부분단어와 품사 태깅 정보를 활용한 형태소 기반의 한국어 단어 벡터 생성

Morpheme-based Korean Word Vector Generation Considering the Subword and Part-Of-Speech Information

저자 (Authors)
윤준영, 이재성
Junyoung Youn, Jae Sung Lee

출처 (Source)
정보과학회논문지 47(4), 2020.4, 395-403 (9 pages)
Journal of KIISE 47(4), 2020.4, 395-403 (9 pages)

발행처 (Publisher)
한국정보과학회
The Korean Institute of Information Scientists and Engineers

URL
http://www.dbpia.co.kr/journal/articleDetail?nodeId=NODE09325339

APA Style
윤준영, 이재성 (2020). 부분단어와 품사 태깅 정보를 활용한 형태소 기반의 한국어 단어 벡터 생성. 정보과학회논문지, 47(4), 395-403.

ISSN 2383-630X(Print) / ISSN 2383-6296(Online)
Journal of KIISE, Vol. 47, No. 4, pp. 395-403, 2020. 4
https://doi.org/10.5626/JOK.2020.47.4.395

부분단어와 품사 태깅 정보를 활용한 형태소 기반의 한국어 단어 벡터 생성

(Morpheme-based Korean Word Vector Generation Considering the Subword and Part-Of-Speech Information)

윤 준 영 †　　이 재 성 ††
(Junyoung Youn)　　(Jae Sung Lee)

요 약 단어 벡터는 단어 사이의 관계를 벡터 연산으로 가능하게 할 뿐 아니라, 상위의 신경망 프로그램의 사전학습 데이터로 많이 활용되고 있다. 영어 등의 언어와는 달리, 한국어는 어절, 형태소, 음절 및 자소 등으로 다양하게 분리할 수 있는 특성 때문에 영어 학습 모델들과는 다른 다양한 단어 벡터 학습 모델들이 연구되어 왔다. 본 연구에서는 한국어 단어 벡터를 학습하기 위한 단위로 우선 어절을 형태소로 분해하고, 이를 음절 및 자소의 부분단어로 분해하여 학습하는 방법을 제안한다. 또한 전처리된 형태소의 의미 및 구조 정보를 활용하기 위해 품사 태그 정보(Part Of Speech)를 학습에 반영하도록 한다. 성능 검증을 위해 단어 유추 평가 및 응용 프로그램 적용 평가를 해 본 결과, 맞춤법 오류가 적은 일반적인 문서에 대해, 형태소 단위로 자소 부분단어 처리를 하고 품사 태그를 추가했을 경우 다른 방법에 비해 우수함을 보였다.

키워드: 단어 벡터, 부분단어, 형태소 벡터, 품사 태그, 사전학습

Abstract Word vectors enable finding the relationship between words by vector computation. They are also widely used as pre-trained data for high-level neural network programs. Various modified models from English models have been proposed for the generation of Korean word vectors, with various segmentation units such as Eojeol(word phrase), morpheme, syllable and Jaso. In this study, we propose Korean word vector generation methods that segment Eojeol into morphemes and convert them into subwords comprising either syllable or Jaso. We also propose methods using Part-Of-Speech tags provided in the pre-processing to reflect semantic and syntactic information regarding the morphemes. Intrinsic and extrinsic experiments showed that the method using morpheme segments with Jaso subwords and additional Part-Of-Speech tags showed better performance than others under the condition that the target data are normal text and not as grammatically incorrect.

Keywords: word vector, subword, morpheme vector, part of speech, pre-training

· 이 논문은 2017년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(No. 2017R1D1A3B03035676)

† 학생회원 : 충북대학교 전기·전자·정보·컴퓨터학부 컴퓨터과학전공 학생
junyoung292@cbnu.ac.kr
†† 종신회원 : 충북대학교 소프트웨어학과 교수(Chungbuk Univ.)
jasonlee@cbnu.ac.kr
(Corresponding author임)

논문접수 : 2019년 9월 5일
(Received 5 September 2019)
논문수정 : 2020년 1월 21일
(Revised 21 January 2020)
심사완료 : 2020년 1월 31일
(Accepted 31 January 2020)

정보과학회논문지 제47권 제4호(2020. 4)

1. 서 론

단어 벡터(word vector)는 자연어 단어를 다차원 실수 벡터로 압축하여 표현한 것으로, 잘 표현된 단어 벡터는 각 단어 사이의 의미적(semantic), 구문적(syntactic) 관계를 추론할 수 있다[1-5]. 벡터 공간(vector space)에 단어를 매핑시킴으로써 유사한 단어는 서로 가까이 표현되고, 산술연산 등을 통해 유추가 가능해지며, 자연어 처리에서 다양한 분야의 입력 속성으로 사용되어 성능향상에 기여할 수 있다[6-10]. 이러한 단어 벡터를 생성하기 위해 인공신경망을 기반으로 단어 벡터를 학습하는 방법들이 많이 연구 되고 있다. 대표적인 단어 벡터 학습 모델로는 주변 단어를 예측하여 학습하는 Word2Vec[3]의 CBOW(Continuous Bag of Words)와 Skip-gram, 단어의 동시발생 확률을 계산하여 벡터로 표현하는 GloVe[4], Word2Vec의 Skip-gram을 확장하여 각각의 단어를 bag-of-word로 보고, 개별 단어가 아닌 n-gram character의 합으로 단어 벡터를 학습하는 FastText[5] 등이 있다. 최근에는 문맥 정보를 고려한 단어 벡터로 ELMo[11], BERT[12], XLNet[13] 등이 연구되고 있으며, 여러 자연언어 처리 프로그램에서 뛰어난 성능을 보이고 있다. 본 논문에서는 간편성의 이유로 여전히 많이 사용되고 있는, 문맥을 고려하지 않은 기존 단어 벡터 생성에 대해 연구한다. 특히, 한국어의 어절 특성을 반영하여 성능을 높이는 방법에 대해 연구한다.

한국어는 형태소 발달 언어(morphologically rich language)이자 형태학적으로 교착어(agglutinative language)에 속한다. 문장에서 띄어쓰기 단위가 어절이며, 어절은 실질적인 의미를 지니는 어간에 문법적 기능을 가진 조사, 어미 등의 형식 형태소가 결합된 형태로 표현되어 비교적 복잡하다. 이러한 방식은 대단히 생산적이어서 학습해야 할 단어 수가 증가하고 이는 문맥의 부족으로 이어진다[14]. 특히 독립된 어절 벡터를 생성하는 경우, 교착어인 한국어의 특성상 어절이 공유하는 형태소의 의미나 구조를 파악할 수 없는 문제점이 있으며, OOV(Out-Of-Vocabulary) 문제가 상대적으로 크게 발생하는 한계에 직면한다. 더욱이 영어의 경우, 단어를 구성하는 문자가 독립된 형태인 알파벳의 조합으로 표현되는 반면, 한글의 어절은 하나 이상의 음절의 조합으로 구성되고, 각각의 음절은 자소의 계층 구조로 이루어져 있다. 한국어 어절의 특성을 고려한 단어 벡터를 생성하기 위해서는, 어절의 형태소적 특성 및 음절을 이루는 자소의 구조를 고려할 필요가 있다. 이런 한국어 특성을 반영하여 부분단어 방법을 적용한 [17,21]의 연구는 기존 방법에 비해 성능향상을 이루었다.

본 논문에서는 이러한 한국어 특성을 좀 더 반영하여, 형태소 기반의 모델을 적용하고 자소 단위의 부분단어 정보를 활용할 수 있는 한국어 단어 벡터 생성 모델을 제안한다. 즉, 한국어 어절을 의미를 가지는 최소 단위인 형태소 단위로 분해하고, 분해된 형태소를 음절 및 자소 등의 부분단어(subword)로 분해하여 학습에 사용한다. 추가로 각 형태소의 의미적, 구조적 정보를 보충하기 위해 품사 태그 정보(Part Of Speech)를 반영하였다.

학습된 단어 벡터의 성능을 평가하기 위해 단어 유추 평가를 진행하였으며, 세밀한 평가를 위해 의미적(semantic), 구문적(syntactic) 평가를 나누어서 평가하였다. 또한 다른 자연어 처리 응용 프로그램에서도 실제 성능향상에 기여하는지를 판별하기 위해, Bi-LSTM 기반의 감성분석(Sentiment Analysis) 모델과 개체명인식(Named Entity Recognition) 모델에 입력 속성으로 사용하여 분류 정확도를 비교하였다. 단, 형태소 분리를 위해, 기존의 형태소 분석기를 전처리로 사용하였으며, 인터넷 댓글과 같이 맞춤법 오류가 많아 형태소 분석 오류가 많은 텍스트를 제외하면, 본 논문에서 제안한 방법이 효과적이었다. (본 실험에서 적용한 형태소 분석기는 Khaiii(Kakao Hangul Analyzer iii)[15]이며, 일반적인 텍스트에 대한 어절 단위 분석 성능이 F1 척도로 약 97%이다.)

2. 관련연구

단어를 벡터로 표현하는 방법으로는 가장 간단한 방법은 one-hot 표기 방법이다. 이 방법은 단어 개수만큼의 차원을 두고, 그 단어에 해당되는 차원만을 1로 표현하고 나머지는 0으로 하는 방법이다. 이 방법은 매우 큰 차원의 벡터가 필요할 뿐만 아니라, 각 단어 사이가 모두 독립적으로 처리되어 그 벡터로는 단어의 유사도 계산이 불가능하다. 또 다른 방법은 단어를 축소하여(혹은 확대할 수도 있음) 각 차원을 실수로 표현한 것이다. 이 논문에서는 전자는 “one-hot 표기 벡터”, 후자를 “단어 벡터”라고 칭한다. 단어 벡터의 학습은 여러 가지가 있을 수 있으며, 한국어 단어 벡터 생성에는 Word2Vec[1-3]의 Skip-gram 모델이나 이를 확장한 모델인 FastText[5]가 많이 쓰이고 있다.

2.1 Skip-gram 모델[3]

Word2Vec의 경우, 대상 단어를 매우 큰 차원의 one-hot 벡터로 표기하고, 그 단어의 문맥 단어들을 다시 one-hot 벡터 또는 벡터들의 합으로 표기한 후, 연산을 통해 그 벡터와 일치할 수 있도록 학습하는 과정에서 단어 벡터를 만들어 낸다. 그림 1은 Word2Vec의 Skip-gram 모델로, 입력 단어 벡터 w의 one-hot벡터를 α_w행렬의 곱으로 선형변환하여 축소된 차원의 은닉

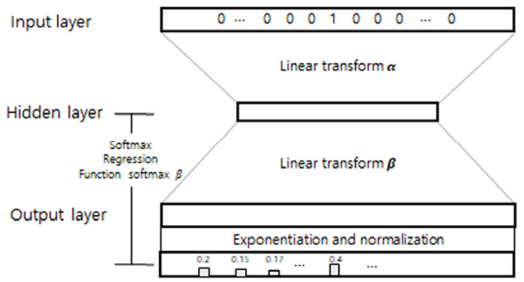

그림 1 Word2Vec의 Skip-gram 모델 계산과정[16]
Fig. 1 Computational procedure of Skip-gram model in Word2Vec[16]

층 벡터를 만들고, 이를 다시 β행렬의 곱으로 선형변환하여 문맥 단어 중의 한 one-hot 벡터로 변환하는 과정을 나타낸다. 이를 수식으로 나타내면 (1)과 같다. 여기에서 $\hat{y}$는 정답 단어 y의 추정치를 나타낸다[16].

$$\hat{y} = softmax(\beta \cdot \alpha_w) \quad (1)$$

이 과정에서 만들어진 은닉층은 필요에 따라 다양한 크기의 차원으로 만들어 낼 수 있으며, 대개 축소된 차원으로 표현하고, 이 벡터를 이용하여 단어 연산을 할 경우, 단어의 의미적 관계나 문법적 관계를 찾아 낼 수 있다[1-3].

2.2 FastText[5] 모델

FastText 모델은 Skip-gram을 확장한 모델로, 대상 단어의 one-hot 벡터를 신경망의 입력으로 하여 주변 문맥에 나타나는 단어들을 예측하는 모델이다. 단, 입력 단어를 n-gram의 bag-of-words로 간주하고, 입력 단어의 n-gram에 대한 one-hot 벡터들을 더하여 신경망의 입력으로 사용하는 점이 다른 점이다. FastText의 경우 미학습 단어를 n-gram으로 처리할 수 있어 기존의 Skip-gram보다 단어 벡터 생성에 효율적으로 알려져 있다[17,18]. 특히 교착어의 특성을 갖는 한국어 단어 벡터 학습시 문맥 부족 문제와 OOV 문제를 완화할 수 있다는 점에서 많이 사용되고 있다[19].

2.3 한국어 단어 벡터 생성에 대한 연구

한국어 단어 벡터 생성은 주로 기존의 Word2Vec 및 GloVe 모델 등을 사용하여 연구하였다[17-22]. Word2Vec 및 GloVe는 학습할 때 독립된 단어를 입력 및 출력에 사용하기 때문에, 교착어인 한국어에 적용할 경우, 한국어 어절의 종류가 많아 어휘 부족 문제에 직면하게 된다. 이를 해결하기 위해 어절을 형태소나 음절 n-gram 등으로 분해하여 학습하는 연구들이 진행되었다.[1)][17]

1) 본 논문에서 사용한 학습 말뭉치의 경우, 어절의 종류는 약 970만개, 형태소 종류는 약 190만개, 음절 종류는 약 3천개로 어절을 분해할 경우 처리해야 할 단위의 수가 급격히 감소한다.

표 1 한국어 부분단어 적용 방법
Table 1 Combinations of Korean subword level

Model	Learning unit	Subword level	Related works
1	Eojeol	Eojeol	Yang et al. 2015[14]
2	Eojeol	morpheme	Lee et al. 2018[20]
3	Eojeol	syllable	Jo et al. 2017[17] Park et al. 2018[21]
4	Eojeol	Jaso	Park et al. 2018[21]
5	morpheme	morpheme	Choi et al. 2016[22]
6	morpheme	syllable	Jo et al. 2017[17]
7	morpheme	Jaso	proposed

기존연구를 학습 단위와 부분단어 분해 수준으로 분류하면 표 1과 같다. 학습 단위(Learning unit)는 단어 벡터 학습에 입력으로 사용되는 단위를 의미하며, 부분단어 수준(subword level)은 입력으로 사용된 단어를 부분단어로 분해하는 정도를 의미한다.

학습 단위는 어절 및 형태소로 정할 수 있다. 학습 단위를 어절로 사용한 연구는 [14], [20], [21]이 있다. [14]는 어절을 학습 단위로 하고 부분단어를 사실상 적용하지 않았다. 이 연구에서는 GloVe, Word2Vec 모델을 적용하여 어절 벡터를 생성한 후, 같은 어근을 공유하는 어절들의 벡터를 합성하여 어근에 대한 벡터를 생성하였다. [20]은 기존 Word2Vec의 Skip-gram을 변형하여 형태소 기반의 어절 벡터 생성에 대한 연구를 하였다. 입력 어절 벡터는 각 어절 형태소 one-hot 벡터의 합으로 정의하고 학습을 하였으며, 학습 후에는 형태소 벡터의 합으로 어절 벡터를 생성하였다. 그 결과 단어 벡터의 성능도 향상 되었다. [21]은 FastText를 확장한 모델을 제안하였으며, 한국어의 어절을 부분단어로 분해하여 학습에 적용하는 두 가지 방법의 접근을 제안하였다. 첫 번째는 음절 n-gram으로 분해하여 학습에 적용한 것이고, 두 번째는 자소 n-gram을 적용한 것이다. 그 결과 기존의 OOV 문제점들을 완화하고 성능 높은 단어 벡터를 생성하였다.

학습 단위를 형태소로 사용한 모델은 [22]와 [17]이 있다. [22]는 형태소 분석을 하고, 독립적 의미를 지니지 못하고 문법적 기능만을 하는 형식 형태소를 제거한 뒤 실질 형태소만을 사용하여 Word2Vec의 Skip-gram을 적용하였다. [17]은 FastText을 이용하여 어절과 형태소를 음절 n-gram character로 분해하여 학습하였고, 음절 수준의 분해가 한국어 단어 벡터 성능향상에 기여할 수 있음을 보였다. 또한 학습에 사용되는 입력 단위로 어절과 형태소를 비교한 결과 형태소 단위의 벡터 생성이 비교적 높은 성능을 보였다.

이외에 [23]은 기존의 영어 모델들이 형태소 발달 언

어에 적합하지 않음을 주장하고, 형태소 발달 언어를 처리하기 위해 어간과 어미를 분리하여 처리하는 방법을 제안하였다. 형태소 발달 언어인 핀란드어를 대상으로 실험을 하여, 제안한 방법이 효과적임을 보였다.

본 연구에서는 전처리로 형태소 분석을 하고 이를 기반으로 형태소를 자소 수준의 부분단어로 분해하여 학습하는 모델을 제안한다. 또한, 전처리 결과 얻을 수 있는 정보인 품사 태그 정보를 활용하는 모델도 제안한다.

3. 모 델

한국어 어절은 문장 내에서 띄어쓰기를 기준으로 분리된 한 단위이다. 형태소는 의미를 가지는 최소의 단위이며, 어절은 다시 형태소로 분해되어, 실질적 의미를 가지는 실질 형태소와, 문법적 역할을 하는 형식 형태소로 구분된다. 형태소는 각 독립된 글자인 음절로 분해할 수 있고, 각 음절은 또 다시 자소로 분해할 수 있다. 한 음절은 3개의 자소(혹은 음소)인 초성, 중성, 종성으로 이루어진다. 음절의 구성에서 초성과 중성은 반드시 필요하나, 종성의 경우 반드시 필요한 요소는 아니다. 본 논문에서는 음절을 자소로 분해 후 종성이 없는 것은 종성 대신 “e”로 표현한다[21].

한국어 단어 벡터 생성을 위해 학습 단위로 사용될 수 있는 것은 어절, 형태소, 형태소 및 품사 태그(POS)로 정할 수 있다. 또한, 학습 시 부분단어를 만드는 최소 단위를 음절 또는 자소로 정할 수 있다. 이 두 단위로 조합 가능한 모델을 만들고 그 명칭을 붙인 것이 표 2이다. EoSyl, EoJa, MorSyl은 [17]과 [21]에서 제안한 모델이며, 본 논문에서는 MorJa, MorSylTag, MorJaTag 모델을 새로 제안한다.

각 모델에서는 n-gram 부분단어를 사용하며, n은 어절 기반의 경우 1~6까지, 형태소 기반은 1~4까지의 값으로 사용했으며, 특수한 경우로 원래 단어(시작과 끝 기호를 단어 앞뒤에 포함)를 학습 단위에 포함하였다[5]. 또 Tag 모델에서는 품사 태그를 학습 단위에 포함시켰다. 각 모델에 대한 설명은 다음과 같다. 단 편의상 bi-gram 부분단어 분해의 예로만 설명한다.

표 2 한국어 단어 벡터 생성 모델
Table 2 Models for Korean word vector generation

Model	Learning unit	Subword level	Related works
EoSyl	Eojeol	syllable	[17], [21]
EoJa	Eojeol	Jaso	[21]
MorSyl	morpheme	syllable	[17]
MorJa	morpheme	Jaso	proposed
MorSylTag	morpheme + POS	syllable	proposed
MorJaTag	morpheme + POS	Jaso	proposed

EoSyl: 어절을 학습의 입력 단위로 사용하며, 어절을 음절(syllable) 단위 부분단어로 분해하여 학습에 적용한다. 어절 “학교에”를 음절 bi-gram 부분단어로 분해하면 다음과 같이 5개의 부분단어가 된다.

{ <학, 학교, 교에, 에>, <학교에> }

EoJa: 어절을 학습의 입력 단위로 사용하며, 어절을 자소(Jaso) 단위의 부분단어로 분해하여 학습에 적용한다. 어절 “학교에”를 자소 bi-gram 부분단어로 분해하면 다음과 같이 11개의 부분단어가 된다.

{ <ㅎ, ㅎㅏ, ㅏㄱ, ㄱㄱ, ㄱㅛ, ㅛe, eㅇ, ㅇㅔ, ㅔe, e>, <ㅎㅏㄱㄱㅛeㅇㅔe> }

MorSyl: 형태소를 학습의 입력 단위로 사용하며, 형태소를 음절(syllable) 단위 부분단어로 분해하여 학습에 적용한다. 형태소 “학교”를 음절 bi-gram 부분단어로 분해하면 다음과 같이 4개의 부분단어가 된다.

{ <학, 학교, 교>, <학교> }

MorJa: 형태소를 학습의 입력 단위로 사용하며, 형태소를 자소(Jaso) 단위 부분단어로 분해하여 학습에 적용한다. 형태소 “학교”를 자소 bi-gram 부분단어로 분해하면 다음과 같이 8개의 부분단어가 된다.

{ <ㅎ, ㅎㅏ, ㅏㄱ, ㄱㄱ, ㄱㅛ, ㅛe, e>, <ㅎㅏㄱㄱㅛe> }

MorSylTag: MorSyl 모델에 형태소의 품사 태그를 추가하여 학습에 사용한다. “학교”와 품사 태그 NNG에 대한 음절 bi-gram 입력은 다음과 같이 5개의 원소가 된다.

{ <학, 학교, 교>, <학교>, NNG }

MorJaTag: MorJa 모델에 형태소의 품사 태그를 추가하여 학습에 사용한다. “학교”와 품사 태그 NNG에 대한 자소 bi-gram 입력은 다음과 같이 9개의 원소가 된다.

{ <ㅎ, ㅎㅏ, ㅏㄱ, ㄱㄱ, ㄱㅛ, ㅛe, e>, <ㅎㅏㄱㄱㅛe>, NNG }

그림 2는 MorSylTag 모델의 학습 예를 나타낸 것이다. 즉, 부분단어 열인 음절 bi-gram, 원래 단어, 품사 태그가 입력으로 one-hot 벡터의 합으로 들어가고, 이를 은닉층으로 선형 변형한 후, 다시 문맥 단어 “에”로 대응 시키는 예이다.

4. 실험 및 토의

4.1 학습 말뭉치 및 모델 하이퍼파라미터

학습에 사용한 말뭉치는 비문과 오탈자를 최소화하기 위해 뉴스기사를 크롤링(crawling)하여 사용하였다. 말뭉치는 총 1,400만 문장으로 이루어져 있으며, 약 2억 개의 어절로 구성되어 있고, 어절의 어휘 수는 약 970만 개, 형태소의 어휘 수는 약 190만개이다. 전처리로 Khaiii[15]를 이용하여 형태소 분석을 하였다. 한국어 단어

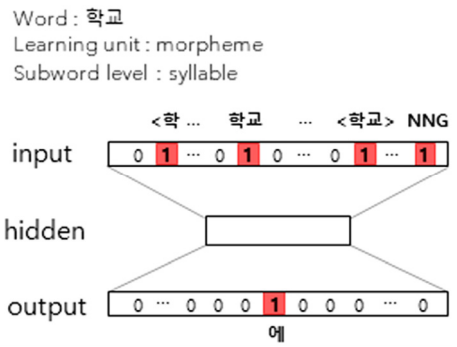

그림 2 “MorSylTag”모델 학습 예
Fig. 2 Example of “MorSylTag”model learning

벡터 생성을 위해 외국어, 특수문자 등은 제거하였고, 지나치게 짧은 문장이나 오류가 포함된 문장은 모두 제거하였다.

학습에 사용된 하이퍼파라미터는 어절 기반 음절 n-gram 분해 모델(EoSyl)을 기준으로 실험을 통해 높은 성능을 보이는 값을 사용하였다. 즉, 단어 벡터의 차원(Dimension)은 300으로, 예측에 사용될 주변 단어의 크기(Window size)는 9, 단어의 최소 등장 횟수는 50으로 하였다. (단, 형태소 기반 모델은 어절 기반 모델에 비해 더 많은 부분단어가 생성되므로, 이를 고려하여 윈도우 크기를 13으로 증가시켰다.) 부분단어인 n-gram의 크기 n은 [17]과 [21]의 실험을 참고하여 학습의 입력 단위와 분해 수준에 따라 다르게 설정하였다. 음절 수준 분해의 경우 어절 기반 학습은 한 개 어절을 포함할 수 있는 1~6, 형태소 기반 학습은 한 개 형태소 포함할 수 있는 1~4로 하였다. 자소 수준 분해의 경우, 입력 단위와 관계없이 1~6으로 설정하였다. 빈도가 높은 단어에 대한 샘플링비율(Frequent subsampling)은 0.0001로 사용하였으며, 5번의 세대(Epoch) 학습을 진행하였다.

4.2 평가

단어 벡터를 평가하는 방법은 크게 두 가지로, 내재평가(intrinsic test)와 외재평가(extrinsic test)로 분류된다[24]. 내재평가는 단어 벡터를 직접적인 연산을 통해 평가하는 방식으로 단어 벡터간의 의미적(semantic), 및 구문적(syntactic) 관계가 얼마나 잘 학습되었는지 평가한다. 외재평가는 단어 벡터를 다른 자연어 처리 응용 프로그램에 사용하였을 경우, 단어 벡터가 성능향상에 기여한 정도를 평가한다.

기존연구와의 비교 평가는 학습 및 평가 데이터, 평가 방법, 점수 산정방식 등에 각각 차이가 있어 어려움이 있다. 따라서 기존연구들 중에 우수한 성능을 보인, [17]의 방법을 EoSyl과 MorSyl, [21]의 방법을 EoSyl과 EoJa[2]로 각각 재현하여 비교한다.

4.2.1 내재평가

내재평가로 단어 유추 평가(Analogy Test)를 진행하였다. 단어 유추 평가는 모델이 단어 벡터 사이의 의미 관계 및 구조를 얼마나 잘 학습했는지 평가하는데, 예를 들어 “서울”- “한국” + “일본”이라는 질의에 대해서 정답 “도쿄”와의 유사도가 얼마나 높게 나타나는지 평가한다. 일반적으로 한국어 단어 유추 평가시 영어 데이터셋을 한글로 번역하여 사용하는 경우가 많지만, 데이터셋에 포함된 영어 단어 의미관계는 한국어와 다른 점이 많고, 구문적 질의는 한국어와 영어의 구문적 구조가 상이하기 때문에 평가에 적합하지 않다고 판단하여 [21]에서 공개한 한국어 단어 유추 테스트 셋을 사용하였다. 이는 총 10,000개의 질의로 구성되어 있으며 의미적(semantic) 관계를 평가하기 위한 5,000개의 질의와 구문적(syntactic) 관계를 평가하기 위한 5,000개의 질의를 포함하고 각각 5개의 범주로 클러스터 되어있다. 그 예는 각각 표 3 및 표 4와 같다.

일반적으로 단어 유추 평가를 위해 평가 셋에 나타난 단어 관계식 Vec(B-A+C)를 계산하여 Vec(D′)을 구하고, 이 벡터 Vec(D′)와 생성된 모든 단어 벡터를 코사인 유사도로 계산하고, 이 순위를 점수로 사용한다. 하지만 본 논문에서 제안하는 모델의 경우, 입력에 사용한 단어의 벡터만을 생성하는 것이 아닌 모든 부분단어에

표 3 의미적 관계 유추 평가 데이터 셋 예[21]
Table 3 Samples of semantic relation analogy test set[21]

Semantic relationships	Word pairs A:B = C:D
국가-수도(Capt)	아테네 : 그리스 = 바그다드 : 이라크
남성-여성(Gend)	왕자 : 공주 = 신사 : 숙녀
인물-국가(Name)	간디 : 인도 = 링컨 : 미국
국가-언어(Lang)	아르헨티나 : 스페인어 = 미국 : 영어
기타(Misc)	부산 : 경상남도 = 대구 : 경상북도

표 4 구문적 관계 유추 평가 데이터 셋 예[21]
Table 4 Samples of syntactic relation analogy test set[21]

Syntactic relationships	Word pairs A:B = C:D
조사결합(Case)	주식 : 주식은 = 자동차 : 자동차는
시제 변이(Tense)	싸우다 : 싸웠다 = 오다 : 왔다
태 변이(Voice)	팔았다 : 팔렸다 = 평가했다 : 평가됐다
종결 변이(Form)	가다 : 가고 = 쓰다 : 쓰고
존칭 변이(Honr)	도왔다 : 도우셨다 = 됐다 : 되셨다

2) EoJa 모델은 [21]의 SISG(ch6+jm)에 해당된다. 단, 구현 차이로 성능의 차이가 있으며, [21]의 구현은 의미적 및 구문적 관계 유추 평가 평균이 각각 0.545, 0.674로 본 구현보다 더 높은 성능을 보였다(표 5 및 표 6 참조).

대하여 벡터를 생성하기 때문에 위와 같은 방식은 적절하지 않다. 따라서 본 실험에서는 COSADD[25]를 기반으로 Vec(D′)와 정답 단어 벡터 Vec(D)의 직접적인 유사도를 구하여 평가하였으며 아래의 식 (2)을 따른다.

$$\text{Cos}(\ \text{Vec(B)}-\text{Vec(A)}+\text{Vec(C)},\ \text{Vec(D)}\ ) \quad (2)$$

표 5는 의미적 관계의 단어 유추 평가 결과를 보여준다. 형태소 기반의 모델은 MorSyl 모델을 제외하고는 기존의 어절 기반 모델보다 높은 성능을 보였다. 또한 입력 단위에 상관없이, 부분단어로 음절분해를 적용한 모델보다 자소분해를 적용한 모델이 높은 성능을 보였다. 특히 품사 태그를 더한 MorSylTag와 MorJaTag 모델이 다른 모델에 비해 높은 성능을 보이는 것으로 보아, 품사 태그를 더하여 의미 및 구조 관계를 보다 명확히 보충할 경우 의미적 관계 유추에 효과적임을 알 수 있다.

의미적 관계 유추 평가 중 하나인 Lang(국가-언어)에서 상대적으로 형태소 기반 모델이 낮은 성능을 보였다. 분석 결과, 해당 테스트 데이터에서 형태소 분석 오류가 많이 나타났고, 그 영향을 확인하기 위해 형태소 분석 오류를 인위적으로 교정한 후 평가를 해본 결과, 형태소 기반 모델 MorSyl, MorJa, MorSylTag, MorJaTag의 성능이 각각 0.511, 0.582, 0.603, 0.619로 향상되어, 대체적으로 어절 기반의 모델보다 높은 성능을 보였다. 이 결과로 판단해 볼 때, 형태소 분석의 성능이 뒷받침된다면, 형태소 기반 모델이 어절 기반 모델보다 더 나은 성능을 보이는 것을 알 수 있다.

표 6은 구문적 관계에 대한 유추 평가 결과를 보인다. 형태소 기반 모델은 학습 이전에 형태소 분석을 통해 구문적 관계에 대한 전처리가 이루어지며, 구성 형태소 벡터의 합을 어절 벡터로 사용한다. 따라서 구문적 관계 유추에서 형태소 기반의 모델(Mor*)이 어절 기반의 모델(Eo*)보다 높은 성능을 보일 수밖에 없다. 그러나, 본 연구에서 제안한 모델인 MorJa나 MorJaTag 모델이 기존의 형태소 기반 모델(MorSyl)보다도 높은 성능을 보인 것은 주목할 만하다.

표 5 의미적 관계 유추 평가 결과
Table 5 Semantic relation analogy test result

Model	Semantic					
	Capt	Gend	Name	Lang	Misc	**Mean**
EoSyl	0.573	0.531	0.433	0.509	0.411	0.491
EoJa	0.584	0.562	0.460	**0.569**	0.453	0.526
MorSyl	0.556	0.528	0.442	0.447	0.397	0.474
MorJa	0.600	0.600	0.510	0.506	0.468	0.537
MorSylTag	0.580	0.596	0.512	0.536	0.472	0.539
MorJaTag	**0.618**	**0.672**	**0.558**	0.527	**0.539**	**0.583**

표 6 구문적 관계 유추 평가 결과
Table 6 Syntactic relation analogy test result

Model	Syntactic					
	Case	Tense	Voice	Form	Honr	**Mean**
EoSyl	0.700	0.535	0.579	0.480	0.544	0.568
EoJa	0.791	0.649	0.658	0.631	0.624	0.671
MorSyl	0.914	0.863	0.832	0.888	0.849	0.869
MorJa	**0.922**	0.884	0.863	0.909	0.872	0.890
MorSylTag	0.857	0.893	0.792	0.937	0.858	0.867
MorJaTag	0.915	**0.940**	**0.896**	**0.958**	**0.923**	**0.926**

4.2.2 외재평가

단어 벡터가 응용 프로그램의 성능향상에 기여하는지 검증하기 위해 영화 감상평의 감성분석(Sentiment Analysis)과 개체명인식(Named Entity Recognition)에 대해 실험하였다. 두 가지 모델 모두 Bi-LSTM 단일 모델을 사용하였으며, 각 모델의 성능 평가는 F1 척도로 측정하였고 결과는 표 7과 같다.

표 7 외재평가 결과
Table 7 Extrinsic test result

Model	Sentiment Analysis	Named Entity Recognition
EoSyl	**0.837**	0.870
EoJa	0.830	0.870
MorSyl	0.824	0.873
MorJa	0.813	0.872
MorSylTag	0.816	0.872
MorJaTag	0.807	**0.874**

감성분석

실험에 사용한 감성분석(Sentiment Analysis)은 영화 감상평 분석으로 감상평 문장을 긍정 또는 부정으로 분류한다. 학습 데이터 셋은 약 20만개의 긍/부정 레이블을 포함하는 네이버 영화 감상평 데이터 셋[26]을 사용하였다.

표 7에서 보듯이 실험 결과, 제안한 형태소 기반 모델들의 성능이 낮게 나왔다. 그 이유는 사용한 데이터 셋이 인터넷 댓글로, 약 35%의 띄어쓰기 오류가 있는 등 대부분이 문법에 맞지 않는 문장으로 작성되어 있고 그에 따라 형태소 분석이 제대로 이루어지지 않았기 때문이다. 표 8은 데이터 셋에 나타난 형태소 분석 오류의 예이다. 원문의 맞춤법 오류가 형태소 분석의 오류로 이어져, 잘못된 형태소 및 자소분해가 이루어졌고, 그에 따라 잘못된 품사 태그가 추가되어 성능 하락이 이루어진 것으로 분석된다.

개체명인식

개체명인식(Named Entity Recognition)은 자연언어

표 8 원문의 문법 오류로 인한 형태소 분석 오류 예
Table 8 Examples of morphological analysis errors caused by grammatical errors in the original text

Original text	Analysis result
이드라마사랑하고잼있는데	이드/NNG + 이/VCP + 라/EC + 마사랑/NNG + 하/XSV + 고/EC + 잼/NNG + 있/VX + 는데/EC
공감가고캐릭터하나하나	공감가/NNG + 고/JKB + 캐릭터하나/NNG + 하나/MAG
또봐도 잼나네요	또/MAG + 봐/NNP + 도/EC + 잼/NNG + 나네/NNP + 요/NNG

문장에서 미등록어로 자주 나타나는 대표적인 개방어휘인 사람, 장소, 기관 등의 이름 개체명 어휘를 찾아내는 것을 목적으로 한다. 본 논문에서 사용한 엑소브레인 개체명인식 데이터는 2016년과 2017년 국어 정보 처리 경진대회 지정분야의 데이터를 정리하여 배포한 것으로 총 1만 문장으로 구성되어 있다[27].

표 7에서 보듯이 학습된 단어를 개체명인식 모델의 입력으로 사용한 결과, 제안한 형태소 기반의 모델이 어절 기반의 모델보다 더 높은 성능을 보이는 것으로 나타났다. 감성분석에 사용된 댓글[26] 데이터와 비교하여 개체명인식에 사용된 엑소브레인[27] 데이터는 띄어쓰기 등의 맞춤법 오류가 상대적으로 적게 나타난다. 전체 데이터의 약 1%정도 추출하여 형태소 분석결과를 수작업으로 검토한 결과 오류율이 약 4%정도로 나타났다. 또한 본 실험에서 단어 벡터 학습에 적용한 형태소 품사 분류 범주가 개체명인식 학습에 사용된 데이터에서 적용된 분류 범주가 다르다. 그럼에도 불구하고 제안된 형태소 기반의 모델로 생성된 단어 벡터가 더 높은 인식 결과를 나타내었다.

4.2.3 토의

그림 3은 전체적인 경향을 보기 위해, 앞에서 제시한 4개의 평가인 의미관계 유추, 구문관계 유추, 감성분석, 개체명인식 결과를 표시한 것이다. 또, 단어 벡터 생성에서 형태소 처리, 자소 처리, 품사 추가의 효과를 표 9에 요약한 것이며, 각 설명은 아래와 같다.

형태소 처리 효과

형태소 분석 오류가 적은 경우 형태소 기반 모델이 어절 기반 모델에 비해 전반적으로 우수한 성능을 보였다. 의미적 관계 유추 평가에서 MorSyl 모델이 어절 기반 모델에 비해 성능이 하락하였지만, 4.2.1에서 설명한 것과 같이 형태소 분석 오류를 인위적으로 교정한 결과 어절 기반 모델보다 우수한 성능을 보였다. 감성분석의 경우 비교적 많은 형태소 분석 오류를 포함하며, 이 경우 어절기반 모델이 형태소 기반 모델보다 우수한 성능을 보였다.[3] 하지만 비교적 오류가 적은 개체명인식에서

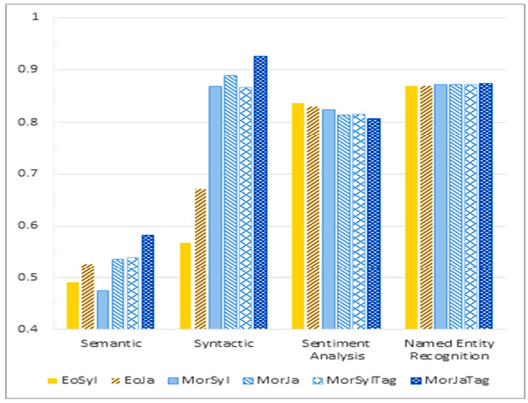

그림 3 4개 평가별 단어 벡터 모델 성능 비교
Fig. 3 Performance comparison of word vector models in four criteria

표 9 모델 변경 효과(유사도 및 F1 차이)
Table 9 Effects of Model Change (Similarity and F1 difference)

	Model Change (From→To)	Sem.	Syn.	Senti. Anal.	Named Entity Recog.
morph	EoSyl→MorSyl	-0.017	**+0.301**	-0.013	**+0.002**
	EoJa→MorJa	**+0.011**	**+0.219**	-0.017	**+0.002**
Jaso	EoSyl→EoJa	**+0.035**	**+0.103**	-0.006	-0.001
	MorSyl→MorJa	**+0.063**	**+0.021**	-0.011	-0.001
POS tag	MorSyl→MorSylTag	**+0.065**	-0.002	-0.009	-0.001
	MorJa→MorJaTag	**+0.046**	**+0.036**	-0.006	**+0.002**

형태소 기반 모델이 우수한 성능을 보여 형태소 분석의 성능이 뒷받침될 경우 본 연구에서 제안하는 형태소 기반 모델이 효과적임을 보였다.

자소 처리 효과

자소분해 모델은 형태소 분석 오류가 적은 데이터의 경우 대체로 우수한 성능을 보였다. 즉, 내재평가에서 우수한 성능을 보였으며, 그 이유는 자소분해 모델이 보다 세밀한 정보를 활용하기 때문으로 보인다. 예를 들어 구문관계 유추의 한 예인, "주식:주식은 = 자동차:자동차는"에서 음절분해 모델은 조사로 분리된 "은"과 "는"을 독립된 단위로 처리하는 반면 자소분해 모델은 "은"과 "는"의 공통 자소인 "ㅡ"와 "ㄴ"을 학습하여 더 우수한 성능을 보이는 것으로 추정된다. 하지만, 형태소 분석 오류가 많은 감성분석(입력 텍스트의 오류 많음)이나 개체명인식(고유명사 처리 오류 많음)의 경우, 자소분해 모델의 성능이 오히려 다소 하락하였다.

3) 오류에 강건한 형태소 분석기(예:Mecab[28], Twitter[29])를 사용하면 감성분석에서도 형태소 기반 모델이 우수할 것으로 추정된다.

품사 추가 효과

품사 태그를 추가한 결과 내재평가의 의미적 관계 유추에서는 자소분해 모델과 음절분해 모델 모두 성능이 향상되었다. 그러나 형태소 분석 오류가 많은 감성분석에서는 두 모델 모두 성능이 하락하였다. 또, 구문적 관계 유추와 개체명인식에서는 음절분해 모델은 성능이 하락하였지만, 자소분해 모델은 향상되었다. 약간의 형태소 분석 오류를 포함하는 경우 자소분해 모델에 품사 태그를 보충해주면, 오류에 민감한 자소분해 모델의 단점을 보완해주어 성능이 향상되는 것으로 추정된다.

5. 결 론

본 논문에서는 한국어 단어 벡터를 효율적으로 생성하기 위한 방법으로 단어 벡터 입력 단위와 부분단어 구성 방법에 대해 연구하였다. 이를 위해 입력 단위를 형태소로 하고 부분단어 구성을 음절 혹은 자소 단위로 하는 방법과 여기에 품사 태그 추가하는 방법을 제안하였고 이를 기존의 방법들과 비교 검증하였다.

제안한 모델의 성능을 검증하기 위해 내재평가와 외재평가를 하였다. 내재평가로는 단어 유추 평가(Analogy test)를 진행하였다. 그 결과 형태소 기반의 모델이 어절 기반의 모델보다 높은 성능을 보였고 품사 태그를 추가하여 단어의 문법 정보를 반영한 결과, 기존 모델의 성능을 능가함을 보였다. 또, 작은 단위인 자소 단위의 부분단어 모델이 음절 단위의 부분단어 모델보다 더 높은 성능을 보였다.

외재평가는 감성분석(Sentiment Analysis) 모델과 개체명인식(Named Entity) 모델에 적용하여 검증하였다. 그 결과 제안된 모델의 단어 벡터가, 맞춤법 오류가 많이 발생하는 인터넷 댓글의 감성분석의 평가에서는 오히려 성능이 하락하는 결과가 나타났지만, 비교적 맞춤법 오류가 적은 데이터로 평가한 개체명인식 모델의 평가에서는 성능을 향상시켰다.

본 논문에서 제안한 모델은 입력 데이터의 맞춤법 오류에 영향을 받을 수 있지만, 맞춤법 오류가 많지 않은 일반적인 문서에 적용하였을 경우, 한국어 단어 벡터 학습에 효과적임을 보였다. 이는 형태소와 같이 의미를 갖는 정확한 단위에 대해 적절한 부분단어 방법을 적용했을 경우, 한국어 단어 벡터 생성이 효과적임을 보여준다. 이러한 방법은 언어 특성에 따라 다를 수는 있지만, 형태소 발달 언어나 띄어쓰기가 없어 적절한 입력 단위를 임의로 정해야 하는 언어 등에 적용해 보는 것도 좋을 것이다.

References

[1] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *Proc. of the ICLR workshop*, 2013.

[2] T. Mikolov, W. Yih, and G. Zweig, "Linguistic regularities in continuous space word representations," *Proc. of the NAACL-HLT*, pp. 746-751, 2013.

[3] T. Mikolov, I. Sutskever, K. Chen, G. Corrado and J. Dean, "Distributed representations of words and phrases and their compositionality," *Proc. of the NIPS*, pp. 3111-3119, 2013.

[4] J. Pennington, R. Socher, and C. D. Manning, "GloVe: Global vectors for word representation," *Proc. of the EMNLP*, pp. 1532-1543, 2014.

[5] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *arXiv preprint arXiv*:1607.04606, 2016.

[6] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv*:1409.0473, 2014.

[7] M. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," *arXiv preprint arXiv*:1508.04025, 2015.

[8] T. Mikolov, Q. V. Le, and I. Sutskever, "Exploiting similarities among languages for machine translation," *arXiv preprint arXiv*:1309.4168, 2013.

[9] A. M. Rush, S. Chopra, and J. Weston, "A neural attention model for abstractive sentence summarization," *arXiv preprint arXiv*:1509.00685, 2015.

[10] S. K. Siencnik, "Adapting word2vec to named entity recognition," *Proc. of the 20th NODALIDA 2015, Vilnius, Lithuania*, no. 109, pp. 239-243, 2015.

[11] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, "Deep contextualized word representations," *arXiv preprint arXiv*:1802.05365, 2018.

[12] J. Devlin, M. Chang, K. Lee and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv*:1810.04805, 2019.

[13] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov and Q. V. Le, "XLNet: Generalized Autoregressive Pretraining for Language Understanding," *arXiv preprint arXiv*:1906.08237, 2019.

[14] H. Yang, Y. Lee, H. Lee, S. Cho, and M. Koo, "A study on word vector models for representing korean semantic information," *Proc. of the KSSS*, Vol. 7, no. 4, 2015. (in Korean)

[15] khaiii(Kakao Hangul Analyzer III), [Online]. Available: https://github.com/kakao/khaiii

[16] B. Wilson, "An overview of word2vec," presentation file, Berlin ML Meetup, 2014.

[17] H. Jo, S. Lee, "Korean word embedding using fasttext," *Proc. of the KISS Conference*, pp. 705-707, Dec. 2017. (in Korean)

[18] H. Cho, C. Seo, S. Kang and E. Youn, "Design and Implementation of word classes Embedding for improving fasttext," *Proc. of the KISS Conference*, pp. 979-981, June. 2018. (in Korean)

[19] C. Park, H. Hwang, C. Lee and H. Kim, "Korean coreference resolution using fastText and pointer networks based on self-matching attention," *Journal of KIISE:Transactions on Computing Practices*, Vol. 24, No. 12, pp. 635-641, Dec. 2018. (in Korean)
[20] D. Lee, Y. Lim, T. Kwon, "Morpheme-based efficient Korean word embedding," *Journal of KIISE*, Vol. 45, No. 5, pp. 444-450, May. 2018 (in Korean)
[21] S. Park, J. Byun, S. Baek, Y. Cho, A. Oh, "Subword-level word vector representations for Korean," *Proc. of the 56th Annual Meeting of the Association for Computational Linguistics*, Jul. 2018. (in Korean)
[22] S. Choi, J. Seol, S. Lee, "On word embedding models and parameters optimized for Korean," *Proc. of the HCLT*, No. 15, 2016. (in Korean)
[23] P. Takala, "Word embedding for morphologically rich languages," *Proc. of the ESANN*, Apr. 2016.
[24] T. Schnabel, I. Labutov, D. Mimno and T. Joachims, "Evaluation methods for unsupervised word embeddings," *Proc. of the EMNLP*, pp. 298-307, 2015.
[25] O. Levy, Y. Goldberg, "Linguistic regularities in sparse and explicit word representations," *Proc. of the eighteenth conference on Computational Language Learning*, pp. 171-180, 2014.
[26] Naver sentiment movie corpus, [Online]. Available: https://github.com/e9t/nsmc/
[27] ETRI, "Exobrain corpus V3.0,"2017. [Online]. Available: https://aiopen.etri.re.kr
[28] Mecab, [Online]. Available: https://github.com/bibreen/mecab-ko-lucene-analyzer
[29] Twitter-korean-text, [Online]. Available: https://github.com/twitter/twitter-korean-text

윤 준 영

2019년 충북대학교 소프트웨어학과 졸업 (학사). 2019년~현재 충북대학교 전기·전자·정보·컴퓨터학부 컴퓨터과학전공 석사과정. 관심분야는 자연어처리, 정보검색, 기계학습

이 재 성

1979년~1983년 서울대학교 컴퓨터공학과 (학사). 1983년~1985년 KAIST 전산학과 (석사). 1985년~1988년 큐닉스 과장 1988년~1993년 마이크로소프트 차장 1995년~1999년 KAIST 전산학과(박사) 1999년~2000년 ETRI 지식정보검색연구팀 팀장. 2000년~현재 충북대학교 컴퓨터 교육과, 소프트웨어학과 교수. 관심분야는 자연어처리, 정보검색, 기계학습