Hindawi
Advances in Multimedia
Volume 2022, Article ID 4414207, 8 pages
https://doi.org/10.1155/2022/4414207

Research Article

Research on Word Vector Training Method Based on Improved Skip-Gram Algorithm

Yachun Tang

College of Information Engineering, Hunan University of Science and Engineering, Yongzhou 425199, China

Correspondence should be addressed to Yachun Tang; 1799@huse.edu.cn

Received 16 November 2021; Revised 22 December 2021; Accepted 27 December 2021; Published 27 February 2022

Academic Editor: Qiangyi Li

Through the effective word vector training method, we can obtain semantic-rich word vectors and can achieve better results on the same task. In view of the shortcomings of the traditional skip-gram model in coding and modeling the processing of context words, this study proposes an improved word vector-training method based on skip-gram algorithm. Based on the analysis of the existing skip-gram model, the concept of distribution hypothesis is introduced. The distribution of each word in the word context is taken as the representation of the word, the word is put into the semantic space of the word, and then the word is modelled, which is better modelled by the smoothing of words and the semantic space of words. In the training process, the random gradient descent method is used to solve the vector representation of each word and each Chinese character. The proposed training method is compared with skip gram, CWE + *P*, and SEING by using word sense similarity task and text classification task in the experiment. Experimental results showed that the proposed method had significant advantages in the Chinese-word segmentation task with a performance gain rate of about 30%. The method proposed in this study provides a reference for the in-depth study of word vector and text mining.

1. Introduction

Nowadays, prior training, word vectors have become necessary modules for many natural language processing tasks and machine learning tasks. Word vectors can be directly used as the input features of downstream text tasks, or they can be combined into text or sentence features to indirectly serve as the input of the model.

With the rise of deep learning in recent years, feature learning methods based on neural networks have brought new ideas for natural language processing [1]. Many researchers have devoted themselves to studying some new word vector models or optimizing them to improve performance. For example, the neural network model based on word vector has improved the performance of multiple natural language processing tasks and even achieved the best results among multiple tasks. The word vector model can generally learn semantic information automatically through large-scale unlabeled prediction.

In the past two decades, the research on Chinese-word segmentation has achieved rich results [2]. The dictionary-based matching methods are adopted in the early stage, such as maximum forward matching and maximum reverse matching. However, there are many word boundary ambiguities and unsigned words in Chinese text owing to the complexity of language. In addition, Chinese has the feature of continuous writing of large character set, so it is impossible to solve the problem effectively by using only the dictionary-based matching method.

It is still a crucial technique to deal with Chinese word in that natural language, and it is also a key step in the processing of other Chinese applications. However, it may be a difficult problem to obtain the "semantics" of words directly through the word vector model according to the characteristics of Chinese. Semantics is the relationship between words or phrases and their meanings. It is of little significance to analyze semantics directly from the word level. If the relationship between word and word can be used effectively, it will certainly improve the expression of Chinese words. Early researchers generally established CWE model by assuming the semantics of words [3]. The model goes

beyond the direct use of the English-word vector model. The improved modeling proposed in this study makes the context of words richer, thus improving the semantics of word representation. Experimental results show that the model's effect exceeds the existing individual word vector model.

Therefore, based on these ideas, this study proposes a skip-gram word vector that can improve the performance of text classification tasks and semantic correlation tasks.

2. Related Work

2.1. Word Vector Representation. In the early stage, the representation of words was generally the representation method with statistical information represented in the latent semantic LSA, such as the single hot code and TF-IDF vector, which did not contain semantic information. Hinton et al. first proposed the distributed representation of words in 1986 [4]. Then, Bagnio [5] proposed an N-gram neural probabilistic language model. In the process of training this model, word vectors are generated incidentally, and the research on word vectors is carried out. Bagnio first represented words as an index in the word list and converted them into D dimensional vectors with a mapping matrix, that is, word vectors. Then, the word vectors of C words mentioned above are concatenated and learned through a multi-layer feed-forward neural network to predict the conditional probability that the central word is the current word in the case mentioned above. In the process of model fitting, the objective of optimization is to make the prediction probability maximum likelihood. Among them, the word vector mapping matrix exists as a parameter. During the training of this neural probabilistic language model, the word vectors are also continuously trained to make them close to the semantics in the corpus. Finally, the language model and word vector are obtained. Once the number of vocabularies was too large, the complexity of the model could not be estimated, and the training difficulty was doubled.

In addition to generating word vectors by means of a probabilistic language model, Colbert and Weston [6] proposed another generation model of word vectors. In this model, input is several words with window C, including a central word and the same number of words above and below. It is mapped to a word vector by a mapping matrix and also by a feed-forward neural network. The output layer has only one neuron to rate the connection between the central word and the context. If all the contexts in the corpus are entered, the model cannot be scored. Therefore, the window context in the corpus is taken as a positive example, and the context in which the central word is replaced is taken as a negative example, so that the word vector can be trained.

As you can see from the methods mentioned above, the word vectors are basically generated from the context of a fixed window. It does not consider the global statistical significance of each word, so Pennington et al. [7] proposed combined statistics and a method of word vector generation. First, the word co-occurrence matrix is calculated, and then the error of the prediction probability of the center word and the inclusion probability of the word co-occurrence matrix is minimized in the local context window, to obtain the optimal word vector. This approach, relatively speaking, emphasizes statistics in the text. But it also contains local context information. Experiments have proved its superiority in word similarity and word analogy and named entity recognition.

The word vector model is designed based on the distribution hypothesis. Therefore, no matter what kind of word vector model is, it will conform to two properties proposed by the distribution hypothesis. First, words with similar context will have similar semantics. Second, the space distance of word vectors will be close. This study mainly discusses the different linguistic characteristics of word vectors through semantic correlation experiments.

The classic measure of semantic relevance is the WordSim353 data set [8]. The data set contains 353-word pairs, each of which is rated between 0 and 10 by at least ten annotators. The higher the score, the more the annotator thinks the two words are semantically related or similar. In the evaluation, for each word pair, the average score of all annotators is used as the reference score x. The cosine distance of two-word vectors of word pairs is taken as the correlation score y of the model. The Pearson correlation coefficient between the two sets of values is then measured. The Pearson correlation coefficient measures the linear correlation between two variables, with values between −1 and 1. If the score scored by the model is the same as the score of the manual call, the score will be higher. Specifically, the Pearson correlation coefficient between x and y is defined as the covariance between x and y.

Word vectors can learn the characteristics of syntax and lexical from unlabeled text, and they are often used as a feature of machine learning systems to improve system performance. In this study, a representative task is selected to complete the task of text classification by taking word vector as the unique feature. The expression ability of word vector can be seen from one side by using word vector as the unique feature.

Based on the average word vector text classification (AVG), the weighted average of the word vectors in the text is directly used as the representation of the document, which is characterized by logistic regression to complete the text classification task. The weight is the word frequency of each word in the document. In this study, IMDB data set [9] was selected for the text classification experiment. The dataset consists of three parts, among them, there are 25,000 documents of training set and test set, which are used for training and testing of text classification. There are 50,000 documents in the unmarked part for training word vectors. Then, the task evaluation index is used as the accuracy of text classification.

2.2. Text Classification. Text classification is another important task in text mining. Its goal is to learn a model that can classify each document into one or more categories in each category. However, with the exponential growth of the

amount of network information, the traditional text classification technology relying on manual completion cannot meet the current needs. As an effective technical means, the automatic text classification method based on a statistical model has become a research hotspot in the industry and has been widely used in various fields of social life.

Text classification is the basic task of text processing. The goal is to classify a group of documents into a fixed number of predefined categories, that is, given a group of categories and a group of text, and text classification is the process of finding the corresponding correct category for each document [10–12]. There are two forms of text classification as follows: single-label classification and multi-label classification. In single-label classification, each document only corresponds to a unique category. In multi-label classification, a document can correspond to one or more categories. A complete text classification task process is shown in Figure 1.

Text classification tasks are as follows:

Document selection is to collect documents in different formats. Data preprocessing refers to the corresponding preprocessing operations for problems such as inconsistent, incomplete, or wrong data, such as word tokenization and stem extraction. Index, that is, text-in-text form is transformed into vector form. Common document representation models include vector space model (VSM), TF-IDF weighting method, latent semantic indexing (LSI), etc. [13, 14]. Feature selection is to select some features with strong functions from the original features without affecting the performance of the classifier. In recent years, automatic classification algorithms have been widely studied when selecting a classification algorithm. From traditional machine learning algorithms such as Bayesian classifier, k-nearest neighbor algorithm (KNN), decision tree, and support vector machine (SVM) to a series of deep learning algorithms based on neural network, how to select the appropriate classification algorithm and how to improve the performance of classifier have always been the research goal [15, 16]. The training model is to learn the parameters of the classifier through a group of documents with predefined labels to obtain appropriate classification results. Performance evaluation is to evaluate the decision-making ability of the classifier according to various evaluation indexes such as accuracy, recall, and F1 score [17, 18]. In the process of model testing, the trained model is generally applied to labeled or unlabeled documents, and the performance of the model is evaluated according to the test results.

3. Skip-Gram Algorithm

3.1. Skip-Gram Vector Representation. The traditional Chinese-word segmentation method relies on dictionary matching, and the greedy algorithm is used to intercept possible maximum length words for limited ambiguity resolution. However, there are two obvious defects in the dictionary-based method, that is, it cannot handle the word boundary ambiguity and unsigned words well. In order to solve these two key problems of Chinese-word segmentation, many research works have focused on the word segmentation-based machine learning Chinese-word segmentation method. Based on the Chinese-word segmentation method of word annotation, the basic assumption is that the internal text of a word is highly cohesive while the boundary of the word is lowly coupled with the external text.

Learning and judging word boundaries through statistical machine learning methods are the mainstream practice of current Chinese-word segmentation. The sequence labeling model is adopted to perform BMES labeling [19]. The traditional statistical machine learning methods, which contains the Chinese-word segmentation method based on the HMM model, the Chinese lexical analysis based on the CRF model of word classification, and the improvement of the Chinese-word segmentation method based on word annotation are proposed [20–22]. Because these methods depend on the features of human design, it takes time to design effective features to compare. However, presentation learning can be introduced into machine learning to reduce manpower and improve efficiency.

Skip-gram model consists of a simple three-layer neural network, including input layer, hidden layer, and output layer, as shown in Figure 2.

It is an effective method to learn high-quality concept vectors from many unstructured data. Skip-gram algorithm can generate a concept vector for each concept. When the position in the sentence is closer to the position of the central concept, the concept vector obtained by the skip-gram algorithm is closer to the concept vector corresponding to the central concept in the concept vector space, that is, the close relationship between the concepts in the sentence can be better reflected according to the relationship between the concept vectors. Therefore, compared with the single hot code, the representation of concept vector can reflect the context information of a concept. In Figure 2, the number of neurons in the hidden layer is D, indicating that the concept is mapped to a D-dimensional real number vector. The dimension of concept vector can be artificially set according to experience.

The process of using skip-gram algorithm to learn the word vector representation of concepts is to maximize the average logarithmic conditional probability q_i for each concept to be learned (trained) W_i (i.e., the central concept, represented by a single hot code) in the corpus, which is expressed as follows:

$$q_i = \frac{1}{m}\sum_{i=1}^{m}\sum_{-h\le k\le h, k\ne 0} \lg q\left(W_{i+k} \mid W_i\right), \tag{1}$$

where h is the size of the training text window, W_{i-k} and W_{i+k} are the first k and last k concepts of W_i, respectively, and m is the total number of concepts in the training sentence. $q(W_{i+k}|W_i)$ is defined by the SoftMax function, as follows:

$$q\left(W_{i+k}|W_i\right) = \frac{\exp\left(u'^{m}_{W_{i+k}} u_{W_i}\right)}{\sum_{W=1}^{n}\exp\left(u'^{m}_{W} u_{W_i}\right)}, \tag{2}$$

where u'^{m}_{W} represents the transpose of each concept vector in the concept table, and n represents the total number of

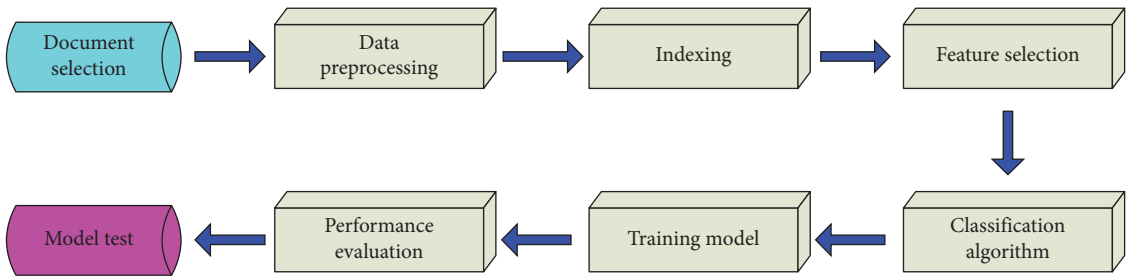

Figure 1: Text classification task flow chart.

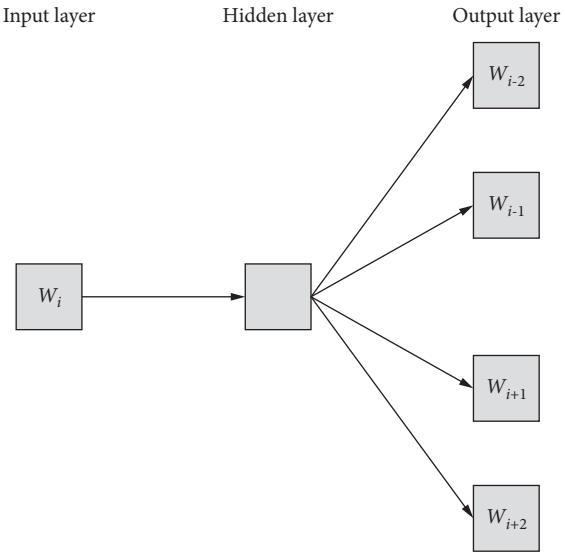

Figure 2: Schematic diagram of skip-gram model.

concepts. Through corpus training, skip-gram model generates concept vectors for each concept in the corpus.

3.2. Skip-Gram Word Vector Representation. The word vector of a word in the context of the target word w is selected in skip-gram model as its context representation. Words are combined with their context to represent a word, which can be expressed as follows:

$$c_l(w_i) = \phi\left(w^{(l)}c_l(w_{i-1}) + w^{(sl)}e(w_{i-1})\right), \tag{3}$$

where w_i is indicated above as $c_l(w_i)$.

$$c_r(w_i) = \phi\left(w^{(r)}c_r(w_{i+1}) + w^{(sr)}e(w_{i+1})\right), \tag{4}$$

where $e(w_{i-1})$ is the word vector of the word w_{i-1}. $w^{(l)}$ is a matrix, which is used to transfer the hidden layer representing the above to the above representation of the next word. φ is a nonlinear activation function.

In this study, the representation x_i of the word w_i is defined as the following formula. The above representation is $c_l(w_i)$, the following representation is $c_r(w_i)$, and the splicing of its word vector is $e(w_i)$:

$$x_i = [c_l(w_i); e(w_i); c_r(w_i)]. \tag{5}$$

3.3. Improved Method. In order to make the representation of the word have more semantic information, the concept of the distribution hypothesis is referred in this study. It is proposed to use the distribution of each word in the context of a word as a representation of the word. Although the word itself still does not have semantic information, the word can be modelled more effectively by using this representation to put the word into the semantic space of the word.

Based on the skip-gram model, the improved method of word vector training is realized in this study. The conditional probability of a word w_j is expressed as follows:

$$\sum_{(w,c\in D)} \sum_{w_{j\in c}} \log P\left(w|w_j\right). \tag{6}$$

In order to train, the simultaneous optimization of the conditional probability of the word w_j and the Chinese character ch_k on the target word w is proposed in this study. The conditional probability is expressed as follows:

$$\sum_{(w,c\in D)}\sum_{w_{j\in c}}\left((1-\beta)\log P\left(w|w_j\right)+\beta\frac{1}{\left|w_j\right|}\sum_{ch_k\in w_j}\log P\left(w|ch_k\right),\right. \tag{7}$$

where ch_k represents the Chinese character word w_j, and $|w_j|$ represents the number of words in the word w_j. The normalization term $(1/|w_j|)$ has the same status in training, which is used to make words with a different number of words. The improved training model structure is shown in Figure 3.

Using the improved training model, not only each word has a corresponding word vector, but also each Chinese character has a corresponding Chinese character vector. The word vector and the Chinese character vector have the same dimension, and the vector representation of the Chinese character and the word is in the same semantic space.

The vector representation of each word and each Chinese character is obtained by the random gradient descent method. The improved training model achieves an improvement over the word representation of the skip-gram model.

4. Experiments and Results

4.1. Evaluation Methods. The word vectors are evaluated from two aspects in this study. First, the linguistic features of word vectors are used to complete the task. Second, the word vector is selected as a feature to improve the performance of task processing. The semantic relevance task is adopted to evaluate the linguistic characteristics of word vectors. The Chinese semantic relevance wordsim-240 dataset and the wordsim-296 dataset were selected for evaluation [23]. Each word pair in the dataset has several digitizers to score it. The higher the score, the more the marker thinks the semantics of the two words are more relevant. In the experimental evaluation, the average of all the scores of the callers is taken as the reference score X. The cosine distance of the word vector of the two words in the word pair is taken as the correlation score Y of the model. Finally, Pearson correlation coefficient ρ_{xy} is calculated to measure the linear correlation between X and Y, which is expressed as follows:

$$\rho_{xy}=\frac{\operatorname{cov}(X,Y)}{\sigma_X\sigma_Y}. \tag{8}$$

The word categorization based on the average word vector is adopted to enhance the word vector as a feature. In the experiment, it is expected to select the Fudan text classification corpus, and the logistic regression model was adopted in text classification tasks.

4.2. Experimental Setup. Since Wikipedia corpus is the best training method for word vectors, Wikipedia Chinese corpus is also selected for improved word vector training. ICTCLAS1 toolkit [24] is used for word segmentation. As the dimension of word vectors generally needs to be 50 dimensions or above, especially when measuring the linguistic characteristics of word vectors, the larger the dimension of word vectors, the better the effect. The context window size is set as 5, and the dimension of word vectors is set as 50.

4.3. Experimental Comparison Method. In order to further prove that the improved word vector training method can improve the semantics of words compared with the original traditional skip-gram word vector model, the main comparison method of this experiment is the traditional skip-gram model, as well as the CWE + P model and SEING model improved by predecessors based on skip-gram model.

4.4. Results. To prove the improvement of the word vector representation by the improved method, the semantic correlation task and text classification task were used to evaluate the training of the improved word vector representation, and the model was compared with that of the leading scholars.

4.4.1. Performance Gain Rate. The absolute values of different evaluation indexes vary greatly. Due to the difference between the indexes, this study can only make a longitudinal comparison of different models within the same index. However, it is difficult to make a horizontal comparison of the performance of a model among different indicators. The relative differences in different evaluation indicators vary greatly. When evaluating the word vector, if the two models are very close in performance values, it will be difficult to quantitatively judge the advantages and disadvantages. This slight difference in performance cannot be caused by the model but may be due to the small number of test sets or errors caused by secondary training.

To solve similar problems, the "performance gain" is used instead of the absolute value of the performance of each task. Performance gain refers to the relative increase in performance of a word vector over a random word vector on a task. The idea of performance gain rate is that each word vector is only compared to the best word vector under the same conditions. According to the special nature of word vector, the performance gain rate of word vector a relative to word vector b is defined as follows:

$$\mathrm{PGR}(a,b)=\frac{P_a-P_{\mathrm{rand}}}{P_b-P_{\mathrm{rand}}}. \tag{9}$$

Word vector a for the same condition, the best word vector best performance gain rate PGR (a, best) can be simply written as the performance gain rate of the word vector a, which is denoted as follows:

$$\mathrm{PGR}(a)=\frac{P_a-P_{\mathrm{rand}}}{P_{\mathrm{best}}-P_{\mathrm{rand}}}. \tag{10}$$

Since the training effect of using Wikipedia corpus to train word vectors is the best, the Wikipedia Chinese is selected for the improved training of word vectors. The word segmentation is performed by the ICTCLAS1 toolkit [25].

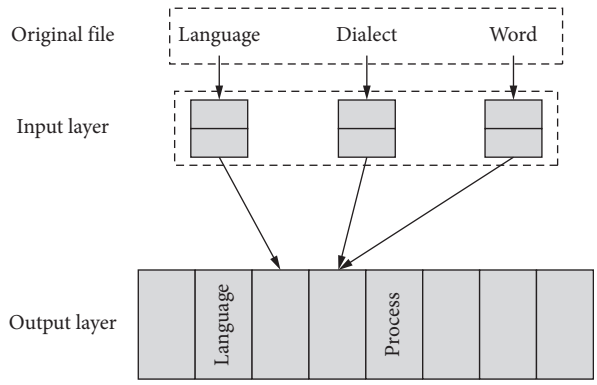

Figure 3: Improved training model structure.

The dimension of word vector needs to select more than 50 dimensions. The context window size is set to 5, and the word vector dimension is set to 50.

4.4.2. Performance on Semantic Relevance Tasks. In the experiment, the training parameters are set to $\beta = 0.5$, that is, the model has a modeling ratio of 1:1 for Chinese characters and words. The experimental results are listed in Table 1.

In order to demonstrate the effectiveness of the proposed vector improvement training, the effects of training under different β are analyzed as a comparison. In the experiment, the values of β from 0 to 0.9 were tried, and the effect diagram is shown in Figure 4.

Through the above figure, the improved word vector training tends to increase first and then decrease. The CWE + P model also has a similar trend. The modeling ratio of Chinese characters at the peak is about 9%, and the performance is 0.5486, which is slightly lower than the improved word vector training. However, the SEING model declined with the increase in the proportion of Chinese character modeling.

4.4.3. Performance on Text Classification Task. In the above part, we verify the improved word vector based on RCNN model from the perspective of semantic related task representation and clarify different semantic features. In this section, the performance effect of text classification task is experimentally verified by the following three models and five-word vectors: AVG, CNN, and RCNN.

The five-word vectors are random word vector, skip gram, CWE + P, SEING, and improved word vector. The models of CWE + P, SEING, and the improved word vectors improve performance consistently with skip gram. The specific results are listed in Table 2.

According to the experiment and analysis, the proposed word vector method makes more words that have a good connection with the context by smoothing the words and Chinese characters. Based on the preservation of the distribution hypothesis, contextual information was adopted to obtain a more efficient word representation in this study. Therefore, the performance of the word-relevant task and the text-category task has a better improvement effect than the previous method. Experiments show that the improved word vector training method based on skip-gram model not only has theoretical significance but also can generate a better word vector model in practical training. The proposed

Table 1: Performance of each model on semantic relevance tasks (X100).

Model	Wordsim-24	Wordsim-29
CWE + P	44.01	53.33
SEING	42.88	49.81
Skip gram	43.61	52.59
Proposed	46.89	54.78

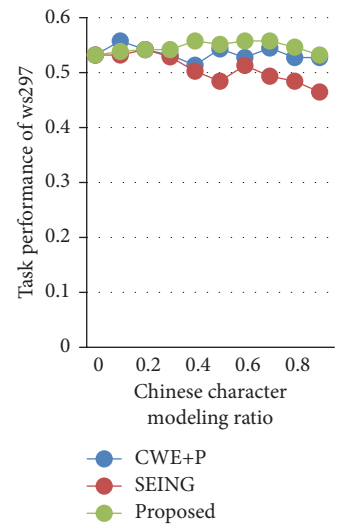

Figure 4: The effect of Chinese character modeling ratio on word meaning.

Table 2: Performance of each model on Chinese-text classification tasks.

Model	AVG	CNN	RCNN
Random word vector	80.55	93.50	94.80
Skip gram	85.51	94.02	95.16
CWE + P	85.99	94.19	95.21
SEING	85.66	94.24	95.32
Proposed	86.33	95.03	95.43

word vector training method is superior to the traditional word vector model both in semantic accuracy and grammatical accuracy [26, 27].

5. Conclusion

An improved training method based on the skip-gram word vector model is proposed in this study. To obtain better representations of Chinese characters, the words in the context of Chinese characters are introduced and the semantic space of words is used to model Chinese characters. Compared with the training method of using English-word vector directly, the performance gain rate of proposed method is increased by 35% in the word segmentation task. On the other hand, due to the increased modeling of Chinese characters in the proposed word vector training, these Chinese characters have established the relationship between some words, which makes the context of words richer and improves the semantic meaning of word expression. In the word sense similarity task, the proposed method is more effective than the existing two models that use Chinese characters to enhance the word representation semantics. In the task of text classification, the word vectors trained by proposed word vectors are also improved when used as features. In addition, large and small noisy corpora still have a great advantage over small and almost noiseless corpora. Therefore, the improved word vector training proposed in this study has a strong advantage in larger corpora. Compared with the skip-gram model, the proposed method can make use of less computational resource overhead to make its training on large-scale corpus possible and does not need the intervention of artificial knowledge, so it is also suitable for the generation of word vectors in other languages. Integrating new methods and ideas to train and apply word vectors has become the focus of this study in the next step. In the future, we can consider expanding the language types of corpora, so that the model can be universal in different languages, and we can consider introducing an external knowledge base to combine a wider range of semantic information and further improve the quality of word vectors [28, 29].

Data Availability

The labeled dataset used to support the findings of this study is available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported in part by the General Project of Hunan Natural Science Foundation (no. 2018JJ2147) and in part by the Youth Project of Hunan Natural Science Foundation (no. 2018JJ3203) and Project of Hunan Science and Technology Department (no. 2019ZK4018) and Hunan University of Science and Engineering Computer Application Special Subject Funding.

References

[1] J. Zhang, X. Hu, Z. Ning et al., "Energy-latency tradeoff for energy-aware offloading in mobile edge computing networks," *IEEE Internet of Things Journal*, vol. 5, no. 4, pp. 2633–2645, 2018.

[2] X. Chen, Z. Shi, X. Qiu, and X. Huang, "Adversarial Multi-Criteria Learning for Chinese Word Segmentation," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pp. 1193–1203, Stroudsburgm, PA, USA, January 2017.

[3] H. Zhi and H. Tao, "An Integrated Model for Effective Saliency Prediction," in *Proceedings of the AAAI Conference on Artificial Intelligence*, AAAI, Palo Alto, CA, USA, February 2017.

[4] G. E. Hinton, "Learning Distributed Representations of Concepts," in *Proceedings of the Eighth Conference of the Cognitive Science Society*, Massachusetts, USA, 1989.

[5] Y. Bengio, R. Ducharme, and P. Vincent, "A neural probabilistic language model," *Journal of Machine Learning Research*, vol. 3, pp. 1137–1155, 2003.

[6] X. Hu, J. Cheng, M. Zhou et al., "Emotion-Aware cognitive system in multi-channel cognitive radio ad hoc networks," *IEEE Communications Magazine*, vol. 56, no. 4, pp. 180–187, 2018.

[7] R. Collobert, J. Weston, L. Bottou, K. Michael, K. Koray, and K. Pavel, "Natural Language Processing (Almost) from Scratch," *Journal of Machine Learning Research*, vol. 12, pp. 2493–2537, 2011.

[8] F. Sun, J.-F. Guo, and Y.-Y. Lan, "A survey on distributed word representation," *Chinese Journal of Computers*, vol. 42, no. 7, pp. 1605–1625, 2019, https://teacher.ucas.ac.cn/~feisun?language=en.

[9] J. Lee, K. Cho, and T. Hofmann, "Fully character-level neural machine translation without explicit segmentation," *Transactions of the Association for Computational Linguistics*, vol. 5, pp. 365–378, 2017.

[10] P. Kumbhar and M. Mali, "A survey on feature selection techniques and classification algorithms for efficient text classification," *International Journal of Science and Research*, vol. 5, no. 5, pp. 1267–1275, 2016.

[11] M. M. Kabir, M. M. Islam, and K. Murase, "A new local search-based hybrid genetic algorithm for feature selection," *Neurocomputing*, vol. 74, pp. 2194–2928, 2011.

[12] L. Manevitz and M. Yousef, "One-class document classification via neural networks," *Neurocomputing*, vol. 70, no. 7-9, pp. 1466–1481, 2007.

[13] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin, "A neural probabilistic language model," *Journal of Machine Learning Research*, vol. 3, pp. 1137–1155, 2003.

[14] C. Lofi, "Measuring semantic similarity and relatedness with distributional and knowledge-based approaches," *Information and Media Technologies*, vol. 10, no. 3, pp. 493–501, 2015.

[15] C. Catal and M. Nangir, "A sentiment classification model based on multiple classifiers," *Applied Soft Computing*, vol. 50, pp. 135–141, 2017.

[16] D. Sarkar, R. Bali, and T. Sharma, "Analyzing movie reviews sentiment," in *Practical Machine Learning with Python*-Springer, Manhattan, NY, USA, 2018.

[17] C. Silberer, V. Ferrari, and M. Lapata, "Visually grounded meaning representations," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 11, pp. 2284–2297, 2017.

[18] L. C. Passaro, A. Bondielli, and A. Lenci, "Learning affect with distributional semantic models," *Italian Journal of Computational Linguistics*, vol. 3, no. 2, pp. 23–36, 2017.

[19] Z. Kai and S. Mao, "Unified framework of performing Chinese word segmentation and part-of-speech tagging," *China Communications*, vol. 9, no. 3, pp. 1–9, 2012.

[20] J. Chung, C. Gulcehre, and K. Cho, "Empirical evaluation of gated recurrent neural networks on sequence modeling," 2014, https://arxiv.org/abs/1412.3555.

[21] K. Xu, R. Ning, and P. Xue, "Extract Chinese unknown words from a large-scale corpus using morphological and distributional evidences," in *Proceedings of the 5th International Joint Conf. on Natural Language Processing (IJCNLP 2011)*, pp. 837–845, Chiang Mai, Thailand, November 2011.

[22] Y. Bin, C. Li, and M. Song, "CHIME: an efficient error-tolerant Chinese pinyin input method," in *Proceedings of the 22nd International Joint Conf. on Artificial Intelligence (IJCAI 2011)*, pp. 2551–2556, Barcelona, Spain, July 2011.

[23] K. Xu and M. Song, "A comparison study of candidate generation for Chinese word segmentation," in *Proceedings of the 7th International Conf. on Natural Language Processing and Knowledge Engineering (NLPKE 2011)*, Tokushima, Japan, November 2011.

[24] Y. Yang, Y. Dan, and W. Lun, "Zero-shot hashing via transferring supervised knowledge," in *Proceedings of the 24th ACM International Conference on Multimedia*, October 2016.

[25] L. Tao, Z. Huang, and J. Wei, "Scalable video event retrieval by visual state binary embedding," *IEEE Transactions on Multimedia*, vol. 18, pp. 1–14, 2016.

[26] D. Q. Nguyen, R. Billingsley, L. Du, and M. Johnson, "Improving topic models with latent feature word representations," *Transactions of the Association for Computational Linguistics*, vol. 3, pp. 299–313, 2015.

[27] B. Chiu, O. Majewska, S. Pyysalo et al., "A neural classification method for supporting the creation of bioverbnet," *Journal of Biomedical Semantics*, vol. 10, no. 1, p. 2, 2019.

[28] X.-H. Phan, C.-T. Nguyen, D.-T. Le, L.-M. Nguyen, S. Horiguchi, and Q.-T. Ha, "A hidden topic-based framework toward building applications with short web documents," *IEEE Transactions on Knowledge and Data Engineering*, vol. 23, no. 7, pp. 961–976, 2011.

[29] J. Lee, W. Yoon, S. Kim et al., "A pre-trained biomedical language representation model for biomedical text mining," *Bioinformatics*, vol. 36, no. 4, pp. 1234–1240, 2020.