M.S. THESIS

Morpheme-based Efficient Korean Word Embedding

형태소 기반 효율적인 한국어 단어 임베딩

2018년 2월

서울대학교 대학원
컴퓨터공학부
이동준

Morpheme-based Efficient Korean Word Embedding

형태소 기반 효율적인 한국어 단어 임베딩

지도교수 권태경

이 논문을 공학석사 학위논문으로 제출함
2017년 10월

서울대학교 대학원
컴퓨터공학부
이동준

이동준의 석사 학위논문을 인준함
2017년 12월

위 원 장	김종권	(인)
부위원장	권태경	(인)
위 원	전화숙	(인)

학위논문 원문 서비스에 대한 동의서

본인의 학위논문에 대하여 서울대학교가 아래와 같이 학위논문 저작물을 제공하는 것에 동의합니다.

1. 동의사항

   ① 본인의 논문을 보존이나 인터넷 등을 통한 온라인 서비스 목적으로 복제할 경우 저작물의 내용을 변경하지 않는 범위 내에서의 복제를 허용합니다.

   ② 본인의 논문을 디지털화하여 인터넷 등 정보통신망을 통한 논문의 일부 또는 전부의 복제배포 및 전송 시 무료로 제공하는 것에 동의합니다.

2. 개인(저작자)의 의무

   본 논문의 저작권을 타인에게 양도하거나 또는 출판을 허락하는 등 동의 내용을 변경하고자 할 때는 소속대학(원)에 공개의 유보 또는 해지를 즉시 통보하겠습니다.

3. 서울대학교의 의무

   ① 서울대학교는 본 논문을 외부에 제공할 경우 저작권 보호장치(DRM)를 사용하여야 합니다.

   ② 서울대학교는 본 논문에 대한 공개의 유보나 해지 신청 시 즉시 처리해야 합니다.

논문 제목 : 형태소 기반 효율적인 한국어 단어 임베딩

학위구분 : **석사**
학 과 : 컴퓨터공학부
학 번 : 2016-21220
연 락 처 : 010-9280-3940
저 작 자 : 이동준 (인)

제 출 일 : 2018년 2월

서울대학교총장 귀하

Abstract

Morpheme-based Efficient Korean Word Embedding

Dongjun Lee
Department of Computer Science & Engineering
The Graduate School
Seoul National University

Word embedding is a strategy of mapping each word from a continuous vector space into one vector. It is the starting point of natural language processing task and greatly impacts the performance. Word2vec and Glove are among the most popular and widely used word embedding models. However, these models have limitations in that it is unable to learn the shared structure of words nor sub-word meanings. This is a serious limitation for morphologically rich languages such as Korean.

In this paper, we propose a new model which is an expansion of the previous skip-gram model to learn the sub-word information. The model 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, by using our trained vectors as an input to the previous text classification model, we tested how much performance has actually been enhanced.

Keywords : word embedding, morphmeme embedding, Korean, neural network, word2vec

Student Number : 2016-21220

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Representing the meaning of a word is one of the fundamental issues in natural language processing(NLP). The simplest method of expressing a word is to treat all words as atomic symbols. In vector space terms, all words are expressed as a very large sparse vector which consists of a single 1 and remaining 0's. Such one-hot encoding method has limitations in that the vector cannot reflect relationships between words. Distributed representations of words in a continuous vector space have been widely used to overcome such limitations. It maps similar words to nearby points in the vector space. Distributed representation can represent relationships between words through vector relation as described in figure 1. It helps to improve the performance of learning algorithms in natural language processing tasks.[2]

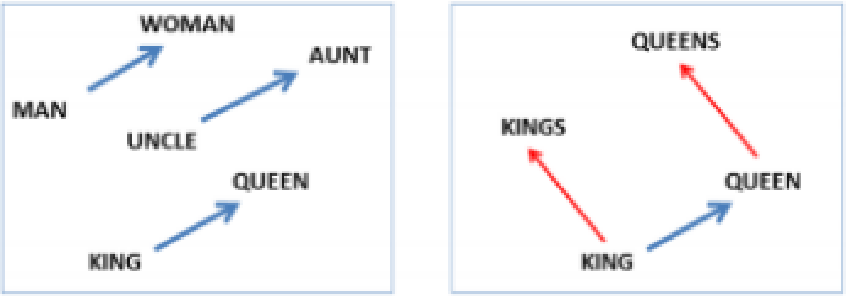


Figure 1: Relationship between words in the distributed representation

Recently, there has been neural network-based word embedding models[4] and their most representative models are word2vec's CBOW(Continuous Bag-Of-Words) and skip-gram[1], and Glove[5]. Such models are especially known for their effective learning of English word vectors. However, their limitations lie in the failure to learn shared structures between different words and sub-word meanings due to learning independent vectors in all words. As a consequence of such limitations, above former models cannot effectively learn words vectors of morphologically rich languages such as Korean. As an agglutinative language, all Korean words consist of a sum of root and affix. For example, it learns words such as "Bab-eul(밥을)," "Bab-eun(밥은)," "Bab-ina(밥이나)," "Bab-do(밥도)," "Bab-man(밥만)" as independent vectors. Hence, the learning process and results are ineffective.

This paper proposes to overcome such limitations with a new model of learning morpheme vectors by expanding the existing skip-gram by defining each word vector as an addition of its morpheme vectors. Then the neural network learns vectors for morphemes. This also allows for the model to learn the word's internal meaning. To measure the effectiveness of learned vectors, word similarity test and word analogy tests were conducted, and a qualitative analysis was carried out by visualizing morpheme vectors in two-dimensional space through PCA(Principal Component Analysis). Moreover, it was experimented to measure how much natural language processing task algorithm performance was actually improved by using learned vectors. This was done by the comparison of classification accuracy by changing only the input word vectors in the existing CNN(Convolutional Neural Network) based text classification model[3].

The paper consists of the following chapters. Chapter 2 introduces the related works of the skip-gram model and its limitations, and other models which attempted to solve the model's limitations. Moreover, it also introduces existing studies of Korean word embedding. Chapter 3 proposes an efficient morpheme-based Korean word embedding model and outlines its advantages. Chapter 4 explains the implementation details and evaluates the performance of the proposed model. Chapter 5 concludes the paper and suggests future works.

Chapter 2

Related Works

Most of the word embedding methods are based on the distributional hypothesis of words that appear in the similar context have a similar meaning. Word embedding model is largely divided into the count-based model which counts the number of words appearing together and the predictive model which predicts words from their nearby words. Among the two, it is the latter[6], especially the neural network-based model, that is widely accepted as the superior. The most representative models of the neural network-based model are word2vec(CBOW or skip-gram) and Glove. The skip-gram model is known as having the highest performance among them.[4]

2.1 Skip-gram model[1]

Skip-gram model predicts surrounding words given the current word. More precisely, it tries to maximize classification of a word based on another word. Each current word is used as an input to a loglinear classifier, and predict words within a window. Skip-gram model architecture is briefly described in Figure 2.

Skip-gram's objective function is to maximize the probability of any context word given to the center word which can be defined as follows.

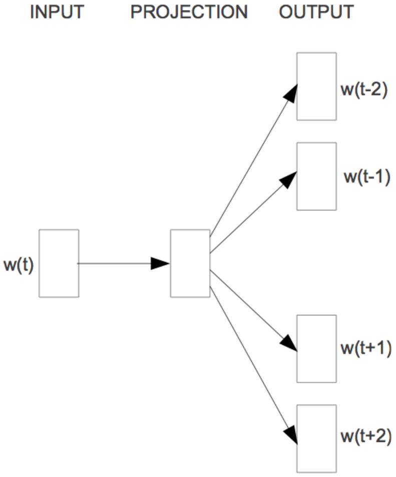


Figure 2: Skip-gram model[2]

$$J(\theta) = \frac{1}{|W|}\sum_{t=1}^{|W|}\sum_{-m\leq j\leq m, j\neq 0}\log P(w_{t+j}|w_t) \tag{2.1}$$

where a given training corpus is represented as a sequence of words $w_1, \ldots, w_{|W|}$ and size of the window is m. The probability of observing context word w_o given to the center word w_c is defined by softmax function:

$$P(w_o|w_c) = \frac{\exp\left(u_o^T v_c\right)}{\sum_{w=1}^{|W|}\exp\left(u_w^T v_c\right)}. \tag{2.2}$$

where u and v are the input and output vectors for w_o and w_c, and $|W|$ is the number of unique words in the training corpus. However, this formulation is computationally to expensive because the cost of computing $\nabla \log P(w_{t+j}|w_t)$ is proportional to W, which is very large.

Therefore, Mikolov et al.[2] used negative sampling method to reduce the computational cost. The main idea of negative sampling is that train binary logistic regressions for a true pair (the center word and the word in its context window) and several random pairs (the center word and random word which is chosen from a noise distribution). Using negative sampling, objective function is defined as

$$J(\theta) = \frac{1}{|W|} \sum_{t=1}^{|W|} J_t(\theta) \tag{2.3}$$

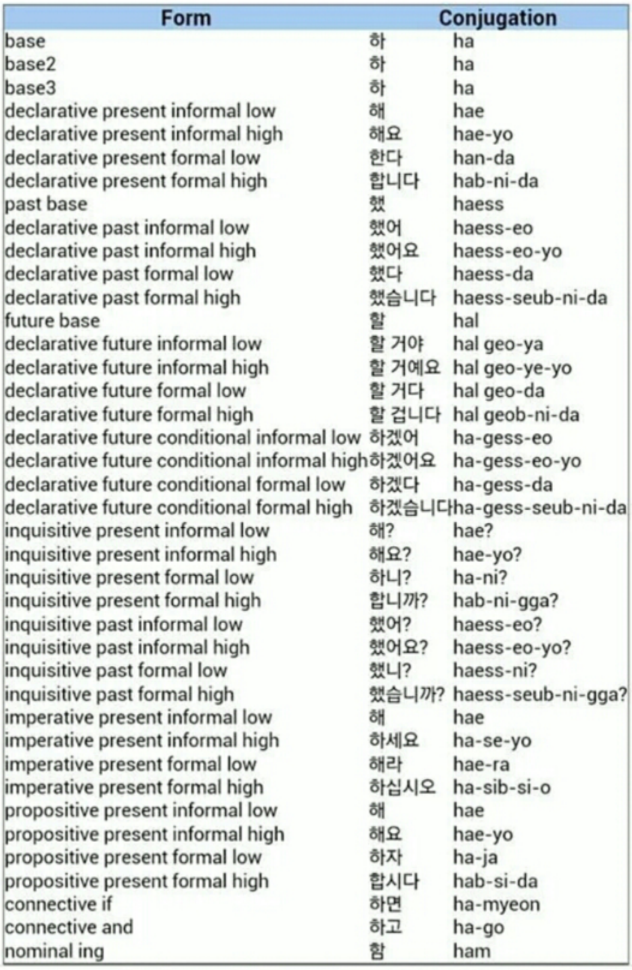
$$J_t(\theta) = \log(u_o^T v_c) + \sum_{i=1}^{k} \mathbb{E}_{w_i \sim P(w)} [\log \sigma(-u_{w_i}^T v_c)]. \tag{2.4}$$

where $P(w)$ is a noise distribution.

2.2 Limitations of the skip-gram model

The skip-gram model learns vectors every word in the training corpus as an independent vector. Therefore, skip-gram can not learn the structural or sub-word information shared by different words. This limitation is critical to morphologically rich languages such as Korean. Korean is an agglutinative language so that all Korean words consist of a root and affix. Hence, various words can be derived from a single root. For example, in English, only ”mountains” can be derived from the root ”mountain”. However in Korean, ”산을”(san-eul), ”산은”(san-eun), ”산도”(san-do), ”산이”(san-yi), ” 산이나”(san-ina), and etc. can be derived from the root ”산”(san). Figure 3 describes how many conjugations can be made for verbs in Korean. Learning independent vectors for every such word is very inefficient.

To overcome such limitations, various methods for learning sub-word



Form	Conjugation	
base	하	ha
base2	하	ha
base3	하	ha
declarative present informal low	해	hae
declarative present informal high	해요	hae-yo
declarative present formal low	한다	han-da
declarative present formal high	합니다	hab-ni-da
past base	했	haess
declarative past informal low	했어	haess-eo
declarative past informal high	했어요	haess-eo-yo
declarative past formal low	했다	haess-da
declarative past formal high	했습니다	haess-seub-ni-da
future base	할	hal
declarative future informal low	할 거야	hal geo-ya
declarative future informal high	할 거예요	hal geo-ye-yo
declarative future formal low	할 거다	hal geo-da
declarative future formal high	할 겁니다	hal geob-ni-da
declarative future conditional informal low	하겠어	ha-gess-eo
declarative future conditional informal high	하겠어요	ha-gess-eo-yo
declarative future conditional formal low	하겠다	ha-gess-da
declarative future conditional formal high	하겠습니다	ha-gess-seub-ni-da
inquisitive present informal low	해?	hae?
inquisitive present informal high	해요?	hae-yo?
inquisitive present formal low	하니?	ha-ni?
inquisitive present formal high	합니까?	hab-ni-gga?
inquisitive past informal low	했어?	haess-eo?
inquisitive past informal high	했어요?	haess-eo-yo?
inquisitive past formal low	했니?	haess-ni?
inquisitive past formal high	했습니까?	haess-seub-ni-gga?
imperative present informal low	해	hae
imperative present informal high	하세요	ha-se-yo
imperative present formal low	해라	hae-ra
imperative present formal high	하십시오	ha-sib-si-o
propositive present informal low	해	hae
propositive present informal high	해요	hae-yo
propositive present formal low	하자	ha-ja
propositive present formal high	합시다	hab-si-da
connective if	하면	ha-myeon
connective and	하고	ha-go
nominal ing	함	ham

Figure 3: Conjugation of Korean verb ”하다”

information of words have been suggested. Cui et al.[7] proposed a model to apply prior knowledge so that syntactically similar words have similar vectors. Luong et al.[8] seperated each word into prefix, root, and suffix, and then composed an RNN(Recurrent Neural Network) to synthesize them. Bojanowski et al.[9] defined each word as a set of n-gram and derived word vector as a sum of its n-gram vectors. Zhang et al. [10] assigned vectors to characters and represented each word as a matrix.

2.3 Existing study on Korean word embedding

Most of the study on Korean word embedding applied existing models such as word2vec and Glove. They improved the embedding performance through additional processing before and after learning.

S. Whan et al.[11] directly applied the existing models(word2vec and Glove) and vectors for the root are generated by synthesizing vectors of words having the same root. This approach also has the limitation that it can't learn sub-word information because existing models are applied in the learning step. Moreover, word synthesizing step gives up too much information of the word. For these reasons, it showed poor performance on a word similarity test and a word analogy test.

S. Choi et al.[12] removed all the morphemes that have no meaning independently through the morphological analyzer and then the existing skip-gram model is applied. By reducing the number of parameters to be learned, it showed better performance than [11]. However, they abandoned learning about various morphemes that do not have independent meaning such as prefixes and suffixes. This is fatal to learning syntactic relations between words because it does not learn the structural information of words.

Chapter 3

Morpheme-based word embedding model

Since the skip-gram model assigns independent vectors to each and every vocabulary in the corpus, it is impossible to learn the structural information of words that are shared by different words. In this paper, we propose a word embedding model that extends the existing skip-gram model using morpheme units to learn sub-word information of words. Each word vector is defined as the sum of the vectors of the morphemes that make up the word. For example, the vector of the word ”밥을” is defined as the sum of the morpheme vectors ”밥”(Noun) and ”을”(Josa). Then the model learns the vector of each morpheme.

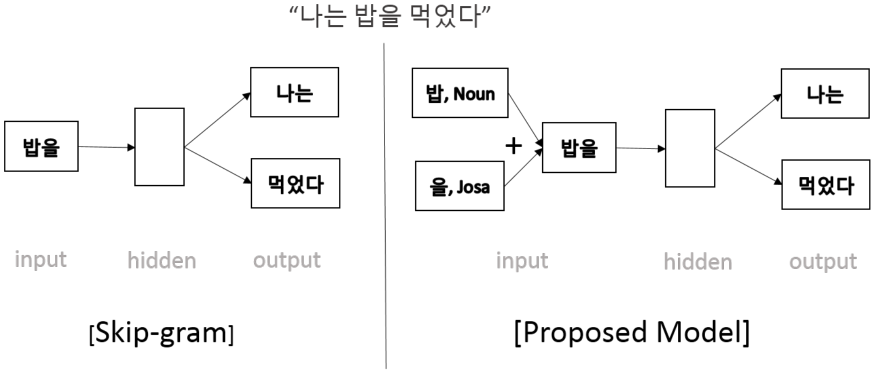


Figure 4: Comparison of skip-gram and the proposed model

The corresponding model can be expressed as follows. When the morphemes forming the word w are $\{m_1,\ \dots,\ m_n\}$ and the vector of morphemes are $\{z_{m_1},\ \dots,\ z_{m_n}\}$, the input vector u_w of word is defined as follows.

$$u_w = \sum_{i=1}^{n} z_{m_i} \tag{3.1}$$

Therefore, the probability that the word o appears around the center word c is defined by softmax function as follows. When z_{c_i}, z_{o_j} are the morpheme vectors of word c, o respectively, and the number of words in the training corpus is $|W|$,

$$P(w_o|w_c) = \frac{\exp\left((\sum z_{o_j})^T v_c\right)}{\sum_{w=1}^{|W|} \exp\left((\sum z_{c_i})^T v_c\right)}. \tag{3.2}$$

As with the skip-gram, when negative sampling is applied to calculate the probability with realistic computational cost, the objective function is defined as,

$$J(\theta) = \frac{1}{|W|} \sum_{t=1}^{|W|} J_t(\theta) \tag{3.3}$$

$$J_t(\theta) = \log\left((\sum z_{o_j})^T v_c\right) + \sum_{i=1}^{k} \mathbb{E}_{w_i \sim P(w)}[\log\sigma\left(-(\sum z_{w_{i_l}})^T v_c\right)]. \tag{3.4}$$

The advantages of the above model are as follows. First, it significantly decreases the number of vectors the model has to learn. While the training corpus used in this paper consists of around 5 million unique words, there were only around 280,000 unique morphemes. Hence, the number vectors that must be learned substantially reduces when learning vectors per morpheme units rather than by word units.

Second, linearity between words can be guaranteed by defining words as the sum of morphemes. For example, in the calculation of word vectors, “밥을” - “밥” + “물” = “물을”, “먹었다” - “먹다” + “맛있다” = “맛있었다” is obvious. This is very effective in reflecting the syntactic relationship between words. Moreover, while the existing models(word2vec, Glove) cannot learn words that did not appear in the training corpus, the proposed model can create word vectors with the sum of its morpheme vectors.

Third, the proposed model is flexible to space errors. This is especially important because spacing errors are very common in the Korean language. Existing learning models are weak against such errors. For example, there is a review saying, “내용이너무좋아요.” The previous model(word2vec or Glove) treats this phrase as a single word and since such words do not appear in the training corpus, the above review cannot be processed. However, the proposed model can assign vectors through the sum of morpheme vectors such as ”내용” + ”이” + ”너무” + ”좋” + ”아요”.

Chapter 4

Experiments

4.1 Training Corpus and Implementation Details

Korean internet news was crawled to construct a training corpus. Around 220 million words form the corpus, around 5 million vocabulary appeared in corpus, and finally there were around 280,000 morphemes.

For comparison with the proposed model, word-level skip-gram and morpheme-level skip-gram were used as the baseline models. For both of our model and the baseline models, we used tensorflow[13] for implementation. Morpheme decomposition and part-of-speech tagging was done by twitter-korean-text[14] library. In addition, following hyparameters were used: 5 window size, 200 dimensions for embedding, 50 minimum number of appearance of words, 0.0001 sampling rate[2] for frequent words, 1.0 learning rate using gradient descent optimizer, and 4 epochs. Implementation of our model and pre-trained morpheme vectors are open sourced.

4.2 Evaluation Methods

The way of evaluating word embedding is largely divided into intrinsic evaluation and extrinsic evaluation.[15] As a process of directly assessing word vectors, the intrinsic evaluation evaluates how well the syntactic and semantic relationships between word vectors are captured. The most representative intrinsic evaluation methods are word similarity test and word analogy test. In the extrinsic evaluation, performance improvement is measured when the word embedding learned by the model is applied to other natural language processing tasks.

For intrinsic evaluation, we conducted word similarity test and word analogy test. In addition, we conducted a qualitative analysis by visualizing morpheme vectors in two dimensional space through PCA(Principal Component Analysis). For extrinsic evaluation, learned vectors from the proposed model were used as input word vectors to a previous CNN-based text classification model[3].

4.3 Intrinsic Evaluation

4.3.1 Word Similarity Test

Word similarity test measures how well the semantic relationship between words have been learned. It is done by composing a series of word pairs and then by comparing the human-evaluated score and cosine similarity between word vectors. Due to the absence of Korean test dataset for word similarity test, this paper used WordSim353[16] composed of English words with Korean translation. WordSim353 consists of 353 pairs of word pairs and contains human-evaluated similarity score between the 2 words for each word pair. This paper excludes cases where Korean translation has destroyed its original meaning. The dataset sample is shown in Table 1.

Table 1: Word similarity test dataset examples

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

To compare the human-evaluated similarity score and cosine similarity of learned word vectors, Spearman coefficient and Pearson coefficient were used to calculate the performance. Figure 5 shows the word similarity test result of skip-gram model and the proposed model. The proposed model outperformed the skip-gram model in both Spearman and Pearson coefficient.

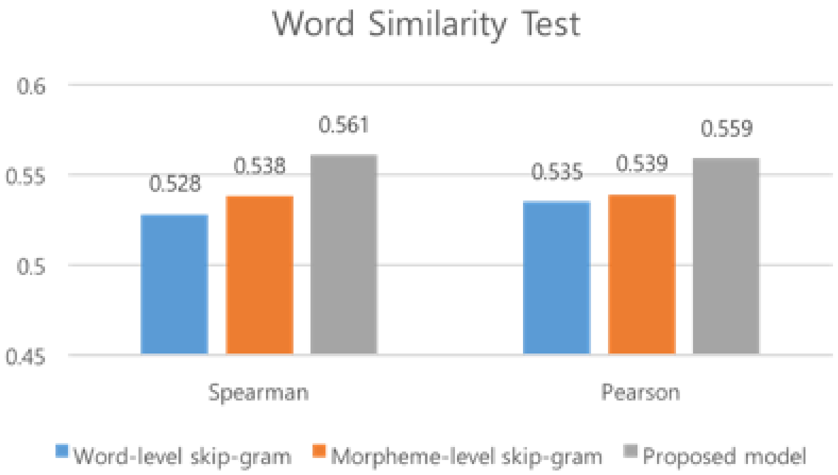


Figure 5: Word similarity test result

4.3.2 Word Analogy Test

Word analogy test evaluates how well the word embedding captures the relationship between words. For example, for the question “Paris” - “France” + “Korea” = “?”, it evaluates whether the model can come up with the word “Seoul”. In the actual analogical stage, such questions would look for the word vector with the highest cosine similarity from the calculated vector. This checks whether the resulting word matches with the ground truth. We built a Korean word analogy test dataset in reference of Google analogy testset[17]. The Google analogy testset is largely divided into semantic relation testset and syntactic relation testset. While it has no difficulty in translating semantic relationships, it is useless in translating syntactic relationships due to the absolute difference in syntactic features of Korean and English. Hence, this paper created its own test dataset reflecting the syntactic features of the Korean language for the experiment. The test dataset consists of 420 semantic relationships and 840 syntactic relationships. Table 2. shows some examples of the this test dataset.

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

The result of word analogy test between the skip-gram model and the proposed model is shown in Figure 6 For both of the semantic relationship and syntactic relationship, the proposed model outperformed skip-gram. The proposed model showed around 12 % higher accuracy for semantic relationship, which it showed 30% higher accuracy for syntactic relationship. This result proves that the proposed model's learned vectors better represent the semantic and syntactic relationships between words. In the case of morpheme-level skip-grams, it is impossible to test for syntactic relationships because it learns the vectors of morphemes rather than words.

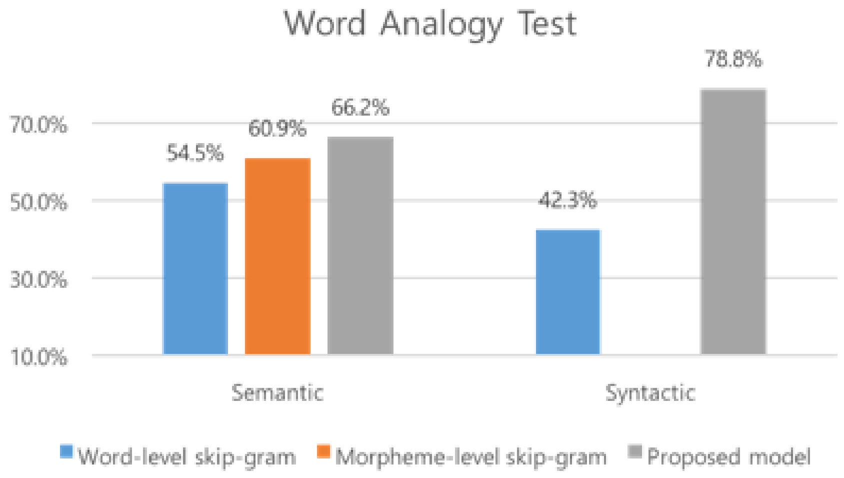


Figure 6: Word analogy test result

4.3.3 Morpheme Vectors Visualization

For a qualitative analysis, we visualized learned morpheme vectors on two-dimensional space through PCA(Principal Component Analysis). For each of the five different parts-of-speech(noun, verb, number, eomi(어미), josa(조

사)), 7 different morphemes were chosen and visualized.

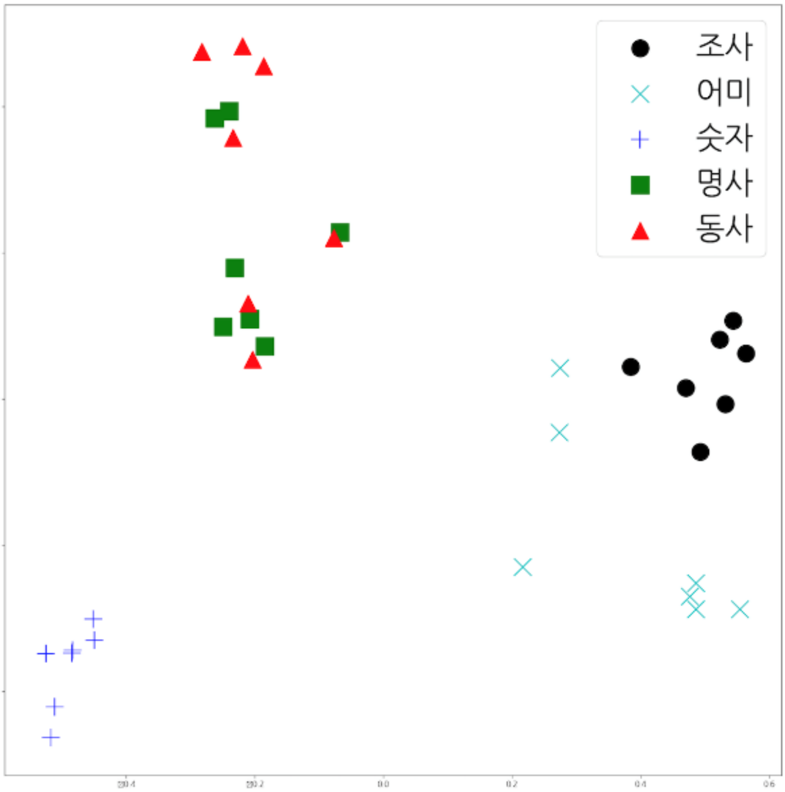


Figure 7: Morpheme vectors visualization

Figure 7 shows the result of this visualization. It shows that the proposed model maps the morpheme vectors well to different spaces according to the part-of-speech. The morphemes with independent meanings (Noun, Verb) were clearly separated from the morphemes without meaning (Josa, Eomi). The differences between number, Josa, and Eomi are also evident.

4.4 Extrinsic Evaluation

Effectively learned word vectors can contribute to performance improvement of various natural language processing tasks. Extrinsic evaluation measures the impact on performance by given vectors when it is given word embedding learned by the model as the input value of other natural language processing tasks.

In this paper, we evaluated how much performance was improved by the word embedding learned by the proposed model when applied to a movie review's sentiment classification for an extrinsic evaluation task. As for dataset, we used Naver movie sentiment corpus[**?**] which consists of about 200,000 reviews. (100,000 positive reviews and 100,000 negative reviews) For the classification model, we used the previously proposed simple yet highly effective CNN(Convolutional Neural Network)-based text classification model[3]. The objective of this task was to classify each movie review as positive or negative.

Table 3: Naver movie sentiment dataset examples

id	document	label
2590934	아이디어가 아주 좋다 재밌다	1
9628757	스토리가 신선하다. 재미있다. 좋다.	1
4886302	말로는 설명할 수 없는 놀랍고 감동적인 영화	1
7439240	저만 별로였던가요 ㅠㅠ 그냥 그랬어요	0
3013859	너무 지루하다. 잠 와 죽는 줄 알았다	0
8159078	재미없다 진심으로 솔직히 재미없다	0

Since skip-gram model learns every word as an independent vector, it cannot handle the words that do not appear in the training corpus. However,

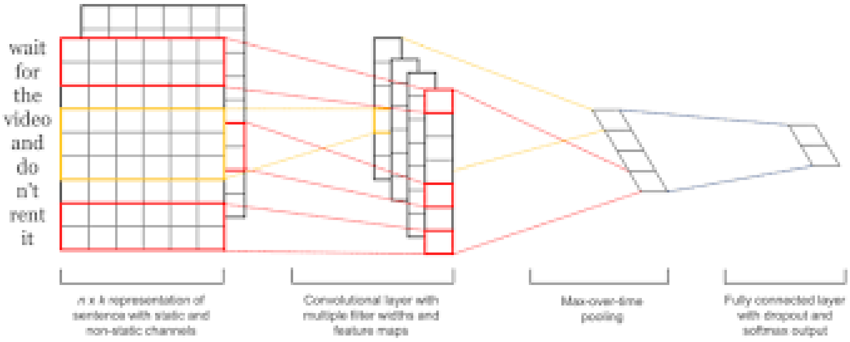

wait
for
the
video
and
do
n't
rent
it
n x k representation of sentence with static and non-static channels
Convolutional layer with multiple filter widths and feature maps
Max-over-time pooling
Fully connected layer with dropout and softmax output

Figure 8: CNN-based text classification model[3]

the proposed model can create vectors to words that do not appear in the training corpus because the model learns morpheme vectors and derive word vectors from them. This is especially important because space errors are very common in the Korean language. The spacing error actually appeared more than 35% in the Naver movie review dataset.

Table 4. shows the number of words processed from Naver movie review dataset based on the number of learned vectors from the skip-gram model's and the proposed model's training corpus. While the number of learned vectors were only a quarter of that of skip-gram, the proposed model could process 5 times more words.

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 handled words	256,049	52,155
The rate of handled words	69%	14%

Figure 9 shows the movie sentiment classification result. As experimented on [3], we tested the static version and the non-static version of CNN-based text classification model. Word vectors are static during CNN training in the static version while they are updated during CNN training in the non-static version. In the static version, the proposed model has much higher classification accuracy then the case of using the learned vector in the skip-gram model. Moreover in the case of non-static version, setting the word vectors learned in the proposed model as the initial value led to a meaningful improvement in accuracy.

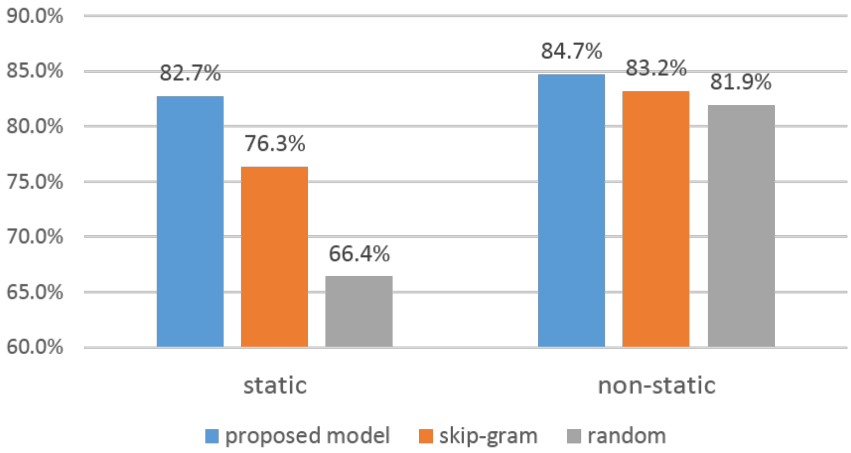


Figure 9: Sentiment classification result[3]

Chapter 5

Conclusion

In this paper, we propose a new model to efficiently learn Korean word embedding which is a morpheme-based extension of the previous skip-gram model. In this model, every word vector is defined as a sum of its morphemes vectors and the neural network learns the morpheme vectors. It can reflect words' internal structure and sub-word information to the vector. To test the efficiency of the embedding, we conducted word similarity test and word analogy test. Learned vectors from the proposed model outperformed the previous skip-gram in both tests. In addition, we proved that they can improve the performance of other natural language processing tasks. Classification accuracy in the previous CNN-based text classification model is significantly improved by only changing the input vectors as the word vectors learned from the proposed model.

The proposed model can be applied not only to Korean language but also to any other languages consisting of morphemes. We expect that it would be especially effective for morphologically rich languages.

As a future work, we will apply the model to the different languages which are rich in word structures such as French and Finnish. Moreover, we will research on a method to improve the embedding model by considering not only the morpheme level information but also syllables and/or character level information.

References

[1] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[2] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems*, pp. 3111–3119, 2013.

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

[4] O. Levy, Y. Goldberg, and I. Dagan, "Improving distributional similarity with lessons learned from word embeddings," *Transactions of the Association for Computational Linguistics*, vol. 3, pp. 211–225, 2015.

[5] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 1532–1543, 2014.

[6] M. Baroni, G. Dinu, and G. Kruszewski, "Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors.," in *ACL (1)*, pp. 238–247, 2014.

[7] Q. Cui, B. Gao, J. Bian, S. Qiu, H. Dai, and T.-Y. Liu, "Knet: A general framework for learning word embedding using morphological knowledge," *ACM Transactions on Information Systems (TOIS)*, vol. 34, no. 1, p. 4, 2015.

[8] T. Luong, R. Socher, and C. D. Manning, "Better word representations with recursive neural networks for morphology.," in *CoNLL*, pp. 104–113, 2013.

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

[10] X. Zhang, J. Zhao, and Y. LeCun, “Character-level convolutional networks for text classification,” in *Advances in neural information processing systems*, pp. 649–657, 2015.

[11] S. Whan *et al.*, “A study on word vector models for representing korean semantic information,” *Phonetics and Speech Sciences*, vol. 7, no. 4, pp. 41–47, 2015.

[12] S. Choi *et al.*, “On word embedding models and parameters optimized for korean,” in *Proceedings of the 28th Annual Conference on Human and Cognitive Language Technology*, 2016.

[13] “Tensorflow.” https://tensorflow.org.

[14] “Twitter-korean-text.” https://github.com/twitter/twitter-korean-text.

[15] T. Schnabel, I. Labutov, D. M. Mimno, and T. Joachims, “Evaluation methods for unsupervised word embeddings.,” in *EMNLP*, pp. 298–307, 2015.

[16] “Wordsim353 - similarity and relatedness.” http://alfonseca.org/eng/research/wordsim353.html.

[17] “Google code archive - word2vec.” https://code.google.com/archive/p/word2vec/source/default/source.

Appendix

Open source: Github

The implementation of the proposed model and test datasets are open sourced at Github (https://github.com/dongjun-Lee/kor2vec).

Train Vectors The proposed model is implemented as a library to learn morpheme vectors using any Korean corpus. In addition, it can be learned by changing the hyperparameters such as vector dimension or window size. Word vector generation codes using morpheme vectors are also open sourced.

Test Dataset The test dataset and codes used for the word similarity test and word analogy test are uploaded. The code visualizing morpheme vectors in two-dimensional space through PCA is also included.

Pre-trained Morpheme Vectors The learned morpheme vectors from the internet news training corpus used in this paper's experiment are available for download.

초 록

단어의 의미를 이해하고 표현하는 것은 자연어 처리에 있어 가장 기초적이면서도 핵심적인 기반이 된다. 이를 위해서 단어를 연속적인 벡터 공간에 표현하는 벡터 공간 모델(vector space model)이 널리 사용된다. 대표적인 단어 임베딩 모델로는 Word2vec이나 Glove가 있다. 이들 모델은 영어에 대해 상당히 효율적인 단어 벡터를 학습한다고 알려져 있다. 하지만 이들의 한계점은 모든 단어에 대해 서로 독립적인 벡터를 학습하기 때문에, 서로 다른 단어들이 공유하는 구조나 단어 내부의 의미를 학습할 수 없다는 것이다. 이러한 한계는 단어의 구조가 풍부한 한국어와 같은 교착어에 치명적으로 작용한다.

본 논문에서는, 이러한 한계점을 극복하기 위해 기존의 skip-gram 모델을 확장하여 각 단어 벡터를 단어를 이루는 형태소들의 벡터의 합으로 정의하고, 형태소들의 벡터를 학습하는 새로운 모델을 제안하였다. 학습된 벡터의 효율성을 측정하기 위해 단어 유사도 평가와 단어 유추 평가를 수행하였다. 또한 학습된 벡터가 다른 자연어 처리 응용의 학습 알고리즘에 얼마나 성능 향상을 가져오는지 실험하기 위하여 기존의 문장 분류 모델에서, 입력 단어 벡터만 변화시켜 분류 정확도를 비교하였다.

주요어 : 단어 임베딩, 형태소 임베딩, 한국어, 인공신경망

학번 : 2016-21220