Hindawi
Mobile Information Systems
Volume 2021, Article ID 5548426, 7 pages
https://doi.org/10.1155/2021/5548426

Research Article

Replacing Out-of-Vocabulary Words with an Appropriate Synonym Based on Word2VnCR

Jeongin Kim, Taekeun Hong, and Pankoo Kim

Department of Computer Engineering, Chosun University, Gwangju 61452, Republic of Korea

Correspondence should be addressed to Pankoo Kim; pkkim@chosun.ac.kr

Received 26 February 2021; Accepted 7 June 2021; Published 17 July 2021

Academic Editor: Jong M. Choi

Copyright © 2021 Jeongin Kim et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The most typical problem in an analysis of natural language is finding synonyms of out-of-vocabulary (OOV) words. When someone tries to understand a sentence containing an OOV word, the person determines the most appropriate meaning of a replacement word using the meanings of co-occurrence words under the same context based on the conceptual system learned. In this study, a word-to-vector and conceptual relationship (Word2VnCR) algorithm is proposed that replaces an OOV word leading to an erroneous morphemic analysis with an appropriate synonym. TheWord2VnCR algorithm is an improvement over the conventional Word2Vec algorithm, which has a problem in suggesting a replacement word by not determining the similarity of the word. After word-embedding learning is conducted using the learning dataset, the replacement word candidates of the OOV word are extracted. The semantic similarities of the extracted replacement word candidates are measured with the surrounding neighboring words of the OOV word, and a replacement word having the highest similarity value is selected as a replacement. To evaluate the performance of the proposed Word2VnCR algorithm, a comparative experiment was conducted using the Word2VnCR and Word2Vec algorithms. As the experimental results indicate, the proposed algorithm shows a higher accuracy than the Word2Vec algorithm.

1. Introduction

Natural languages are those developed naturally in human groups over a long history allowing intentions to be conveyed or opinions exchanged [1]. Unlike artificial languages, natural languages often have ambiguities, omission of words, or paraphrasing. Furthermore, because they require social knowledge, they are extremely difficult to understand for computers [2]. When a natural language is processed using a computer, it is called natural language processing (NLP), and understanding and emulating a natural language in a computer environment is one of the research objectives of NLP [3]. NLP implements an emulation of human language using a machine such as a computer [4]. NLP is deeply related with language studies, as well as the science of language recognition, which is the investigation into the internal mechanism of language [4]. Mathematical and statistical tools are frequently used in the implementation of natural languages, and an NLP uses machine learning tools to a large extent. In NLP, a morpheme analysis refers to analysing word segments of a sentence in terms of morphemes, which are the smallest semantic unit [5]. The problematic parts in a morpheme analysis include disambiguation, new word processing, and morpheme analysis errors caused by out-of-vocabulary (OOV) words, typographical mistakes, and improper word spacing [6–10], which can be seen as critical weaknesses of such an analysis. Among the errors of the morpheme analysis previously mentioned, the most typical problem in analysing a natural language is the replacement of an OOV word with an improper but semantically similar word. This study aims to replace OOV words that produce morpheme analysis errors with semantically similar words. The extraction of replacement word candidates for an OOV word is conducted using the fact that the probability of the replacement word appearing along with surrounding neighboring words of the OOV word is high [7]. The replacement word of an OOV word is selected by measuring the semantic similarities of the replacement word candidates and the surrounding neighboring words of the OOV word and selecting a replacement

word candidate that has the highest similarity. Using this method, an OOV word is replaced with a semantically similar word to improve the performance, resolve morpheme analysis errors, and allow documents to be more intelligently analysed.

2. Related Works

2.1. Replacing Out-of-Vocabulary Word. A dictionary-based word replacement method for an OOV word compares the text string of the OOV word and the text string of a word found in the dictionary, measures a substring or the subsequence length of the two respective words, and replaces the OOV word with a registered word from the dictionary [6]. An N-gram-based word replacement method for an OOV word replaces the OOV word by measuring the probabilistic value of the replacement word for the written part of the OOV word based on other words near the word sequence containing the OOV word [8]. The N-gram technique is a typical probabilistic method. The N-gram method obtains the probability by basically applying learning data. When obtaining or classifying a value of a given word or syllable in a sentence, this method uses n surrounding words or syllables as data [9]. A co-occurrence word-based replacement method of an OOV word uses learning data, in which co-occurring words are generated as word pairs, and chooses a word that is paired with a neighboring word of the OOV word as the replacement word, thereby replacing the OOV word [10]. A co-occurrence word analysis is based on the theory that the correlation of two words is high when the two words are frequently found within a certain range, such as in the same document, paragraph, or sentence. Furthermore, this analytical method has been found to measure the topic similarity of a document more objectively compared with a cocitation analysis or topic classification, which are other measuring methods that can be used to observe a knowledge structure of a certain topic domain. A word-embedding-based replacement method for an OOV word converts words written in a document into vectors and replaces the OOV word with a replacement word having a similar vector value [11].

2.2. Word2Vec. Word-embedding is a technique for learning the vector expressions for every word in a given corpus. Word-embedding facilitates the measurement of similarities between multiple words and an inference based on vector computations using the vectorised semantics [12–17]. Furthermore, mappings can be applied in real numbers on a vector space and can also be expressed with a small number of dimensions [11]. Machine learning methods using such a word-embedding method include the Word2Vec method. Word2Vec was presented in a study conducted by Google in 2013 and is a continuous word-embedding-based learning method created by numerous researchers led by Mikolov [18]. The Word2Vec method is not much different from a conventional neural network language model method but facilitates learning several times faster compared with a conventional method by greatly reducing the number of computations. Hence, it has become a word-embedding method used by most researchers [19]. Word2Vec proposes two network models for learning, namely, a continuous bag of words (CBOW) model and a skip-gram model [20]. The word-embedding-based OOV word replacement method converts an OOV word into a vector value, selects words having similar values as the OOV word on the vector space as replacement word candidates, measures the cosine similarity values of the OOV word and replacement word candidates, and selects a replacement word candidate having a cosine value of close to 1 in order to replace the OOV word. The word-embedding method vectorised words from a large learning dataset quickly and replaced the OOV words. However, it has a problem in that an OOV word can be replaced with a word having the opposite meaning because it does not consider the semantic relationship of the OOV word or the replacement word.

3. Word-to-Vector and Conceptual Relations (Word2VnCR) Algorithm

This section provides a detailed description of the Word2VnCR algorithm used for replacing an OOV word with a morpheme analysis error with a semantically similar word. Figure 1 shows the overall system structure of the Word2VnCR algorithm.

The NUS Short Message Service (SMS) Corpus collected by the National University of Singapore was used for the data in the experiments [21]. Because the texts of the NUS SMS Corpus are sentences composed in a natural language, a preprocessing process has to be applied in order for the computer to understand and analyse the sentences. The preprocessing process is carried out in sequential order of tokenization, part-of-speech tagging, noun extraction, and OOV word extraction. Tokenization is the segmentation of a sentence based on a single word. For English, tokenization is applied based on a blank space. A tokenized text undergoes part-of-speech tagging using the Python Natural Language Toolkit (NLTK). The Python NLTK is an NLP library created in the Python program language for an NLP and provides functions such as part-of-speech tagging, an N-gram, WordNet hierarchy, and WordNet-based similarity [22–24]. This study uses the Python NLTK for part-of-speech tagging of words because the goal of the study is to resolve the problem of more accurately replacing OOV words with semantically similar words rather than applying an accurate part-of-speech tagging of the words. During the noun extraction process, only the words tagged as a noun (NN), plural noun (NNS), proper noun (NNP), or plural proper noun (NNPS) are extracted from the part-of-speech-tagged NNS SMS text. Only noun-type words are extracted because the Python NLTK performs the tagging of such words best among the different types of words. Therefore, the approach developed in this paper also only extracts noun-type words to increase the performance of the OOV word extraction. The OOV word extraction process applies Python Enchant to determine whether a noun-type word extracted from the NUS SMS text is an OOV word. Table 1 shows the preprocessing process of NUS SMS text.

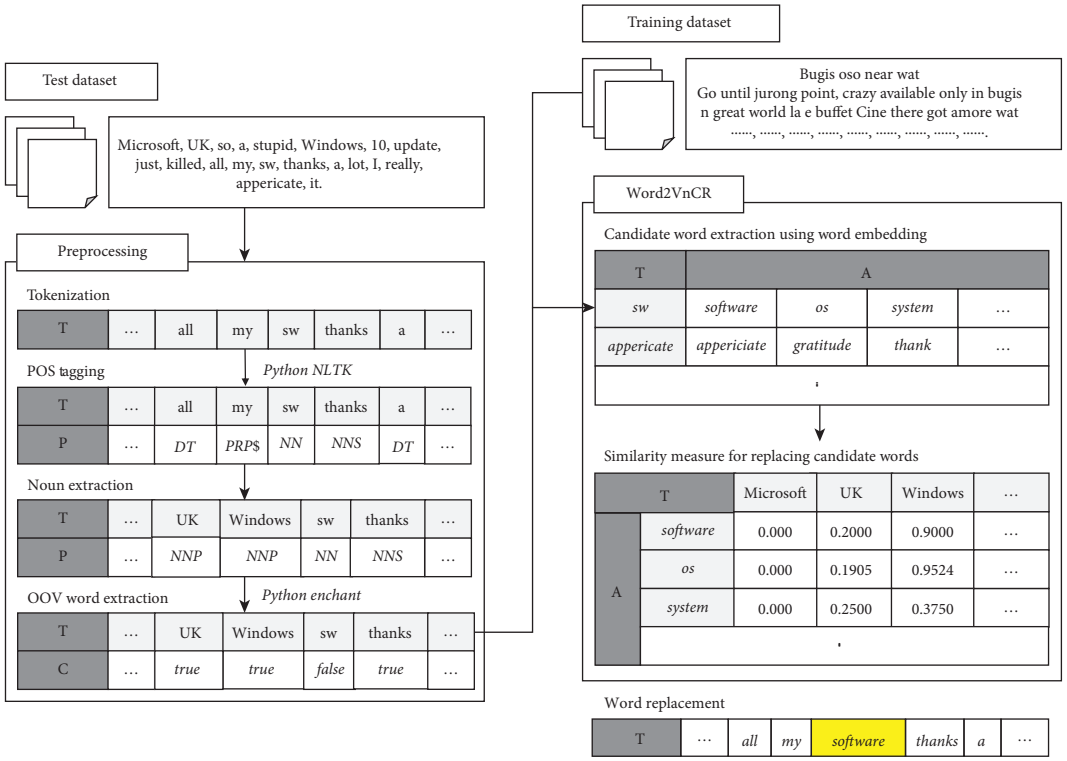

Figure 1: Overview of the Word2VnCR system.

Table 1: A sample of preprocessing tasks on NUS SMS text data.

Process	Example
NUS SMS data	Microsoft UK so a stupid Windows 10 update just killed all my sw thanks a lot I really appericate it
Tokenization	Microsoft, UK, so, a, stupid, Windows, 10, update, just, killed, all, my, sw, thanks, a, lot, I, really, appericate, it.
POS tagging	Microsoft/NNP UK/NNP so/RB a/DT stupid/JJ Windows/NNP 10/CD update/NN just/RB killed/VBD all/DT my/PRP sw/NN thanks/NNS a/DT lot/NN I/PRP really/RB appericate/NN it/PRP
Noun extraction	Microsoft/NNP UK/NNP Windows/NNP update/NN sw/NN thanks/NNS lot/NN appericate/NN
OOV word extraction	Microsoft/true UK/true Windows/true update/true sw/false thanks/true lot/true appericate/false

The OOV words "sw" and "appericate" extracted from an NUS SMS text through a preprocessing are replaced with semantically similar words by applying the Word2VnCR algorithm in written order in the text. As shown on the right side of Figure 1, the selection of a replacement word for an OOV word is applied to find the replacement words of the extracted OOV words. The Word2VnCR algorithm uses a word-embedding method to find replacement word candidates and measures the semantic similarities of the extracted candidate words and the neighboring words of the OOV word. In the candidate replacement word extraction process of OOV words based on word-embedding, the CBOW model learning method of Word2Vec is applied in order to generate the replacement word candidates of the OOV words "sw" and "appericate." The input values in the CBOW model learning method are in a one-hot encoded matrix [0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 1; 0; 0; 0; 0; 0; 0; 0] for the OOV word "sw" and one-hot encoded matrix [0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 1; 0] of the OOV word "appericate." The input value X is multiplied by the weight matrix W, thereby generating an h vector of the hidden layer. Here, W refers to a matrix transformed from the learning dataset, and V and N are the quantity of data and length of the text of the learning dataset. In other words, the 13th and 19th rows of matrix W become an h vector of the hidden layer. The h vector generated through the product of the input value X and the weight matrix W is multiplied by a weight matrix W', in which the row and column sizes are switched; a softmax calculation is then conducted and the output layer y of weight matrix W is generated. In this way, the replacement word candidates of the OOV words "sw" and "appericate" are extracted from the learning dataset. The probability of a word when applying the embedding method can be defined through softmax using the following equation [25]:

$$y_k = \frac{\exp(a_k)}{\sum_{i=1}^{n} \exp(a_i)}. \tag{1}$$

In equation (1), $\exp(a_k)$ and $\exp(a_i)$ are exponential functions, n is the number of words in the output layer, and y_k is the kth output. As shown in equation (1), the numerator

of the softmax function is an exponential function of the input word a_k, and the denominator is the sum of the exponential functions of all input words. Figure 2 shows the process of generating the learning data when the word-embedding method of the Word2VnCR algorithm is applied for the learning dataset of the NUS SMS.

For an input of the OOV word "sw," if the word-embedding method is applied based on the neighboring words of "sw," the following learning data are generated: (my, thanks⟶software), (my, thanks⟶OS), (my, thanks⟶system). Based on this, if "sw" is entered as an input value, "software," "os," "system," "computer," and "pc" are outputted as replacement word candidates of the OOV word "sw." In this paper, when the word-embedding method of Word2VnCR was applied using the NUS SMS Corpus as the learning data, "software," "os," "system," "computer," and "pc" were extracted as the replacement word candidates of "sw," and "appreciate," "gratitude," and "thank" were extracted for "appericate." The semantic similarity was measured between the replacement word candidates of the OOV words "sw" and "appericate" extracted using the word-embedding method and the noun words of the sentence in which the OOV words appeared. The Wu–Palmer (WUP) metric similarity measurement method, a semantic similarity measurement approach of WordNet that can be used to semantically analyse the relationship between concepts, was applied for measuring the semantic similarity. Table 2 shows the results of calculating the WUP similarity value for the replacement word candidates of OOV words and the noun words in the sentence.

According to the semantic similarity measurement results of the Word2VnCR algorithm, "software" was selected from among the replacement word candidates "software," "os," "system," "computer," and "pc" for the OOV word "sw," and "gratitude" was selected among the replacement word candidates "appreciate," "gratitude," and "thank" for the OOV word "appericate." Accordingly, an example sentence containing the OOV words, "Microsoft UK so a stupid Windows 10 update just killed all my sw thanks a lot I really appericate it" was revised as "Microsoft UK so a stupid Windows 10 update just killed all my software thanks a lot I really gratitude it" by replacing the OOV words "sw" and "appericate" with "software" and "gratitude," respectively.

4. OOV Word Replacement Method Using Word2VnCR

This section describes the experiment conducted on the Word2VnCR algorithm described in Section 3, in which OOV words were replaced with semantically similar words, along with the analysis results. The experiment results of the Word2VnCR algorithm used in this paper were comparatively evaluated against those of the Word2Vec algorithm, which represents the word-embedding algorithms. The NUS SMS Corpus consists of 55,963 texts in total. Among them, 44,770 texts, corresponding to approximately 80% of the total number, were used as the learning data of the NUS SMS, and 11,193 texts, or approximately 20%, were used as the test dataset of NUS SMS. Table 3 shows the composition of the NUS SMS test dataset, which consists of 116,195 words in total including 30,211 noun words and 10,526 OOV words. A total of 11,631 words were classified as OOV words among the noun words of the NUS SMS test dataset.

However, to accurately resolve the problem of OOV words using semantically similar terms, in this paper, 1,105 OOV words having a spacing error were excluded. In the experiments conducted on the Word2VnCR and Word2Vec algorithms, the replacements of the OOV words extracted from the test dataset were found, and after replacing the OOV words with the replacement words, it was determined whether the replacement words matched the replacement words of the answer dataset. In this paper, the performance was evaluated by focusing on the accuracy because the aim was to accurately determine whether the OOV words were replaced with the proper replacement words. The experiment results of the Word2VnCR and Word2Vec algorithms are shown in Table 4.

In the results of the comparative experiments of the Word2VnCR and Word2Vec algorithms targeting the NUS SMS test dataset, 87.09% and 90.48% recall rates and accuracy levels of 50.15% and 45.41% were achieved, respectively. It was determined that the recall of the Word2Vec algorithm was higher because its word-embedding method was different from that of the Word2VnCR algorithm, resulting in a replacement of more OOV words. The accuracy of the Word2VnCR algorithm was higher because the similarities between the OOV and replacement words were measured using the semantic similarity method in the Word2VnCR algorithm, whereas they were measured based on the cosine similarity method using the angle of two words located within the vector space for the Word2Vec algorithm. In other words, although the Word2Vec algorithm replaced more OOV words, they were replaced with words having different meanings. It was therefore confirmed that the Word2VnCR algorithm proposed in this paper achieved a higher accuracy than the Word2Vec algorithm when replacing the OOV words. In addition, the experiment results for NUS SMS dataset can be found in Figure 3 for the proposed method suggested in this paper by extracting five OOV words for each sentence.

Figure 3 shows that the fewer the OOV words in the text, the more accurately the suitable words could be replaced. However, the semantic similarity could not be measured when the number of OOV words is increased.

5. Conclusions

In this paper, a novel Word2VnCR algorithm was proposed for the replacement of an OOV word with a semantically similar word when an error in the morpheme analysis occurs. This algorithm applies a method of extracting replacement word candidates having a similar meaning as the OOV word and a method for measuring the semantic similarity between the replacement word candidates and the neighboring words of the OOV word to select a replacement word of the OOV word. For theWord2VnCR algorithm, experiments were conducted to determine whether the replacement word matches the word of the answer dataset after finding a

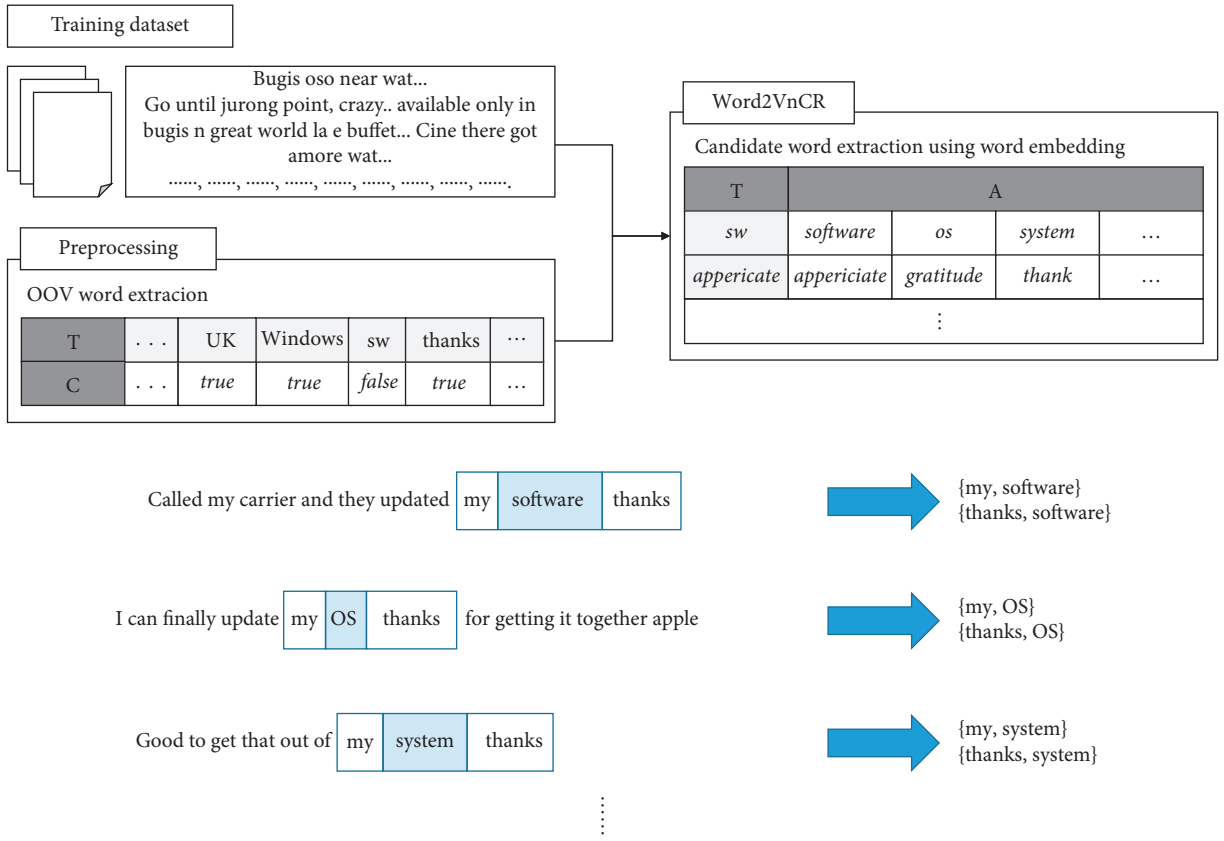

Figure 2: A sample of word-embedding training dataset in Word2VnCR.

Table 2: A sample of extraction of candidate words and semantic similarity scores for OOV words.

OOV word	Extracted candidate words using WordNet	Extracted nouns from a sentence	Word sense of nouns from a sentence	Semantic similarity
SW	software#n#1	Microsoft	—	—
		UK	UK#n#1	0.2000
		Windows	Windows#n#1	0.9000
		update	update#n#1	0.4706
		thanks	thanks#n#2	0.333
		lot	—	—
		appericate	—	—
		Sum of semantic similarity		1.9036
appericate	os#n#3	Sum of semantic similarity		1.9031
	system#n#2	Sum of semantic similarity		1.5151
	computer#n#1	Sum of semantic similarity		1.0042
	pc#n#1	Sum of semantic similarity		0.9122
	appreciate#v#1	Sum of semantic similarity		—
	gratitude#n#1	Sum of semantic similarity		1.3305
	thank#v#1	Sum of semantic similarity		—

Table 3: General composition of experimental NUS SMS dataset.

	Total words	Number of nouns	Number of words classified as OOV	Total OOV words
NUS SMS dataset	116,195	30,211	11,631	10,526

Table 4: Comparison results of Word2VnCR and Word2Vec.

	Total OOV words (A)	Number of words replaced (B)	Number of successful words (C)	Recall (%) (B/A)	Accuracy (%) (C/B)
Word2VnCR	10,526	9,167	4,597	87.09	50.15
Word2Vec	10,526	9,524	4,325	90.48	45.41

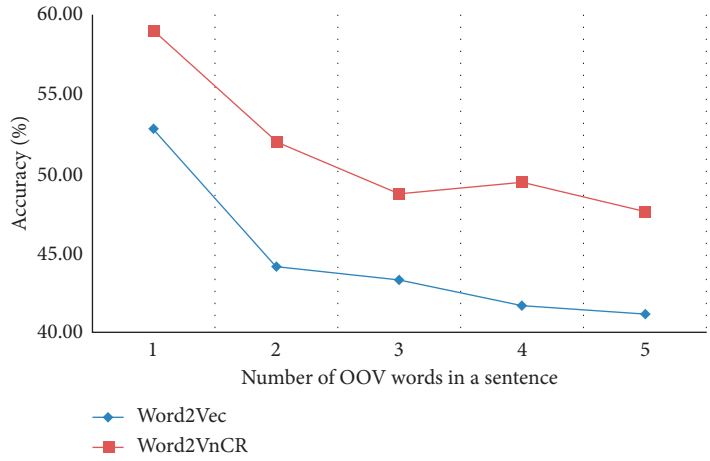

Figure 3: Comparison of Word2VnCR and Word2Vec on NUS SMS dataset.

replacement word of an OOV word extracted from the test dataset and replacing the OOV word with a semantically similar word. The experimental results demonstrated that the Word2VnCR algorithm replaces OOV words with semantically similar words at a higher accuracy than theWord2Vec algorithm. Therefore, it was determined that the Word2VnCR algorithm is the most effective for replacing OOV words with semantically similar words. Finally, the Word2VnCR proposed in this paper exhibited a high accuracy when replacing OOV words with semantically similar words. However, the experimental results are affected by how the learning dataset is constructed because replacement word candidates of an OOV word cannot be accurately extracted if the word-embedding learning of the learning dataset is not properly applied. Therefore, an additional study should be conducted to replace OOV words semantically based on learning data with additional text containing a small number of OOV words, whereby the neighboring words of the OOV terms are composed of semantic words.

Data Availability

The data used to support the findings of this study are available at https://doi.org/10.25540/WVM0-4RNX.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this article.

Acknowledgments

This work was supported by the Ministry of Education of the Republic of Korea and the National Foundation of Korea (NRF-2019S1A5A2A03049825) and National Research Foundation of Korea (NRF) grant funded by the Korea Government (MSIT) (No. NRF-2020R1A2C2007091).

References

[1] T. Atkins and M. Escudier, *Mechanical Engineering Dictionary*, Oxford University Press, Oxford, UK, 2007.

[2] Computational Terminology Compilation Committee, *Computer Internet IT Glossary*, Iljinsa, Seoul, South Korea, 2005.

[3] Natural Language, https://en.wikipedia.org/wiki/Natural_language.

[4] Natural Language Processing, https://en.wikipedia.org/wiki/Natural_language_processing.

[5] Part of Speech, https://en.wikipedia.org/wiki/Part_of_speech.

[6] L. Yongho, "Traffic classification using the automated training set composition with the LCS algorithm," *Journal of KISS: Information Networking*, vol. 1, no. 41, 2013.

[7] H. Tom, S. Yehezkel, and L. Ltay, *Learning TensorFlow*.

[8] T. Yoonshik, "Self-organizing n-gram model for automatic-word spacin," M.Sc. thesis, Gyeongpook National University, Daegu, South Korea, 2007.

[9] K. Devyani, "Fake news classification on twitter using flume, N-gram analysis, and decision tree machine learning technique," in *Proceedings of the International Conference on Computational Science and Applications*, Cagliari, Italy, January 2020.

[10] S. A. Morris, "Mapping research specialties," *Annual Review of Information Science and Technology*, vol. 42, no. 1, pp. 213–295, 2008.

[11] Y. Youngshin, "Relationship analysis of disease and biomarker/microorganisms using word embeddin," M.Sc. thesis, Hallym University, Chuncheon, South Korea, 2017.

[12] Y. Byeonghun, "Correlation analysis of chronic obstructive pulmonary disease (copd) and its biomarkers using word embedding," Master's Thesis, Hallym University, Chuncheon, South Korea, 2018.

[13] E. Frank, "Domain-specific keyphrase extraction," in *Proceedings of the 16th International Joint Conference on Artificial Intelligence*, pp. 668–673, Morgan Kaufmann, Stockholm, Sweden, 1999.

[14] K. Zhang, "Keyword extraction using support vector machine," in *Proceedings of the 7th International Conference on Web-Age Information Management (WAIM2006)*, pp. 85–96, Hong Kong, China, 2006.

[15] H. J. B. Keith, "Phraserate: an html keyphrase extracto," no. 16, Technical Report, Riverside, CA, USA, 2002.

[16] Y. Matsuo et al., "Keyword extraction from a single document using word Co-occurrence statistical information," *International Journal on Artificial Intelligence Tools*, vol. 13, no. 1, pp. 157–169, 2004.

[17] A. Hulth, "Improved automatic keyword extraction given more linguistic knowledge," in *Proceedings of the 2003 Conference on Emprical Methods in Natural Language Processing*, Sapporo, Japan, 2003.
[18] T. Mikolov, "Efficient estimation of word representations in vector space," in *Proceedings of the International Conference on Learning Representations*, Scottsdale, AZ, USA, May 2013.
[19] Theory of Word2Vec, http://blog.naver.com/eun9659/221233326276.
[20] M. Tmoams, "Distributed representations of words and phrases and their compositionality," 2013, https://arxiv.org/abs/1310.4546.
[21] C. Tao, "Creating a live, public short message service corpus: the NUS SMS corpus," in *Language Resources and Evaluation*Springer, Berlin, Germany, 2013.
[22] G. A. Miller, "WordNet: a lexical database for english," *Communications of th ACM*, vol. 38, no. 11, 1995.
[23] G. A. Miller, "Introduction to wordnet: an on-line lexical database," *International Journal of Lexicography*, pp. 235–244, 1990.
[24] B. Steven, *NLTK: the natural language toolkit*, in *Proceedings of the COLING/ACL 2006 Interactive Presentation Session*, pp. 214–217, Sydney, Australia, July 2006.
[25] L. Weiyang, "Large-margin softmax loss for convolutional neural networks," *Proceedings of the 33rd International Conference on Machine Learning*, vol. 48, pp. 507–516, 2016.