Word Embedding에 PCA를 적용한 개체명 인식 모델을 위한 효율적인 학습방법 연구

A Study on Efficient Training Method for Named Entity Recognition Model with Word Embedding Applied to PCA

저자 (Authors)
송은영, 최회련, 이홍철
Eun-Young Song, Hoe-Ryeon Choi, Hong-Chul Lee

출처 (Source)
대한산업공학회지 45(1), 2019.2, 30-39(10 pages)
Journal of the Korean Institute of Industrial Engineers 45(1), 2019.2, 30-39(10 pages)

발행처 (Publisher)
대한산업공학회
Korean Institute Of Industrial Engineers

URL
http://www.dbpia.co.kr/journal/articleDetail?nodeId=NODE07612573

APA Style
송은영, 최회련, 이홍철 (2019). Word Embedding에 PCA를 적용한 개체명 인식 모델을 위한 효율적인 학습방법 연구. 대한산업공학회지, 45(1), 30-39

Journal of the Korean Institute of Industrial Engineers
Vol. 45, No. 1, pp. 30-39, February 2019.
ISSN 1225-0988 | EISSN 2234-6457
https://doi.org/10.7232/JKIIE.2019.45.1.030

<Application Research>

Word Embedding에 PCA를 적용한 개체명 인식 모델을 위한 효율적인 학습방법 연구

송은영 · 최희련 · 이홍철†

고려대학교 산업경영학과

A Study on Efficient Training Method for Named Entity Recognition Model with Word Embedding Applied to PCA

Eun-Young Song · Hoe-Ryeon Choi · Hong-Chul Lee

Department of Industrial Management Engineering, Korea University

The Bidirectional LSTM CRF model used for Named Entity Recognition takes much time to train Named Entity. The hyper-parameters of Word Embedding used as input data in this model affect performance and training time. However, there is very little research on the number of dimensions, which is one of the parameters of Word Embedding. In this paper, we obtain proper number of 4-Word Embeddings (fastText, GloVe, skip-gram, CBOW) considering performance and training time in Bidirectional LSTM CRF which can input large amount of data. Next, apply the PCA to the word vector in Word Embedding to reduce the dimension to small dimensional (10 dimensions) intervals. Therefore, applying PCA to conventional Word Embedding and training Word Embedding with small dimensional intervals shows that the model can be trained by maintaining or improving performance based on stable training time in fewer dimensions than conventional Word Embedding.

Keywords: Word Embedding, Named Entity Recognition, Principal Component Analysis, Text Mining

1. 서 론

개체명(Named Entity)은 문서 또는 문장에서 고유한 의미를 내포하고 있는 명사 또는 숫자 표현으로, 개체명 인식은 개체명을 추출하고 추출된 개체명을 의미에 따라 분류하는 자연어 처리 연구 분야 중 하나이다(Lee and Jang, 2010). 자연어 처리는 컴퓨터가 인간이 사용하는 언어를 이해할 수 있게 하는 것으로 언어를 수치적으로 표현할 수 있어야 한다. 수치적 표현으로는 one-hot vector 방식이 있으나, 이 방식은 단어 간의 관계를 표출해 내지 못하는 단점을 가지고 있다. 분산표상(distributed representations) 개념을 사용하여 단어벡터를 미리 정의된 R차원의 연속형 실수값으로 표현할 수 있는 Word Embedding 방법으로 one-hot vector의 단점을 극복하였다(Bengio *et al*., 2003). 초반의 통계기반 기계학습을 적용한 개체명 인식 연구는 Word Embedding을 적용하면서 다양한 딥러닝 모델 연구로 확장하고 있다(Young *et al*., 2017).

초창기 Word Embedding은 차원의 저주를 극복하기 위해 단어의 분산표상을 만드는 방식인 NNLM(Neural Network Language Model)이 제시되었고, 이 방법은 RNNLM(Recurrent Neural Network Language Model)으로 발전하였다(Mikolov *et al*., 2013). 이 두 방법은 단어 벡터화 학습에는 좋은 방법이지만, 다량의 학습 파라미터 및 데이터로 인한 장시간의 학습시간을 요구하는 방법으로 이를 보완하기 위해 Continuous Word Embedding 학습모형인 Word2Vec이 발표되었다. Word2Vec(Mikolov *et al*., 2013c)은 2가지 네트워크 모델을 제시하고 있다. 하나는 CBOW(continuous bag-of-words) 모델이고, 다른 하나는 skip-gram 모델이다.

이 논문은 과학기술정보통신부와 한국정보화진흥원 주관의 "2018 빅데이터 플래그십 선도사업(실증확산)"의 지원을 받아 수행되었음.
† 연락저자 : 이홍철 교수, 02841, 서울특별시 성북구 안암로 145 고려대학교 산업경영공학부, Tel : 02-3290-3389, Fax : 02-929-5388, E-mail : hclee@korea.ac.kr
2018년 9월 17일 접수; 2018년 11월 19일 수정본 접수; 2018년 11월 21일 게재 확정.

이후에 GloVe(Pennington *et al*., 2014) 및 fastText(Bojanowski and Grave *et al*., 2017) 등의 방법들이 제안되었다.

개체명 인식 모델은 어떠한 Word Embedding 방법을 사용하는지에 따라 해당 모델의 성능에 영향을 미치는 것으로 알려져 있다(Choi and Cha, 2016). 따라서 다양한 Word Embedding 방법을 사용하여 모델의 성능과 관련된 다양한 연구가 진행되었다. Yu and Ko(2017)는 개체명 인식에 적합한 단어 표상을 확장시키는 Word Embedding을 개체명 인식에서 높은 성능을 보이는 Bidirectional LSTM CRF 모델에 적용하여 모델의 성능을 향상시키는 연구를 하였다. 또한, Lebret and Collobert(2014)는 Word Embedding을 Hellinger PCA를 통해 단어 동시발생 행렬(co-occurrence matrix) Word Embedding을 개체명 인식 모델에 적용하여 계산량은 줄이고, NNLM과 비슷한 성능을 얻었다.

하지만 개체명 인식 모델에 Word Embedding을 적용한 연구들은 몇 가지 한계점을 내포하고 있다. 그중 하나는 대부분 성능 위주의 평가(Lebret and Collobert, 2014; Santos and Guimaraes, 2015; Young *et al*., 2017)로 모델에 따른 학습시간 문제를 다루고 있는 연구들이 부족하다는 점이다(Lee *et al*., 2006; Lebret *et al*., 2013). 이 점을 고려한다면, 메모리가 제한된 장치에서 모델의 성능을 유지하며 학습시간을 줄이는 것은 모델의 유용성을 향상시킬 수 있다(Raunak, 2017). 또 다른 하나는 특정 도메인을 대상으로 모델 자체의 학습 파라미터는 고정 후, 여러 개의 Word Embedding 방법을 적용(Seok *et al*., 2015; Ghannay *et al*., 2016)하거나 window size와 같은 파라미터를 조정한 연구(Nooralahzadeh *et al*., 2018)는 존재하나, Word Embedding의 차원(number of dimensions) 파라미터 조정에 관한 연구(Patel and Bhattacharyya, 2017)는 부족하다는 것이다. 대부분의 기존 연구(Schnabel *et al*., 2015; Ghannay *et al*., 2016; Zhai *et al*., 2016)는 Word Embedding 차원 파라미터를 실험을 통해 임의로 정하였다. 또는 기존의 경험(Levy and Goldberg, 2014; Levy *et al*., 2015; Ghannay *et al*., 2016)을 기반으로 학습하거나, 대표적으로 사용되는 300에서 500차원 사이를 적용하였다(Patel and Bhattacharyya, 2017, Lam *et al*., 2018). 그 밖의 연구로는 큰 차원(예 : 100차원) 단위로만 학습하여 성능이 높은 Word Embedding의 차원을 정하였다(Pennington *et al*., 2014).

따라서 본 논문에서는 학습 데이터 셋을 생성하는 과정에서 1단계로 큰 차원 단위로 최적의 Word Embedding 차원을 찾은 후, 2단계에서는 1단계에서 찾은 Word Embedding에 PCA를 적용하여 성능 및 학습시간을 모두 고려한 효율적인 개체명 인식 모델 학습 방법을 제안한다. 본 논문에서 적용한 도메인은 국내 4개 지역 교통사고 제보데이터에서의 사고지점에 대한 개체명 인식이며, 사용된 개체명 인식 모델은 최근 개체명 인식에서 좋은 성능을 얻고 있는 Bidirectional LSTM CRF를 사용하였다. CRF는 성능에 비해 많은 학습시간이 필요하므로(Lee *et al*., 2006), 이를 보완하기 위해서도 차원축소를 통한 학습시간 단축이 필요하다. 얻어진 Word Embedding에 대한 평가는 특정 도메인 모델 내에서의 성능 및 학습시간에 대한 평가 방식인 외재성 평가를 사용하였다(Lee *et al*., 2018).

본 논문의 구성은 다음과 같다. 제 2장에서는 본 연구에서 제안하는 방법을 서술하며, 제 3장에서는 데이터 전처리 및 형태소 분석과 코퍼스 구축을 설명한다. 제 4장에서는 Word Embedding 및 PCA에 대해 서술한다. 제 5장에서는 개체명 인식 모델을 서술하며, 교통사고 제보데이터를 대상으로 실험한 결과를 설명하고, 마지막 제 6장에서는 결론 및 기대효과 그리고 한계점 및 추후 연구방향에 대해 기술한다.

2. 제안 방법

본 논문에서는 특정 도메인으로 국내 4개 지역인 인천, 광주, 제주 및 강원지역의 교통사고 제보데이터를 대상으로 하며, 제안하는 효율적인 학습 방법의 구조는 <Figure 1>과 같다.

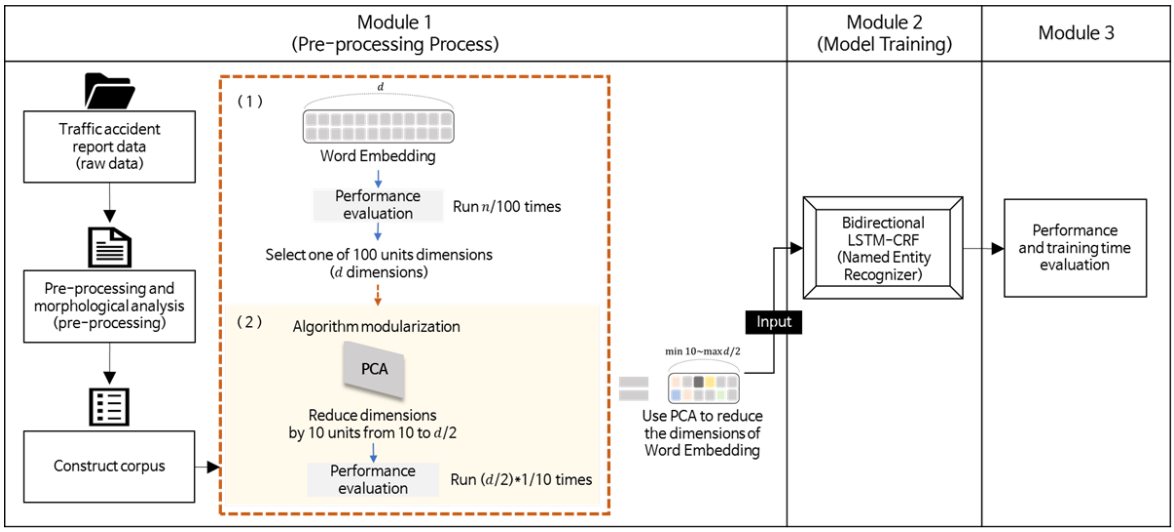


Figure 1. Suggestion Method

제안방법의 구조는 3개의 모듈로 구성되며, 첫 번째 모듈은 대상 도메인 데이터의 전처리 및 Word Embedding 생성단계이다. 데이터 전처리는 원시데이터(raw data)를 입력받아 전처리를 거쳐 형태소 분석과 코퍼스를 구축하고, 개체명 인식기에 사용될 Word Embedding을 생성한다. 본 논문에서 제안하는 Word Embedding은 (1) 100차원 단위의 큰 차원생성을 통한 최적의 Word Embedding을 찾아내고, (2) Word Embedding에 PCA로 10단위부터 $d/2$까지 축소된 차원을 적용한다. (2)는 Word Embedding 코드에 PCA 알고리즘을 추가하여 (1)에서 찾은 Word Embedding을 기반으로 차원축소 된 모든 Word Embedding을 한 번에 구할 수 있도록 모듈화 하였다. 본 논문에서 차원축소 시 사용한 차원의 범위인 $d/2$는 Raunak(2017)가 제안한 Word Embedding 차원축소 시 축소할 차원크기의 최댓값인 $d/2$을 의미한다. 두 번째 모듈은 Bidirectional LSTM CRF을 적용한 개체명 인식 모델에 Word Embedding을 학습시킨다. 이때 학습시간은 이 모듈에서 수행되는 시간을 말한다. 마지막 모듈은 PCA 차원축소를 적용한 Word Embedding을 사용하여 Bidirectional LSTM CRF의 성능 및 학습시간을 평가한다. 본 논문의 학습방법은 교통사고 제보데이터에서 Word Embedding에 PCA의 차원축소를 적용한 방법이다. 이 방법은 기존 연구에 비해, 모델의 성능 및 학습시간을 동시에 고려하여 개체명 인식 모델을 효율적으로 학습할 수 있도록 하는 방법을 제안한다.

3. 데이터 전처리 및 형태소 분석과 코퍼스 구축

개체명 인식 모델의 효율적인 학습 방법 연구에 사용한 데이터는 도로교통공단에서 제공해 준 교통 제보데이터로 2017년 1월부터 6월까지의 인천, 광주, 제주 및 강원지역 총 15,193개 사고 제보로 이루어졌다.

교통사고 제보데이터는 각 지역별 제보 형식을 가지고 있으며 2014년 1월 1일부터 전면적으로 도로명 주소가 시행되었음에도 불구하고 교통사고 제보데이터의 교차로나 지역명이 도로명 주소로 일관되게 기록되어 있지 않았다. 따라서 Word Embedding을 만들기 위해 교통사고 제보데이터를 정형화하는 전처리 과정이 필요하였다. 제보 내의 화살표 같은 특수문자는 대체할 수 있는 조사로 바꿔서 표현하는 등의 제보 형식 단일화와 같은 지점에 대한 다양한 명칭을 도로명 주소로 통일하였으며, 숫자는 대응되는 한글로 변환해주었다.

전처리가 된 데이터는 <Table 1>과 같으며, 데이터는 4개의 열인 데이터 번호, 날짜, 시간, 사고 내용으로 구성된다.

다음으로 전처리가 된 사고 내용 데이터를 형태소 단위로 나눴다. 형태소 분석기의 종류를 결정하는 것은 분석할 데이터나 적용 도메인에 따라 달라질 수 있다(Seo *et al*., 2017). Word Embedding도 형태소 분석기에 따라 다르게 만들어지기 때문에, 어떤 형태소 분석기를 사용하느냐가 중요하다. 본 연구에서는 KoNLPy(Park and Cho, 2014)의 클래스 중 하나인 한나눔(Hannanum) 형태소 분석기를 사용하였다. 다른 형태소 분석기와는 달리, 한나눔 형태소 분석기는 문장을 분석하여 형태소 단위로 나눠줄 때, 다양한 형태소 후보군을 제시하고 사용자가 이를 선택할 수 있도록 하였다. 한나눔 형태소 분석기를 사용함으로써, 문장들이 형태소 분석기의 분석에만 의존하지 않도록 하였다.

또한, 형태소 분석기들은 신조어나 줄임말로 인해 미등록단어 문제에 직면하고 있다(Kim *et al*., 2014). 일반적으로 이 문제를 해결하기 위해서 형태소 분석기의 사용자 사전(user dic)에 단어를 추가하는 방법이 있다. 본 연구에서는 고속도로 또는 일부 지역의 도로명 및 교차로명을 사용자 사전에 추가해 줌으로써 한계점의 일정부분을 해결하였고, 특정 도메인의 개체명 인식에서 성능을 향상시킬 수 있도록 하였다(Kim *et al*., 2018).

이후, 형태소 단위로 나뉜 전처리 데이터로 코퍼스를 구축하였다. 코퍼스 구축 시, 개체명 인식을 위한 태거(tagger)를 만들었다. 기본적으로 개체명 인식에서 BIO 태거를 사용한다. 여기서 B(Beginning)는 개체명의 시작, I(Inside)는 B뒤로 이어지는 개체명의 토큰을 의미하며, O(Outside)는 B와 I에 속하지 않는 것을 말한다(Ratinov and Roth, 2009). 하지만 본 연구에서는 교통사고 지점의 방향이나 위치를 표현하기 위해 교통사고 제보데이터에 맞는 개체명 태거를 따로 생성하였다(2017년 빅데이터 플래그십 시범사업 최종보고서, 2018). 생성된 태거는 LOC_DIR, LOC_FROM, LOC_TO, LOC_FIN, LOC_PO, LOC_FDO, CHARO 및 O까지 포함하여 총 8개이며, 이에 따라, 형태소 기반으로 나뉜 문장들을 태깅하였다. 생성된 코퍼스는 <Figure 2>와 같다(Song *et al*., 2017).

Table 1. Formulated Data

NUM	Date	Time	Traffic accident report data
1	20170101	0102	경명대로 계산역사거리에서 계산삼거리방향.....
2	20170205	1128	남동대로 길병원사거리에서 남동경찰서사거리.....
…	…	…	…
n-1	20170509	2222	문학사거리 약 20m 못 간 지점 일차로에서.....
n	20170628	2305	아암대로 고속종점지하차도에서 낙섬사거리.....

```
...
△△고속도로 LOC_FDC
○○ LOC_DIR
방향 O
◇◇ 터널 LOC_FIN
부근 O
1차로 CHARC
에서 O
승용차 CAR
사고처리 O
하 O
고 O
있 O
어 O
7Km구간 O
정체
...
```

Figure 2. Tagged Corpus

4. Word Embedding 방법과 주성분분석

본 장에서는 Word Embedding 방법 및 차원축소 방법인 주성분분석(principal component analysis, PCA)에 대해 살펴본다.

4.1 Word Embedding 방법

Word Embedding 방법으로 Word2Vec의 CBOW와 skip-gram, GloVe 및 fastText가 있다. Word2Vec은 <Figure 2>와 같이, CBOW (continuous bag-of-words)와 skip-gram 두 가지 모델이 있다. 먼저, CBOW는 문장의 주변 단어를 보고 타겟단어를 예측하는 모델이다. 예를 들면, '오늘 도서관에 가서 __을 빌렸다.'라는 문장에서 주변 단어를 보고 빈칸에 '책'이라는 단어를 예측하는 것을 말한다. CBOW는 NNLM과 유사하며, 단어를 한 개의 요소만 1이고 나머지는 0인 N차원의 벡터로 표현할 수 있는 one-hot encoding 벡터로 변환한 후, 단어들을 똑같은 위치에 사영(projection)시킨다. 이 벡터들의 평균을 사영층(projection layer)의 입력 값으로 사용하고 가중치 행렬을 곱하여, 이를 출력층(output layer)으로 보내 에러를 줄여가는 모델이다.

Skip-gram은 CBOW와는 반대 모델이다(Jeong *et al*., 2018). 앞서 예로든 문장에서 CBOW이 주변 단어를 보고 책이란 단어를 예측했던 것과는 반대로 skip-gram은 책이란 단어에서 나머지 단어들을 예측하는 것이다. 이는 큰 데이터 셋에서 사용할 때 더 유용한 것으로 알려졌으며, 현재까지는 skip-gram이 CBOW보다 더 좋은 결과를 내는 추세이다.

GloVe는 문서 전체의 공기정보를 잘 반영하도록 하여, 단어간 유사도를 반영하지 못했던 Word2Vec의 단점을 극복했다. Pennington *et al*.(2014)은 Word2Vec의 negative sampling의 값을 변경해가며, GloVe와 Word2Vec의 CBOW와 skip-gram을 비교하였고, GloVe의 정확성이 더 높았다.

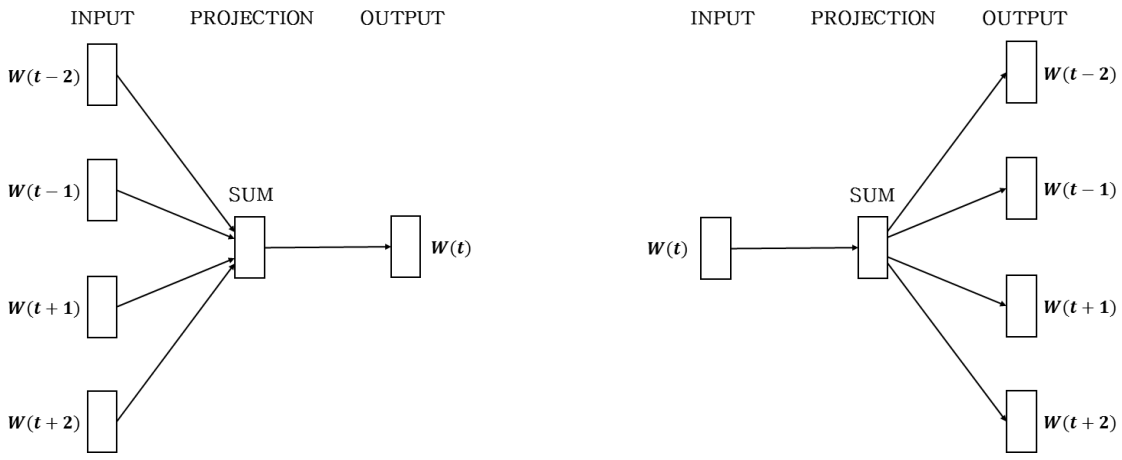
FastText는 skip-gram이 각 단어의 내부 구조를 무시한다는 한계점을 개선하기 위해 부분단어(subword)를 이용하는 방법으로 제안되었다(Bojanowski and Grave *et al*., 2017). 이 모델은 단어를 bag of character n-gram으로 나타냈으며, 자기 자신도 n-gram에 포함 시킨다는 특징이 있다(Bojanowski and Grave *et al*., 2017). 예를 들어, '아시아드경기장' 역을 n = 3일 때를 기준으로 character n-gram으로 표현한다면 아래와 같다.

<아시, 아시아, 시아드, 아드경, 드경기, 경기장, 기장>,
<아시아드경기장>

FastText는 부분 단어를 사용해서 OOV(out of vocabulary) 문제를 처리할 수 있는 장점이 있으며, 현재까지 다른 Word Embedding 방법보다 빠르고, 큰 코퍼스에도 빠르게 단어 표상을 계산할 수 있다고 알려져 있다(Kim and Lee, 2018).

4.2 주성분분석(Principal Component Analysis, PCA)

주성분분석은 고차원 데이터를 저차원 데이터로 축소시키는 차원축소 방법의 하나이다(Shlens, 2014). 이 방법은 기존의 변수로 새로운 변수를 만들어 내는 특징추출(feature extraction) 기



(a) CBOW (b) skip-gram

Figure 3. CBOW & skip-gram Architecture(Figure source : Mikolov *et al*., 2013c)

법이며, 지도학습에서 특징추출의 목적은 클래스 분류의 최대화인 반면에, 비지도학습에서 특징추출의 목적은 정보의 손실을 최소화하는 것이다. 본 논문의 목적은 후자이며, 동시에 이러한 차원축소를 통해 학습시간을 빠르게 하고, 성능을 유지하는 것에 초점을 맞추려 하였다.

주성분분석은 두 가지 방법이 있으며, 하나는 선형 방법인 PCA이고, 다른 하나는 비선형 방법인 Kernel PCA이다. 본 논문에서는 두 방법 중 선형 PCA를 사용하였다.

선형 PCA는 다음과 같다. 데이터들의 평균을 '0'으로 만들어 데이터 정규화를 시킨다. 다음으로 데이터들의 분산이 가장 큰 축을 찾고, 그 축의 수직인 축을 반복적으로 찾아 나가는 대각화 과정(diagonalization)을 거친다. 그 결과로 고윳값들(eigenvalues)이 나온다. 그중 가장 큰 고윳값(eigenvalue)의 고유벡터(eigenvectors)가 바로 PCs(principal components)가 된다.

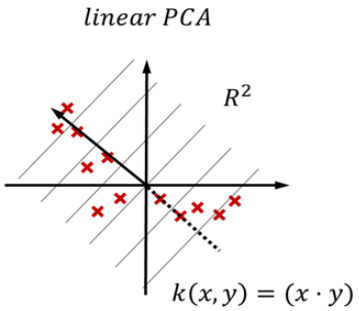


Figure 4. Linear PCA(Figure source : Scholkopf *et al*., 1999)

본 논문에서는 위에서 설명한 4가지의 Word Embedding 방법에 이를 통해 얻은 단어들의 벡터 값을 기반으로 작은 차원 단위로 PCA 차원축소를 적용하여 교통사고 제보데이터 개체명 인식 모델에서 효율적인 학습 방법을 제안한다.

5. 실험 결과 및 성능 평가

본 장에서는 PCA로 차원축소 시킨 Word Embedding을 교통사고 제보데이터 개체명 인식 모델에 적용하여 얻은 성능 및 학습시간의 결과를 살펴본다. 본 연구에서 사용한 개체명 인식모델인 Bidirectional LSTM CRF는 파이썬 3.6과 TensorFlow를 사용하여 개발하였으며, 실험은 Intel(R) Core(TM) i7-7700 CPU @ 3.60GHz 모델의 CPU와 64GB의 메인메모리 그리고 NVIDIA사의 GeForce GTX 1080 Ti인 GPU 환경에서 수행하였다.

5.1 개체명 인식 모델

본 논문에서는 Bidirectional LSTM CRF 모델을 이용하여 Word Embedding 방법과 Word Embedding 차원의 크기를 바꿔가면서 개체명 인식 실험을 진행하였다. Bidirectional LSTM CRF 모델은 개체명 인식과 같이 순차적인 데이터를 레이블링(labeling)하는데 특화되어 있다(Lee, 2015). 또한, <Figure 5>에 보이는 것처럼 각 단어를 표현하기 위해 문자 임베딩 벡터로 개별 Word Embedding 벡터를 확장하였다. 이러한 방법을 사용하면, 특정 도메인 내의 단어를 학습할 수 있다는 장점이 있다(Lample *et al*., 2016).

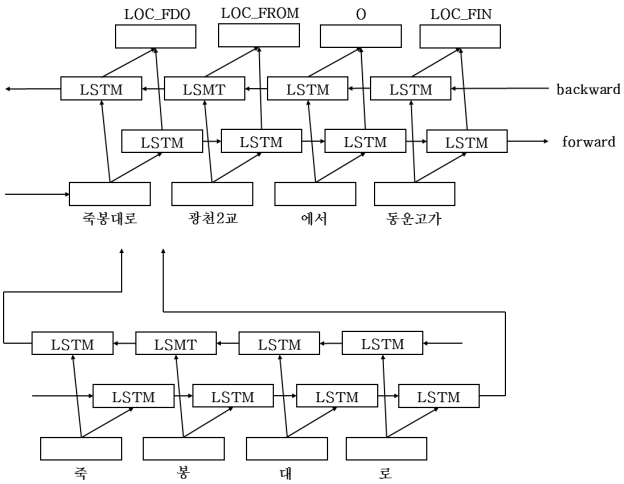


Figure 5. Bidirectional LSTM CRF Model Used in the Experiment

모델에 사용된 LSTM은 RNN의 장기의존성 문제를 해결하기 위해 제안된 RNN 모델 중 하나이다. LSTM은 <Figure 6>에서와 같이, input gate, forget gate, output gate 3개의 층과 1개의 cell로 이루어져 있다(Yu and Ko, 2017).

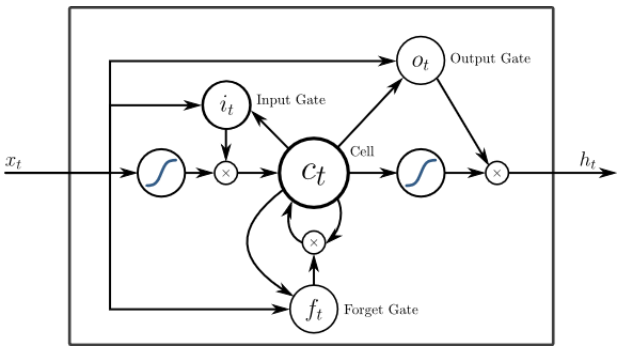


Figure 6. Long Short-Term Memory(Figure source : Graves *et al*., 2013)

LSTM의 진행과정은 다음 식(Colah.githyb.io, 2015)과 같다. 아래 식에서 C는 cell, i는 input gate, o는 output gate, f는 forget gate, h는 hidden state를 의미한다. 식 (1)은 forget gate 층에서 일어나는 과정으로 cell에서 어떤 정보를 버릴 것인지 결정한다. 다음 식 (2), 식 (3)에서 input gate를 통해 새로운 정보 중 어떤 것을 저장할지 결정한다. 또한, 어떤 값을 업데이트할 것인지 정한다. 식 (4)는 식 (2)와 식 (3)을 통해 얻은 값으로 C_{t-1}을 C_t로 업데이트한다. 마지막으로 output gate로 어떤 결과를 내보낼지 정하는 과정이 식 (5)와 식 (6)에서 이루어진다(Graves, 2013).

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (1)$$
$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (2)$$
$$\widetilde{C} = \tanh(W_C \cdot [h_{t-1,} x_t] + b_C) \quad (3)$$
$$C_t = f_t * C_{t-1} + i_t * \widetilde{C}_t \quad (4)$$
$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (5)$$
$$h_t = o_t * \tanh(C_t) \quad (6)$$

이러한 과정을 양방향으로 진행하는 것이 Bidirectional LSTM 이며, 여기에 CRF(conditional random field)를 연결하여 단어 결괏값 간의 의존성을 추가한 모델이 Bidirectional LSTM CRF 이다. Bidirectional LSTM CRF은 결괏값으로 개체명 인식의 태그 값과 태그를 얻는다.

Word Embedding을 넣어 Bidirectional LSTM CRF를 학습할 때, 하이퍼파라미터인 hidden size는 Word Embedding 차원으로 하였고, dropout값은 0.5로 지정하였으며, 경사하강법 알고리즘으로는 Adam optimizer를 사용하였다.

5.2 실험 결과

기존 연구와 같이 교통사고 제보데이터 개체명 인식 모델의 학습 파라미터는 고정한 후, 학습시간을 고려하지 않고 Word Embedding의 100차원에서 400차원까지 100차원 단위로 성능만 평가한 경우에 Word2Vec의 CBOW와 skip-gram, GloVe 및 fastText의 window size는 3, 차원 d가 300일 때 가장 높은 성능수치를 보였다. 이 Word Embedding을 각각을 교통사고 제보데이터 개체명 인식 모델에 적용시켰을 경우에, 4가지의 Word Embedding 방법 중 fastText가 94.6%로 성능이 제일 높았다(<Table 2> 참조).

앞장에서도 언급한 것처럼 본 논문의 적용 대상인 교통사고 제보데이터를 적용한 개체명 인식 모델은 학습시간이 많이 걸리는 모델이다. 이 모델에 <Table 2>에서 찾은 Word Embedding 차원을 10차원 단위로 PCA 차원축소 하여 기존에 찾은 차원의 절반 차원까지 학습하여 얻은 성능 및 학습시간을 <Table 3>과 <Table 4>에 나타내었다. 이를 본 논문의 제안 방법과 기존 Word Embedding과 차이가 있는지 확인하기 위해 기존 Word Embedding 방법을 10차원 단위로 학습한 결과도 함께 나타내었다. 결과적으로 10차원 단위로 PCA 차원축소 한 Word Embedding 방법과 기존 Word Embedding 방법이 성능 및 학습시간에서 차이가 있었다. 참고로 두 표에서 성능 및 학습시간이 기존 Word Embedding보다 우수하면 진하게 표시하였다. <Table 3>과 <Table 4>를 보면, GloVe는 10차원 단위로 PCA로 차원축소 시켰을 때 모든 차원에서 기존 Word Embedding보다 성능은 높고 학습시간이 짧았다. 특히, 다른 Word Embedding 방법들보다 GloVe에 차원축소 후 학습시켰을 때 성능이 향상되었다. Word2Vec의 skip-gram은 모든 차원에서 학습시간은 짧아졌지만, 성능은 특정 차원에서만 높았다. fastText와 Word2Vec의 CBOW는 특정 차원에서 성능이 높고, 학습시간은 짧았다. 또한, <Table 4>에서 10차원별 학습시간의 총합은 PCA로 차원축소 한 Word Embedding을 적용한 모델의 학습시간의 합이 기존 Word Embedding을 모델에 적용한 학습시간 합보다 작았다.

Table 2. Result When Number of Dimensions of Word Embedding is 300

Word Embedding-300D	F1 score(%)	Training Time(sec(s))
fastText	94.6*	1171.7
GloVe	94.2	1161.9
Word2Vec(skip-gram)	93.5	1190.9
Word2Vec(CBOW)	93.7	1269.5

Table 3. When Training in This Model, The Performance of 10 Dimensional Intervals of 300D + PCA Word Embedding Method and Conventional Word Embedding Method(%)

W.E[1]	fastText		GloVe		Word2Vec(skip-gram)		Word2Vec(CBOW)	
Dim.[2]	Origin	300D+PCA[3]	Origin	300D+PCA	Origin	300D+PCA	Origin	300D+PCA
10	91.3	90.8	89.6	**92.2**	89.0	**89.4**	89.0	87.2
20	92.5	**92.7**	91.3	**92.4**	91.2	88.5	90.2	**90.6**
30	91.2	**92.6**	91.7	**93.1**	90.8	90.7	91.2	**91.3**
40	92.9	**93.0**	92.3	**92.3**	90.9	**91.9**	91.3	90.6
50	93.2	92.4	92.1	**93.6**	91.8	**92.1**	91.7	91.3
60	93.5	93.3	91.0	**93.2**	91.6	**92.6**	91.9	**92.5**
70	93.6	93.5	91.8	**92.8**	90.9	**92.9**	92.2	92.0
80	93.8	93.2	90.3	**93.2**	92.1	**92.3**	92.6	91.6
90	94.0	93.9	91.7	**93.2**	92.9	92.3	92.2	92.1
100	93.2	**94.2**	92.8	**93.6**	91.9	92.6	93.0	92.8
110	93.3	**94.0**	91.3	**93.2**	92.5	91.9	93.0	91.5
120	94.5	94.2	92.7	**93.1**	92.8	92.3	93.0	92.0
130	94.0	**94.2**	92.8	**93.0**	92.0	**92.8**	92.6	**92.9**
140	93.7	**94.3**	93.1	**94.1**	93.0	92.7	92.1	**92.7**
150	94.0	**94.1**	92.8	**94.3**	92.5	92.2	93.1	92.7

[1]Word abbreviation for 'Word Embedding.'
[2]Word abbreviation for 'Size of Dimensionality.'
[3]PCA applied to Origin(300 dimensions) to reduce the dimension.

Table 4. When Training in This Model, The Training Time of 10 Dimensional Intervals of 300D+PCA Word Embedding Method and Conventional Word Embedding Method(sec(s))

W.E[1]	fastText		GloVe		Word2Vec(skip-gram)		Word2Vec(CBOW)	
Dim.[2]	origin	300D+PCA[3]	origin	300D+PCA	origin	300D+PCA	origin	300D+PCA
10	849.7	**808.2**	810.8	**800.2**	859.5	**850.5**	820.4	**804.7**
20	889.5	**815.9**	819.3	**805.6**	827.1	878.5	825.9	**823.5**
30	817.6	835.6	830.4	**823.6**	887.1	**838.8**	840.3	**839.8**
40	829.0	855.8	854.4	**833.7**	907.0	**849.9**	851.4	852.7
50	842.0	856.5	865.9	**855.2**	962.3	**864.6**	926.4	**870.3**
60	854.7	863.5	880.3	**876.3**	946.8	**879.9**	922.5	**879.5**
70	867.8	871.5	893.4	**878.8**	985.7	**889.9**	953.4	**893.0**
80	966.1	**890.9**	907.2	**892.1**	905.8	925.3	909.1	**903.1**
90	985.8	**902.7**	909.0	**904.4**	918.4	925.5	919.5	**918.4**
100	995.7	**911.5**	926.6	**918.0**	951.7	1000.5	994.1	**929.5**
110	1009.8	**930.1**	938.3	**926.1**	955.6	1045.4	945.4	948.8
120	1031.2	**942.0**	944.5	**939.3**	1044.7	**1015.9**	952.2	957.2
130	1041.4	**952.8**	957.7	**950.4**	970.5	1052.7	965.6	969.0
140	1054.4	**967.0**	971.4	**966.5**	984.6	1048.0	976.7	983.9
150	1074.2	**992.3**	981.6	**978.1**	996.4	**994.0**	990.7	**987.3**
SUM[4]	14108.9	13396.3	13490.8	13348.3	14103.2	14059.4	13793.6	13560.7

[4]Total of training time.

<Table 5>는 <Table 3>과 <Table 4>를 통해, 본 논문에서 제안한 방법을 기반으로 특정 도메인의 개체명 인식 모델 학습에 적합한 Word Embedding 방법 및 차원을 나타내었다. 이는 fastText의 140차원과 GloVe의 150차원이었다. 기존 300차원에서 10차원 단위로 PCA 차원축소 하였을 때, 차원축소를 140으로 적용한 fastText에서 성능이 94.3%로 기존과 유사한 성능을 내었고, 학습시간은 기존 1171.7(s)에서 967(s)로 204(s)가 빨라졌다. 마찬가지로 GloVe도 10차원 단위로 PCA로 차원축소하면서 찾았을 때, 150차원일 때 성능이 기존 94.2%에서 0.1% 높아진 94.3%로 더 높았고, 학습시간은 기존 1161.9(s)에서 978.1(s)로 183.8(s)가 빨랐다. 이러한 차이가 얼마나 의미가 있는지는 이에 대한 수학적 평가 척도가 아직 없다. 따라서 이 차이가 얼마나 의미가 있는지를 표현하기 위해 추가실험을 진행하였다.

앞의 <Table 4>의 총 학습시간 결과와 <Table 2>와 <Table 5>의 결과를 함께 보면, 기존 Word Embedding이 개체명 인식 모델에서 10차원부터 300차원까지 10차원 단위로 30번 학습한 것보다 본 논문의 제안 방법과 같이 100차원에서 400차원까지 100차원 단위로 4번 학습하여 300차원을 찾은 후, 그 다음에 10차원부터 150차원까지 10차원씩 차원을 변경하여 학습한 것이 최대 19번 안에 총 학습시간을 단축시키고, 개체명 인식 모델을 효율적으로 학습시킬 수 있는 Word Embedding을 찾을 수 있음을 확인하였다.

Table 5. The Performance and Training Time When Data was Learned by Word Embedding Method and Number of Dimensions Determined through the Proposed Method

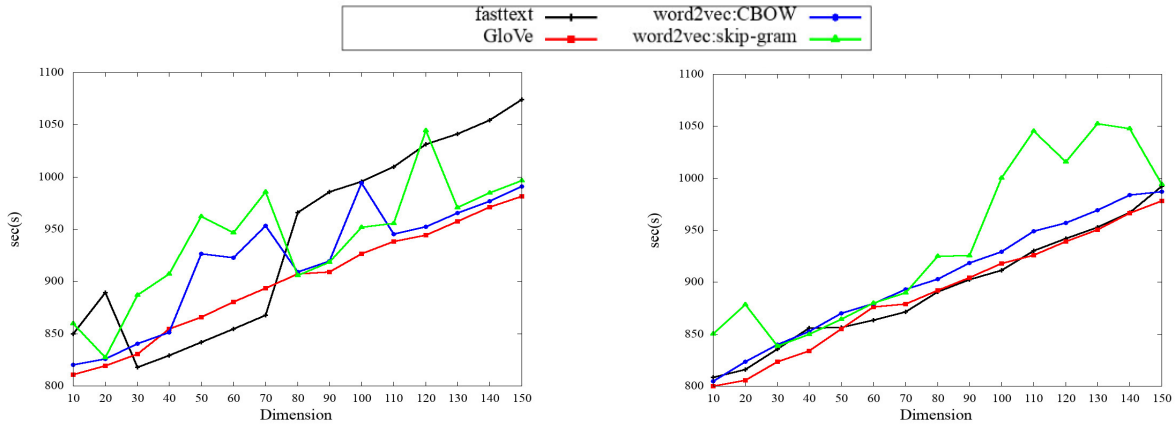
Word Embedding	F1 score(%)	Training Time(sec(s))
fastText-140D	94.3	967.0(s)
GloVe-150D	94.3	978.1(s)

추가실험 결과인 <Table 6>은 데이터 크기에 따른 학습시간의 차이를 보이기 위해 기존 15,193개에서 데이터 크기를 약 10,000개 정도 늘린 25,000개로 두고 기존 Word Embedding (fastText)에 차원 별로 학습시간을 나타내었다. 데이터 크기가 15,193일 때, fastText 300차원의 학습시간은 1171.7(s) 이었고, 150차원일 때의 학습시간은 1074.2(s) 이었다. 반면, 데이터 크기가 25,000일 때, fastText 300차원의 학습시간은 2900.6(s)이었고, 150차원일 때의 학습시간은 2410.7(s)이었다. 이를 통해 데이터의 크기가 증가함에 따라 Word Embedding 차원 크기 간의 학습시간의 차이가 크게 나는 것을 확인하였다.

Table 6. In fastText, Training Time by Number of Dimensions according to the Size of Data(sec(s))

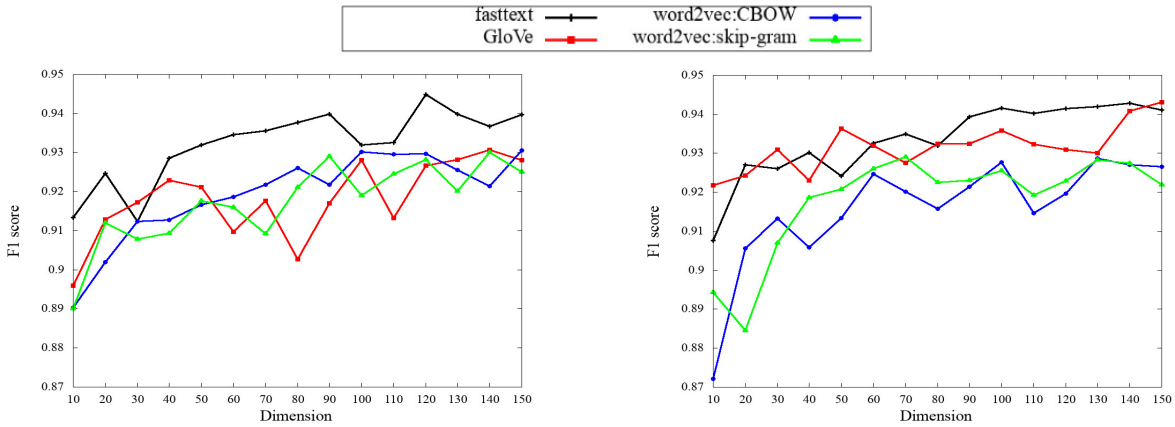
Dim. of fastText \ Data Size	15,193	25,000
300D	1171.7	2900.7
150D	1074.2	2410.7
100D	995.7	2240.0
50D	842.0	1933.2

본 논문의 교통사고 제보데이터 개체명 인식 모델은 성능을 유지하면서 학습시간을 단축시켜줘야 할 뿐만 아니라, 안정적인 학습시간도 고려해야 한다. 따라서 PCA를 통한 Word Embedding의 작은 차원 단위의 차원축소가 기존의 Word Embedding의 동일한 차원 기준으로 학습시간의 안정화 측면에서 차이가 있음을 보이기 위해 다음 <Figure 7>과 같이 그래프로 시각화하였다.



(a) Origin (b) 300D+PCA

Figure 7. Graph of Measured Training Time by 10 Demensional Intervals



(a) Origin (b) 300D+PCA

Figure 8. Graph of Measured Performance by 10 Dimensional Intervals

<Figure 7>에서는 PCA를 적용했을 때, Word2Vec의 CBOW와 skip-gram, GloVe 및 fastText에서의 각 차원마다의 학습시간을 그래프로 나타내었다. 기본적으로 PCA의 차원축소 방법을 적용하지 않은 <Figure 7>(a)의 경우, GloVe를 제외한 모든 Word Embedding 방법이 학습시간 측면에서 차원이 증가함에 따라 안정적으로 증가하지 않았다. 반면에, PCA를 적용하여 차원축소를 하여 학습한 경우에는 GloVe는 물론이고, fastText, Word2Vec의 CBOW가 차원이 증가함에 따라 학습시간이 안정적으로 증가하였다.

<Figure 8>에서는 PCA를 적용했을 때, Word2Vec의 CBOW와 skip-gram, GloVe 및 fastText에서의 각 차원마다의 성능을 그래프로 나타내었다. 기본적으로 PCA의 차원축소 방법을 적용하지 않은 <Figure 8>(a)와 <Figure 8>(b)를 비교하였을 때, fastText와 GloVe가 차원의 크기에 따라 성능이 안정적으로 증가하였다. 또한, <Figure 8>(a)에서 GloVe는 매 차원에서 4개의 Word Embedding 방법 중에서 가장 낮은 성능을 보였고, 성능이 안정적이지도 않았으나, <Figure 8>(b)에서 GloVe는 fastText 다음으로 높은 성능 및 안정적인 성능을 보였다.

6. 결론 및 향후 연구

본 논문은 인천, 광주, 제주 및 강원지역의 국내 4개 도시에서 2017년 1월부터 6월까지 수집된 교통사고 제보데이터의 사고지점 개체명 인식 모델에 대해 성능과 학습시간을 동시에 평가하는 효율적인 학습 방법을 제안하였다.

제안 방법의 첫 단계로 기본적인 전처리 과정 및 형태소 분석을 진행하여 이를 코퍼스로 구축하였으며, Word Embedding은 다음 두 단계를 거쳐 생성하였다.

1단계 : 100단위씩의 고차원 증가 방법을 사용하여 Word Embedding을 생성

2단계 : 생성된 Word Embedding으로 10단위씩 $d/2$차원까지 PCA 차원 축소를 적용함

두 번째 단계는 Word Embedding에 PCA를 적용하여 생성된 Word Embedding을 Bidirectional LSTM CRF 모델에 입력하여

학습하였고, 마지막 단계는 모델에 대해 성능과 학습시간을 동시에 고려하여 평가하였다. 제안 방법의 검증을 위해서 특정 도메인인 교통사고 제보데이터의 사고지점을 대상으로 개체명 인식 모델에 적용하여 다음과 같은 결과를 얻었다.

4개의 Word Embedding에 PCA를 적용한 개체명 인식의 성능은 선택된 기존의 Word Embedding을 적용하였을 때의 성능과 비교하였을 때, 인식 성능은 유사하였지만, 차원축소를 적용한 Word Embedding을 사용하였을 때 모델의 안정적인 학습시간을 기반으로 성능은 유지 또는 높이며 학습시간은 단축되는 결과를 얻었다. 따라서 본 논문과 같이 대용량 데이터인 경우, 안정적인 학습시간을 기반으로 성능 및 학습시간을 함께 고려해야 하는 모델에 효율적임을 입증할 수 있었다. 특히, 교통사고 제보데이터의 개체명 인식 모델은 fastText, GloVe에 PCA로 작은 차원 단위로 차원축소를 하였을 때, 다른 Word Embedding에 비해 안정적으로 학습시간이 변화하면서 높은 성능을 보임을 알 수 있었다. 또한, GloVe는 PCA를 통한 작은 차원 단위 차원축소를 적용하였을 때, 모든 차원에서 성능은 향상되었고, 학습시간은 줄어 특정 도메인 모델에서는 PCA를 통한 차원축소의 효과를 보였다.

본 논문에서는 성능 및 학습시간을 동시에 평가할 수 있는 수리적 척도의 부재로 방법론의 성능을 특정 도메인에 의존적인 외재성 평가를 기준으로 진행하였다. 따라서 모델에서 Word Embedding을 학습 코퍼스와 함께 넣어 학습시킬 때, 외재성 평가 기준에서 성능 및 학습시간을 평가할 수 있는 수학적 척도에 관한 연구가 향후 필요하다. 또한 개체명 인식 모델의 효율적인 학습방법을 위해 본 논문에서 제안한 PCA를 적용한 Word Embedding과 window size를 고려하는 등의 향후 연구가 필요하다.

참고문헌

Bengio, Y., Ducharme, R., Vincent, P., and Jauvin, C. (2003), A Neural Probabilistic Language Model, *Journal of Machine Learning Research*, **3**, 1137-1155.

Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. (2017), Enriching Word Vectors with Subword Information, *Transactions of the Association for Computational Linguistics*, **5**, 135-146.

Colah.github.io. (2015), Understanding LSTM Networks-colah's blog. [online] Available at : http://colah.github.io/posts/2015-08-Understanding-LSTMs/ [Accessed 3 Sep. 2018].

Choi, Y.-S. and Cha, J.-W. (2016), Korean Named Entity Recognition and Classification using Word Embedding Features, *Journal of KIISE*, **43**(6), 678-685.

Graves, A. (2013), Generating Sequences With Recurrent Neural Networks, CoRR, arXiv preprint arXiv : 1308.0850.

Graves, A., Mohamed, A., and Hinton, G. E. (2013), Speech recognition with deep recurrent neural networks, 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, 6645-6649.

Ghannay, S., Favre, B., Esteve, Y., and Camelin, N. (2016), Word Embedding Evaluation and Combination, *LREC*.

Jeong, J.-Y., Mo, K.-H., Seo, S.-W., Kim, C.-Y., Kim, H.-D., and Kang, P.-S. (2018), Unsupervised Document Multi-Category Weight Extraction based on Word Embedding and Word Network Analysis : A Case Study on Mobile Phone Reviews, *Journal of Korean Institute of Industrial Engineers*, **44**(6), 442-451.

Jo, H.-S. and Lee, S.-G. (2017), Korean Word Embedding using fastText, *Proceedings of The Korea Information Science Society Conference*, 705-707.

Kim, Y.-S. and Lee, S.-W. (2018), Combinations of Text Preprocessing and Word Embedding Suitable for Neural Network Models for Document Classification, *Journal of KIISE*, **45**(7), 690-700.

Kim, H.-J., Cho, S.-Z., and Kang, P.-S. (2014), KR-WordRank : An Unsupervised Korean Word Extraction Method Based on WordRank, *Journal of the Korean Institute of Industrial Engineers*, **40**(1), 18-33.

Kim, J.-K., Lee, C.-H., and Lee, J.-M. (2018), Construct fashion shopping object name recognition dictionary for Natural Language Processing of intelligent chatbot in shopping mall, *Journal of the Korean Institute of Industrial Engineer Spring Conference*, 2744-2749.

Lebret, R., Legrand, J., and Collobert, R. (2013), Is deep learning really necessary for word embeddings?, *Idiap*, 1-9.

Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., and Dyer, C. (2016), Neural Architectures for Named Entity Recognition, *HLT-NAACL*, arXiv preprint arXiv : 1603.01360.

Lam, M. (2018), Word2Bits-Quantized Word Vectors, Corr. abs/1803.05651.

Lebret, R. and Collobert, R. (2014), Word Embeddings through Hellinger PCA, *EACL*, 482-490.

Lee, C.-K., Hwang, Y.-G., Oh, H.-J., Lim, S.-J., Heo, J., Lee, C.-H., Kim, H.-J., Wang, J.-H., and Jang, M.-G. (2006), Fine-Grained Named Entity Recognition using Conditional Random Fields for Question Answering, *AIRS,* 581-587.

Lee, C.-K. and Jang, M.-G. (2010), Named Entity Recognition with Structural SVMs and Pegasos algorithm, *Korean Journal of Cognitive Science*, **21**(4), 655-667.

Lee, C.-K. (2015), Named Entity Recognition using Long Short-Term Memory Based Recurrent Neural Network, Proceedings of the 2015 Conference of Korean Institute of Information Scientists and Engineers, 645-647.

Lee, D.-J., Lim, Y.-B., and Kwon, T.-K. (2018), Morpheme-based Efficient Korean Word Embedding, *Journal of KIISE*, **45**(5), 444-450.

Levy, O. and Goldberg, Y. (2014), Dependency-Based Word Embeddings, *ACL*, 302-308.

Levy, O., Goldberg, Y., and Dagan, I. (2015), Improving Distributional Similarity with Lessons Learned from Word Embeddings, *TACL*, **3**, 211-225.

Mikolov, T., Karafiat, M., Burget, L., Cernocky, J., and Khudanpur, S. (2010), Recurrent Neural Network based Language Model, Eleventh Annual Conference of the International Speech Communication Association, 581-587.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G., and Dean, J. (2013c), Distributed Representations of Words and Phrases and their Compositionality, Advances in neural information processing systems, arXiv preprint arXiv : 1310.4546.

Mikolov, T., Chen, K., Corrado, G. S. and Dean, J. (2013), Efficient Estimation of Word Representations in Vector Space, *CoRR*, arXiv preprint arXiv : 1301.3781.

Nooralahzadeh, F., Øvrelid, L., and Lønning, J. T. (2018), Evaluation of Domain-specific Word Embeddings using Knowledge Resources,

Proceedings of the Eleventh International Conference on Language Resources and Evaluation, 1438-1445.

Schnabel, T., Labutov, I., Mimno, D. M., and Joachims, T. (2015), Evaluation methods for unsupervised word embeddings, *EMNLP*, 298-307.

Seok, M., Song, H.-J., Park, C.-Y., Kim, J.-D., and Kim, Y.-S. (2015), Comparison of NER Performance Using Word Embedding, *Advanced Science and Technology Letters*, **120** (AIA 2015), 784-788.

Park, E.-J. and Cho, S.-Z. (2014), KoNLPy : Korean natural language processing in Python, *The 26th Annual Conference on Human and Cognitive Language Technology*, 133-136.

Patel, K. and Bhattacharyya, P. (2017), Towards Lower Bounds on Number of Dimensions for Word Embeddings, *IJCNLP,* 31-36.

Pennington, J., Socher, R., and Manning, C. D. (2014), Glove : Global Vectors for Word Representation, The 2014 Conference on Empirical Methods on Natural Language Processing.

Ratinov, L. and Roth, D. (2009), Design Challenges and Misconceptions in Named Entity Recognition, *In Proceedings of the Thirteenth Conference on Computational Natural Language Learning*, 147-155.

Raunak, V. (2017), Simple and Effective Dimensionality Reduction for Word Embeddings, arXiv preprint arXiv : 1708.03629v3.

Santos, C. N. and Guimaraes, V. (2015), Boosting Named Entity Recognition with Neural Character Embeddings, *NEWS@ACL*, 25-33.

Shlens, J. (2014), A Tutorial on Principal Component Analysis, arXiv preprint arXiv : 1404.1100.

Scholkopf, B., Smola, A., and Muller, K. R. (1997), Kernel principal component analysis, *ICANN*, 583-588.

Seo, D.-S., Mo, K.-H., Park, J.-S., Lee, G.-C., and Kang, P.-S. (2017), Word Sentiment Score Evaluation based on Graph-Based Semi-Supervised Learning and Word Embedding, *Journal of the Korean Institute of Industrial Engineers*, **43**(5), 330-340.

Song, E.-Y., Kim, K.-H., Choi, H.-R., and Lee, H.-C. (2017), Named Entity Recognition of Deep Learning based Traffic Accident Spot, 2017 *Korea Industrial Technology Association Fall Conference*, 2421-2425.

Yu, H.-Y. and Ko, Y.-J. (2017), Expansion of Word Representation for Named Entity Recognition Based on Bidirectional LSTM CRFs, *Journal of KIISE*, **44**(3), 306-313.

Young, T., Hazarika, D., Poria, S., and Cambria, E. (2017), Recent Trends in Deep Learning Based Natural Language Processing, arXiv preprint arXiv : 1708.02709.

Zhai, M., Tan, J., and Choi, J.-H. (2016), Intrinsic and extrinsic evaluations of word embeddings, In Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, *AAAI Press*, 4282-4283.

저자소개

송은영 : 아주대학교 교통시스템공학과, 산업공학과에서 2016년 학사학위를 취득하고 고려대학교 산업경영공학과 석사과정에 재학 중이다. 주요 연구분야는 AI, NLP, Text Mining, Data Mining이다.

최희련 : 단국대학교 산업공학과에서 1993년 학사학위를 취득하고 고려대학교 산업공학과에서 석사 및 박사수료를 하였다. 2007년부터 고려대학교 산업경영공학부 강사로 재직하고 있으며, 연구분야는 생산시스템, 인공지능 및 Blockchain이다.

이홍철 : 고려대학교 산업공학부에서 1983년 학사, Texas Arlington 대학교 산업공학과에서 1988년 석사학위를 취득하고 Texas A&M대학교에서 산업공학 박사학위를 취득하였다. 현재 BK21 Plus 제조, 물류 분야 사업팀장을 역임하고 있다. 1996년부터 고려대학교 산업경영공학부 교수로 재직하고 있다. 연구분야는 AI, 생산공학시스템, 시뮬레이션이다.