Hindawi
Scientific Programming
Volume 2022, Article ID 8252492, 9 pages
https://doi.org/10.1155/2022/8252492

Research Article

A Short Text Similarity Calculation Method Combining Semantic and Headword Attention Mechanism

Mingyu Ji and Xinhai Zhang

School of Information and Computer Engineering, Northeast Forestry University, Harbin 150040, China

Correspondence should be addressed to Mingyu Ji; jimingyu@nefu.edu.cn

Received 21 November 2021; Revised 8 March 2022; Accepted 1 May 2022; Published 21 May 2022

Academic Editor: Pengjiang Qian

Short text similarity computation plays an important role in various natural language processing tasks. Siamese neural networks are widely used in short text similarity calculation. However, due to the complexity of syntax and the correlation between words, siamese networks alone cannot achieve satisfactory results. Many studies show that the use of an attention mechanism will improve the impact of key features that can be utilized to measure sentence similarity. In this paper, a similarity calculation method is proposed which combines semantics and a headword attention mechanism. First, a BiGRU model is utilized to extract contextual information. After obtaining the headword set, the semantically enhanced representations of the two sentences are obtained through an attention mechanism and character splicing. Finally, we use a one-dimensional convolutional neural network to fuse the word embedding information with the contextual information. The experimental results on the ATEC and MSRP datasets show that the recall and $F1$ values of the proposed model are significantly improved through the introduction of the headword attention mechanism.

1. Introduction

In machine learning, text similarity is a type of similarity learning, and is a hot research area in the field of natural language processing (NLP). Its influence in several fields such as question answering systems, information retrieval, machine translation, and text classification is becoming increasingly significant [1]. For example, the calculation of the matching degree between query items and documents in retrieval systems and of question and candidate answers in question answering systems are based on text similarity. So, research on semantic similarity calculation is highly significant for the development of NLP-based systems.

However, the calculation of text similarity is a challenging task. As a few short words can contain complex and subtle content, anthropological linguistics is a very esoteric subject. Seemingly different sentences can express very similar semantics, so text should not only be analyzed on different degrees of granularity but also on a deeper level within specific linguistic contexts. Previous studies were limited to the use of traditional statistic models for text similarity calculations, such as the Term Frequency-Inverse Document Frequency (TF-IDF) model based on literal matching, the BM25 model, and latent semantic analysis based on semantic matching [2–4]. However, these models are based on keyword information for matching, which only allows the extraction of shallow information and ignores deep semantic information [5]. Methods based on neural network models use word2vec and other methods to convert words into word vectors, train the model to obtain the feature representation of the sentence, and then use fully connected layers or editing distance equations to calculate the similarity. Hu et al. [6] used convolutional neural networks to model two sentences, and calculated their similarity through the extracted semantic vectors. Sundermeyer et al. [7] applied a long short-term memory (LSTM) to the field for literary NLP. LSTMs solve the problem of traditional recurrent neural networks for long-distance information dependencies of input sequences. Zhu et al. [8] proposed a bidirectional LSTM network based on a siamese network

structure to calculate text similarity; their network traverses the entire text using two LSTM models and comprehensively considers the context information accompanying each word.

In the field of deep learning, current methods for comparing the similarity of two sentences are mainly divided into three types: siamese network frameworks, interactive network frameworks, and pretrained models [9]. The common approach involving siamese networks is to evaluate sentence similarity by mapping the two sentences through the same encoder, comparing them, and evaluating their similarity through the calculation of a loss function [10–13]. This method of using siamese networks to share the parameters can reduce the computation time greatly but does not take into account the interactive relationship between the sentence encoding vectors. It is also difficult to measure the contextual importance of words, which results in poor accuracy. Studies on interactive network frameworks dealing with text similarity include ESIM, BiMPM, and DIIN [14, 15]. In these approaches, the two sentences are first encoded using neural network, the similarity between word sequences in the text is calculated through some complex attention mechanism to formulate an interaction matrix, and the interaction information is finally integrated. However, global information such as syntax and inter-sentence relationships is ignored. Using pretrained models for text similarity-related tasks can lead to good results, as demonstrated by BERT [16] and XLNet [17]. These models are trained on a large-scale corpus and then fine-tuned on a target dataset pertaining to a specific field. However, the pretrained models have too many parameters and it is difficult to change the network structure, which limits their applicability.

The attention mechanism can be abstracted to improve the attention focus on a specific part of the data. The attention mechanism was first adopted to the image processing field to allow focusing on key information in specific image regions. Bahdanau et al. [18] first introduced the attention mechanism into natural language processing tasks, aiming to align the output of the target end with the input of the source end to improve the accuracy machine translation. Subsequently, scholars have proposed various attention mechanisms for different tasks. For example, Cheng et al. [19] proposed a one-way self-attention mechanism in reading tasks to analyze the correlation between current and previous words. He et al. [20] and Shan et al. [21] found that in recommendation systems, an attention mechanism can capture the long-term and short-term interest of users effectively and improve the accuracy of the system. Tan et al. [22] used an attention mechanism based on BiLSTM and CNN to represent separately the question and candidate answers semantically in a Q&A system and answer selection tasks, and used cosine similarity for fusion. The results showed that only a word-level attention mechanism leads to good results.

The main contributions of this paper can be summarized as follows: 1. A semantic similarity calculation method is proposed based on the siamese network structure and combining a convolutional neural network (CNN) and a bidirectional gated recurrent unit (BiGRU). The BiGRU network is used to extract contextual information, and then the CNN network is used to fuse the word embedding information with the contextual information. 2. An attention mechanism based on the headword is proposed, and the output of the BiGRU is weighted and updated to enhance the influence of the headword of the sentence.

2. Methods

The proposed HA-RCNN model for calculating text similarity consists of three components: (1) A sentence encoder. We use a BiGRU to extract the contextual information and combine it with the word embedding information to obtain a representation of each word in the sentence. (2) A headword-based attention mechanism. We use the nouns or verbs that reflect the main information of the sentence as headwords. After obtaining the set of headwords, the output of the BiGRU is weighted and updated. (3) Information fusion. In this part, the word sequences obtained after splicing are fused. Finally, we use cosine similarity as the evaluation function to determine the similarity of the two texts. Figure 1 gives an illustration of the proposed HA-RCNN model.

2.1. Sentence Encoding. Recurrent neural networks (RNNs) are the most common and effective method for dealing with sequences [23]. Through the interconnection between the nodes of the hidden layer, the previous memory is factored in the current output to capture contextual information. However, gradient disappearance and gradient explosion may occur during the training process, so only a small amount of context information can be captured. GRU networks use different functions to control the state of the hidden layer and screen useful information in the sequence, which avoids the gradient explosion problem.

The GRU is a variant of LSTM. Compared with LSTMs, GRU models have a simpler network structure, but their effect is the same as that of LSTM, which leads to greatly reduced training times. GRUs merge the input gate and the forget gate into a single structure called the update gate.

GRUs use a gating mechanism to control input, memory, and other information to make predictions at the current time step. A GRU has two gates, a reset gate and an update gate. Intuitively speaking, the reset gate determines how to combine the new inputs with previous memory, while the update gate defines the amount of previous memory taken into account for the current time step. The special feature of these two gating mechanisms is that they can preserve the information contained in long-term sequences, which will not be lost over time if it is not relevant to the current prediction. If the reset gate is set to 1 and the update gate is set to 0, a standard RNN model is obtained. The update equation of the GRU is as follows:

$$z_t = \sigma\left(w_z x_t + u_z h_{t-1} + b_z\right), \tag{1}$$

$$r_t = \sigma\left(w_r x_t + u_r h_{t-1} + b_r\right), \tag{2}$$

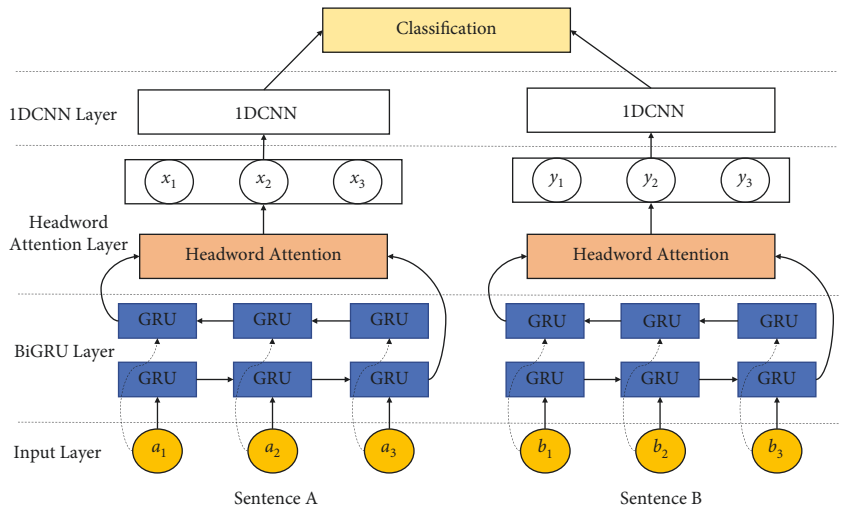

Figure 1: Model structure.

$$h_t' = \tanh\left(w_c x_t + u_c\left(r_t \odot h_{t-1}\right) + b_c\right), \tag{3}$$

$$h_t = z_t \odot h_{t-1} + \left(1 - z_t\right) \odot h_t', \tag{4}$$

$$y_t = \sigma\left(W_0 \cdot h_t\right), \tag{5}$$

where x_t is the current input; σ is a sigmoid function; h_{t-1} and h_t are the hidden states at the previous and current moment, respectively; h_t' is the candidate state at the current moment; and y_t is the current output. Equations (1) and (2) pertain to the update and reset gate, respectively.

In GRUs, information can only be transmitted one-way. In practice, each word may have a dependency relationship with words in its context, so in this paper a BiGRU network is adopted. A BiGRU is composed of a forward and a backward GRU. It traverses the text in two directions and obtains contextual information bidirectionally, thus overcoming the single-direction processing limitation of plain GRUs. The process is shown in Figure 2.

The sentence sequence $S = (w_1, w_2, w_3, \ldots, w_l)$ is obtained through the embedding layer, where L is the sentence length, and w_i is the i-th word in the sentence. $C_L(w_i)$ and $C_R(w_i)$ represent the contextual information on the left and right side of word w_i, respectively. $C_L(w_i)$ and $C_R(w_i)$ are obtained by training the forward and backward GRU, respectively, as shown below:

$$\begin{aligned} C_L(w_i) &= \tanh\left(W_L * C_L(w_{i-1}) + W_{SL} * e(w_{i-1})\right), \\ C_R(w_i) &= \tanh\left(W_R * C_R(w_{i-1}) + W_{SR} * e(w_{i+1})\right). \end{aligned} \tag{6}$$

In the above equations, $e(w_{i-1})$ represents the word embedding of word w_{i-1}; and $C_L(w_{i-1})$ represents the vector representation of the contextual information on the left side of w_{i-1}; W_L represents the transformation matrix of the contextual information vector; and W_{SL} is the matrix that combines the current word vector with the left contextual vector of the next word. C_R is calculated in a similar way.

After extracting the context information using the BiGRU, the contextual information and word embedding

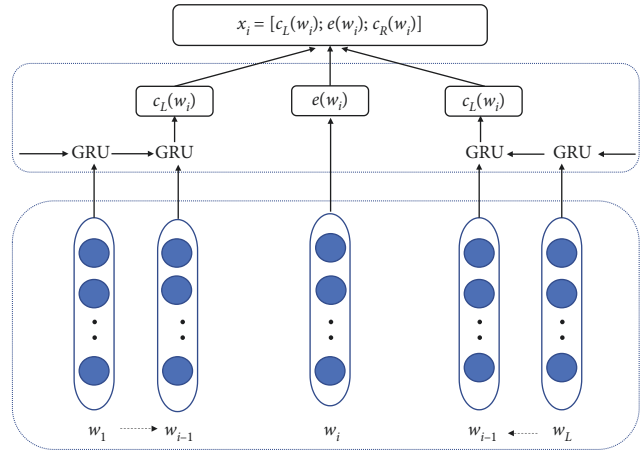

Figure 2: Processing of BiGRU extracting context information.

information are spliced together. Finally, we obtain the semantic representation of the i-th word w_i in the word sequence as $x_i = [C_L(w_i); e(w_i); C_R(w_i)]$.

2.2. Headword-Based Attention Mechanism. In previous studies, attention mechanisms were used to enhance the expression of local information. However, these mechanisms usually take into account the number of occurrences of certain words in a sentence from a traditional statistical perspective, resulting in an increase in the weight of some unimportant words.

Our approach is based on the assumption that the nouns or verbs in the sentence reflect the main information of the sentence, and consider them as headwords. For example, the information expressed in the sentence "Does Ant Check Later require a credit check?" is mainly expressed through the words "require," "Ant Check Later," and "credit check." In the sentence "When will the deposit rate go up?", the information is expressed mainly through "go up" and "deposit rate."

To obtain the headwords, we use the Language Technology Platform (LTP) to analyze the sentence syntactically. As an example, for the sentence "How do I

apply for quota in Huabei?," we obtain the result shown in Figure 3.

The meanings of corresponding tags are shown in Table 1.

After analysis, "apply" is identified as the main verb and is extracted as word_V of the sentence. If the subject or object of word_V is a noun or a noun phrase, it is assigned as the primary noun word_V. If the rhetorical and juxtaposed elements of word_N also contain nouns, they are also added to word_N. word_N and word_V form the headword of the sentence. In addition, if the main verb cannot be extracted through syntactic analysis, the headword is extracted directly through the part of speech. Therefore, there may be multiple word_N instances. For example, in Figure 3, the subject of "apply" is "*I*" and the object is "quota." Because "*I*" is a personal pronoun rather than a noun or noun phrase, the object "quota" is a noun, and the noun "Huabei" is a modifier of "quota," the headwords of the sentence in Figure 3 are {apply, quota, and Huabei}.

After the headwords have been obtained, they are denoted as $v_{HW} = (S_1, S_2, \ldots, S_l)$, where l is the number of words in the set. We use v_{HW} to update the weighting output of the forward (C_L) and backward (C_R) GRUs. Specifically, for each vector in v_{HW}, the similarity with C is calculated separately to obtain the maximum value v_t. The calculation method is as follows:

$$v_t = \max\left\{\text{cossin}\left(c\left(w_i\right), s_j\right)\right\}, \quad 1 \le j \le l. \tag{7}$$

By updating c with v_t, we obtain the information enhancement representation based on the attention of the headwords.

2.3. Information Fusion. CNNs extract the local information of the text through a fixed-size convolution kernel, and use a pooling layer to reduce the amount of calculation and retain key information. Because the convolution kernel has a fixed window, it is always possible that some important information will be lost. Although this problem can be solved using multiple windows of different sizes, this solution will lead to an increase of the number of calculations.

We use a one-dimensional CNN network (1DCNN) to fuse the information of the spliced word sequences. The calculation process is as follows:

$$y_i = \text{CNN}\left(x_i\right), \quad i \in [1, L], \tag{8}$$

where y_i represents the feature representation corresponding to x_i after 1D convolution processing. The calculation process is shown in Figure 4.

Finally, after obtaining the vector representation of the two sentences, we use their cosine distance to determine whether the two texts are semantically similar. The corresponding equation is:

$$\text{similarity} = \frac{\sum_{i=1}^{n} S_L^i \times S_R^i}{\sqrt{\sum_{i=1}^{n} \left(S_L^i\right)^2} \times \sqrt{\sum_{i=1}^{n} \left(S_R^i\right)^2}}. \tag{9}$$

3. Experiment

We conducted experiments to demonstrate the effectiveness of the proposed HA-RCNN model. In this section, the experimental datasets and evaluation criteria are first introduced, followed by a detailed analysis of the experimental results.

3.1. Datasets. Two datasets were used to verify the performance of the model, as follows:

(a) The Ant Financial NLP Competition (ATEC) dataset was obtained from Ant Financial's 2018 competition. Each pair of sentences in the dataset comes from questions received by an intelligent customer service and was labeled with "1" to indicate that the two sentences are semantically similar, and 0 when the sentences were not similar.

(b) The Microsoft Research Paraphrase Corpus (MSRP) is a collection of sentence pairs obtained from news on the web. As in the ATEC dataset, each pair of sentences was labeled with a 0 or a 1 for dissimilarity or similarity, respectively.

In the ATEC dataset, the training set contains 100,000 sentence pairs and the test set contains 10,000 sentence pairs. During preprocessing, we found that the ratio of positive and negative samples in ATEC was significantly unbalanced at about 4.5 : 1. In order to avoid the impact of sample imbalance on the experiment, we selected 32250 pairs of sentences for training and 6450 pairs of sentences for testing, with positive and negative samples accounting for half of each subset. The MSRP dataset contains 5803 sentence pairs, including 4077 pairs in the training set and 1726 pairs in the test set. Due to the small number of samples in MSRP, we did not segment the dataset. The standard format of the two datasets is shown in Table 2.

In the experiments, we used accuracy, precision, recall, and $F1$ as the evaluation criteria, calculated as follows:

$$\begin{aligned} \text{accuracy} &= \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}, \\ \text{precision} &= \frac{\text{TP}}{\text{TP} + \text{FP}}, \\ \text{recall} &= \frac{\text{TP}}{\text{TP} + \text{FN}}, \\ F1 &= 2 * \frac{\text{pre} * \text{rec}}{\text{pre} + \text{rec}}, \end{aligned} \tag{10}$$

where TP is the number of positive samples predicted as positive samples; TN is the number of negative samples predicted as negative samples; FP is the number of negative samples predicted as positive samples; and FN is the number of positive samples predicted as negative samples.

3.2. Experimental Results and Analysis. In order to prove the effectiveness of HA-RCNN, we compared it with state-of-the-art models used for the same application.

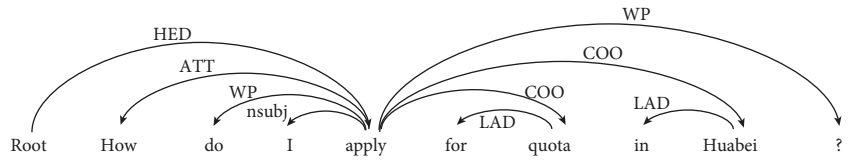

Figure 3: Results of syntax analysis.

Table 1: The meaning of each tag.

Tag	Description
ATT	Attribute
COO	Coordinate
HED	Head
LAD	Left adjunct
WP	Punctuation

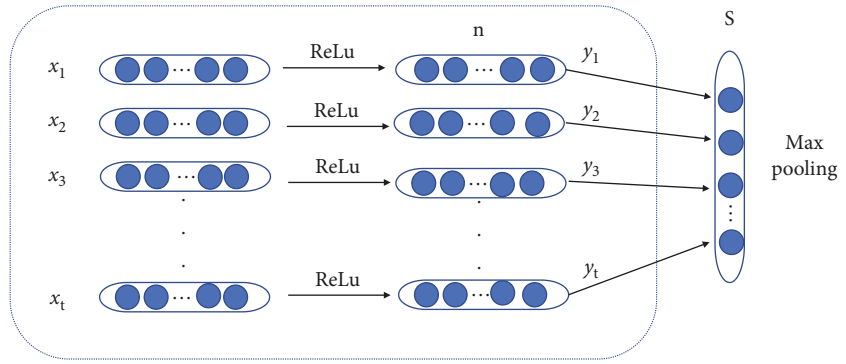

Figure 4: 1DCNN and max-pooling.

Table 2: Standard format of datasets.

Dataset	Sentence A	Sentence B	Label
ATEC	我的花呗账单可以提前还吗 (can I pay my bill in advance)	怎么推迟花呗的还款日期 (how to postpone the repayment date of the bill)	0
	花呗额度怎么提升 (how can I increase Huabei's quota)	花呗额度能不能提额 (can Huabei's quota be raised)	1
MSRP	Air commodore quaife said the hornets remained on three-minute alert throughout the operation	Air commodore john quaife said the security operation was unprecedented	0
	Still, he said, "I'm absolutely confident we're going to have a bill"	"I'm absolutely confident we're going to have a bill," frist, R-Tenn., said Thursday	1

MMNF [24]: This model uses the Jaccard coefficient based on the part of speech, TF-IDF and the Word2Vec-CNN model to measure sentence similarity through weighted calculation.

BiGRU + Dilated [25]: This model uses constituency parsing and dilated convolution to reduce the missing elements in long sentences and increase the important information in short sentences. At the same time, the receptive field is extended to capture semantic relevance in two-dimensional space.

Tree-lSTM [26]: The model uses a control input to model the relationship between the two inputs. To calculate the sentences' similarity, their semantic representation is embedded into a dense vector through syntactic parsing and other operations.

CNN-lSTM [27]: This model is based on the siamese neural network structure. A CNN and LSTM are used to obtain the local and global information of the text, respectively.

Multi-Feature [28]: This model evaluates the similarity of two sentences in terms of words, word order, and word vectors, and introduces word vectors in traditional statistical-based discriminative method to make judgments, taking into account the structural information of the sentences.

The results of the comparative experiment are shown in Figures 5 and 6.

As can be seen from the data in the diagram, the performance of the CNN-LSTM model is poor. Although the text is analyzed from both local and global perspectives, the model focuses only on few factors and ignores the influence

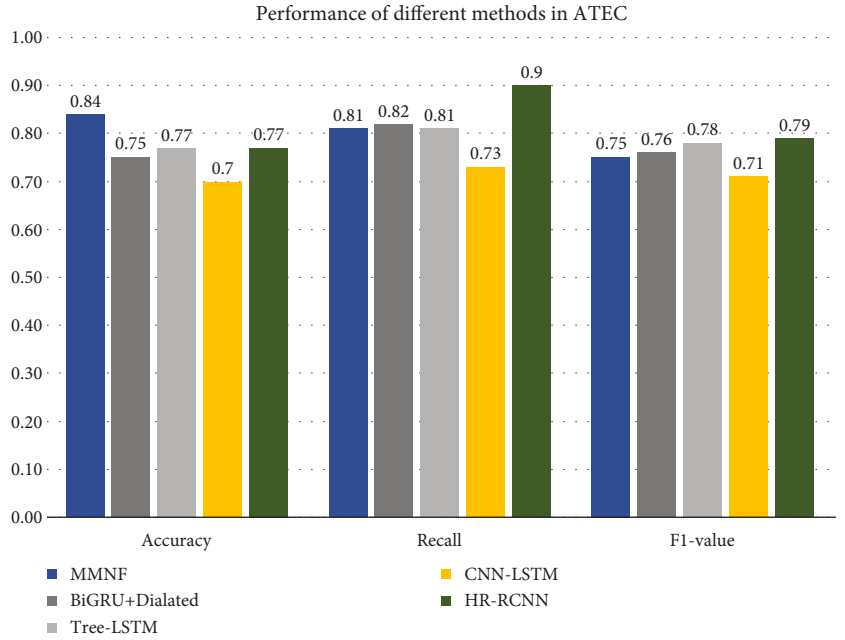

Figure 5: Performance of different methods on the ATEC dataset.

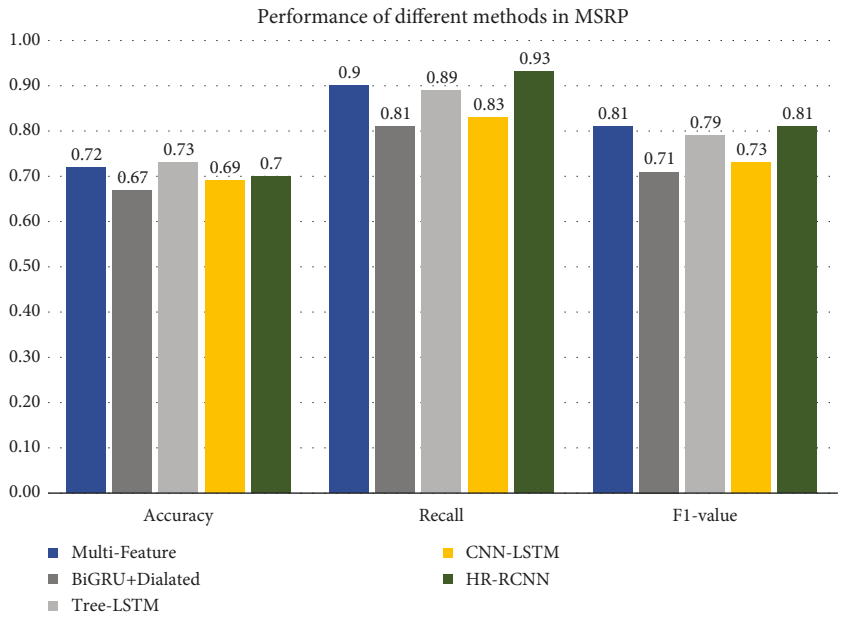

Figure 6: Performance of different methods on the MSRP dataset.

of sentence structure and syntactic information. Compared with the CNN-LSTM model, the Tree-LSTM and BIGRU + Dialated control word vectors through operations such as syntax analysis, taking into account the influence of sentence structure on text similarity evaluation. The MMNF model extracts text features by combining different machine learning algorithms and a CNN network, maximizing the performance of the model through a continuous adjustment of the weights. Its accuracy is the highest in several comparative experiments. The recall rate and $F1$-value of the HA-RCNN model are better than those of the other models, but its accuracy is worse than that of the MMNF model. The comparison of the $F1$-values shows that the HA-RCNN model achieves excellent results on the ATEC dataset.

On the MSRP dataset, the performance of the BiGRU + Dialated and CNN-LSTM models was poor. The Multi-Feature model performed well due to its combination of multiple parameters, aided by the use of word vectors. The recall rate of HA-RCNN model was again the highest, and its $F1$-value was also excellent.

It can be seen that the HA-RCNN model lags behind the other two models in accuracy. By observing the confusion matrix of the experiment, we see that the number of TP samples is small, which leads to the low accuracy of the

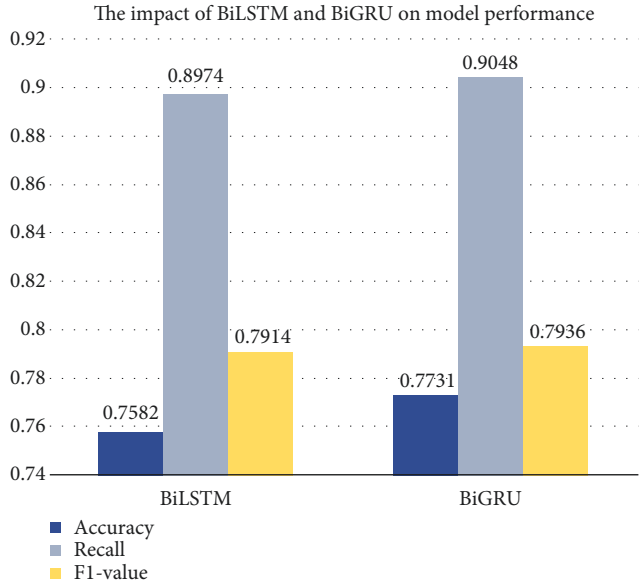

Figure 7: The impact of BiLSTM and BiGRU on model performance.

Table 3: Effects of different attention mechanisms on model performance.

Methods	Accuracy	Recall	$F1$-value
SA [29]	0.7522	0.8631	0.7737
CSA [30]	0.7582	0.8656	0.7795
S2SA [31]	0.7685	0.8919	0.7912
HA	0.7731	0.9048	0.7936

model. We believe that there are two reasons for this phenomenon. First, the sample size of the training and test sets provided by the MSRP dataset is small, and the proportion of positive and negative samples in the dataset is unbalanced. This makes it difficult to achieve accurate classification of positive sample data. Another reason is that the model in this paper is more complex and it is difficult to train the desired effect in small datasets.

We replaced the BiGRU in the model with a BiLSTM and observed the change in model performance on the ATEC dataset, as shown in Figure 7.

It is evident that the model performance is hardly affected by the change, but the model is trained faster.

In order to verify the effectiveness of the proposed method, we introduce the attention mechanism proposed by other scholars into the model of this paper for comparison. The results are shown in Table 3.

These three attention mechanisms allow the model to learn to determine the importance of each word by increasing the weight of important words and reducing the weight of unimportant words. To avoid interfering with weight assignment, no lexical rules were introduced.

To test whether the proposed attention mechanism can capture important information in sentences, we randomly selected a pair of sentences from the dataset and observed how they affected the output of the mechanism. The results are shown in Figure 8.

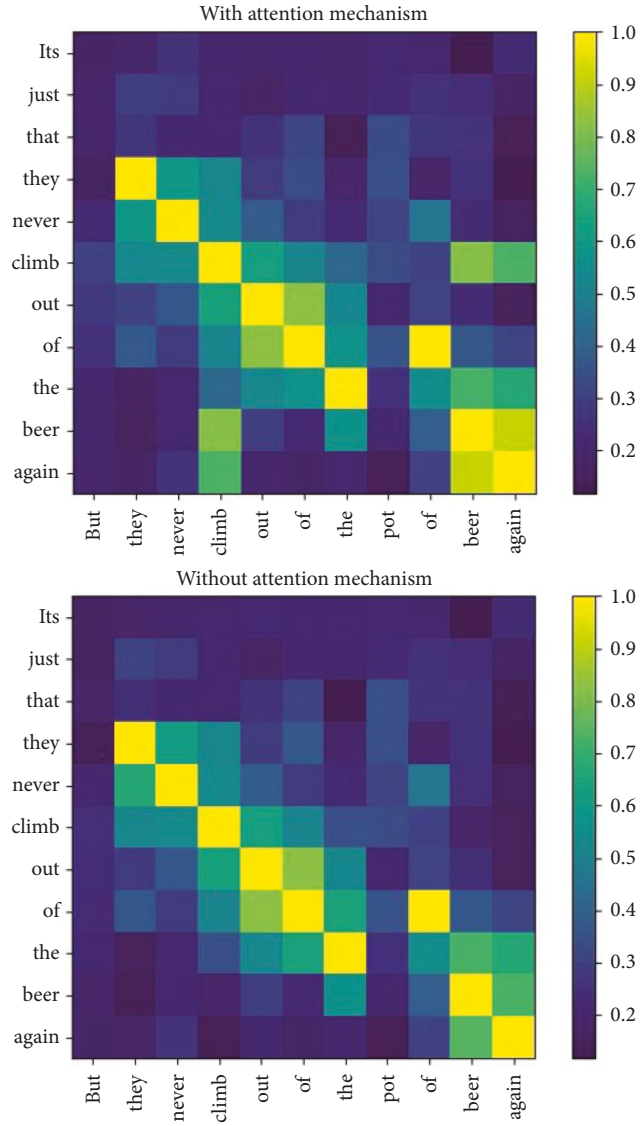

Figure 8: Attention visualization for a pair of sentences.

It can be seen that after the attention mechanism, the semantic expression of the important elements in the sentence is enhanced, as is the correlation between them and other important elements. However, non-important elements of the sentence have little effect.

4. Conclusion

In this paper, we proposed a model based on a siamese network, integrating semantic information and a headword attention mechanism to learn sentence representations. Our model obtains a semantically enhanced representation through the headword attention mechanism, which increases the influence of key information in the sentence. In order to verify the performance of the model, we conducted experiments on the ATEC and MSRP datasets. Compared with other models, our model achieved relatively excellent performance in the recall and $F1$ metrics.

In future work, we will study the impact of multi-level attention mechanisms on model performance, and incorporate external knowledge into our model.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The Fundamental Research Funds for Central Universities (Grant no. 2572015CB32) and National Natural Science Foundation of China (Grant no. 61901103).

References

[1] N. H. Tien, N. M. Le, Y. Tomohiro, and I. Tatsuya, "Sentence modeling via multiple word embeddings and multi-level comparison for semantic textual similarity," *Information Processing & Management*, vol. 56, no. 6, p. 102090, 2019.

[2] I. Arroyo-Fernández, C.-F. Méndez-Cruz, G. Sierra, J.-M. Torres-Moreno, and G. Sidorov, "Unsupervised sentence representations as word information series: revisiting TF-IDF," *Computer Speech & Language*, vol. 56, pp. 107–129, 2019.

[3] M. Murata, H. Nagano, R. Mukai, K. Kashino, and S. Satoh, "BM25 with exponential IDF for instance search," *IEEE Transactions on Multimedia*, vol. 16, no. 6, pp. 1690–1699, 2014.

[4] C. Yadav and A. Sharan, "A new LSA and entropy-based approach for automatic text document summarization," *International Journal on Semantic Web and Information Systems*, vol. 14, no. 4, pp. 1–32, 2018.

[5] G. Chen, X. Shi, M. Chen, and L. Zhou, "Text similarity semantic calculation based on deep reinforcement learning," *International Journal of Security and Networks*, vol. 15, no. 1, pp. 59–66, 2020.

[6] B. Hu, Z. Lu, H. Li, and Q. Chen, *Convolutional neural network architectures for matching natural language sentences Advances in Neural Information Processing Systems*, Google, Montreal, Canada, 2015.

[7] M. Sundermeyer, R. Schlüter, and H. Ney, "LSTM neural networks for language modeling," in *Proceedings of the 13th Annual Conference of the International Speech Communication Association 2012*, Portland, United states, September 2012.

[8] W. Zhu, T. Yao, J. Ni, B. Wei, and Z. Lu, "Dependency-based Siamese long short-term memory network for learning sentence representations," *PLoS One*, vol. 13, no. 3, Article ID e0193919, 2018.

[9] S. Peng, H. Cui, N. Xie, S. Li, J. Zhang, and X. Li, "LEnhanced-RCNN: An Efficient Method for Learning Sentence similarity," in *Proceedings of the Web Conference 2020 - Proceedings of the World Wide Web Conference*, Chunghwa Telecom, Taipei, Taiwan, April 2020.

[10] W. Bao, W. Bao, J. Du, Y. Yang, and X. Zhao, "Attentive siamese LSTM network for semantic textual similarity measure," in *Proceedings of the 2018 International Conference on Asian Language Processing (IALP)*, Bandung, Indonesia, November 2018.

[11] C. H. Shih, B. C. Yan, S. H. Liu, and B. Chen, "Investigating siamese lstm networks for text categorization," in *Proceedings of the network 9th Asia-Pacific Signal and Information Processing Association Annual Summit and Conference(APSIPA ASC)*, DOLBY, Kuala Lumpur, Malaysia, December 2017.

[12] M. Nicosia and A. Moschitti, "Accurate sentence matching with hybrid siamese networks," in *Proceedings of the network 26th ACM International Conference on Information and Knowledge Management*, ACM SIGWEB, Singapore, November 2017.

[13] T. Ranasinghe, C. Orăsan, and R. Mitkov, "Semantic textual similarity with siamese neural networks," in *Proceedings of the neuraInternational Conference Recent Advances in Natural Language Processing(RANLP)*, Varna, Bulgaria, September 2019.

[14] Q. Chen, X. Zhu, Z. Ling, S. Wei, H. Jiang, and D. Inkpen, "Enhanced Lstm for Natural Language Inference,"for natural: 1609.06038," 2016, https://arxiv.org/abs/1609.06038.

[15] Z. Wang, W. Hamza, and R. Florian, "Bilateral Multi-Perspective Matching for Natural Language Sentences," 2017, https://arxiv.org/abs/1702.03814.

[16] Q. T. Ho, N. Q. K. Le, and Y. Y. Ou, "Fad-Bert: Improved prediction of FAD binding sites using pre-training of deep bidirectional transformers," *Computers in Biology and Medicine*, vol. 131, Article ID 104258, 2021.

[17] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, and Q. V. Le, "nXlnet: Generalized Autoregressive Pretraining for Language Understanding," *For Advances in Neural Information Processing Systems*, Citadel, Vancouver Canada, 2019.

[18] D. Bahdanau, K. Cho, and Y. Bengio, "gNeural Machine Translation by Jointly Learning to Align and Translate," 2014, https://arxiv.org/abs/1409.0473.

[19] J. Cheng, L. Dong, and M. Lapata, "Long Short-Term Memory-Networks for Machine reading," 2016, https://arxiv.org/abs/1601.06733.

[20] X. He, Z. He, J. Song, Z. Liu, Y.-G. Jiang, and T.-S. Chua, "Nais: neural attentive item similarity model for recommendation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 12, pp. 2354–2366, 2018.

[21] Z.-P. Shan, Y.-Q. Lei, D.-F. Zhang, and J. Zhou, "NASM: nonlinearly attentive similarity model for recommendation system via locally attentive embedding," *IEEE Access*, vol. 7, pp. 70689–70700, 2019.

[22] M. Tan, C. Dos Santos, B. Xiang, and B. Zhou, "Improved Representation Learning for Question Answer Matching," in *Proceedingsof the mprov54th Annual Meeting of the Association for Computational Linguistics*, Amazon, Berlin, Germany, August 2016.

[23] W. Che, Y. Shao, T. Liu, and Y. Ding, "Semeval-2016 task 9: Chinese semantic dependency parsing," in *Proceedings of the SemEval 2016 - 10th International Workshop on Semantic Evaluation, Proceedings, ACL Special Interest Group on the Lexicon (SIGLEX)*, San Diego, CA, USA, June 2016.

[24] P. Zhang, X. Huang, Y. Wang, C. Jiang, S. He, and H. Wang, "Semantic similarity computing model based on multi model fine-grained nonlinear fusion," *IEEE Access*, vol. 9, pp. 8433–8443, 2021.

[25] M. Ji, C. Wang, and G. Liu, "Measurement of sentence similarity based on constituency parsing and dilated convolution," *International Journal of Computer Applications in Technology*, vol. 64, no. 3, pp. 252–259, 2020.

[26] K. S. Tai, R. Socher, and C. D. Manning, "Improved Semantic Representations from Tree-Structured Long Short-Term Memory Networks," 2015, https://arxiv.org/abs/1503.00075.
[27] E. L. Pontes, S. Huet, A. C. Linhares, and J. M. Torres-Moreno, "Predicting the Semantic Textual Similarity with Siamese CNN and LSTM," 2018, https://arxiv.org/abs/1810.10641.
[28] M. Farouk, "Measuring text similarity based on structure and word embedding," *Cognitive Systems Research*, vol. 63, pp. 1–10, 2020.
[29] A. Fadel, I. Tuffaha, and M. Al-Ayyoub, "Tha3aroon at NSURL-2019 Task 8: Semantic Question Similarity in Arabic," 2019, https://arxiv.org/abs/1912.12514.
[30] S. Zheng, F. Chen, and X. Wang, "Semantic matching for short texts: a cross attention mechanism," *Journal of Physics: Conference Series*, vol. 1757, 2021.
[31] Z. Li, H. Chen, and H. Chen, "Biomedical text similarity evaluation using attention mechanism and siamese neural network," *IEEE Access*, vol. 9, pp. 105002–105011, 2021.