Asia Pacific Science Press

Journal of Advances in Engineering and Technology
ISSN(O):3005-7507
ISSN(P):3005-7493
Vol. 2 No.4 (2025)

Vocabulary Semantic Similarity Calculation in Natural Language Processing

Huixiang Xiao[1], Kaige Zheng[2], Xiangyu Li[3]*

1.Chongqing University of Technology, Chongqing, 402160, China

2.Graduate School of Techno Design, Kookmin University, Seoul, 02707, Korea

3.Department of Electronic Engineering, Shanghai Jiao Tong University, Shanghai, 200240, China

Corresponding author: Xiangyu Li, xiangyuli@sjtu.edu.cn

Abstract: Natural language processing (NLP) is a critical research direction in artificial intelligence, where the calculation of vocabulary semantic similarity is the foundation and core work. However, existing calculation methods are faced with problems, e.g., the inability to extract important semantic information. Failure to address this issue can compromise the accuracy of semantic similarity measures in NLP applications. To this end, in this paper, a vocabulary semantic similarity calculation model based on word vectors and convolutional neural networks (CNNs) was proposed. The word vector model was improved using long short-term memory (LSTM) networks, and important semantics were extracted using convolutional layers and ensured semantic order through bidirectional Gated Recurrent Unit. The structure of the Siamese neural network was used to ensure consistency in text encoding. The experimental findings have shown that the proposed model has the highest F1 value in different datasets. In the original Chinese natural language inference (OCNLI) dataset, the Pearson correlation coefficient of the model was 0.021 and 0.018 higher than that of the LSTM network and CNNs, respectively. The accuracy of the similarity calculation in the two datasets was 92.4% and 96.5%, respectively. According to these results, the semantic similarity prediction value of the proposed model can be closer to the true value, and the prediction performance of the model is more excellent.

Keywords: Semantic Similarity; Convolutional Neural Network; Gated Recurrent Unit; Natural Language Processing; Word Vector

Published: Dec 19, 2025

DOI: https://doi.org/10.62177/jaet.v2i4.946

1.Introduction

Natural language processing (NLP) is one of the most important and challenging applications in artificial intelligence, where the progress is catalysed by foundation models. There are lots of opportunities as well as challenges for those valuable information[1]. However, with so much information available, it's hard to avoid personal information overload! The proliferatio of information makes it difficult for people to filter and process a large amount of information, increasing the difficulty of using information[2]. At the same time, a large amount of junk information not only hinders the efficiency of decision making of individuals, companies, and even countries, but may also be exploited and harmful to society, such as online rumors[3-4].

The calculation of vocabulary semantic similarity (VSS) refers to measuring the degree of similarity in meaning between two word objects, which can transform abstract vocabulary similarity relationships into numerical values that can be processed by computers[5]. Natural language processing (NLP) can effectively solve these problems, and VSS calculation is one of the key technologies. Calculating lexical semantic similarity (SS) is widely used in fields such as information retrieval, text classification, and others. After calculating the SS between two words, NLP systems can better understand the content of the text and improve the accuracy and computational efficiency of NLP systems[6].

1.1 Related Work

Existing methods for calculating VSS include word vectors, semantic maps, and deep learning-based methods such as BERT, RoBERTa, and XLNet. However, these methods suffer from low accuracy in calculating SS, insufficient interpretability, and lack of consideration of word order. Ismail et al. developed a new alignment word space method to increase the accuracy of SS calculation for vocabulary. This method combined alignment-based and vector space-based similarity calculation methods to represent words as vectors in a semantic network, and generated an aligned word space matrix for the text based on the word space. The experiment showed good performance in word-order processing of input text and vector models. The accuracy of the result reached 0.7212[7].Pan et al. found that language model training incurred high computational and time cost and therefore proposed a vocabulary enhancement method based on elastic boundaries and multi-sample learning. This method embedded the constraint relationships implied by the vocabulary in neural words, classified the vocabulary constraint set, and allowed positive and negative samples to learn from each other. The benchmark accuracy is shown to be improved to 75% by evaluating the lexical similarity of neural word embeddings[8]. Dai et al. developed a new semantic association model to enhance their understanding of the intrinsic connections between English vocabulary. A tree kernel function was introduced to extract relationships between words, combining dictionary-based and corpus-based methods for calculating VSS to create a dataset of related vocabulary. The experimental results indicated that the more common-sense phenomena there were, the higher the correlation value. This method had the highest classification accuracy for different vocabulary relationships, reaching 85.57%[9]. Ahmad F et al. proposed a hybrid method for calculating SS between sentences. This method considered semantic information including vocabulary databases, word embeddings, and corpus statistics, as well as implicit word order information, to model human common-sense knowledge. Experiments indicated that the highest correlation value could be obtained in both word and sentence similarity, and compared with methods that only use word vectors or based on WorldNet, it could improve by up to 32%, with a Pearson correlation coefficient of 0.8953[10]. Chauhan S et al. proposed an evaluation method combining semantic and syntactic similarity by integrating TF-IDF and word embeddings. Experiments indicated that this method could effectively improve the recognition accuracy of machine translation for natural language[11]. Osth et al. introduced a fused letter position model that combines absolute and relative letter position representations. The experiment demonstrated that the model exhibited moderate capacity to capture the variability of individual words in the false alarm rate[12]. Zhang et al. developed a novel approach using deep learning for part-of-speech-based encoding and context-based decoding. Their study demonstrated that this method could efficaciously reduce the amount of data transmitted and enhance the semantic accuracy between transmitted and recovered messages[13].

1.2 Contributions and Paper Organization

Existing studies have investigated VSS calculation and word vector improvement from multiple perspectives and have achieved meaningful progress. However, existing approaches still suffer from issues such as limited accuracy in semantic similarity computation and insufficient consideration of word order. To address these challenges, this paper proposes a novel VSS calculation model that integrates word vectors with convolutional neural networks (CNNs) and long short-term memory (LSTM) networks. Specifically, the LSTM network is used to enhance the quality of word vector representations, while a combination of CNNs and bidirectional gated recurrent units (Bi-GRUs) is employed to effectively capture sequential dependencies and key contextual semantics before and after vocabulary. This design aims to improve both the accuracy and computational efficiency of VSS calculation.

The main contributions of this paper are summarized as follows:

A novel VSS computation model is developed by integrating word vectors, CNNs, and GRUs to effectively address the low-accuracy limitations and neglect of word order in existing methods.

An LSTM-based enhancement of word vector representations is introduced, which enables better extraction of sequential and contextual semantic information.

Extensive experiments conducted on benchmark datasets (LCQMC and OCNLI) demonstrate that the proposed model achieves higher accuracy (92.4% and 96.5%) and greater stability compared with baseline models such as LSTM and CNN.

The rest of the paper is organized as follows: Section 2 discusses the proposed VSS calculation model, specifically highlighting the word-vector-enhanced LSTM network approach with a CNN-BiGRU model, to obtain semantic usage and context information. Section 3 verifies the efficiency and stability of the proposed model by testing with LCQMC, OCNLI datasets.

The results are compared to the baseline models including CNNs, GRUs, and LSTMs. Finally, Section 4 summarizes the findings of this paper and provides ample scope for future research.

2.Equations and Mathematical Expressions

2.1 Word Vector-based VSS calculation

Vocabulary is the basic unit of semantic expression, including all words and phrases in the language. Calculating the VSS can improve the performance of information retrieval systems and optimize the quality of machine translation[14]. This paper is based on the Skip-gram model proposed by Mikolov et al. which predicts contextual words from a given central vocabulary. The structure of the model consists of an input layer, a hidden layer, and an output layer. The model is trained using the Skip-gram approach to obtain word vector representations. During training, a window is slid on the text data and each sliding generates a training sample, which includes a central word and a series of contextual words[15]. The goal of the model is to maximize the sum of conditional probabilities to predict these contextual words given a central word. The maximization function is calculated as

$$Q = -\frac{1}{n}\sum_{n=1}^{n}\sum_{-c\le j\le c}\log P(w_{n+j} \mid w_n) \tag{1}$$

where n means the number of words, C represents the size of the sliding window, $P(w_{n+j} \mid w_n)$ represents the conditional probability, w_n is the center word, and w_{n+j} is the adjacent word. The conditional probability calculation is expressed as[16]

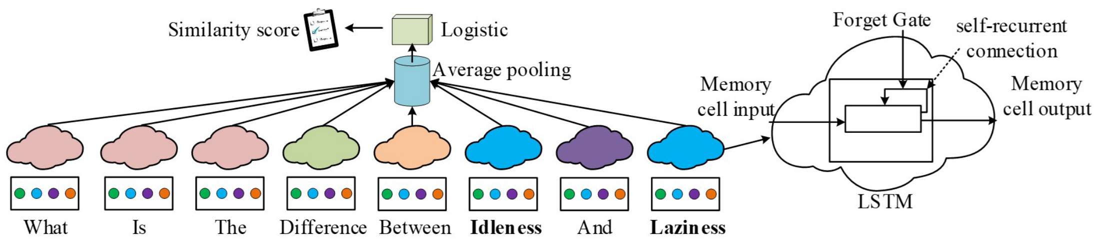
$$P(w_{n+i} \mid w_n) = \frac{\exp(w_{n+i}^{(2)} \cdot w_n^{(1)})}{\sum_{k=1}^{V}\exp(w_k^{(2)} \cdot w_n^{(1)})} \tag{2}$$

where V represents the total number of adjacent words, k represents adjacent words, $w_{n+j}^{(2)}$ represents the row vector of the adjacent word composition matrix, and $w_n^{(1)}$ represents the row vector of the word composition matrix.

To obtain the SS of different vocabulary, the word vector space of the preceding and following vocabulary can be calculated, and the distance between the spatial vectors can be used to calculate it. This paper used the large-scale Chinese question matching corpus (LCQMC) dataset from Harbin Institute of Technology and the original Chinese natural language inference (OCNLI) dataset from Alibaba to obtain the relevant corpus. Stop words are removed from the data by removing words that have no actual semantic or informational value to improve the efficiency of text analysis. Research segments continuous text into independent words according to rules. The remaining data preprocessing also includes XML tag filtering, punctuation removal, and number filtering. Based on the Skip-gram model, this study uses the LSTM network to extract the sequence information of the vocabulary, feedback its contextual vocabulary, and increase the accuracy of the SS calculation.

During training, the complete sentence of the vocabulary to be calculated is input into the model and the specific structure of the model is depicted in Figure1.

Figure1: LSTMs learning word pair co-occurrence utterance computation process



The input is the same sentence or two different sentences with similar words appearing, and the output is the final SS score of the vocabulary. The self-feedback connection refers to the process in a neural network where the output of a neuron is fed back to the same neuron as input, forming a closed loop. This connection method enables neurons to process time-series data and transmit information between multiple time steps. The processed data are input into the average pooling layer, which achieves feature dimension reduction by taking the average of all values within the local receptive domain, reducing the size of the feature map and decreasing computational complexity. Enter the logistic regression layer again, convert the input data into probability, and finally obtain the similarity score of vocabulary. The activation value of the input gate of the storage unit in the LSTM model is calculated as[17]

$$in_t = \sigma(W(h_{t-1}, x_t) + b_i) \tag{3}$$

where in_t represents the activation value of the input gate, σ means the Sigmoid activation function, W means the weight matrix of the input gate, h_{t-1} represents the hidden state of the previous time step, x_t represents the input at time t, and b_i represents the bias term of the inputting gate. The candidate value for the state of memory neurons is

$$C_t^{\%} = \tanh(W_C(h_{t-1}, x_t) + b_C) \tag{4}$$

where the hyperbolic tangent function has a range of $[-1,1]$, W_C is the weight matrix of the candidate memory neuron state, and b_C is the bias term of the candidate memory neuron state. The activation value of the forgetting gate is calculated as

$$f_t = \sigma(W_f(h_{t-1}, x_t) + b_f) \tag{5}$$

where f_t, W_f , and b_f represent the activation value, the weight matrix, and the bias term of the forgetting gate, respectively. The memory neuron state at the current time step can be calculated using (3),(4) and (5) as

$$C_t = in_t \cdot C_t^{\%} + f_t \cdot C_{t-1} \tag{6}$$

where C_t represents the memory neuron state at the current time step, and C_{t-1} represents the memory neuron state at the previous time step. Based on the memory neuron state at the current time step, the activation value and output of the output gate are calculated as[18]

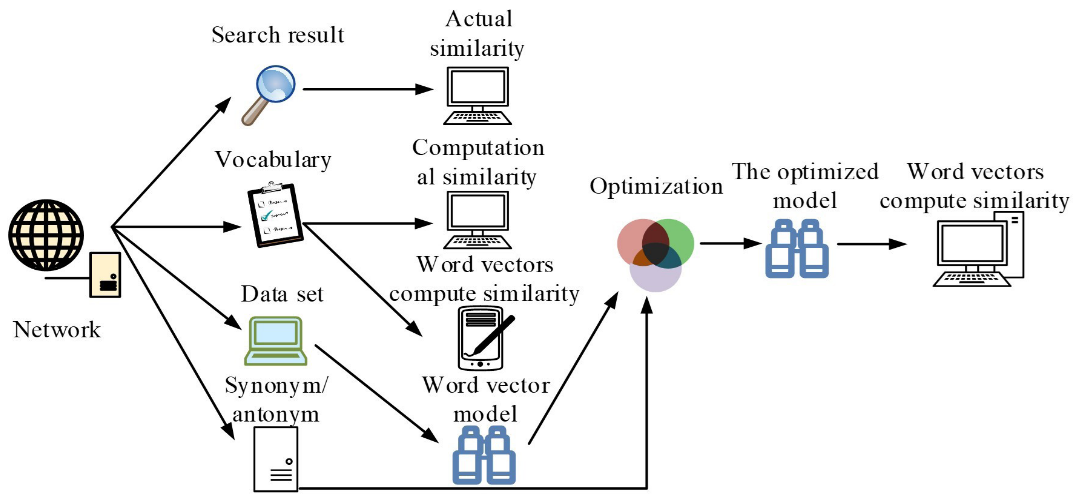
$$\begin{cases} \text{out}_t = \sigma(W_{\text{out}}(h_{t-1}, x_t) + b_{\text{out}}), \\ h_t = \text{out}_t \cdot \tanh(C_t), \end{cases} \tag{7}$$

where out_t, W_{out} , and b_{out} represent the activation value, the weight matrix, and the bias term of the output gate, respectively. h_t represents the final output of the registration code. After passing the final output through the average pooling layer and logistic regression layer, the SS score of the vocabulary is calculated as

$$S = \sum_{i=1}^{10} i \cdot P_i \tag{8}$$

where S means the SS score of vocabulary, i represents the category label, and P_i means the probability that the category label is i . The SS calculation process to improve the Skip-gram model is shown in Figure2.

Figure2: Flow of semantic similarity calculation with improved Skip-gram model



Sentences with similar meanings are crawled from the Internet to calculate the similarity of Chinese or English words, train the improved model, and judge the model training results based on the relevant results. After training is complete, synonyms or antonyms are entered into the model and random gradient descent is utilized to optimize the word vector space. The optimized word vector model is utilized to calculate new word vectors, and finally the SS between the two words is calculated based on the word vectors. The examples of vocabulary relation types extracted from semantic parsing are presented in Table1.

Table1. Vocabulary relation types extracted from semantic parsing of event-related entities

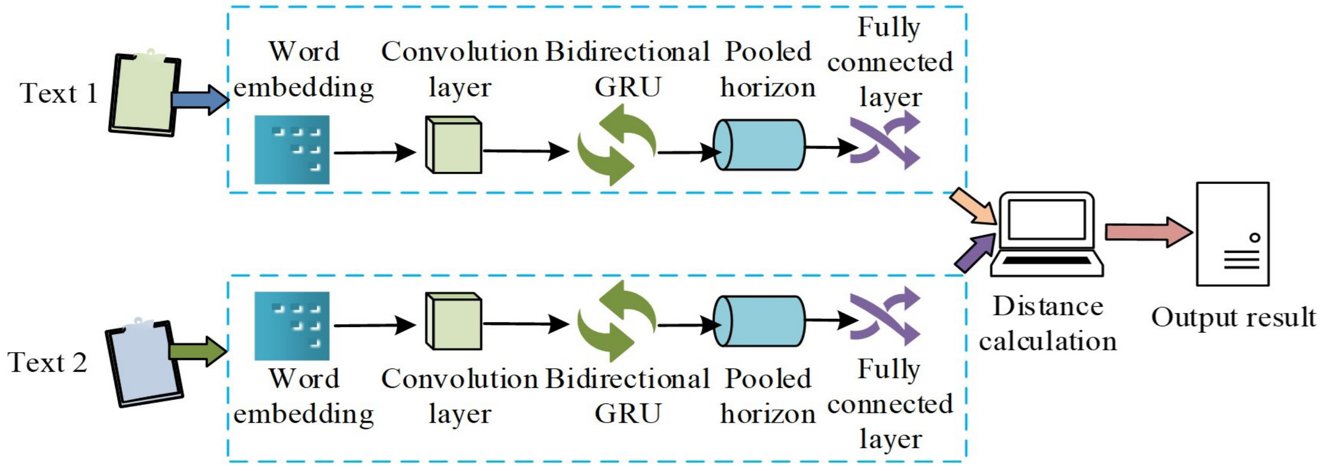
Label	Type	Relation	Target
Soccer Player	SoccerPlayerType	patientOf	Event red card
Referee	SoccerRefereeType	agentOf	Event red card
Event red card	EventType		

2.2 CNN and GRU-based Similarity Calculation

As vocabulary may have the characteristics of polysemy and multiple meanings, it is not enough to calculate SS based solely on individual words in text. It is also necessary to connect with the context to extract semantic information from the vocabulary, further increasing the accuracy of the SS calculation[19]. However, traditional LSTM can only extract sequence information and cannot extract key semantic information.

To address these issues, a CNN-bidirectional gated recurrent unit (CNN-BiGRU) SS calculation model was proposed. As shown in Figure3, the model contains two identical neural networks. The model contains two identical neural networks, each with identical parameters and weights. After inputting a single text or two identical texts, the model enters the embedding layer to transform the text, extracts text features in the convolutional layer, and the extracted features enter the bidirectional Gated Recurrent Unit layer to learn and extract the position information of vocabulary in the context. The extracted feature vectors are then passed into the pooling layer to reduce dimensionality. Subsequently, they are fed into the fully connected layer for complete vectorization. Finally, the similarity score is calculated by computing the distance between the resulting vocabulary vectors. As the input required for the model is string text, it is necessary to convert the lexical content of the text into the corresponding numerical information before entering the embedding layer for conversion. Data preprocessing operations such as stop word removal, tag filtering, and word segmentation are also required., tag filtering, and word segmentation are also required. The preprocessed data are then normalized to fix the length of a single text to 20, which means that the number of words in the sentence is 20. If the vocabulary exceeds 20, remove the excess vocabulary and if it is less than 20, fill it with a zero vector. After processing, a word vector table is used to map the corresponding text content to dense high-dimensional vectors.

Figure3: CNN-BiGRU vocabulary semantic similarity computation model



Significant semantic information is encoded in high-dimensional dense vectors output by embedding layers in convolutional networks. These vectors are then transformed with the use of convolution kernels. The length of convolutional kernels in the model is the same as that of dense vectors. The kernels move backward along the vector, performing local sections of width 3. The resulting one-dimensional (1D) vectors are compressed semantic features. The computation of the convolutional feature maps is given by[20]

$$\mathrm{map} = \mathrm{le} - \mathrm{k} + 1 \tag{9}$$

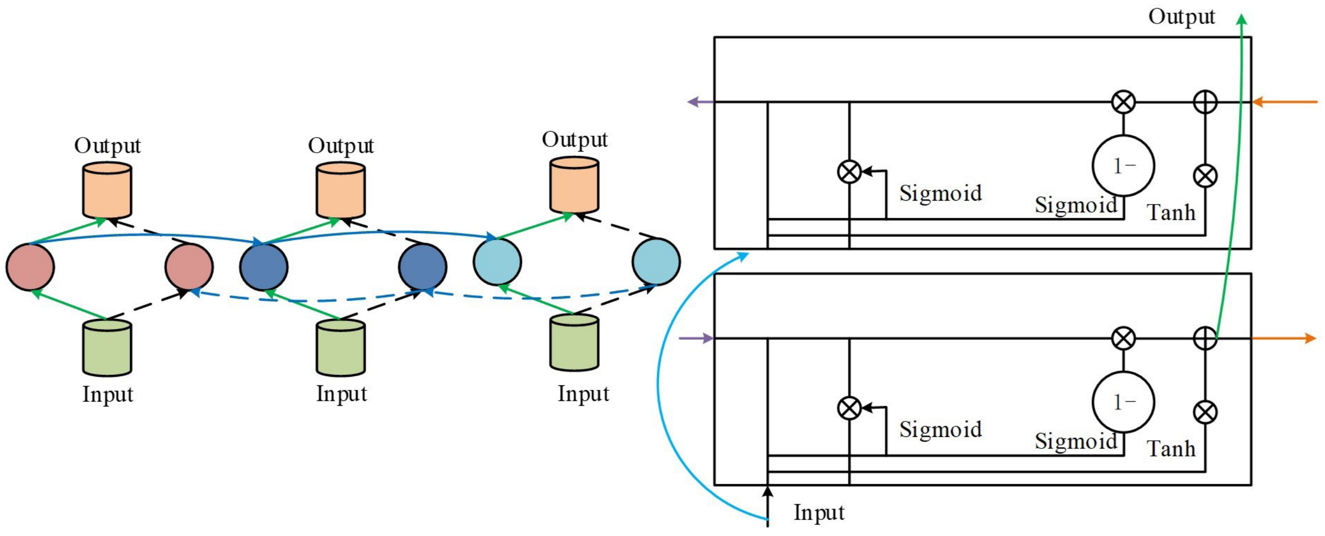
which represents the convolutional feature map, le represents the dimensionality of the convolutional kernel, k represents the convolutional kernel, where the stride of each movement of the convolutional kernel is 1. In convolutional layers, models can assign higher weights to relevant semantic information, enhancing the ability to extract important semantic information. Both bidirectional gated recurrent units and LSTM are transformed from recurrent neural networks, but LSTM has a more complex structure and higher computational cost. Therefore, research is being conducted to use bidirectional gated recurrent units instead of LSTM.

The specific structure of BiGRU is shown in Figure4.In the forward direction, the state update of the gated recurrent unit is output sequentially from the previous moment to the next moment, shifting from the previous text to the following text, and the semantics of the vocabulary can be associated with the content of the previous sentence. In the opposite direction, the gated recurrent unit associates the input of the next moment with the content of the previous moment, allowing the model to link the semantics of the vocabulary with the content of the preceding and following words, thereby extracting important semantic information from the sentence. Although the structure of the CNN-BiGRU SS calculation model is a twin neural network, the structures and parameters of the two neural networks are exactly the same, and the input data are different. Therefore, the two types of output data have consistency and can be used for distance calculation to determine their SS. This paper uses the Manhattan distance method to calculate the distance between vocabulary vectors, and the Manhattan distance is written as

$$d = \sum_{i=1}^{n} |b_i - a_i| \tag{10}$$

where n represents the dimension of the lexical space, b_i and a_i represent the position of the point a and point b in the dimensional space i, respectively. In the similarity calculation layer, backpropagation can be used to determine the difference between the calculated similarity and the actual similarity, and continuously adjust the relevant parameters of the twin neural network to improve its SS calculation ability. When the SS calculation value of two words is less than 1, and the two words in the dataset are similar, the model adjusts the parameters through backpropagation to gradually approach 1 in the calculation result. When two words are different and the actual SS calculation value is not 0, the model back propagates the results and continuously adjusts the weight parameters to gradually approach 0.

Figure4: Specific structure of the bidirectional gating recurrent unit



3. Experimental Results

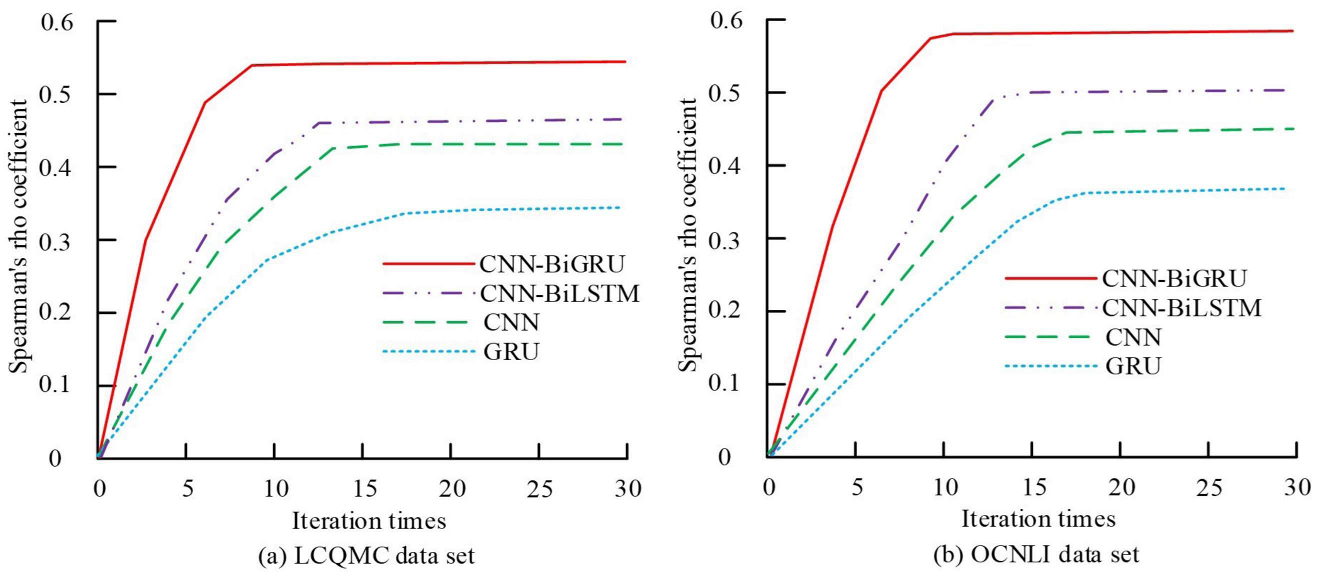
3.1 Effectiveness and Stability Analysis

The experimental configuratio selected for this paper was Windows 10 64 bit operating system, with an Intel i5 12700H CPU and 16GB of memory. The selected datasets for this paper were Baidu's LCQMC dataset and Alibaba's OCNLI dataset. The data sets were broken into training and testing sets in an 8:2 ratio. The comparative models used in this paper included CNN, GRU, LSTM, and CNN-BiLSTM.

The effectiveness comparison of the optimized model is indicated in Figure5.In Figure5 (a), the CNN-BiGRU model converged the fastest and reached the convergence state after 10 iteratios. The maximum Spearman coefficient was 0.07, 0.11, and 0.19 higher than that of CNN-BiLSTM, CNN, and GRU, respectively. In Figure5 (b), the model variation curve was basically the same, but the maximum value of the Spearman coefficient increased, and the maximum value of the CNN-BiGRU model increased by 0.05 compared to Figure 5.

The stability analysis of the optimized model is shown in Figure6.In Figure6 (a), five sets of samples were randomly selected from the training and testing sets, and the Spearman coefficients remained stable above 0.50. Pearson coefficients varied within the range of [0.453, 0.524], indicating that the improved model had good stability. In Figure6 (b), the Spearman and Pearson coefficients of each sample group increased, with similar changes in magnitude.

Figure5: Comparison of the effectiveness of the improved model



The performance comparison of different models is shown in Figure7.In Figure7 (a), the maximum Pearson coefficient of the CNN-BiGRU model was 0.69, which was 0.05, 0.13, and 0.11 higher than CNN-BiLSTM, LSTM, and CNN, respectively. In Figure7 (b), the maximum Pearson coefficients of each model increased, but the convergence position remained unchanged. The maximum values of the CNN-BiGRU model were 0.013, 0.021, and 0.018 higher than those of other models, respectively.

3.2 Comparative Analysis

The results of the VSS calculation for different models are indicated in Table2.It is shown that the effectiveness of the CNN model is better than that of the LSTM and GRU models in both datasets, and the performance of the model that combined CNN and LSTM was further improved. However, among all models, the CNN-BiGRU model proposed in this paper had the best performance, with recall rates 5.7%, 10.0%, 8.6%, and 1.8% higher than CNN, GRU, LSTM, and CNN-BiLSTM, respectively. In the OCNLI dataset, the performance of the model improved by 9.2% compared to the LCQMC dataset, with the largest increase. The similarity calculation accuracy of the CNN-BiGRU model was the highest in both datasets, at 92.4% and 96.5%, which was 8.6% and 12.2% higher than the lowest LSTM.

Figure6: Stability analysis of the improved model

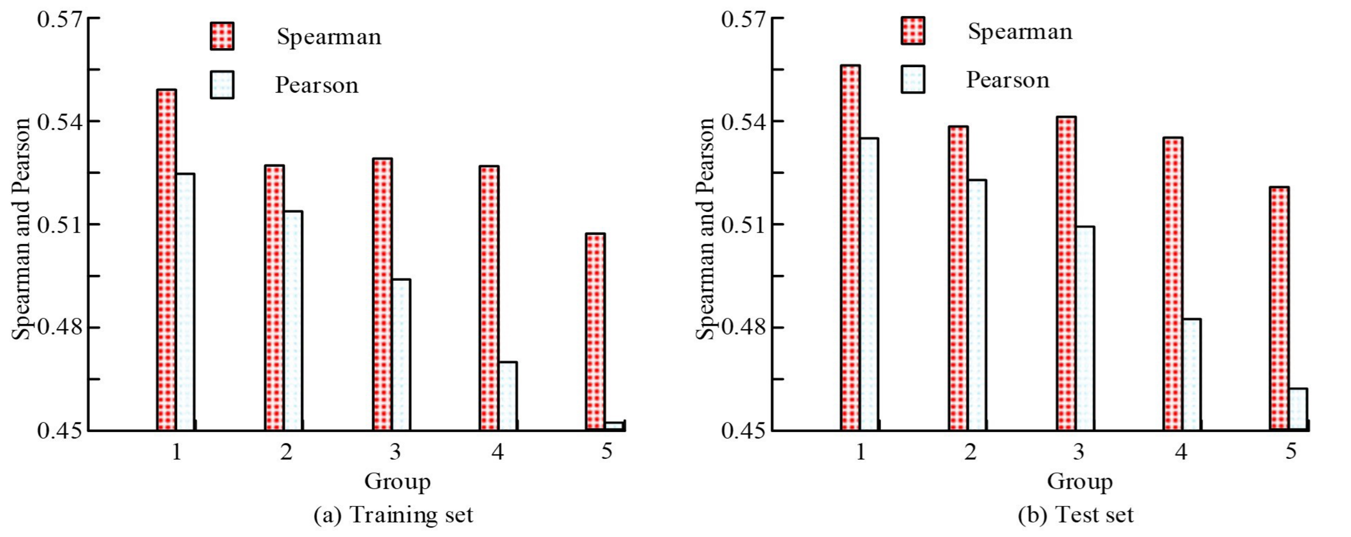


Table2. Performance metrics of different models on LCQMC and OCNLI datasets

Model	**Recall/%**		**Precision/%**		**F1/%**	
	LCQMC	**OCNLI**	**LCQMC**	**OCNLI**	**LCQMC**	**OCNLI**
CNN-BiGRU	88.2	97.4	92.4	96.5	90.2	98.4
CNN	82.5	95.3	87.7	91.2	85.8	96.8
GPU	78.2	94.1	85.3	90.5	81.9	95.5
LSTM	79.6	92.8	83.8	84.3	81.6	89.1
CNN-BiLSTM	86.4	96.5	90.9	94.4	88.4	97.2

The comparison of similarity scores for different vocabulary before and after model improvement is indicated in Table3. Pairs like Important-Significant and Abundant-Plentiful are semantically very close to each other, which explains their high scores. Nonetheless, the baseline model (Before improvement) did not fully detect their similarity with scores of 7.2 and 6.8, which are much lower than the human scores of 9.2 and 8.8. The difference hints that the baseline model, that only uses co-occurrences, might not grasp deep semantic equivalence. After enhancement, the proposed model corrected the estimates of 9.5 and 9.0 closer to the human score. The remaining words with high similarity had a lower probability of appearing simultaneously, and the evaluation of high scores was more cautious, resulting in a decrease in score. After improvement, the model significantly reduced the correlation between the probability of scoring and the number of words that appear simultaneously.

The results of the analysis, i.e., reduction in dimensionality with PCA, of the vocabulary similarity distribution before and after the model improvement are illustrated in Figure8.

As shown in Figure8 (a), the original model suffered from low similarity scores due to a small number of training samples of certain words, which actually clustered words. In Figure8 (b), the model mitigated scores of words with high similarity towards manual scoring, and pulled scores of words with high frequency co-occurrence but low similarity apart in distribution.

Figure7: Performance comparison of different models

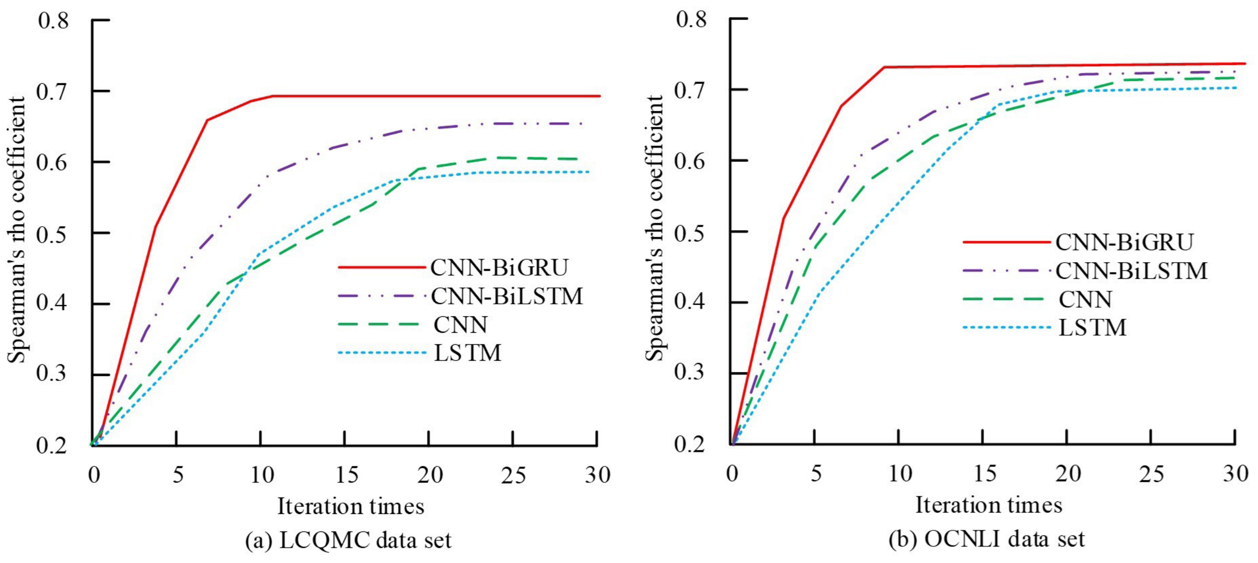


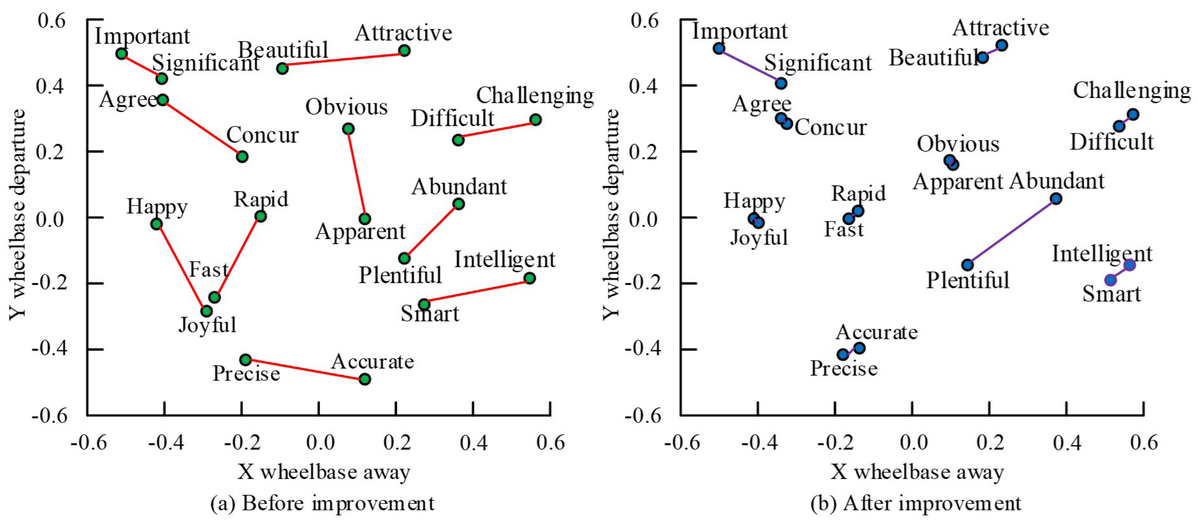
Table3. Comparison of similarity scores of different words before and after model improvement

Serial number	Synonym	Manual scoring	Before improvement	After improvement
1	Happy-Joyful	9.5	6.7	10.0
2	Fast-Rapid	8.4	5.1	9.9
3	Beautiful-Attractive	7.6	4.8	8.7
4	Important-Significant	9.2	7.2	9.5
5	Intelligent-Smart	7.3	5.4	8.8
6	Difficult-Challenging	8.2	6.5	9.5
7	Agree-Concur	9.3	6.0	10.0
8	Obvious-Apparent	8.5	5.9	10.0
9	Abundant-Plentiful	8.8	6.8	9.0
10	Precise-Accurate	8.9	5.3	9.5

4.Conclusions and Future Directions

In this paper, a VSS calculation model based on word vectors and CNNs was proposed to address the issues of low accuracy and lack of consideration of word order in existing methods. The experiment showed that the CNN-BiGRU model had the fastest convergence speed and the maximum Spearman coefficient was 0.07, 0.11, and 0.19 higher than CNN-BiGRU, CNN, and GRU, respectively. In different samples, the Spearman coefficient of the model could remain stable above 0.50, and the variation range of Pearson coefficient was [0.453, 0.524], indicating that the improved model had good stability. The maximum Pearson coefficient was 0.013, 0.021, and 0.018 higher than other models, respectively. The recall rate of the CNN-BiGRU model was 5.7%, 10.0%, 8.6%, and 1.8% higher than CNN, GRU, LSTM, and CNN-BiLSTM, respectively. The accuracy of the similarity calculation was 92. 4% and 96. 5% in the two datasets, which was 8.6% and 12.2% higher than the lowest LSTM. The improved model significantly increased the scoring of highly similar vocabulary, making it closer to manual scoring, significantly reducing the correlation between the probability of scoring and the number of words that appear simultaneously, and reducing scoring errors.

Figure8: Comparison of vocabulary similarity distribution before and after model improvement



In the future, research shall include not only the training and evaluation of common corpora, but also application-related corpora, to increase the generalization and domain adaptability of the model. Running the proposed model on specialized tasks in medicine, law, finance, etc., can demonstrate its ability to handle diverse linguistic structures. These fields exhibit distinct semantic distributions, syntactic structures, and contextual dependencies compared to general-language corpora. For example, medical texts contain technical acronyms and implicit causal relationships; legal documents emphasize the use of precise legal terminology and formal logic; financial news incorporates numerical expressions and domain-specific idioms. Such tasks can expose the model's limitations in processing domain-shifted input and can require adaptation strategies such as domain-specific fine-tuning, multi-task learning, or integration of external knowledge sources, e.g., medical ontologies, legal knowledge graphs. A systematic evaluation in these scenarios could provide deeper insight into the model's transferability, semantic sensitivity, and generalization in real-world natural language processing tasks.

Improving the interpretability of the similarity scoring mechanism is another direction of improvement. Despite their accuracy, deep learning models are typically black-boxes. Utilizing attention mechanisms or saliency visualizations could show what contextual aspects are taken into account when making semantic judgments, most prominently. the model's ability to adapt to low-resource environments and tested on noisy — user-generated text such as online reviews/social media would illustrate further scalability real-world applications.

Funding

No

Conflict of Interests

The authors declare that there is no conflict of interest regarding the publication of this paper.

Reference

[1] Giabelli, A., Malandri, L., Mercorio, F., et al. (2022). Embeddings evaluation using a novel measure of semantic similarity. Cognitive Computation, 14(2), 749–763. https://doi.org/10.1007/s12559-021-09934-5

[2] Yang, D., & Yin, Y. (2022). Evaluation of taxonomic and neural embedding methods for calculating semantic similarity. Natural Language Engineering, 28(6), 733–761. https://doi.org/10.1017/S1351324922000295

[3] Vakulenko, M. O. (2023). Semantic comparison of texts by the metric approach. Digital Scholarship in the Humanities, 38(2), 766–771. https://doi.org/10.1093/llc/fqac078

[4] Oussalah, M., & Mohamed, M. (2022). Knowledge-based sentence semantic similarity: Algebraical properties. Progress in Artificial Intelligence, 11(1), 43–63. https://doi.org/10.1007/s13748-021-00296-8

[5] Wingfield, C., & Connell, L. (2023). Sensorimotor distance: A grounded measure of semantic similarity for 800 million concept pairs. Behavior Research Methods, 55(7), 3416–3432. https://doi.org/10.3758/s13428-023-02118-1

[6] Triandini, E., Fauzan, R., Siahaan, D. O., et al. (2022). Software similarity measurements using UML diagrams: A systematic literature review. Register: Jurnal Ilmiah Teknologi Sistem Informasi, 8(1), 10–23. https://doi.org/10.29303/j.regist.v8i1.3867

[7] Ismail, S., Shishtawy, T. E. L., & Alsammak, A. K. (2022). A new alignment word-space approach for measuring semantic similarity for Arabic text. International Journal on Semantic Web and Information Systems (IJSWIS), 18(1), 1–18. https://doi.org/10.4018/IJSWIS.300729

[8] Pan, J. S., Wang, X., Yang, D., et al. (2024). Flexible margins and multiple samples learning to enhance lexical semantic similarity. Engineering Applications of Artificial Intelligence, 133, 108275–108294. https://doi.org/10.1016/j.engappai.2024.108275

[9] Dai, B. (2024). Relational analysis of college English vocabulary - A reflection based on semantic association network modeling. Applied Mathematics and Nonlinear Sciences, 9(1), 64–82. https://doi.org/10.2478/amns.2024.1.00006

[10] Ahmad, F., & Faisal, M. (2022). A novel hybrid methodology for computing semantic similarity between sentences through various word senses. International Journal of Cognitive Computing in Engineering, 3(6), 58–77. https://doi.org/10.1080/23311916.2022.2124236

[11] Chauhan, S., Kumar, R., Saxena, S., et al. (2024). Semsyn: Semantic-syntactic similarity based automatic machine translation evaluation metric. IETE Journal of Research, 70(4), 3823–3834. https://doi.org/10.1080/03772063.2023.2288606

[12] Osth, A. F., & Zhang, L. (2024). Integrating word-form representations with global similarity computation in recognition memory. Psychonomic Bulletin & Review, 31(3), 1000–1031. https://doi.org/10.3758/s13423-023-02293-4

[13] Zhang, Y., Zhao, H., Wei, J., et al. (2022). Context-based semantic communication via dynamic programming. IEEE Transactions on Cognitive Communications and Networking, 8(3), 1453–1467. https://doi.org/10.1109/TCCN.2022.3176618

[14] Asudani, D. S., Nagwani, N. K., & Singh, P. (2023). Impact of word embedding models on text analytics in deep learning environment: A review. Artificial Intelligence Review, 56(9), 10345–10425. https://doi.org/10.1007/s10462-023-10409-8

[15] Rodriguez, P. L., & Spirling, A. (2022). Word embeddings: What works, what doesn't, and how to tell the difference for applied research. The Journal of Politics, 84(1), 101–115. https://doi.org/10.1086/715618

[16] Mars, M. (2022). From word embeddings to pre-trained language models: A state-of-the-art walkthrough. Applied Sciences, 12(17), 8805–8817. https://doi.org/10.3390/app12178805

[17] Eminagaoglu, M. (2022). A new similarity measure for vector space models in text classification and information retrieval. Journal of Information Science, 48(4), 463–476. https://doi.org/10.1177/01655515211069131

[18] Ichien, N., Lu, H., & Holyoak, K. J. (2022). Predicting patterns of similarity among abstract semantic relations. Journal of Experimental Psychology: Learning, Memory, and Cognition, 48(1), 108. https://doi.org/10.1037/xlm0000948

[19] Gao, Q., Huang, X., Dong, K., et al. (2022). Semantic-enhanced topic evolution analysis: A combination of the dynamic topic model and word2vec. Scientometrics, 127(3), 1543–1563. https://doi.org/10.1007/s11192-022-04337-8

[20] Zhang, Y., Zhang, C., & Hu, F. (2025). Optimization of science and technology project management system based on hybrid semantic similarity evaluation framework. Journal of Computational Methods in Sciences and Engineering, 25(4), 3384–3396. https://doi.org/10.1177/14727978251319396