단어의 위치정보를 이용한 Word Embedding

황현선[O,†], 이창기[†], 장현기[††], 강동호[††]

강원대학교[†], SK 주식회사 C&C[††]

{hhs4322, leeck}@kangwon.ac.kr, hktopx77@gmail.com, eastsky21@sk.com

Word Embedding using word position information

Hyunsun Hwang[O,†], Changki Lee[†], HyunKi Jang[††], Dongho Kang[††]

Kangwon National University[†], SK holdings C&C[††]

요 약

자연어처리에 딥 러닝을 적용하기 위해 사용되는 Word embedding은 단어를 벡터 공간상에 표현하는 것으로 차원축소 효과와 더불어 유사한 의미의 단어는 유사한 벡터 값을 갖는다는 장점이 있다. 이러한 word embedding은 대용량 코퍼스를 학습해야 좋은 성능을 얻을 수 있기 때문에 기존에 많이 사용되던 word2vec 모델은 대용량 코퍼스 학습을 위해 모델을 단순화 하여 주로 단어의 등장 비율에 중점적으로 맞추어 학습하게 되어 단어의 위치 정보를 이용하지 않는다는 단점이 있다. 본 논문에서는 기존의 word embedding 학습 모델을 단어의 위치정보를 이용하여 학습 할 수 있도록 수정하였다. 실험 결과 단어의 위치정보를 이용하여 word embedding을 학습 하였을 경우 word-analogy의 syntactic 성능이 크게 향상되며 어순이 바뀔 수 있는 한국어에서 특히 큰 효과를 보였다.

주제어: Word Embedding, Word2vec, GLoVe

1. 서론

자연어처리를 위한 기존의 기계학습 모델에서는 단어를 One-hot의 형태로 표현하여 차원의 수가 많아져서 딥 러닝(Deep Learning)에 적용하기 어렵다는 문제점이 있었으나 word embedding을 이용하여 저 차원에서 단어를 표현하며, 딥 러닝의 사전학습 효과를 얻을 수 있게 되었다[1,2].

Word embedding은 단어를 벡터 공간상에 표현하는 것으로 One-hot의 표현 보다 저 차원으로 단어를 표현하며, 기존의 방법으로는 어려웠던 단어 유사도를 코사인 유사도로 쉽게 구할 수 있다는 장점이 있다. 이러한 word embedding을 구할 때에는 대용량 코퍼스로부터 NNLM(Neural Network Language Model)[1]을 학습시켜 얻게 된다. 그러나 이러한 NNLM 모델은 복잡하고 속도가 느려 대용량 코퍼스를 학습하는 데에 많은 시간이 소모된다는 단점이 있어, word embedding을 빠르게 학습 시키는 다양한 모델들이 연구되었다. 본 논문은 이러한 word embedding 학습 모델들 중에서 word2vec[3]의 CBOW(Continuous Bag-Of-Words) 모델을 단어의 위치정보를 볼 수 있게 개선한 모델을 제안한다. 실험 결과 기존의 word2vec 모델들 보다 syntactic 성능이 향상되는 것을 볼 수 있었다.

2. 관련 연구

Word embedding은 대용량 코퍼스를 이용하여 문장에서 사람이 사용하는 단어의 패턴을 학습하게 된다. 이에 따라 word embedding의 성능은 학습데이터의 양이 중요하게 되는데, 기존의 NNLM 모델로 대용량 코퍼스를 학습하기에는 많은 시간이 소모됨에 따라 다양한 word embedding 학습 모델들이 연구되었다.

2.1 Word2vec

Word2vec[3]은 기존의 NNLM 모델을 단순화 시키는 것과 동시에 비슷하게 사용되는 단어들은 벡터 공간상에서도 유사하게 표현되도록 학습하는 모델이다. 그림 1은 word2vec의 CBOW 모델과 skip-gram 모델을 나타낸 그림이다. CBOW 모델은 기존의 NNLM 모델에서 hidden layer를 제거하였고 projection layer에서 주위 단어 벡터들(그림1에서 w(t-2), w(t-1), w(t+1), w(t+2))을 단순히 element wise sum을 한 뒤 단순 신경망을 통하여 단어 w(t)를 예측하게 된다. Skip-gram 모델은 이와 반대로 단어 벡터 w(t)를 이용하여 주위 단어들(그림1에서 w(t-2), w(t-1), w(t+1), w(t+2))을 예측하게 된다. Word2vec은 이러한 방법을 통해 학습속도가 빠르며 문장에서 비슷하게 사용되는 단어들은 벡터 공간상에서도 유사하게 표현된다는 장점이 있으나, 문장에서의 단어 위치정보를 이용할 수 없다는 단점이 있다.

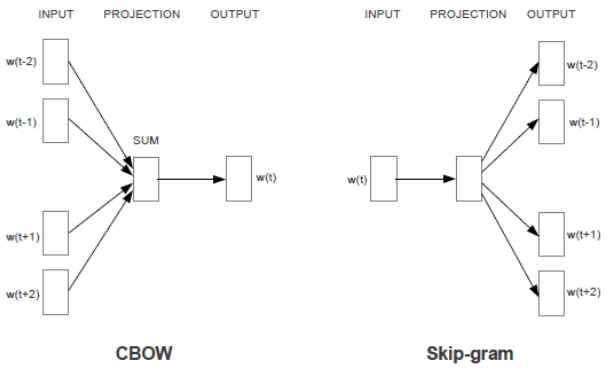

그림 1. Word2vec의 모델[3]

2.2 GLoVe

GloVe[4]는 word2vec과 달리 전체 학습데이터에서 등

장하는 단어들의 통계 정보를 활용하는 모델이다. GLoVe는 word embedding 학습 전에 학습데이터에서 각 단어들끼리의 한 문장에서 동시에 등장하는 확률들을 계산하며 그 예시는 표 1과 같다. 이때 단어의 수가 V개이면 해당 확률 표는 (V x V)의 크기를 가지게 된다. 이후 각각의 단어 벡터들은 서로 다른 단어에 대한 벡터 유사도를 표 1과 같은 동시 등장 확률과 비슷한 분포가 되도록 학습하게 된다. 이는 전체 학습데이터의 통계 정보를 가지고 학습하여 빠른 학습속도를 가지며 높은 성능을 낼 수 있다는 장점이 있으나, 학습 시 많은 메모리가 필요하다는 단점이 있다.

표 1. 한 문장에서 동시에 등장하는 단어들의 확률 예시

Probability and Ratio	$k = solid$	$k = gas$	$k = water$
$P(k\|ice)$	1.9×10^{-4}	6.6×10^{-5}	3.0×10^{-3}
$P(k\|steam)$	2.2×10^{-5}	7.8×10^{-4}	2.2×10^{-3}
$P(k\|ice)/P(k\|steam)$	8.9	8.5×10^{-2}	1.36

3. 단어의 위치정보를 이용한 Word Embedding

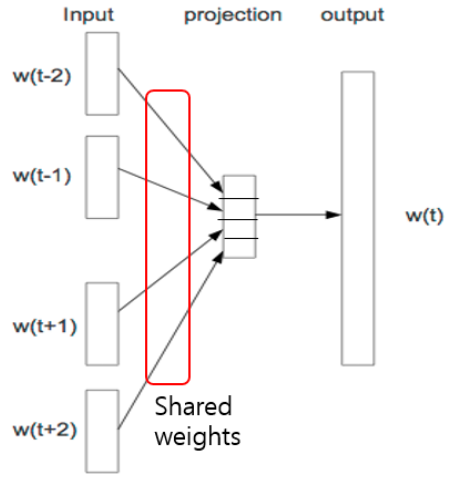

그림 2. 단어의 위치정보를 이용하는 word embedding 학습 모델

Word2vec의 CBOW 모델은 기존의 NNLM 모델을 단순화시켜 학습속도가 빠르고, 단어를 벡터 공간상에 효과적으로 표현할 수 있었다. 그러나 이 모델은 입력 단어 벡터들을 element wise sum을 하여 주위 단어들의 위치정보를 볼 수 없다는 단점이 있다. 이는 항상 어순이 고정되어있는 영어와 같은 언어에서는 영향이 적으나 한국어 같은 어순이 바뀔 수 있는 언어에 대해서는 문제가 생길 수 있다. 본 논문에서는 word2vec의 CBOW 모델을 단어의 위치정보를 볼 수 있게 적용한 모델을 제안한다.

그림 2는 본 논문에서 제안하는 단어의 위치정보를 이용하는 word embedding 학습 모델이다. 그림 1의 CBOW 모델에서 각 단어 벡터들을 element wise sum 하는 부분을 concatenate 하도록 수정하여, 입력으로 들어가는 주위 단어 벡터들(그림2에서 w(t-2), w(t-1), w(t+1), w(t+2))의 순서가 바뀌면 다른 입력으로 인식하게 만들어 문장에서의 해당 단어들의 위치정보를 볼 수 있게 만드는 효과가 있다. 그러나 이러한 방법은 CBOW 모델에 비하여 모델이 복잡해지는 효과가 있어 학습속도가 느려지게 된다.

4. 실험 및 결과

capital-common-countries
그리스/nnp 아테네/nnp 이라크/nnp 바그다드/nnp
그리스/nnp 아테네/nnp 태국/nnp 방콕/nnp
그리스/nnp 아테네/nnp 중국/nnp 베이징/nnp

capital-world
나이지리아/nnp 아부자/nng 가나/nnp 아크라/nnp
나이지리아/nnp 아부자/nng 알제리/nnp 알제/nnp
나이지리아/nnp 아부자/nng 요르단/nnp 암만/nnp

currency
알제리/nnp 디나르/nnp 앙골라/nnp 콴자/nng
알제리/nnp 디나르/nnp 아르헨티나/nnp 페소/nng
알제리/nnp 디나르/nnp 아르메니아/nnp 드램/nnp

city-in-state(korea)
수원시/nnp 경기도/nnp 화성시/nnp 경기도/nnp
수원시/nnp 경기도/nnp 성남시/nnp 경기도/nnp
수원시/nnp 경기도/nnp 광명시/nnp 경기도/nnp

그림 3. 한국어 word-analogy 평가데이터 예시

본 논문에서 제안한 단어의 위치정보를 이용한 word embedding 학습 모델과 다른 word embedding 학습 모델들과의 비교를 위해 한국어 데이터와 영어 데이터로 실험을 하였다. 한국어 학습 데이터는 10년치 금융 도메인 뉴스 기사를 크롤링 하여 형태소 분석기[5]를 사용하여 형태소 분석을 한 뒤 사용하였다. 영어 학습 데이터의 경우 Reuters에서 10년치 금융 도메인 뉴스 기사를 구매하여 NLTK python 패키지[6]의 토큰 분리기를 사용하여 토큰 분리 후 사용하였다. 이때 데이터 크기에 따른 성능 비교를 위해 영어 학습 데이터의 경우 소규모 데이터와 대규모 데이터로 나누어 실험을 하였다. 학습 파라미터의 경우 window size는 3으로, learning rate는 0.025으로, 학습 반복 횟수는 5로 고정하였다. GLoVe 모델은 다른 word embedding 학습 모델들과는 형태가 많이 달라 learning rate는 0.05으로, 학습 반복 횟수는 10으로 사용하였으며 word embedding의 차원은 50, 100, 200, 400으로 실험하였다. 성능 평가에는 word2vec의 word-analogy Top-1 accuracy를 사용하였으며, 한국어 word embedding 성능 평가의 경우, 영어 평가 데이터를 번역한 것과 한국어에 맞게 수정한 데이터를 사용하였으며 예시는 그림 3과 같다. 각각의 데이터 별 word embedding의 단어사전은 학습 모델들 모두 같으며, word-analogy의 threshold는 한국어는 100,000, 영어는 50,000을 사용하였다.

표 2는 word embedding 학습 모델 별 각각의 학습데이터에 대한 word-analogy 성능 표이다. 한국어 학습 데이터는 4억1900만 형태소 가량 되며, 영어 학습 데이터의 경우 각각 1억 8000만 토큰, 20억 9000만 토큰 가량이다. 실험 결과 본 논문에서 제안한 단어의 위치정보를 이용한 word embedding 학습 모델이 semantic 성능의 경우 CBOW 모델과 비슷하거나 낮았지만 syntactic 성능은

CBOW, skip-gram, GLoVe 모델보다 높은 것을 확인 할 수 있었다. 이는 단어의 위치정보를 추가 함에 따라 생긴 효과로 볼 수 있으며, 특히 영어에 비해 한국어의 성능이 크게 오른 것을 볼 수 있다. GLoVe 모델의 경우 전체 학습 데이터에 대한 통계 정보를 이용하여 semantic 성능이 높은 것을 볼 수 있었으며 단순히 단어 출현 빈도에 따른 학습으로 syntactic 성능이 낮은 것을 볼 수 있었다.

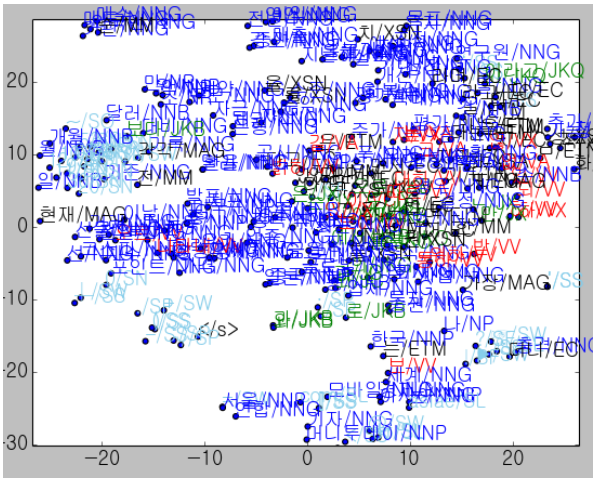

그림 4. CBOW로 학습한 한국어 word embedding

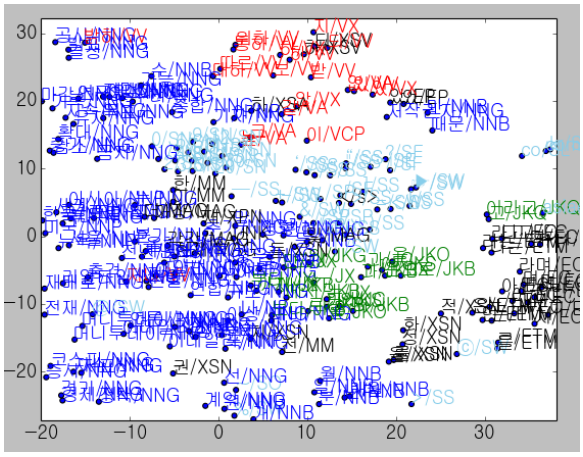

그림 5. 단어의 위치정보를 이용하여 학습한 한국어 word embedding

그림 4와 5는 50차원의 한국어 word embedding을 t-Distributed Stochastic Neighbor Embedding(t-SNE)을 이용하여 2차원으로 나타낸 것이다. 그림 4의 word embedding은 word2vec의 CBOW로 학습한 것이며 품사 구분 없이 분포가 나타나는 것을 볼 수 있다. 그러나 그림 5의 단어의 위치정보를 이용하여 학습 시킨 word embedding의 경우 같은 품사끼리 뭉쳐있는 것을 볼 수 있다. 이것은 단어의 위치정보를 넣어 단순한 단어 출현 빈도 뿐만 아니라 해당 단어의 문장에서의 위치적 역할까지 학습하였다고 볼 수 있다.

단어의 위치정보를 넣은 word embedding이 한국어 자연어처리에 미치는 영향을 확인하기 위해 [7]의 한국어 의존 구문 분석에 적용하였다. 한국어 뉴스기사 및 wiki 데이터(총 29.18억 형태소)를 학습 데이터로 사용하여 word2vec의 CBOW 모델과 본 논문에서 제안한 단어의 위치정보를 이용한 word embedding 학습 모델을 각각 학습시켰다. 의존 구문 분석 모델 및 학습데이터와 평가데이터는 [7]의 연구와 동일하게 사용하였다. 실험 결과 word2vec의 CBOW 모델로 학습한 word embedding을 사용한 경우 UAS 90.27%[7]의 성능이 나왔으나, 단어의 위치정보를 이용한 word embedding을 사용한 경우 UAS 90.49%로 한국어 의존 구문 분석의 성능이 0.22% 오른 것을 확인하였다.

5. 결론

본 논문은 기존의 word embedding 학습기중 하나인 word2vec의 CBOW 모델을 단어의 위치정보를 이용하게 개선하였다. 실험 결과 기존의 word embedding 학습 모델들 보다 syntactic에 강함을 보이며 한국어 형태소의 경우 벡터 공간상에서 비슷한 품사끼리 모이는 것을 볼 수 있었다. 향후 연구로는 각 word embedding 학습 모델들을 word-analogy 뿐만 아니라 개체명 인식, 의미역 결정 등 다른 자연어처리에 미치는 영향을 비교할 예정이다.

감사의 글

이 논문은 SK주식회사C&C의 지원을 받아 연구되었음.

참고문헌

[1] Yoshua Bengio, et al. A neural probabilistic language model. In NIPS, 2001.

[2] Ronan Collobert, et al. Natural language processing (almost) from scratch. The Journal of Machine Learning Research, 12, 2011.

[3] Tomas Mikolov, et al. Distributed representations of words and phrases and their compositionality. Advances in Neural Information Processing Systems. 2013.

[4] J. Pennington, R. Socher, C. D. Manning, “Glove: Global Vectors for Word Representation,” EMNLP2014

[5] 이창기, “Structural SVM을 이용한 한국어 띄어쓰기 및 품사 태깅 결합 모델”, 정보과학회논문지 : 소프트웨어 및 응용, 40(12), pp826-832, 2013.

[6] http://www.nltk.org/

[7] 이창기, 김준석, 김정희, "딥 러닝을 이용한 한국어 의존 구문 분석", 제26회 한글 및 한국어 정보처리 학술대회, pp. 87-91, 2014.

표 2. Word emedding 학습 모델 별 성능표

모델	차원	한국어(4억1900만 형태소)		영어(1억8000만 토큰)		영어(20억9000만 토큰)	
		semantic	syntatic	semantic	syntatic	semantic	syntatic
본 논문의 모델	50	2.65	10.16	6.41	34.03	15.69	47.62
	100	4.19	14.06	7.53	43.26	26.82	60.27
	200	4.19	13.28	7.72	**44.73**	30.29	**64.47**
	400	4.14	**16.41**	6.5	42.24	29.59	62.76
Word2vec(CBOW)	50	1.84	2.34	8.98	18.57	16.61	25.62
	100	2.82	1.56	10.10	27.41	22.96	36.86
	200	3.87	4.69	11.13	30.12	29.31	45.23
	400	4.10	4.69	9.64	30.30	30.12	48.62
Word2vec(skip-gram)	50	5.78	7.03	12.58	17.43	26.91	21.31
	100	7.00	5.47	16.79	24.67	39.96	29.92
	200	7.38	6.25	19.88	29.27	51.76	37.55
	400	6.51	6.25	19.5	28.77	**55.18**	43.32
GLoVe	50	5.88	3.12	7.02	19.37	10.65	26.26
	100	11.17	1.56	18.29	27.43	21.27	38.33
	200	16.60	2.34	24.70	30.90	32.67	45.27
	400	**18.69**	3.12	**27.13**	30.46	39.13	44.47