 간다.”라는 문장일 경우, 문법에 맞지 않는 문장은 “학



[그림 3] 정확도 측정 결과

교에 나는 간다.”와 같이 문법에 맞는 문장의 문맥 순서를 변경하여 만든다. 오·탈자가 포함된 문장은 “난는 학교에 간다.”와 같이 문법에 맞는 문장에서 일부 자소를 추가 하거나, “나느 학교에 간다.”와 같이 일부 자소를 삭제하여 문장을 만든다.

4.2 실험 결과

실험은 세종말뭉치에서 제공하는 한글 문장을 태그별로 나누어 둔 파일을 사용하였고, 문법에 맞지 않는 문장과 오·탈자가 포함된 문장은 세종말뭉치의 문장에서 임의로 변경하여 생성하였다. [그림 3]은 이들을 사용하여 나온 결과가 세종말뭉치에 존재하는 태그와 일치하는지 판단하여 정확도를 측정하는 과정이다.

[표 1]은 각 모델별 문장에 따라 정확도를 비교한 표이다. 문법에 맞는 문장의 경우, HMM의 정확도는 85.06%, LSTM의 정확도는 84.70%, CNN-LSTM의 정확도는 85.66%이며, 문법에 맞지 않는 문장의 경우, HMM의 정확도는 81.21%, LSTM의 정확도는 80.93%, CNN-LSTM의 정확도는 83.02%이다. 오·탈자가 포함된 문장의 경우는 HMM의 정확도는 77.95%, LSTM의 정확도는 83.73%, CNN-LSTM의 정확도는 84.94%로 모든 경우에서 CNN-LSTM이 정확도가 가장 높은 것을 확인할 수 있다.

4.3 결과 분석

CNN-LSTM이 HMM, LSTM보다 문법에 맞는 문장의 경우 0.60%, 0.96%, 문법에 맞지 않는 문장의 경우 1.81%, 2.09%, 오·탈자가 포함된 문장의 경우 6.99%, 2.21% 만큼 높은 정확도를 보여주었다.

CNN-LSTM은 단어의 정보를 최대한 활용하기 위해 필터가 다른 여러 컨볼루션 망을 사용하여 벡터를 생성한다. 이 벡터로 학습이 이루어지기 때문에 높은 정확도를 얻을 수 있다. 또한 오·탈자가 포함된 단어의 경우에도 필터를 통한 벡터 값이 원단어의 벡터 값과 유사한 형태를 가지기 때문에 높은 정확도를 얻을 수 있었다.

모델	HMM	LSTM	CNN-LSTM
문법에 맞는 문장	85.06%	84.70%	85.66%
문법에 맞지 않는 문장	81.21%	80.93%	83.02%
오·탈자가 포함된 문장	77.95%	82.73%	84.94%

[표 1] 각 모델별 정확도 비교 (오차 ±0.05%)

5. 결론 및 향후 과제

본 논문에서는 CNN-LSTM 모델을 사용하여 OOV를 더 정확하게 처리하는 형태소 분석기 구현 방안을 제안한다. 실험 결과 OOV를 가진 문장뿐만 아니라 OOV를 가지고 있지 않은 문장에 대해서도 형태소 분석의 정확도가 0.66%에서 6.99%까지 개선되었다.

NLP 분야에서 한국어는 불규칙적인 문법 요소와 모호성 때문에 영어에 비해 뒤처지고 있는 실정이다. 한국어의 형태소 분석이 발전한다면 답보 상태에 있는 한국어의 자연어 처리에서 큰 발전을 기대할 수 있다.

향후 과제로는 LSTM 모델은 문장 길이가 길고 층이 깊으면 인코더(encoder)가 압축해야 할 정보가 많아져서 정보 손실이 일어나 형태소 분석의 정확도가 떨어지는 문제점이 있다. 이를 해결하기 위해 LSTM을 앞에서 뒤, 뒤에서 앞을 모두 고려하는 양방향(bidirectional) 네트워크인 Bi-LSTM(Bidirectional LSTM) 모델로 대체한다면 많은 형태소를 가진 단어가 존재하는 문장의 형태소 분석이 더욱 정확해질 것이다.

감사의 글

“본 연구는 과학기술정보통신부 및 정보통신기술진흥센터의 SW중심대학지원사업의 연구결과로 수행되었음”(2016-0-00017)

참 고 문 헌

[1] Tom Young, Devamanyu Hazarika, Soujanya Poria, Erik Cambria. Recent Trends in Deep Learning Based Natural Language Processing, 2017

[2] Scott M. Thede, Mary P. Harper. A Second-Order Hidden Markov Model for Part-of-Speech Tagging, 1999

[3] 이재성, 한국어 형태소 분석을 위한 3단계 확률 모델, 2011

[4] Zazo R1, Lozano-Diez A1, Gonzalez-Dominguez J1, Toledano DT1, Gonzalez-Rodriguez J1, Language Identification in Short Utterances Using Long Short-Term Memory (LSTM) Recurrent Neural Networks, 2016

[5] 강승식, 김영택, 사전 정보에 기반한 효율적인 한국어 형태소 분석기의 설계 및 구현, 1991

[6] Longlu Qin, POS tagging of Chinese Buddhist texts using Recurrent Neural Networks, 2015