한국어 특질을 고려한 단어 벡터의 Bi-LSTM 기반 개체명 모델 적용

남석현O, 함영균, 최기선

한국과학기술원
obiwan96@kaist.ac.kr, hahmyg@kaist.ac.kr, kschoi@kaist.ac.kr

Application of Word Vector with Korean Specific Feature to Bi-LSTM model for Named Entity Recognition

Sukhyun NamO, Younggyun Hahm, Key-Sun Choi
KAIST

Deep learning의 개발에 따라 개체명 인식에도 neural network가 적용된 연구가 활발히 일어나고 있다. 영어권 개체명 인식에서는 F1 score 90%을 웃도는 성능을 내는 연구들이 나오고 있다. 하지만 한국어는 영어와 언어적 특질이 많이 달라 이를 그대로 적용시키는 데는 어려움이 있어 영어권 개체명 인식기에 비해 비교적 낮은 성능을 보인다. 본 논문에서는 "하다" 접사의 동사형이 보존된 워드 임베딩을 사용하고 한국어 개체명의 특징을 담은 one-hot 벡터를 추가하여 한국어의 특질에 보다 적합한 데이터를 deep learning 기술에 적용하였다.

주제어: 개체명 인식

1. 서론

개체명 인식(Named Entity Recognition)은 문서로부터 개체명(Named Entity)을 추출하고, 추출된 개체명의 종류를 분류하는 자연언어처리의 한 분야이다. 개체명은 문서에서 나타나는 고유한 의미를 가지는 명사나 숫자 표현 등으로, 대체로 하나 이상의 단어가 결합되어 구성된다. 본래는 시간이나 수식 표현까지 포함하는 포괄적인 개념이나, 본 연구에서는 인명(PS), 지명(LC), 기관명(OG)의 고유명사와 숫자(DT), 시간(TI)을 대상으로 한정한다. 표 1은 개체명이 표시된 문장의 예시이다.

'<부천 영화제 사무국:OG>'은 <내달 3일:DT> <오후 3시30분~5시30분:TI> <부천:LC> <중동:LC>신도시 <중앙공원:LC>에서 'PiFan 사랑 걷기 대회'를 개최한다.

표 1. 개체명이 표시된 문장 예

개체명 인식에 대한 연구는 1995년, MUC-6(the Sixth Message Understanding Conference)[1]에서 처음 촉발되었다. MUC-6에 참가한 많은 시스템들은 특정 언어에 제한된 규칙과 제한된 입출력방법을 사용하여 다른 언어나 시스템에 적용하지 못하는 문제가 있었으나, 이후 기계학습 방식과 BIO 태깅의 통일된 입출력방식을 도입하여 체계적으로 연구되어오고 있는 분야이다. 이 때, BIO태깅이란 개체명의 시작을 "B"로, 개체명이 이어지고 있는 경우에는 "I"로, 개체명이 아닌 경우에는 "O"로 태깅하는 방식을 일컫는다. 정보 검색, 질의응답 시스템 등 매우 다양한 분야의 시스템에서도 개체명 인식이 사용됨에 따라 성능향상을 목표로 현재까지도 연구되어오고 있다.

최근에는 자연언어처리뿐만 아니라 많은 분야에서 괄목할 만한 성과를 보여주고 있는 딥러닝 기술의 개발로 개체명 인식에서도 이를 이용한 연구가 진행되고 있다. 문장 내 단어의 의미를 분산된 에너지 벡터로 표현할 수 있는 연구인 단어 임베딩(word embedding) 방법론의 개발로[2] 단어를 벡터화 시킨 후 deep neural network를 적용시킬 수 있게 되었고, 상당한 성능을 보여주고 있다. 영어의 경우에는 딥러닝을 이용하여 90%를 웃도는 높은 정확도를 보이는 연구들이 나오고 있다[3][4]. 하지만 한국어는 언어적 특질이 영어와 달라 영어 개체명 인식에서 쓰인 기술을 그대로 적용하기에는 어려움이 있고 따라서 영어권 개체명 인식에 비해 다소 낮은 성능을 보이고 있다. 또한 최근 한글 개체명 인식에도 딥러닝 기술을 적용한 연구가 나오고 있다[5]. 이에 본 논문에서는 한국어의 특징을 분석하여 딥러닝 기술을 한국어에 알맞게 적용시켜 더 높은 성능을 보이고자 하였다.

본 논문에서는 기존의 BiLSTM을 이용한 영어 개체명 인식 모델[6]을 구현하고 이에 대한 다음의 추가 작업을 통해 성능이 더 향상될 수 있음을 보였다.

1) "하다" 접사의 동사형이 보존된 단어 임베딩 사용
2) 한국어 개체명의 특징을 담은 one-hot 벡터 추가
3) 평가데이터 오류 수정

2. 개체명 학습 모델

본 논문에서는 딥러닝의 다양한 모델 중에서도 최근의 영어권과 한국어 개체명 인식에서 모두 가장 높은 성능을 보이고 있는 BiLSTM을 기반으로 하여 CRF방식을 결합한 BiLSTM-CRF 방식을 이용하였다. 그림 1은 BiLSTM을 사용한 개체명 인식기의 흐름도를 보여주고 있다.

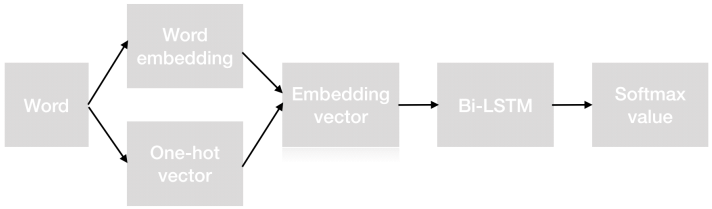

그림 1. 개체명 인식 BiLSTM 흐름도

개체명 인식은 POS 태깅이 완료된 데이터를 대상으로 이루어진다. POS태깅이 되어있는 데이터 셋에 대해 각 단어를 워드 임베딩과 one-hot벡터를 합친 임베딩 벡터로 변환시킨다. 학습데이터의 임베딩 벡터를 BiLSTM을 통해 학습시켜 BiLSTM 모델을 생성한다. 학습된 모델에 테스트 데이터의 임베딩 벡터를 입력하여 나온 벡터를 softmax 함수에 적용시켜 개체명 인식 결과를 얻을 수 있다.

2.1 “하다” 접사의 동사형이 보존된 워드 임베딩 사용

본 논문에서는 한국어 위키피디아를 대상으로 skip-gram모델의 word2vec[7]을 사용하였다. 벡터 값의 차원은 100, 300, 500으로, 창크기는 3, 5로 변화시키며 어떤 환경에서 생성된 워드 임베딩 데이터를 이용하였을 때 성능이 가장 높은지 확인하여보았다. 생성된 데이터는 약 26만여 개의 단어를 포함한다.

기존의 한국어 자연언어 처리에서 워드 임베딩을 활용하는 연구들은 워드 임베딩을 생성하면서 형태소 분석을 한 후 생성하였다. 그 때문에 동사표현인 ‘출생하’를 예로 들면 ‘출생/NNG’와 ‘하/XSV’의 각각의 임베딩을 생성하는 연구가 이루어졌다[5][8]. 본 논문에서는 개체명 인식에 있어서는 개체명 앞 혹은 뒤에 동사가 있는 것이 중요한 것으로 판단하여, 이러한 경우 ‘출생하/VV’에 대한 임베딩을 생성하는 차이를 두었다.

이러한 워드 임베딩 생성 방식에 차이를 두었을 때 명사형을 가져왔을 때 발생하는 오차를 없앨 수 있다. 예를 들어 문장에서 ‘조정하다’와 같은 경우 ‘조정’이 ‘조정하’의 동사적 의미로 쓰인 것인데, ‘조정/NNG’의 워드 임베딩을 가져오게 되면 운동 종목인 조정의 워드 임베딩을 가져와 오류가 발생한다. 표 2에 명사로 분석을 하게 되면 뜻이 달라지는 단어들을 평가 데이터에서 찾아 일부를 예시로 나타내었다. “하다” 접사의 동사형이 보존된 워드 임베딩을 사용하면 이러한 오류를 없앨 수 있으며, 이에 대한 성능 비교를 3장에 수행하였다. 2016 국어 정보 처리 경진대회에서 제공한 워드임베딩도 위키피디아를 대상으로 워드 임베딩을 제작하였기 때문에 성능 비교 대상으로 사용하였다.

지적, 자부, 주도, 구가, 전역, 투구, 선사, 대패, 구상, 주도, 대신

표 2. 명사형으로 분석하였을 때 동사형과 의미가 달라지는 단어의 예시

임베딩 데이터를 사용함에 있어서 워드 임베딩 대상 corpus에 존재하지 않아 embedding vector가 존재하지 않는 단어에 대해서 같은 차원의 벡터를 임의의 값으로 채워서 사용하는 경우가 많다[6]. 본 연구에서는 임의의 값으로 채운 백터를 사용하여보기도 하였고, 일관적으로 같은 차원의 zero vector를 사용하여보기도 하였다.

2.2 한글 one-hot 벡터

영어의 경우, 워드임베딩 이외에도 추가적인 자질로서 품사태그 정보와 대문자로 시작하는지 등을 워드 벡터에 포함시켰을 때 성능이 향상되는 연구가 이루어졌다[4][6]. 이 때, 영어의 경우 개체명은 대체로 대문자로 시작하기 때문에 이러한 정보가 개체명 인식에 굉장히 도움이 된다. 이에 따라 한국어의 경우에도 한국어의 자질을 워드 벡터에 포함할 필요성이 있다. 본 논문에서는 한국어 개체명의 특징을 담고 있는 one-hot 벡터를 생성하였고, 이를 개체명 모델에 적용하여 비교평가를 수행하였다.

데이터 셋에서 개체명으로 태깅된 단어들의 특징을 분석하여서 공통적으로 어떤 특징이 있는지 분석하여보았다. 특히 한국어는 어원이 한자인 이루어진 단어가 많아, 개체명에 공통적으로 포함되어있는 글자가 많다. 다음은 각 개체명 종류 별로 이를 분석한 것이며, 괄호 안은 평가 데이터 셋에서 해당 조건을 만족하는 어휘가 개체명일 확률을 의미한다.

LC - ‘국’(71%), ‘동’(41%), ‘구’(35%), ‘시’(34%), ‘도’(19%)로 끝남
OG - ‘팀’(26%), ‘회’(35%)로 끝남
DT - ‘지나’(90%), ‘올해’(97%), ‘월’(98%), ‘일’(97%), ‘년’(94%)을 포함
TI - ‘오전’(100%), ‘오후’(100%), ‘분’(72%), ‘시’(49%)를 포함

이때, 50%가 넘는 특징들만 one-hot 벡터의 특징으로서 사용하였다. 다음은 그 결과 선택된 특징들이다.

- 글자 수가 셋 이상이다.
- 한글이 아닌 숫자로 이루어져있다.
- ‘지나’, ‘올해’, ‘월’, ‘일’, ‘년’을 포함한다.
- ‘국’으로 끝난다.
- ‘오전’, ‘오후’, ‘분’을 포함한다.

또한 품사태그 정보를 확인하여 일반명사인 NNG와

의존명사인 NNB를 묶어서 일반 명사, 외래어인 SL, 고유명사인 NNP, 동사인 VV, 그 외의 5가지로 나누어 vector 정보를 추가하여주었다. 또한 괄호 속에 들어가있는 단어들도 이에 대한 정보를 표기하여 총 11차원의 one-hot vector를 생성하였다. One-hot 벡터는 워드 임베딩 벡터의 끝에 추가하여서 최종 임베딩 벡터를 생성하였다.

2.3 BiLSTM

BiLSTM은 순차적 데이터 활용에 있어서 가장 많이 쓰이는 딥러닝 모형인 LSTM을 두 개를 함께 학습시켜, 각 데이터에 대해 왼쪽(forward)뿐만 아니라 오른쪽(backword) 데이터를 고려하도록 보완한 모델이다. 특히나 앞뒤 문맥을 모두 고려해야하는 자연언어처리에서 높은 성능을 보이는 알고리즘이다. BiLSTM 구현에는 구글에서 오픈소스로 공개한 기계학습 라이브러리인 텐서플로우(TensorFlow)[9]를 활용하였다. Hidden dimension이 256이고 drop out을 0.5로 설정한 LSTM을 2 layer로 생성하여 backword, frontword cell로 설정하였다. Optimizer로는 Adam optimizer을 사용하였으며, batch size는 128로 설정하여 생성된 BiLSTM을 train data로 학습시키며 매번 학습된 BiLSTM으로 test data에 적용시켜 인식결과를 확인하였다. 매번 F1 score를 측정하여 만약 최대치를 기록하면 학습된 BiLSTM을 저장시키는 방식으로 50 epoch 학습시켰다.

2.4 CRF

개체명 인식을 실제로 사용하기 위해서는 각각의 형태소를 태깅하는 것만으로는 부족하다. BIO태깅을 하기 위해서는 하나의 개체명을 추출해낼 수 있어야하는데, '연세대학교 원주캠퍼스'를 예로 들었을 때, 연세대학교와 원주캠퍼스를 같은 개체명으로 인식하여 '연세대학교 B-OG 원주캠퍼스 I'로 태깅해야 한다. BiLSTM은 각각의 형태소를 태그 벡터로서 출력하며 LC, PS, OG, DT, TI, O 중 어느 것인지 결정할 뿐 하나의 개체명이 무엇인지 알아내지 못하는 문제가 있다. 이 때문에 CRF를 추가할 필요성이 있었다. CRF에는 CRF++ Tool[10]을 활용하였다.

2.5 개체명 사전 활용

태깅 결과를 분석한 결과 개체명의 boundary를 제대로 찾지 못해 틀린 경우가 전체 오류의 6.1%를 차지했다. 이에 개체명 사전을 활용하여 이미 찾은 개체명의 boundary를 확정짓는 함수를 추가하였다. 개체명 사전의 데이터 베이스는 [11]에서 사용한 데이터 베이스를 활용하였다. 개체명인 단어의 다음 단어가 명사(NN)이며 개체명이 아닌 경우, 개체명에 다음 단어를 포함시켜 개체명 사전에 검색하였을 때 존재한다면 그 단어도 개체명에 포함시키도록 하는 함수를 추가하여주었는데, F1 score 기준 추가하기 전보다 1%낮아져 효과가 없었다.

3. 성능 평가 및 분석

3.1 성능 평가

데이터 셋은 2016 국어 정보 처리 경진대회[12]에서 제공한 데이터 셋을 활용하였다. 학습 데이터는 3,555문장으로 이루어져있으며, 실험 데이터는 501문장으로 이루어져있다.

3.1.1 성능 평가 방식

성능 평가 방식은 두 가지를 이용하였다. 첫 번째는 각각의 형태소에 대해 개체명 태깅이 맞았는지를 확인하여 F1 score를 측정하는 방법이다 (3.1.2절과 3.1.3절). 두 번째는 찾아낸 개체명과 정답 데이터 셋을 비교하여 개체명을 정확히 찾아내었는지 확인하여 F1 score를 측정하는 방식이다 (3.1.4절). 두 번째 방식은 2016 국어 정보 처리 경진대회에서 채점에 사용된 프로그램을 그대로 사용하였다.

워드 임베딩 데이터		F1(%)
2016 국어 정보 처리 경진대회 워드 임베딩		76.88
"하다" 접사의 동사 표현을 보존한 워드 임베딩	300차원, 창 크기 3	**83.82**
	300차원, 창 크기 5	83.11
	500차원, 창 크기 3	83.64
	500차원, 창 크기 5	83.29
	100차원, 창 크기 3	83.26
	300차원, 창 크기 3 + one-hot 벡터	**84.19**

표 3. 워드 임베딩 데이터 별 개체명 인식기 성능

3.1.2 워드 임베딩에 따른 개체명 인식 성능

위키피디아로부터 워드 임베딩 데이터를 생성할 때 차원과 창 크기를 변화시켜가며 성능비교를 하여보았다. 또한, 성능 평가를 위하여 2016 국어 정보 처리 경진대회에서 함께 제공한 워드 임베딩 데이터를 이용하였을 때와도 비교를 하여보았다. 결과는 표3과 같다.

대체적으로 2016 국어 정보 처리 경진대회 워드 임베딩 데이터를 이용했을 때보다 6% 가량 높아 훨씬 좋은 성능을 보임을 알 수 있다. 또한 300차원과 창 크기는 3일 때 성능이 가장 높아 해당 워드 임베딩을 이후로도 사용하였다.

3.1.3 One-hot벡터 및 워드 임베딩 외의 벡터

3.1.1의 결과에서는 one-hot 벡터를 추가하지 않고 임베딩 벡터를 생성할 때 워드 임베딩에 없는 단어들에 대해서는 같은 차원의 랜덤 벡터를 생성하여 사용하였다. 한국어 개체명의 특징을 분석하여 생성한 one-hot 벡터를 추가하였을 때 F1 score가 84.19%로 소량 증가하였다. 워드 임베딩에 없는 단어들에 대하여 랜덤 벡터가 아닌 zero 벡터로 대체를 한 경우는 84.73%로 가장 높은 성능을 기록하였다.

3.1.4 CRF

BiLSTM의 결과는 각각의 형태소가 어느 개체명에 포함되는지 벡터로 출력된다. CRF를 추가하였을 때와의 성능 비교를 위해 같은 개체명이 연속해서 태깅되면 같은 개체명이라 정하여 json 데이터를 형성하여 평가하였다.

CRF를 사용하지 않았을 때 66.2%, CRF를 사용하였을 때 72.8%로 성능이 향상되었음을 확인하였다..

3.2 오류 분석

3.1.3에서 방식1의 F1 score가 84.73%일 때 기준으로 테스트 데이터 셋에 있는 17,394개의 단어 중 642개의 단어에 대해 태깅이 잘못 된 것을 확인하였다. 이를 분석해본 결과, 오류를 표 4와 같이 4가지로 분류할 수 있었다. 전체 오류 중 나타난 빈도와 예시도 함께 표기하였다.

구분	예시
개체명의 경계 인식이 잘못된 경우(5.6%)	'20세기 폭스사'는 전체가 하나의 기관명인 개체명인데, 폭스사만 OG로 태그.
지역명과 기관명을 혼용한 경우(6.1%)	'홍대' 같은 경우는 대학교로서의 기관명일 수도 있고, 지역으로서의 홍대를 나타낼 수도 있다. '홍대'가 학습데이터 셋에서는 LC로 태그되어 있는데 OG로 태그.
데이터 셋의 오류로 인해 발생한 경우(9.8%)	'경찰'의 경우, 데이터 셋에서 'OG'라고 태그되어 있는 경우도 있고 'O'로 태그 되어 있는 경우도 있다.
그 외(78%)	'투수진'을 사람으로 태그 하는 등 개체명 태그 자체의 오류

표 4. 오류 분류 및 예시

3.2.1 데이터 셋 수정

데이터 셋의 오류로 인해 성능이 낮게 측정되는 경우가 예상보다 많아 테스트 데이터에서 명확하게 데이터 셋의 오류로 판단되는 사례를 수정하였으며 그 경우는 표 5와 같다. 수정한 데이터 셋을 이용하여 성능을 측정하여본 결과 86.3%을 기록하여 기존보다 1.6% 향상되었다. 수정된 전체 데이터는 총 41사례로 [13]에서 공개하였다.

구분	변경 사례
개체명 태그 누락된 경우 수정	전: '<LG:OG>는 <7일:DT> **잠실구장**에서 계속된' 후: '<LG:OG>는 <7일:DT> **<잠실구장:LC>**에서 계속된'
품사태그 수정	전: '태 NNP 어 NNP 난 NNP' 후: '태어나 VV ㄴ ETM'
잘못된 개체 인식	전: '비디오점 <체인 씨네타운:OG>이' 후: '비디오점 체인 <씨네타운:OG>이'

표 5. 테스트 데이터 노이즈 제거 작업 사례

4. 결론 및 향후 과제

본 논문에서는 한국어 개체명 인식의 성능 향상을 위해 기존의 BiLSTM을 이용한 영어 개체명 인식 시스템을 구현한 후 "하다" 접사의 동사형이 보존된 자체 워드 임베딩을 제작, 한국어 특징에 맞는 one-hot 벡터 추가를 통하여 성능이 더 향상되었음을 보였다. 하지만 개체명의 경계를 정확하게 찾지 못해 개체명을 잘못 태깅하는 경우가 있어 사전을 이용하여 개체명의 경계를 정확하게 찾는 시도를 하여보았으나, 오히려 성능을 낮추는 결과를 보였다. 개체명 사전을 이용하여 BiLSTM의 결과에 나온 개체명의 경계 인식 연구가 향후 진행된다면 한글 개체명 인식의 성능을 더욱 효과적으로 증대시킬 수 있을 것으로 보인다.

사사

본 연구는 과학기술정보통신부 및 정보통신기술진흥센터의 SW중심대학 지원사업의 연구결과로 수행되었음(2016-0-00018)

참고 문헌

[1] http://cs.nyu.edu/faculty/grishman/muc6.html

[2] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, Jeffrey Dean, "Distributed Representations of Words and Phrases and their Compositionality", In Advances on Neural information Processing Systems, 2013.

[3] Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, Chris Dyer, "Neural Architectures for Named Entity Recognition", 2016.

[4] Jason P.C. Chiu , Eric Nichols "Named Entity Recognition with Bidirectional LSTM-CNNs", 2015.

[5] 나승훈, 민진우, "문자 기반 LSTM CRF를 이용한 개 체명인식",한국컴퓨터종합학술대회논문집,2016.

[6] Vinayak Athavale, Shreenivas Bharadwaj, Monik Pamecha, Ameya Prabhu, Manish Shrivastava, "Towards Deep Learning in Hindi NER: An approach to tackle the Labelled Data Scarcity", 2016

[7] https://code.google.com/archive/p/word2vec

[8] 최윤수, 차정원, "Word Embedding 자질을 이용한 한국어 개체명 인식 및 분류", 2016

[9] https://www.tensorflow.org

[10] https://taku910.github.io/crfpp/

[11] Jeong-Uk Kim, "KoEL: Korean entity linking system using sentence features and extended entity relation", KAIST, Master Thesis, 2017

[12] https://sites.google.com/site/2016hclt

[13]https://github.com/machinereading/KoreanNERCorpus