품사 임베딩과 음절 단위 개체명 분포 기반의 Bidirectional LSTM CRFs를 이용한 개체명 인식

유홍연○, 고영중
동아대학교 컴퓨터공학과
{hongyeon1408, youngjoong.ko}@gmail.com

Named Entity Recognition Using Bidirectional LSTM CRFs Based on the POS Tag Embedding and the Named Entity Distribution of Syllables

Hongyeon Yu○, Youngjoong Ko
DongA University, Department of Computer Engineering

요 약

개체명 인식이란 문서 내에서 인명, 기관명, 지명, 시간, 날짜 등 고유한 의미를 가지는 개체명을 추출하여 그 종류를 결정하는 것을 말한다. 최근 개체명 인식 연구에서는 bidirectional LSTM CRFs가 가장 우수한 성능을 보여주고 있다. 하지만 LSTM 기반의 딥 러닝 모델은 입력이 되는 단어 표상에 의존적이기 때문에 입력이 되는 단어 표상을 확장하는 방법에 대한 연구가 많이 진행되어지고 있다. 본 논문에서는 한국어 개체명 인식을 위하여 bidirectional LSTM CRFs모델을 사용하고, 그 입력으로 사용되는 단어 표상을 확장하기 위해 사전 학습된 단어 임베딩 벡터, 품사 임베딩 벡터, 그리고 음절 기반에서 확장된 단어 임베딩 벡터를 사용한다. 음절 기반에서 단어 기반 임베딩 벡터로 확장하기 위하여 bidirectional LSTM을 이용하고, 그 입력으로 학습 데이터에서 추출한 개체명 분포를 이용하였다. 그 결과 사전 학습된 단어 임베딩 벡터만 사용한 것보다 4.93%의 성능 향상을 보였다.

주제어: 개체명 인식, 단어 표상, 음절, bi-LSTM-CRFs

1. 서론

개체명(Named Entity)이란 문서 내에서 인명, 기관명, 지명, 시간, 날짜 등 고유한 의미를 가지는 단어를 말한다. 이러한 개체명을 문서로부터 추출하여 개체명의 종류를 결정하는 것이 개체명 인식(Named Entity Recognition)이다[1].

최근 개체명 인식 연구에서는 순차 레이블링(Sequence Labeling)에 특화된 Long Short-Term Memory(LSTM)기반의 bidirectional LSTM Conditional Random Fields(bi-LSTM-CRFs) 모델이 가장 우수한 성능을 보이고 있다. bi-LSTM-CRFs는 LSTM 출력 계층(Output Layer)의 개체명 태그 사이에 전이 확률을 추가하고, 입력 문자열을 양방향으로 받는 모델이다[2,3].

LSTM 기반의 모델은 각 단어를 잘 표현하는 단어 임베딩 벡터(word embedding vector)를 입력으로 받기 때문에, 단어 표상(word representation)에 의존적이다. 따라서 이러한 단어 표상을 잘 만들기 위한 많은 연구들이 진행되고 있다. 대표적인 방법으로는 대량의 말뭉치를 이용하여 사전 학습된(pretrained) 단어 임베딩 벡터를 활용하거나[2-4], 단어를 구성하고 있는 문자들의 임베딩 벡터(Character embedding vector)로부터 단어 임베딩 벡터를 유도해내는 방법들이 연구 되고 있다. 이러한 유도 방법 들은 LSTM과 CNN을 사용하는 방법들이 있으며, 최근에 가장 좋은 성능을 보여주고 있다[5-10].

따라서 본 논문에서는 한국어 개체명 인식을 위해 bi-LSTM-CRFs를 이용하고, 입력으로 사용되는 단어 표상을 확장하기 위해 사전 학습된 단어 임베딩 벡터, 품사 임베딩 벡터, 그리고 음절 기반에서 확장된 단어 임베딩 벡터를 사용한다.

단어 표상 확장을 위해 첫 번째로 사전 학습된 단어 임베딩 벡터가 사용된다. 대부분의 개체명은 미등록어이기 때문에 학습 데이터에 나오지 않은 개체명을 잘 분류하는 것에는 한계가 있다. 따라서 [4]에서 사용한 것과 같이 대량의 말뭉치

이 논문은 2015년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(No. NRF-2015R1D1A1A01056907)

를 이용하여 단어 임베딩 벡터를 사전 학습하고, 단어 집합을 확장시킨다.

두 번째로는 사전 학습된 품사 임베딩 벡터를 사용하여 단어 표상을 확장한다. 개체명 인식에서는 품사의 시퀀스 또한 중요하기 때문에 품사를 잘 표현하는 임베딩 벡터가 중요하다. 품사 임베딩 벡터를 사전 학습하기 위해 대량의 말뭉치를 형태소 분석을 하고, 단어를 삭제한 뒤 품사만 학습하여 사용한다.

세 번째는 음절 기반 임베딩 벡터로부터 단어 기반 임베딩 벡터로 유도하여 단어 표상을 확장하는 방법이다. 임베딩 벡터 확장을 위한 입력인 음절 임베딩 벡터로는 각 음절별로 학습 코퍼스에 나온 개체명의 분포를 벡터로 만들어 사용하였다.

사전 학습된 단어 임베딩 벡터에 품사 임베딩 벡터와 음절 기반 단어 임베딩 벡터를 확장한 결과 4.93%의 성능 향상을 얻었으며, 성능 평가를 위한 개체명 말뭉치로는 2016년 국어 경진대회에서 배포한 4,056문장을 사용하였다.

본 논문의 구성은 2장에서 관련 연구를 소개하고, 3장에서는 bi-LSTM-CRFs모델과 단어 표상 확장 방법을 소개한다. 4장에서는 확장된 단어 표상의 학습 결과를 분석하고, 마지막 5장에서 결론에 대해 기술한다.

2. 관련 연구

개체명 인식에서는 기계학습 방법이 많이 이용되어 왔으며, 주로 HMM(Hidden Markov Model), SVM(Support Vector Machine), CRFs(Conditional Random Fields), Structural SVM 등이 있다[1,2]. 하지만 최근 들어 딥러닝(Deep learning) 기법 중 순차 레이블링에 특화 되어있는 LSTM 기반 방법인 bi-LSTM-CRFs가 개체명 인식에서 가장 우수한 성능을 보여주고 있다[2-3,5-8].

LSTM 기반 방법은 입력되는 단어 표상에 의존적이기 때문에 단어 표상을 확장하는 방법에 대한 연구들이 수행되고 있다. 단어 표상을 확장하는 방법으로는 대용량 말뭉치에 단어를 사전 학습하여 사용하거나, 단어의 패턴이나 추가적인 자질(feature)을 벡터로 표현하여 확장하는 방법들이 연구 되어 왔다[2-5]. 또한 최근 영어 개체명 인식에서는 단어를 문자 단위 임베딩 벡터로부터 단어 단위 임베딩 벡터로 유도하는 방법이 연구 되고 있으며[3,5-10], 이 유도 방법을 [5]에서 한국어 개체명 인식에 적용하여 음절 단위 임베딩 벡터로부터 단어 단위 임베딩 벡터로 유도하는 연구를 진행하여 개체명 인식에서 가장 좋은 성능을 보이고 있다.

음절 단위 임베딩 벡터로부터 단어 단위 임베딩 벡터로 유도하는 방법은 LSTM기반 방법과 CNN(Convolutional Neural Networks)기반 방법, 그리고 LSTM과 CNN을 결합하여 사용하는 방법이 있다[5].

LSTM 기반 방법에서는 한 단어를 구성하고 있는 음절 열을 입력으로 받고, forward의 마지막 상태와 backward의 마지막 상태를 결합하여 단어를 표현하는 임베딩 벡터를 만들어 내는 방법이다[5-6,9].

CNN 기반 방법에서는 한 단어를 구성하는 음절 열을 양 끝이 padding된 단어의 처음부터 끝까지 K개의 convolution filter를 적용하고 Max pooling을 취하여 단어를 표현하는 벡터를 만들어 내는 방법이다[5,7-8,10].

[5-7]에서는 LSTM과 CNN으로 만들어낸 단어 임베딩 벡터를 결합하여 단어표상에 추가하여 확장함으로써 좋은 성능을 보이고 있다.

본 논문에서는 단어 표상을 확장하기 위해 사전 학습된 단어 임베딩 벡터에 품사 임베딩 벡터를 추가하고, 음절 임베딩 벡터로부터 확장된 단어 임베딩 벡터를 추가한다.

품사 임베딩 벡터는 대량의 말뭉치를 형태소 분석한 결과에 단어를 삭제하고 품사를 하나의 단어 단위로 하여 word2vec[11]의 CBOW(Continuous bag-of-words) 모델을 이용하여 학습하고, 그 결과로 나오는 벡터를 사용한다.

음절 임베딩 벡터로부터 단어 임베딩 벡터를 유도하는 방식에는 bidirectional LSTM을 사용하고, 그 음절 열의 입력으로 학습 데이터에서 추출한 음절단위 개체명 분포를 이용한다.

3. 제안 방법

3.1. bi-LSTM-CRFs 모델

3.1.1. Recurrent Neural Networks(RNN)

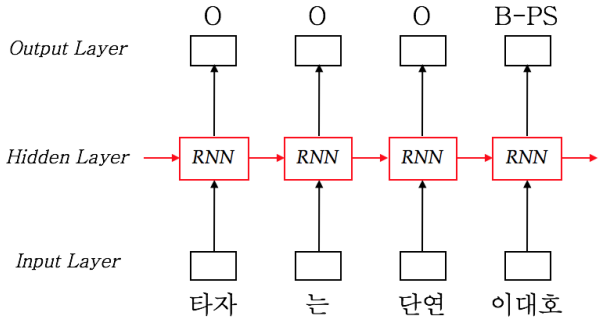

그림 1. Recurrent Neural Networks 모델

그림 1에서 보이는 것처럼 RNN은 단어 열$(x_1, x_2, ..., x_t)$을 입력 계층에서 입력으로 받고, 그 입력은 은닉 계층$(h_1, h_2, ..., h_t)$을 거쳐 입력을 잘 표현하는 벡터$(y_1, y_2, ..., y_t)$가 출력 계층으로 출력된다. RNN 구조를 아래와 같은 식으로 정의할 수 있고, 아래 식에서 U, V, W는 각 계층의 weight 행렬이다.

$$h_t = tanh(Ux_t + Vh_{t-1})$$
$$y_t = softmax(Wh_t)$$

RNN은 위의 수식과 그림 1에서 알 수 있듯이 이전 상태를 다음 상태로 계속 전달하는 모델이다. 따라서 이론상으로는 이전 상태를 기억하여 장기 의존성(long-range dependencies)을 다룰 수 있다. 하지만 실제적으로 위치상 멀리 있는 정보를 많이 잃어버리는 문제인 그래디언트 소멸 문제(Vanishing gradient problem)가 존재하기 때문에 장기 의존성을 유지할 수 없는 문제점이 있다.

3.1.2. Long Short-term Memory(LSTM)

LSTM은 RNN의 그래디언트 소멸 문제를 해결하여 장기 의존성을 잘 학습할 수 있는 특별한 모델이다. 이 문제를 해결하기 위하여 LSTM에서는 RNN의 은닉 계층 노드에 3개의 gate(input, output, forget)와 1개의 memory cell을 이용한다.

LSTM의 memory cell은 전체적인 상태를 기억하여 다음 상태로 전달하는 역할을 한다. forget gate를 이용하여 cell의 상태에서 어떤 정보를 제거할지 결정하고, input gate로 cell에서 어떤 정보를 갱신할지 결정한다. 마지막으로 output gate를 이용하여 cell의 어떤 정보를 전달할지 결정하여 장기 의존성을 유지한다. 본 논문에서는 [6]과 같이 forget gate를 사용하지 않았다. LSTM 구조의 전체적인 수식은 다음과 같다.

$$i_t = sigmoid(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i)$$
$$c_t = (1-i_t)\odot c_{t-1} + i_t \odot tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c)$$
$$o_t = sigmoid(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o)$$
$$h_t = o_t \odot tanh(c_t)$$

위 식에서 i, c, o, h는 각각 input gate, memory cell, output gate, hidden state이다. 그리고 $\odot$는 element-wise product이며, W는 weight 행렬을 b는 bias를 나타낸다.

3.1.3. bidirectional LSTM CRFs(bi-LSTM-CRFs)

bi-LSTM-CRFs는 그림 2에서 보이는 것과 같이 LSTM에 입력 문자열을 양방향으로 받아서 각 단어 별로 은닉 계층의 결과를 얻고, 그 결과 간의 의존성을 추가한 모델이다. 본 논문에서는 bi-LSTM-CRFs를 이용하여 한국어 개체명 인식 실험을 진행한다.

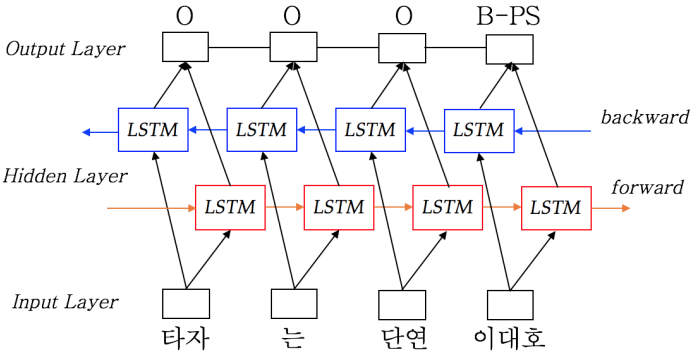

그림 2. bidirectional LSTM CRFs 모델

3.2. 단어 표상 확장

LSTM 기반의 모델은 각 단어의 임베딩 벡터를 입력으로 받기 때문에 단어 표상에 의존적이다. 따라서 본 논문에서는 그림 3과 같이 사전 학습된 단어 임베딩 벡터, 품사 임베딩 벡터, 음절 기반 단어 임베딩 벡터를 사용하여 단어 표상을 확장한다.

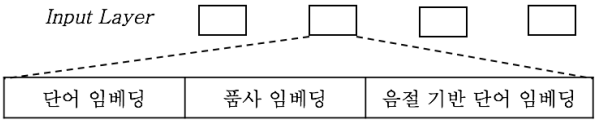

그림 3 . 확장된 단어 표상

3.2.1. 단어 임베딩

본 논문에서는 단어 임베딩 벡터를 잘 만들기 위하여, 사전 학습을 진행한다. 단어 임베딩을 사전 학습하기 위해 3GB의 뉴스 말뭉치를 사용하였고, word2vec의 CBOW 모델을 사용하였다. 이때 학습이 되는 단어의 단위는 형태소와 품사태그가 결합된 형태로 사용하였다. 예를 들면 아래와 같다.

- 단어 : “타자”, “이대호”
- 임베딩 단위 : “타자/NNG”, “이대호/NNP”

3.2.2. 품사 임베딩

개체명 인식에서는 품사의 시퀀스 또한 중요하기 때문에 품사를 잘 표현하는 자질 임베딩 벡터가 중요하다. 따라서 본 논문에서는 단어 임베딩 벡터를 사전 학습하는 것처럼 품사 단위의 사전 학습 결과를 품사 임베딩 벡터로 사용하였다. 품사 임베딩 벡터를 사전 학습하기 위하여 3GB의 뉴스 말뭉치를 형태소 분석 후 단어를 삭

제 한 뒤 품사를 임베딩의 단위로 사용하였다. 품사 임베딩 벡터 학습의 모델로는 단어 임베딩 벡터를 학습할 때와 같은 word2vec의 CBOW모델을 사용하였다.

- 문장 : "타자/NNG 는/JX 단연/MAG 이대호/NNP"
- 품사 단위 문장 : "NNG JX MAG NNP"

3.2.3. 음절 기반 단어 임베딩

음절 기반 단어 임베딩이란 각 단어를 표현하기 위해 음절 단위의 임베딩 벡터를 기반으로 단어 단위의 임베딩 벡터로 확장한 벡터를 말한다. 단어는 음절의 시퀀스이기 때문에 음절 단위의 임베딩 벡터의 확장은 단어를 표현하기에 적합하다. 본 논문에서는 그림 4에서 보이는 것과 같이 bidirectional LSTM을 이용하고, forward의 마지막 상태와 backward의 마지막 상태를 결합하여 단어 단위 임베딩 벡터로 사용하였다.

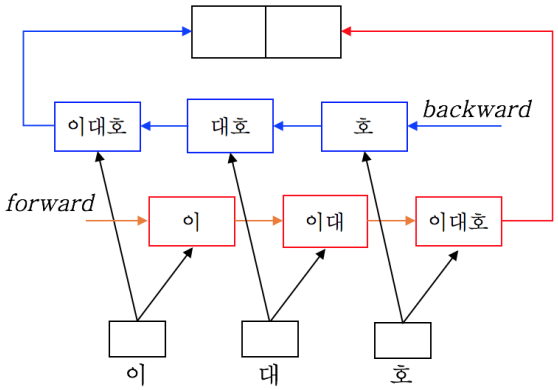

그림 4. 음절 기반 단어 임베딩 벡터를 위한 bi-LSTM

입력되는 음절 임베딩 벡터로는 음절별로 각 개체명 태그별 분포를 벡터로 만들어서 사용하였다.

표 1 . 음절 단위 개체명 분포 벡터

	B					I					O
	PS	OG	LC	DT	TI	PS	OG	LC	DT	TI	
타	0.09	0.09	0.09	0.09	0.09	0.09	0.09	0.09	0.09	0.09	0.09
자	0.09	0.09	0.09	0.09	0.09	0.09	0.10	0.09	0.09	0.09	0.09
는	0.09	0.09	0.09	0.09	0.09	0.09	0.09	0.09	0.10	0.09	0.10
단	0.09	0.09	0.09	0.09	0.09	0.09	0.09	0.10	0.09	0.09	0.09
연	0.10	0.09	0.09	0.09	0.09	0.09	0.09	0.09	0.09	0.09	0.09
이	0.14	0.09	0.08	0.08	0.08	0.11	0.09	0.08	0.08	0.08	0.09
대	0.09	0.11	0.09	0.09	0.09	0.09	0.10	0.09	0.09	0.09	0.09
호	0.11	0.09	0.09	0.09	0.09	0.09	0.09	0.09	0.09	0.09	0.09

예를 들어 "타자/O 는/O 단연/O 이대호/B-PS" 문장에서 각 음절에 대응되는 음절 임베딩 벡터는 표 1과 같다. 표 1에서 보이는 것과 같이 만들어진 벡터는 11차원을 가진다(개체명 태그 5개 * BI정보 2개 + O 태그 = 11). 최종적으로 모델에 입력 될 때는 분포 벡터에 softmax를 이용하여 확률로 변환 후 사용한다. 표 1의 값들은 지면상 반올림하여 표기하였다.

3.2.4. 전체 구성도

본 논문에서 제안하는 한국어 개체명 인식에 사용되는 bi-LSTM-CRFs의 전체 구성도는 그림 5와 같다.

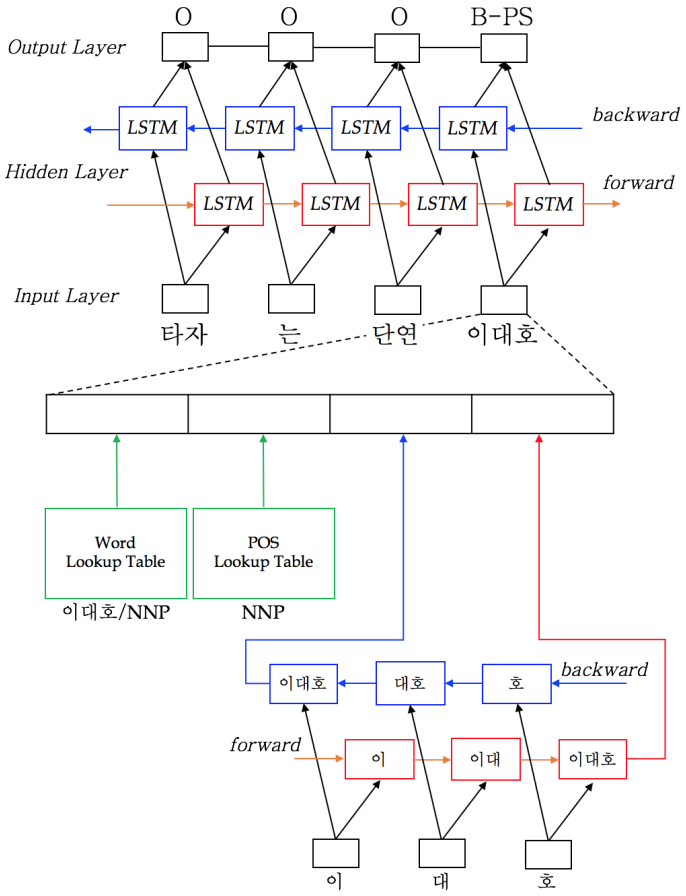

그림 5. 단어, 품사, 음절 임베딩 결합한 전체 구성도

4. 실험

4.1. 실험 환경

제안한 한국어 개체명 인식에서 단어 표상 확장 방법의 성능 평가를 위해 bi-LSTM-CRFs를 TensorFlow[12]로 구

현하여 사용하였다.

개체명 인식 평가 데이터로는 2016년 국어 정보 처리 시스템 경진 대회에서 배포한 개체명 인식 말뭉치 4,056 문장을 사용 하였다. 4,056 문장 중 3,244 문장을 학습 데이터로, 812 문장을 평가 데이터로 사용 하였다. 개발 데이터로는 학습 데이터 중에서 랜덤으로 500 문장을 추출하여 사용하였다. 전체적인 실험 성능은 개발 데이터에서 가장 좋은 성능을 보인 97 epoch로 평가하였다.

4.2. 단어 임베딩 벡터 실험

단어 임베딩 벡터 실험에서는 단어의 차원을 64로 하였고, 3GB의 뉴스 말뭉치로 word2vec의 CBOW모델을 사용하여 학습한 결과를 사용하였다. word2vec의 파라미터로는 window size, iteration을 각각 5로 사용하였다. 단어 임베딩 벡터를 랜덤으로 초기화한 실험보다 사전 학습된 임베딩 벡터를 사용한 실험의 성능이 3.16% 높았다.

표 2 . 단어 임베딩 벡터 실험 결과(F1)

	Test
random	71.43
pretrain	**74.59**

4.3. 품사 임베딩 벡터 실험

품사 임베딩 벡터 실험에서는 사전 학습된 품사 임베딩 벡터가 one-hot입력보다 좋은 자질이 될 수 있음을 보인다. 품사 임베딩 벡터에 16차원을 사용하였고, word2vec의 파라미터로 window size와, iteration은 각각 5를 사용하였다. 품사 임베딩 벡터 실험에서는 품사를 one-hot으로 추가한 실험보다 사전 학습된 품사 임베딩 벡터를 사용한 실험이 0.84%증가하였고, 사전 학습된 단어 임베딩 벡터 만 사용한 결과 대비 2.99%증가하였다.

표 3 . 품사 임베딩 벡터 실험 결과(F1)

	Test
pretrain	74.59
pretrain+pos(one-hot)	76.74
pretrain+pos(pretrain)	**77.58**

4.4. 음절 기반 단어 임베딩 벡터 실험

음절 기반 단어 임베딩 벡터 실험에서는 음절 입력을 랜덤 벡터, 사전 학습된 음절 임베딩 벡터, 개체명 분포 임베딩 벡터, 그리고 사전 학습된 음절 임베딩 벡터와 음절 단위 개체명 분포 임베딩 벡터를 결합한 벡터로 실험을 진행하였다. 랜덤 벡터와 사전 학습된 음절 임베딩 벡터는 64차원을 사용 하였고, 음절 단위 개체명 분포 임베딩 벡터는 11차원을 사용하였다. 그 결과 랜덤으로 입력한 것 보다 사전 학습된 벡터와 음절 단위 개체명 분포 벡터를 결합한 성능이 1.94%향상하여 가장 좋은 성능을 보였다. 본 실험에서 개체명 분포 임베딩 벡터를 단독으로 사용한 경우의 성능이 높지는 않지만, 사전 학습된 음절 단위 임베딩 벡터에 결합하여 사용한 음절 단위 개체명 분포 벡터가 의미 있는 벡터임을 증명 하였다.

표 4 . 음절 기반 단어 임베딩 벡터 실험 결과(F1)

	Test
pretrain+pos(pretrain)	77.58
pretrain+pos(pretrain)+charLSTM(random)	78.02
pretrain+pos(pretrain)+charLSTM(pretrain)	78.44
pretrain+pos(pretrain) +charLSTM(distribution)	77.72
pretrain+pos(pretrain) +charLSTM(pretrain+distribution)	**79.52**

4.5. 최종 실험 결과 비교

최종 실험 결과로는 총 세 종류의 실험 중 가장 높은 성능을 모아서 비교한다. 결과적으로 사전 학습된 단어 임베딩 벡터만 사용한 실험 대비 품사 임베딩 벡터와 음절 기반 단어 임베딩 벡터를 추가한 실험의 성능이 4.93% 증가 하였다.

표 5 . 최종 실험 결과 비교(F1)

	Test
pretrain	74.59
pretrain+pos(pretrain)	77.58
pretrain+pos(pretrain) +charLSTM(pretrain+distribution)	**79.52**

아래의 그림 6은 총 세 종류의 실험에 따른 F1 성능을 나타낸 것이다.

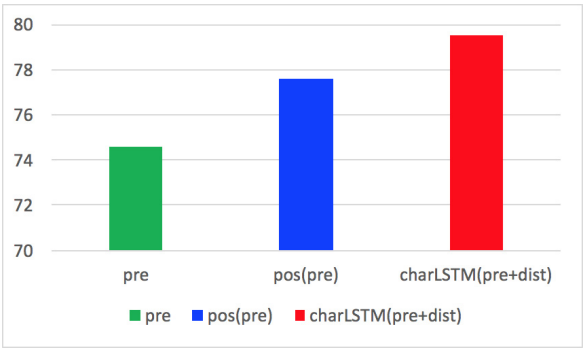

그림 6. 최종 실험 결과 비교(F1)

5. 결론

본 논문에서는 한국어 개체명 인식을 위해 bi-LSTM-CRFs에 입력으로 들어가는 단어 표상을 확장하는 방법을 이용하였다. 단어 표상을 확장하기 위하여 사전 학습된 단어 임베딩 벡터, 사전 학습된 품사 임베딩 벡터, 그리고 음절 기반 단위 임베딩 벡터를 사용하였다. 그 결과 품사 임베딩 벡터와 음절 기반 단어 임베딩 벡터를 추가한 모델이 사전 학습된 단어 임베딩 벡터만을 사용한 모델에 비해 4.93% 증가한 높은 성능을 얻을 수 있었다.

참고문헌

[1] 이창기, 김준석, 김정희, 김현기, "딥 러닝을 이용한 개체명 인식", *한국정보과학회 동계학술발표회 논문집*, No.12, pp.423-425, 2014.

[2] 이창기, "Long Short-Term Memory 기반의 Recurrent Neural Network를 이용한 개체명 인식", *한국컴퓨터종합학술대회 논문집*, No.6, pp.645-647, 2015.

[3] Zhiheng Huang, Wei Xu, Kai Yu, "Bidirectional LSTM-CRF Models for Sequence Tagging", *arXiv:1508.01991*, 2015.

[4] R Collobert, J Weston, L Bottou, M Karlen, K Kavukcuoglu, P Kuksa, "Natural Language Processing (almost) from Scratch", *The Journal of Machine Learning Research*, Vol.12, No.8, pp.2493-2537, 2011.

[5] 나승훈, 민진우, "문자 기반 LSTM CRF를 이용한 개체명 인식", *한국컴퓨터종합학술대회 논문집*, 2016.

[6] Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, Chris Dyer, "Neural Architectures for Named Entity Recognition", *arXiv:1603.01360*, 2016.

[7] Xuezhe Ma, Eduard Hovy, "End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF", *arXiv:1603.01354*, 2016.

[8] Jason P.C. Chiu, Eric Nichols, "Named Entity Recognition with Bidirectional LSTM-CNNs", *arXiv:1511.08308*, 2015.

[9] Wang Ling, Isabel Trancoso, Chris Dyer, Alan W Black, "Character-based Neural Machine Translation", *arXiv:1511.04586*, 2015.

[10] CN dos Santos, B Zadrozny, "Learning Character-level Representations for Part-of-Speech Tagging", Vol.5, No.2014, pp.3830-3838, *ICML*, 2014.

[11] word2vec, [Online]. Available: http://code.google.com/archive/p/word2vec/

[12] TensorFlow, [Online]. Available: https://www.tensorflow.org/