ISSN 2383-630X(Print) / ISSN 2383-6296(Online)
Journal of KIISE, Vol. 45, No. 5, pp. 444-450, 2018. 5
https://doi.org/10.5626/JOK.2018.45.5.444

형태소 기반 효율적인 한국어 단어 임베딩
(Morpheme-based Efficient Korean Word Embedding)

이 동 준 † 임 유 빈 † 권 태 경 ††
(Dongjun Lee) (Yubin Lim) (Ted "Taekyoung" Kwon)

요 약 기존의 word2vec(continuous bag-of-words 또는 skip-gram)이나 Glove 등의 단어 임베딩 모델은 단어의 구조나 단어 내부의 의미를 학습하지 못한다. 이는 한국어과 같은 교착어들을 학습하는데 있어서 큰 한계로 작용한다. 본 논문에서는, 기존의 skip-gram 모델을 확장하여 단어 벡터를 형태소들의 벡터의 합으로 정의하고, 형태소들의 벡터를 학습하는 새로운 모델을 제안하였다. 학습된 벡터의 성능을 평가하기 위하여 단어 유사도 평가와 단어 유추 평가를 수행하였고, 다른 자연어 처리 응용에 학습한 벡터를 사용함으로써 얼마나 성능이 향상되는지 실험하였다.
키워드: 단어 임베딩, 형태소 임베딩, 한국어, 인공신경망

Abstract Previous word embedding models such as word2vec and Glove are not able to learn the internal structure of words. This is a serious limitation for agglutinative languages with morphology such as Korean. In this paper, we propose a new model which is an expansion of the previous skip-gram model. This defines each word vector as a sum of its morpheme vectors and hence, learns the vectors of morphemes. To test the efficiency of our embedding, we conducted a word similarity test and a word analogy test. Furthermore, using our trained vectors on other NLP tasks, we tested how much performance actually had been enhanced.
Keywords: word embedding, morpheme embedding, Korean, neural network

1. 서 론

자연어 처리에 있어서 단어의 의미를 이해하고 표현하는 것은 가장 기초적이면서도 핵심적인 기반이 된다. 이를 위해서 단어를 연속적인 벡터 공간에 표현하는 벡터 공간 모델(vector space model)이 널리 사용된다. 이러한 벡터 표현은 유사한 단어를 서로 가까운 벡터로 매핑함으로써 자연어 처리에서 학습 알고리즘의 성능을 향상시키는데 큰 도움을 준다[1].

최근에는 인공신경망(neural network)를 기반으로 단어 벡터를 학습하는 방법들이 연구되었으며[2], 가장 대표적인 모델로는 word2vec의 CBOW(Continuous Bag-Of-Words)와 skip-gram[3], Glove[4]가 있다. 이러한 모델들은 영어에 대해 상당히 효율적인 단어 벡터를 학습한다고 알려져 있다. 하지만 이들의 한계점은 모든 단어에 대해서 서로 독립적인 벡터를 학습하기 때문에, 서로 다른 단어들이 공유하는 구조나 단어 내부의 의미에 대해 학습할 수 없다는 것이다. 이러한 한계 때문에 기존의 모델들은 단어 내부의 구조가 풍부한 한국어와 같은 교착어에 대해 효율적으로 단어 벡터를 학습할 수 없다.

본 논문에서는 이러한 한계점을 극복하기 위해, 기존의 skip-gram 모델을 확장하여 각 단어 벡터를 해당 단어를 이루는 형태소들의 벡터의 합으로 정의하고, 형태소들의 벡터를 학습하는 새로운 모델을 제안하였다.

† 비 회 원 : 서울대학교 컴퓨터공학부
djlee@mmlab.snu.ac.kr
yblim@idb.snu.ac.kr
†† 종신회원 : 서울대학교 컴퓨터공학부 교수
(Seoul Nat'l Univ.)
tkkwon@snu.ac.kr
(Corresponding author임)
논문접수 : 2017년 7월 19일
(Received 19 July 2017)
논문수정 : 2018년 1월 27일
(Revised 27 January 2018)
심사완료 : 2018년 1월 30일
(Accepted 30 January 2018)

정보과학회논문지 제45권 제5호(2018. 5)

이를 통해 모델이 단어 내부의 의미를 학습할 수 있도록 하였다. 학습된 벡터의 효율성을 측정하기 위해 단어 유사도 평가(word similarity test)와 단어 유추 평가(word-analogy test)를 수행하였다. 또한 학습된 벡터가 다른 자연어 처리 응용의 학습 알고리즘에 얼마나 성능 향상을 가져오는지 실험하기 위하여 기존의 문장 분류(text classification) 모델[5]에서 입력(input) 단어 벡터만 변화시켜 분류 정확도(classification accuracy)를 비교하였다.

본 논문의 구성은 다음과 같다. 2장에서 기존의 연구들을 소개하고, 3장에서 형태소 기반의 새로운 모델의 구조와 이의 장점에 대해 설명한다. 4장에서는 학습된 벡터의 성능을 평가하고, 5장에서는 결론 및 향후 연구에 대해 다룬다. 본 논문의 구현 및 실험 데이터셋은 Gtihub에 오픈소스로 공개되어 있다.

2. 관련 연구

대부분의 단어 임베딩 방법은 비슷한 문맥을 가지는 단어가 비슷한 의미를 갖는다는 분포 가설(distributional hypothesis)에 기반한다. 단어 임베딩 모델은 크게 단어가 함께 출현하는 횟수를 세는 빈도 기반(count based) 모델과 단어를 그 주변 단어들로부터 예측하는 예측(predictive) 모델로 나누어진다. 이 중에서는 예측 모델이 더 좋은 성능을 보이는 것으로 알려져 있다[6]. 특히 인공 신경망(neural network) 기반의 예측 모델이 최근에 가장 각광받고 있다[2].

인공 신경망 기반의 예측 모델 중 대표적인 것으로는 word2vec(CBOW 또는 skip-gram), Glove가 있다. 이들 중에서는 skip-gram 모델의 성능이 가장 뛰어난 것으로 알려져 있다[2].

2.1 Skip-gram 모델[1]

Skip-gram 모델은 현재 주어진 단어에 대하여 주변에 등장하는 단어들을 예측한다. 즉, skip-gram 모델의 목적 함수(loss function)는 다음과 같이 정의된다. 학습 말뭉치(training corpus)가 단어의 배열 $w_1, \dots w_T$이고, 윈도우 크기(window size)가 s일 때,

$$J(\Theta) = \frac{1}{T}\sum_{t=1}^{T}\sum_{-s\le j\le s, j\neq 0} logP(w_{t+j}|w_t).$$

위에서 주어진 단어 c에 대해 단어 o가 주변에 등장할 확률은 softmax 함수로 다음과 같이 정의할 수 있다. u, v가 각각 단어에 대한 입력(input) 벡터와 출력(output) 벡터이고, W가 학습 말뭉치에 등장하는 서로 다른 어휘의 개수 일 때,

$$P(w_o|w_c) = \frac{\exp(u_o^T v_c)}{\sum_{w=1}^{W}\exp(u_w^t v_c)}.$$

하지만 이처럼 softmax 함수를 사용하면, $\nabla logP(w_{t+j}|w_t)$를 계산하는데 매우 큰 값인 W에 비례하는 연산 비용(computational cost)이 들기 때문에, 학습이 현실적으로 불가능하다. 따라서 위 확률을 근사적으로 계산하기 위하여 negative sampling을 사용한다. Negative sampling을 사용하면 목적 함수(loss function)은 다음과 같이 정의된다. k가 negative sample의 개수, P(w)가 noise distribution일 때,

$$J(\theta) = \frac{1}{T}\sum_{t=1}^{T} J_t(\theta),$$

$$J_t(\theta) = \log\sigma(u_o^T v_c) + \sum_{i=1}^{k}\mathbb{E}_{w_i\sim P(w)}\left[\log\sigma(-{u_{w_i}}^T v_c)\right].$$

2.2 Skip-gram 모델의 한계

Skip-gram 모델은 학습 말뭉치에 등장하는 각각의 어휘에 대하여 서로 독립적인 벡터를 학습하기 때문에, 서로 다른 단어들이 공유하는 구조적 정보는 학습할 수 없다는 한계가 존재한다. 이러한 문제점은 한국어와 같은 교착어에 대해 더욱 큰 한계로 작용한다. 한국어에서 단어(해당 논문에서 단어는 띄어쓰기 단위인 어절을 의미한다.)는 하나의 어근과 하나 이상의 접사로 이루어진다. 예를 들어, '한국어' 라는 어근에 대하여 '한국어는', '한국어가', '한국어도', '한국어만' 등 다양한 형태의 단어가 존재한다. 기존의 skip-gram 모델은 이들에 대해 모두 서로 독립적인 벡터를 학습하기 때문에 매우 비효율적이다.

이러한 한계점 때문에 단어의 구조적 정보에 대해 학습하기 위한 다양한 방법들이 제시되었다. [7]에서는 각각의 단어를 character n-gram들의 합으로 정의하고 character n-gram에 대한 벡터를 학습하는 방법을 제안하였다. [8]에서는 각각의 단어를 접두사, 어근, 접미사로 분리하고, RNN(Recursive Neural Network)을 사용하여 단어 벡터를 학습하는 방법을 제안하였다. [9]에서는 형태가 유사한 단어들이 비슷한 벡터를 가지도록 사전 지식(prior knowledge)을 적용하는 방법을 제안하였다.

2.3 한국어 단어 임베딩에 대한 연구

한국어 단어 임베딩에 대한 연구는 주로 word2vec이나 Glove 등의 기존의 모델을 적용하되, 학습 전후에 추가적인 처리를 통하여 성능을 향상시키는 방법들에 대해 진행되었다. [10]에서는 기존의 모델을 그대로 적용한 후에, 서로 같은 어근을 가지는 단어들의 벡터를 합성하여 어근에 대한 벡터를 생성하였다. [11]에서는 형태소 분석기를 사용하여 독립적으로 의미를 지니지 못하는 형태소를 모두 제거한 후에, 기존의 모델들을 적용하였다. 이들은 모두 결국 어근에 대한 벡터만 학습하기 때문에, 어근과 어미, 조사 등의 관계나 각 형태소의 의미에 대해 학습하지 못해 부분 문자열 정보를 제대로 활용하지 못한다는 한계가 있다.

3. 형태소 기반 단어 임베딩 모델

Skip-gram 모델은 말뭉치(corpus)에 등장하는 모든 각각의 어휘에 대하여 독립적인 벡터를 할당(assign)하기 때문에, 서로 다른 단어가 서로 공유하는 단어의 구조적 정보에 대해서는 학습이 불가능하다.

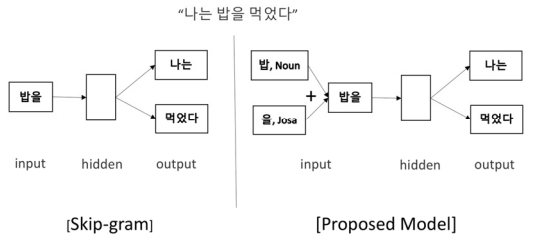

그림 1 skip-gram 모델과 제안된 모델 비교
Fig. 1 Comparison of skip-gram and the proposed model

따라서 본 논문에서는 그림 1과 같이 단어의 구조적 정보를 학습하기 위하여 형태소 단위를 사용하는, 기존의 skip-gram 모델을 확장한 단어 임베딩 모델을 제안하였다. 각 단어 벡터는 그 단어를 이루는 형태소들의 벡터의 합으로 정의된다. 예를 들어, 단어 '밥을'의 벡터는 형태소 벡터 '밥'(Noun)과 '을'(Josa)의 합으로 정의된다. (서로 다른 품사의 형태소를 구분하기 위하여 각각의 형태소가 품사태그를 포함하도록 하였다.) 그리고 모델은 각 형태소의 벡터를 학습하게 된다. 해당 모델을 수식으로 표현하면 다음과 같다.

단어 w를 이루는 형태소가 $\{m_1, ..., m_n\}$이고 각 형태소의 벡터가 $\{z_{m_1}, ..., z_{m_n}\}$일 때, w의 input 벡터 u_w는 다음과 같다.

$$u_w = \sum_{i=1}^{n} z_{m_i}$$

따라서 주어진 단어 c에 대하여 단어 o가 주변에 등장할 확률은 softmax 함수로 다음과 같이 정의된다. z_{c_i}, z_{o_j}가 각각 단어 c와 단어 o의 형태소 벡터들일 때,

$$\mathrm{P}(w_c|w_o) = \frac{\exp((\sum z_{o_j})^T v_c)}{\sum_{w=1}^{W} \exp((\sum z_{c_i})^T v_c)}.$$

Skip-gram과 마찬가지로 현실적인 연산 비용(computational cost)으로 확률을 계산하기 위하여 negative sampling을 적용하면, 목적 함수(loss function)은 다음과 같이 정의된다.

$$\mathrm{J}(\theta) = \frac{1}{\mathrm{T}} \sum_{t=1}^{\mathrm{T}} J_t(\theta),$$

$$J_t(\theta) = \log \sigma\left(\left(\sum z_{o_j}\right)^T v_c\right) + \sum_{i=1}^{k} \mathbb{E}_{w_i \sim P(w)} [\log \sigma(-(\sum z_{w_{i_i}})^T v_c)].$$

해당 모델이 가지는 장점은 다음과 같다. 첫째, 모델이 학습하여야 하는 파라미터의 수를 현저하게 감소시킨다. 본 논문에서 사용한 학습 말뭉치(training corpus)에서 등장한 어휘의 개수는 약 500만개였으나, 서로 다른 형태소의 개수는 약 28만개였다. 따라서 형태소 단위로 벡터를 학습함으로써 단어 단위로 벡터를 학습할 때에 비해 학습해야 하는 벡터의 수가 현저하게 줄어든다.

둘째, 단어를 형태소의 합으로 정의함으로써 단어 간의 선형성(linearity)를 보장할 수 있다. 예를 들어, 단어 벡터의 연산에서 '밥을' - '밥' + '물' = '물을', '먹었다' - '먹다' + "맛있다" = "맛있었다" 등이 보장된다. 이는 단어 사이의 구문론적(syntactic) 관계를 반영하는데 아주 효과적이다. 또한 학습 말뭉치에 등장하지 않은 단어에 대해서 기존의 모델들(word2vec, Glove)은 학습이 불가능하지만, 제안된 모델로는 형태소 벡터의 합으로 단어 벡터를 생성할 수 있다.

셋째, 띄어쓰기 오류에 대해 강하다. 한국어의 띄어쓰기 오류는 아주 흔하게 발생하기 때문에 이는 매우 중요하다. 기존 단어 기반의 학습 모델로는 이러한 띄어쓰기 오류에 대처할 수 없다. 예를 들어, "내용이너무좋아요"라는 리뷰가 있다. 기존의 모델로는 이 문장이 하나의 단어로 취급되고, 이러한 단어가 학습 말뭉치(training corpus)에 등장하지 않기 때문에 해당 리뷰를 처리할 수 없다. 하지만 제안된 모델에서는 해당 리뷰에 대하여 "내용" + "이" + "너무" + "좋" + "아요"와 같이 형태소 벡터의 합으로 벡터를 할당할 수 있다.

4. 실험

4.1 학습 말뭉치(training corpus) 및 구현(implementation)

학습을 위한 말뭉치로 인터넷 뉴스 기사를 크롤링(crawling)하여 사용하였다. 말뭉치는 총 약 2억2천개의 단어로 구성되었으며, 말뭉치에 등장하는 어휘의 수는 약 5백만개, 형태소의 수는 약 28만개였다.

본 논문에서는 제안한 모델과 비교를 위하여, 기존의 단어 단위 skip-gram과 형태소 단위 skip-gram을 사용하였다. 형태소 단위 skip-gram에서는 형태소 분석기를 사용하여 형태소 단위로 말뭉치를 구성한 후에 기존의 skip-gram 모델을 적용하였다. 본 논문에서 제안한 모델과 기준 모델(baseline model)들은 모두 tensorflow [12]로 구현되었다. 그리고 단어를 형태소로 분해하기 위하여 트위터(twitter) 형태소 분석기[13]를 사용하였다. 학습에 사용된 하이퍼파라미터(hyperparameter)들은 다음과 같다. 형태소 및 단어 벡터의 차원(dimension)은 200으로, 윈도우 크기(window size)는 5로 정의하였다. 단어의 최소 등장 빈도수는 50으로 정의하였으며, 빈도가 높은 단어에 대한 샘플링 비율(sampling rate)[1]은

0.0001을 사용하였다. 목적 함수 최적화(loss function optimization)를 위하여 tensorflow의 gradient descent optimizer를 사용하였고, 4번의 세대(epoch)를 진행하였다.

최소 등장 빈도수로 필터링한 후, 말뭉치에 등장하는 어휘의 수는 약 20만개, 형태소의 수는 약 6만개였다.

4.2 평가 방법

단어 임베딩의 성능을 평가하는 방법은 크게 내재성 평가(intrinsic test)와 외재성 평가(extrinsic test) 2가지로 분류된다[14]. 내재성 평가는 단어 벡터를 직접적으로 평가하는 방식으로, 생성된 단어 벡터간의 구문론적(syntactic), 의미론적(semantic) 관계가 얼마나 잘 학습되었는지 평가한다. 대표적인 방법으로는 단어 유사도 평가(word similarity test)와 단어 유추 평가(word analogy test)가 있다. 외재성 평가에서는 모델이 학습한 단어 임베딩을 다른 자연어 처리 문제에 적용하였을 때 어떤 성능 향상을 가져오는지 평가한다.

논문에서는 내재성 평가를 위하여 단어 유사도 평가와 단어 유추 평가를 수행하였고, 형태소 벡터를 2차원 좌표에 시각화함으로써 정성적 분석(qualitative analysis)를 수행하였다. 외재성 평가 방식으로는 [5]에 제안된 CNN(Convolutional Neural Network) 기반 문장 분류 모델에 본 논문에서 학습한 단어 임베딩을 적용하여 실험하였다.

4.3 내재성 평가(Intrinsic Test)

4.3.1 단어 유사도 평가(Word Similarity Test)

단어 유사도 평가는 일련의 단어 쌍을 미리 구성한 후에, 사람이 평가한 점수와 단어 벡터간의 코사인 유사도(cosine similarity)간의 비교를 통해 단어 사이의 의미론적 관계를 얼마나 잘 학습하였는지 측정하는 방법이다. 본 논문에서는 단어 유사도 평가를 위한 한국어 테스트셋이 존재하지 않아, 영어 단어로 구성된 WordSim 353[15] 테스트셋을 한국어로 번역하여 사용하였다. WordSim353은 353개의 단어 쌍으로 구성되어 있으며, 두 단어 사이의 의미론적 관계가 여러 전문가들에 의해 평가된 값이다. Soccer - Football 같이 한국어로 번역할 때 본래의 의미를 유지하지 못하는 경우는 제외하였다. 데이터셋의 예시는 표 1에 나타나 있다. WordSim353 테스트셋의 단어들은 모두 명사이기 때문에 모든 단어가 하나의 형태소로 구성되어 있다.

표 1 단어 유사도 평가 데이터셋 예시
Table 1 Word similarity test dataset examples

Word 1	Word 2	Score
Computer	News	4.47
Tiger	Cat	8.00
Mars	Water	2.94

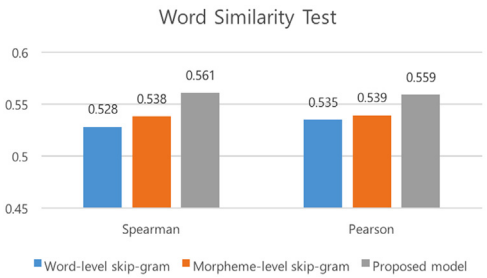

그림 2 단어 유사도 평가 결과
Fig. 2 Word similarity test result

사람이 평가한 점수와 학습한 단어 벡터간의 코사인 유사도를 비교하기 위하여 Spearman 상관계수와 Pearson 상관계수를 계산하여 성능을 측정하였다. 그림 2는 skip-gram 모델과 제안한 모델의 단어 유사도 평가 결과이다. Spearman, Pearson 상관계수 모두 제안한 모델이 기존의 skip-gram 모델에 비해 더 높은 성능을 보였다.

4.3.2 단어 유추 평가(Word Analogy Test)

단어 유추 평가에서는 단어 임베딩이 단어 사이의 의미 관계를 얼마나 잘 학습했는지 평가한다. 예를 들어, "갑과 을의 관계는 병과 정의 관계와 같다"라는 의미론적 유추에서 "'갑' - '을' + '정' = ?" 이라는 질의에 대해 '병'을 도출해낼 수 있는지 평가한다. 실제 유추 단계에서는 완전히 일치하는 벡터를 찾기보다는 질의에 가장 코사인 유사도가 높은 단어 벡터를 찾게 된다.

단어 유추 평가를 위한 테스트셋으로는 [3]에서 제안된 Google analogy 테스트셋을 참고하여 제작하였다. Google 테스트셋은 크게 구문론적 질의와 의미론적 질의로 분류되는데, 의미론적 질의는 번역하여 사용하는데 문제가 없으나, 한국어와 영어의 구문론적 구조가 전혀 상이하기 때문에 구문론적 질의는 사용할 수 없다. 따라서 자체적으로 한국어의 구문론적 특성을 반영한 테스트셋을 자체적으로 생성하여 평가에 사용하였다. 테스트셋은 의미론적 질의 420개와 구문론적 질의 840개, 총 1,260개의 질의로 구성하였다. 테스트셋의 예시는 표 2에 나타나 있다.

그림 3과 표 3은 skip-gram 모델과 제안한 모델의 단어 유추 평가 결과를 나타내고 있다. 형태소 단위 skip-gram에서는 형태소 단위로만 벡터를 학습하기 때문에 구문론적 관계에 대해서는 실험할 수 없다. 구문론적 관계에서 실험되는 단어들은 두개 이상의 형태소로 이루어져있기 때문이다. 의미론적 질의와 구문론적 질의 모두에 대하여 제안한 모델이 더 높은 정확도를 보였다. 제안한 모델이 의미론적 질의에 대해서 정확도가 10% 이상 높았으며, 구문론적 질의에 대해서는 정확도가 30% 이상 높았다.

표 2 단어 유추 평가 데이터셋 예시
Table 2 Word analogy test dataset examples

Semantic relationships	Word pairs
Capital - Country	파리-프랑스, 독일-베를린
Man - Woman	신랑-신부, 아들-딸
Syntactic relationships	**Word pairs**
Noun - Noun + Josa(조사)	밥-밥을, 물-물을
Adjective - Adverb	부드러운-부드럽게, 용감한-용감하게
Verb - Verb + 습니다	강조했다-강조했습니다, 잡혔다-잡혔습니다

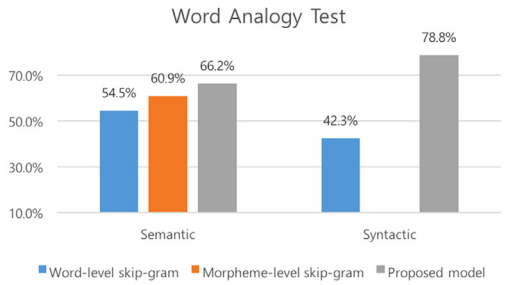

그림 3 단어 유추 평가 결과
Fig. 3 Word analogy test result

표 3 구문론적 관계에 대한 단어 유추 평가 결과
Table 3 Syntactic word analogy test result

Syntactic relationships	Skip-gram	Proposed model
Noun - Noun + Josa	53.3%	100%
Noun + Josa - Noun + different Josa	58.1%	93.3%
Adjective - Adverb	20.9%	19.0%
Verb - Verb+습니다	36.7%	98.1%

4.3.3 형태소 벡터 시각화(Visualization)

정성적 분석을 위하여, 학습된 형태소 벡터에 대해 PCA(Principal Component Analysis)를 사용하여 2차원 공간에 시각화하였다. 다섯개의 품사(조사, 어미, 숫자, 명사, 동사)에 대하여 각각 7개의 임의의 형태소를 선택하여 시각화를 진행하였다. 그림 4는 모델이 품사에 따라 형태소를 서로 다른 공간에 잘 매핑시키고 있음을 보여준다. 독립적으로 의미를 가지는 형태소(동사, 명사)와 의미를 가지지 않는 형태소(조사, 어미)가 확연히 구분되었으며, 숫자/조사/어미 간의 구분도 확연하게 나타났다.

4.4 외재성 평가(Extrinsic Test) - 영화 감상평의 감정 분류

효과적으로 학습된 단어 임베딩은 다양한 자연어 처리 응용에 적용되어 성능 향상에 기여할 수 있다. 외재

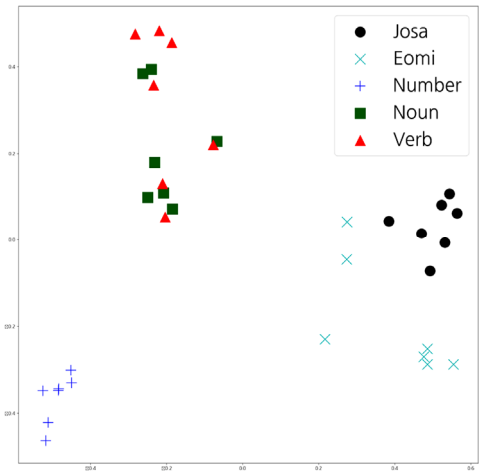

그림 4 형태소 벡터 시각화
Fig. 4 Visualization of morpheme vectors

성 평가는 모델이 학습한 단어 임베딩을 다른 자연어 처리 문제의 입력(input) 값으로 주었을 때, 주어진 벡터들이 성능에 주는 영향을 측정한다.

본 논문에서는 외재성 평가를 위하여 영화 감상평의 감정 분류 문제에 대하여 제안한 모델이 학습한 단어 임베딩이 얼마나 성능 향상을 가져오는지 실험하였다. 데이터셋으로는 약 20만개의 감상평으로 구성된 NAVER 영화 감상평 데이터셋[16]을 사용하였다. 실험에서는 기존에 제안된 단순하면서도 다양한 도메인에서 뛰어난 성능을 보인 그림 5의 CNN(Convolutional Neural Network) 기반의 문장 분류 모델[5]을 사용하였으며, 각 감상평이 긍정인지 부정인지 분류하는 것을 목적으로 하였다.

Skip-gram 모델은 각각의 단어를 독립적인 벡터로 학습하기 때문에, 학습 말뭉치에 등장하지 않는 단어에 대해서는 대응할 수 없다. 하지만 제안한 모델은 형태소 벡터를 학습하고 그를 기반으로 단어 벡터를 생성하기 때문에 학습 말뭉치에 등장하지 않는 단어에 대해서도 벡터를 생성할 수 있다. 이는 특히 한국어의 띄어쓰기 오류가 아주 흔하게 발생하기 때문에 중요하다. 실제로 NAVER 영화 감상평 데이터셋에서 띄어쓰기 오류가 존재하는 리뷰의 비율이 35% 이상이었다.

표 4는 제안한 모델과 skip-gram 모델로 학습 말뭉치에서 학습한 벡터의 수, 그리고 이를 토대로 NAVER 영화 감상평 데이터셋에서 처리할 수 있는 단어의 수를 나타내고 있다. 제안한 모델이 Skip-gram에 비하여 학습한 벡터의 수는 약 1/4에 불과했지만, 처리할 수 있었던 단어의 수는 약 5배였다.

그림 6은 영화 감상평 감정 분류 정확도에 대한 결과이다. [5]에서 실험했던 것과 마찬가지로, 초기 단어벡터

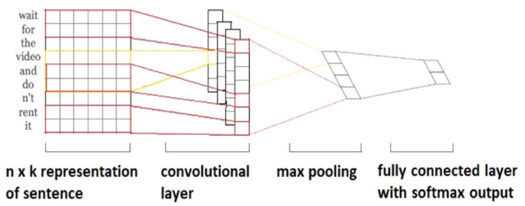

그림 5 CNN 기반 문장 분류 모델[5]
Fig. 5 CNN based text classification model [5]

표 4 네이버 영화 감상평 데이터셋에서 처리한 단어 수
Table 4 The number of words handled in Naver movie sentiment corpus

	Proposed model	Skip-gram
The number of learned vectors	51,987	215,764
The number of executed words	256,049	52,155
The rate of executed words	69%	14%

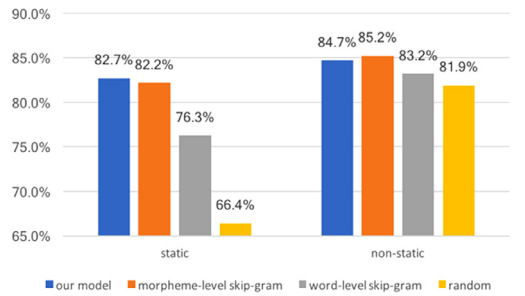

그림 6 감정 분류 결과
Fig. 6 Sentiment classification result

를 업데이트 하지 않는 정적(static) 실험과 초기 단어 벡터를 업데이트 하는 동적(nonstatic) 실험을 진행하였다. 정적 실험의 경우, 제안한 모델에서 학습한 단어 벡터를 사용한 경우가 skip-gram 모델에서 학습한 벡터를 사용한 경우에 비해 훨씬 높은 분류 정확도를 보였다. 단어 벡터가 업데이트 되는 동적 실험의 경우에도 제안한 모델에서 학습한 단어 벡터를 초기값으로 설정함으로써 유의미한 정확도 향상을 가져왔다.

5. 결 론

본 논문에서는 교착어의 특성을 가지는 한국어에 대해서 효율적인 단어 임베딩을 학습하기 위하여 기존의 skip-gram 모델을 확장한 새로운 모델을 제안하였다. 단어 내부의 구조를 반영하기 위하여 각 단어 벡터를 단어를 이루는 형태소들의 벡터의 합으로 정의하고 형태소의 벡터를 학습하였다. 학습된 벡터의 성능은 단어 유사도 평가(word similarity test)와 단어 유추 평가(word analogy test)를 사용하여 평가하였다. 두 테스트 모두 제안된 모델이 기존 skip-gram의 성능을 능가하였다. 또한, 제안된 모델로 학습된 벡터가 다른 NLP task에 효율적으로 사용될 수 있는지 실험하기 위하여 CNN 기반의 문장 분류(text classification) 모델[5]을 사용하였다. 실험 결과, 기존 skip-gram으로 학습된 벡터를 사용했을 때 보다 제안된 모델로 학습된 벡터를 사용했을 때 분류 정확도(classification accuracy)가 향상되는 것을 확인하였다.

본 논문에서 제안한 모델은 한국어 뿐 아니라, 어떠한 언어에도 적용할 수 있다. 특히 단어의 구조가 많은 언어들에 대해 더욱 효율적일 것으로 기대된다.

향후 연구로는 단어 내부의 구조가 많은 다른 언어들에 대해 같은 모델을 적용하여 학습하고 실험할 것이다. 또한 형태소 단위 뿐 아니라 음절(syllable) 또는 자모(character) 단위의 정보를 함께 고려하여 임베딩의 성능을 향상시키는 방법에 대해 연구할 것이다.

References

[1] Mikolov, Tomas, et al., "Distributed representations of words and phrases and their compositionality," *Advances in neural information processing systems*, pp. 3111-3119, 2013.

[2] Levy, Omer, Yoav Goldberg, and Ido Dagan, "Improving distributional similarity with lessons learned from word embeddings," *Transactions of the Association for Computational Linguistics*, Vol. 3, pp. 211-225, 2015.

[3] Mikolov, Tomas, et al., "Efficient estimation of word representations in vector space," *arXiv preprint arXiv*: 1301.3781, 2013.

[4] Pennington, Jeffrey, Richard Socher, and Christopher D. Manning, "Glove: Global Vectors for Word Representation," *EMNLP*, Vol. 14, pp. 1532-1543, 2014.

[5] Kim, Yoon, "Convolutional neural networks for sentence classification," *arXiv preprint arXiv:1408.5882*, 2014.

[6] Baroni, Marco, Georgiana Dinu, and Germán Kruszewski, "Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors," *ACL (1)*, pp. 238-247, 2014.

[7] Bojanowski, Piotr, et al., "Enriching word vectors with subword information," *arXiv preprint arXiv*: 1607.04606, 2016.

[8] Lazaridou, Angeliki, et al., "Compositionally Derived Representations of Morphologically Complex Words in Distributional Semantics," *ACL (1)*, pp. 1517-1526, 2013.

[9] Cui, Qing, et al., "Knet: A general framework for learning word embedding using morphological knowledge," *ACM Transactions on Information Systems (TOIS)*, Vol. 34, No. 1, No. 4, 2015.
[10] Whan, Sook, "A Study on Word Vector Models for Representing Korean Semantic Information," *Journal of the Korean Society of Speech Sciences*, Vol. 7, No. 4, pp. 41-47, 2015. (in Korean)
[11] Sanghyuk Choi et al., "On Word Embedding Models and Parameters Optimized for Korean," *HCLT*, No. 15, 2016. (in Korean)
[12] Tensorflow, [Online]. Available: https://www.tensorflow.org/
[13] Twitter-korean-text, [Online]. Available: https://github.com/twitter/twitter-korean-text
[14] Schnabel, Tobias, et al., "Evaluation methods for unsupervised word embeddings," *EMNLP*, pp. 298-307, 2015.
[15] WordSim353, [online]. Available: http://alfonseca.org/eng/research/wordsim353.html
[16] Naver sentiment movie corpus, [Online]. Available: https://github.com/e9t/nsmc/

부 록

본 논문의 모델에 대한 구현, 그리고 실험에 사용된 데이터셋은 GitHub에 오픈소스로 공개되어 있다. (https://github.com/dongjun-Lee/kor2vec)

형태소 벡터 학습 논문에서 제시한 학습을 위한 모델의 구현이 공개되어 있다. 해당 소스코드를 사용하여 임의의 한국어 corpus에 대해 형태소들에 대한 벡터를 학습할 수 있다. 또한 윈도우 크기(window size)나 단어 벡터의 차원(dimension) 등에 대한 hyperparameter를 변경하여 학습할 수 있다.

테스트 데이터셋 본 논문에서 사용한 단어 유사도 평가(word similarity test) 및 단어 유추 평가(word analogy test)에 사용된 테스트 데이터셋 및 코드가 공개되어 있다. 또한 PCA를 이용하여 2차원에 형태소 벡터를 시각화하는 코드 또한 공개되어 있다.

학습된 형태소 벡터 본 논문에서 실험에 사용한, 인터넷 뉴스 학습 말뭉치(training corpus)로 학습된 형태소들의 벡터가 공개되어 있다.

이 동 준
2016년 서울대학교 컴퓨터공학부 학사
2018년 서울대학교 컴퓨터공학부 석사
2018년~현재 SAP 재직. 관심분야는 기계학습, 딥러닝, 자연어 처리

임 유 빈
2015년 한동대학교 전산전자공학부 학사
2015년~현재 서울대학교 컴퓨터공학부 석박사 통합과정. 관심분야는 딥러닝, In-DB 분석

권 태 경
1993년 서울대학교 컴퓨터공학 학사. 1995년 서울대학교 컴퓨터공학 석사. 2000년 서울대학교 컴퓨터공학 박사. 2004년~현재 서울대학교 컴퓨터공학부 교수