Proceedings of Machine Learning Research vol 278:1–11, 2025 2025 2nd International Conference on Machine Learning and Intelligent Computing

Research on Chinese Text Similarity by Fusing Deep and Shallow Features

Chengfang Lu* 2023710700@YNAGTZEU.EDU.CN
School of Computer Science, Yangtze University, Jingzhou, Hubei Province 434000, China

Gang Li 2023710695@YNAGTZEU.EDU.CN
School of Computer Science, Yangtze University, Jingzhou, Hubei Province 434000, China

Linjie Hou 2024720774@YNAGTZEU.EDU.CN
School of Computer Science, Yangtze University, Jingzhou, Hubei Province 434000, China
Corresponding author

Editors: Nianyin Zeng, Ram Bilas Pachori and Dongshu Wang

Abstract

Existing Chinese text similarity calculation methods typically focus on a single dimension, resulting in insufficient information integration and difficulty in comprehensively merging semantic, feature, and structural information. To address this issue, a Chinese text similarity calculation model that integrates deep and shallow similarities has been proposed. The model first utilizes a Siamese neural network to obtain dynamic vector representations of the texts, further extracting features and calculating deep semantic similarity. Next, based on traditional edit distance algorithms, an improved component-weighted edit distance algorithm is designed by introducing tokenization and assigning weights to different parts of speech, to more accurately reflect the lexical-level shallow features and structural information of the texts. Finally, by linearly weighting and fusing deep semantic similarity with shallow feature similarity, a more comprehensive text similarity evaluation is achieved. Experimental results show that in experiments based on Chinese STS-B and Chinese SICK datasets, the Spearman correlation coefficients improved by 4.34 and 3.76, respectively, compared to the baseline model Siamese-RoBERTa. This model effectively enhances the performance of Chinese short text similarity calculation and better aligns with the expression habits of Chinese texts.

Keywords: Similarity Model; Feature Fusion; Semantic Representation; Character-level Matching; Improved Edit Distance; Analytic Hierarchy Process (AHP)

1. Introduction

In the field of Natural Language Processing, the calculation of Chinese text similarity plays an important role in tasks such as information retrieval (Zeng, 2020), machine translation (Lv et al., 2025), question answering systems (Pu et al., 2020), and scoring of subjective questions (Shen, 2021). The calculation of text similarity aims to evaluate the similarity or relevance between different texts. Although this technology has been widely applied in practical applications, it still faces many challenges.

Early research in text similarity primarily relied on traditional machine learning models. One widely used approach was Term Frequency-Inverse Document Frequency (TF-IDF) (Arroyo Fernández et al., 2019), which represents text as numerical vectors based on word frequency statistics and measures similarity using cosine distance. However, TF-IDF is limited to surface-level lexical patterns and fails to capture contextual semantics. Another classical method, Latent Semantic Analysis

(LSA) (Yadav and Sharan, 2018), addresses this issue by applying singular value decomposition (SVD) to uncover latent semantic structures in text. While LSA improves upon TF-IDF, it still struggles with synonymy and polysemy.

With the advancement of deep learning techniques, text similarity computation has entered a new era. Deep learning, owing to its automatic feature extraction capability and strong robustness, has been widely adopted for semantic similarity tasks. For instance, Kleenankandy and Nazeer (2021) utilized RNNs to learn dependency relationships between words for sentence similarity assessment. Meanwhile, Rakhlin (2016) proposed the Text-CNN model, which applies convolutional and pooling layers to capture local textual features. However, these methods often struggled with long-sequence inputs due to vanishing or exploding gradient problems. To address this limitation, Long Short-Term Memory (LSTM) networks were introduced, incorporating gating mechanisms to effectively model long-range dependencies. Sundermeyer et al. (2012) demonstrated that LSTMs could better capture contextual information in sequential text data. Further improvements were achieved with Bidirectional LSTMs (BiLSTMs), which process input sequences in both forward and backward directions, leveraging richer contextual representations and significantly enhancing model performance.

In recent years, Transformer-based models have revolutionized NLP with their superior semantic modeling capabilities. A landmark development is BERT (Bidirectional Encoder Representations from Transformers) (Devlin et al., 2018), which employs a bidirectional Transformer encoder with multi-layer feature extraction and relative position encoding to effectively capture contextual semantics. Another significant advancement comes from Siamese network architectures, which process paired inputs through identical subnetworks to compute similarity scores, demonstrating both computational efficiency and strong generalization. Notable implementations include BiLSTM-based Siamese networks for semantic comparison (Zhu et al., 2018), Sentence-BERT (SBERT) for enhanced semantic matching (Reimers and Gurevych, 2019) and attention-augmented BiGRU variants for improved context modeling (Chen et al., 2022). Zhang and Li (2024) propose SRoberta-SelfAtt, a RoBERTa-based Siamese model with self-attention for text similarity, achieving superior performance on three benchmark datasets. Zu et al. (2024) proposed a method for measuring semantic similarity in Chinese short texts using cosine similarity, enhanced by part-of-speech weighting and word order adjustments, which demonstrated strong performance.

Existing text similarity methods often rely on single features or neglect Chinese linguistic characteristics. To address these limitations, this study proposes a hybrid model combining deep semantic features (via Siamese neural networks) with enhanced shallow lexical features (via weighted edit distance adapted for Chinese syntax). This approach improves accuracy while better capturing Chinese-specific patterns.

2. Construction of the Similarity Fusion Model

The proposed model first processes text pairs through a Siamese Network (RoBERTa+BiGRU) to compute deep semantic similarity via cosine similarity. Simultaneously, Stanford NLP performs tokenization and POS tagging, with AHP determining word weights for shallow lexical similarity calculation. The final similarity score combines both metrics through linear weighting. The overall workflow of the model is illustrated in Figure 1.

The comprehensive similarity score is calculated as shown in Equation (1).

$$CompSim = \alpha \cdot Sim_{\cos ine} + (1 - \alpha) \cdot Sim_{edit} \tag{1}$$

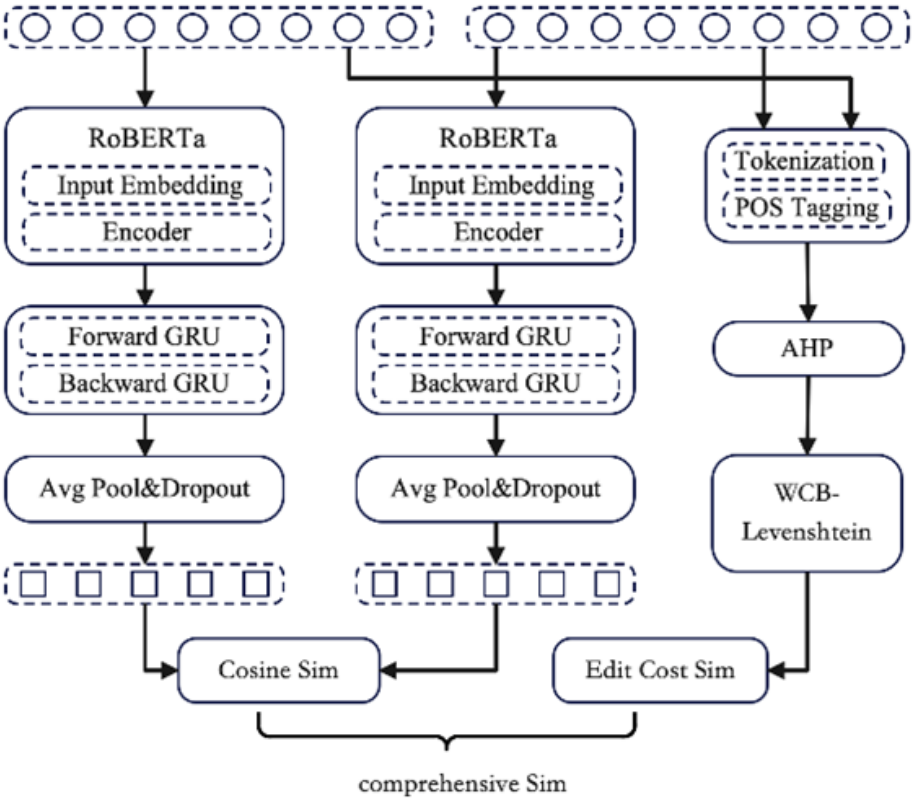

Figure 1: Similarity Fusion Model

In the equation, $Sim_{\cos ine}$ represents the semantic similarity of the model, Sim_{edit} denotes the edit cost similarity of the model, α is the similarity weighting factor, and $CompSim$ represents the final comprehensive similarity score to be calculated.

2.1. Deep-level Semantic Similarity Model

The model employs a Siamese network to map variable-length texts into a fixed-dimensional vector space, ensuring semantically similar texts are close and dissimilar ones are distant. A schematic diagram of the Siamese network structure is illustrated in Figure 2.

In the word vector acquisition stage, this paper utilizes the RoBERTa model (Wang et al., 2025) to convert input texts into dynamic word vector representations. As an improved version of BERT, RoBERTa enhances model performance and robustness by removing the next sentence prediction (NSP) task, employing dynamic masking strategies, and increasing training data volume and batch size. In terms of model architecture, its structure remains essentially identical to BERT, as shown in Figure 3.

The processed word vectors are obtained by concatenating the first and last layer outputs of RoBERTa, preserving both local details and global semantic information. These word vectors are then fed into a BiGRU model for further processing. The BiGRU extends the standard GRU (Zhang et al., 2020) into a bidirectional architecture, consisting of forward and backward GRUs, which enables simultaneous capture of contextual information from both directions. The computational

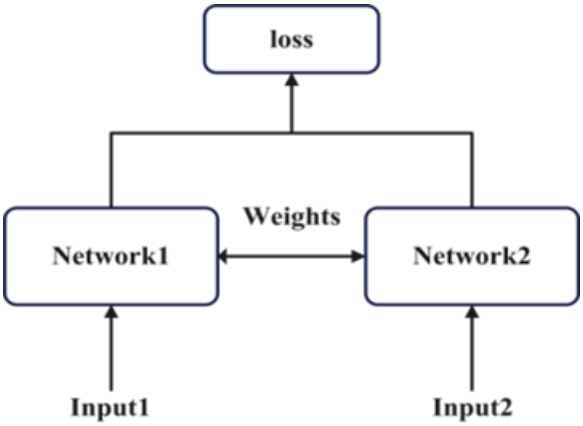

Figure 2: Twin network structure

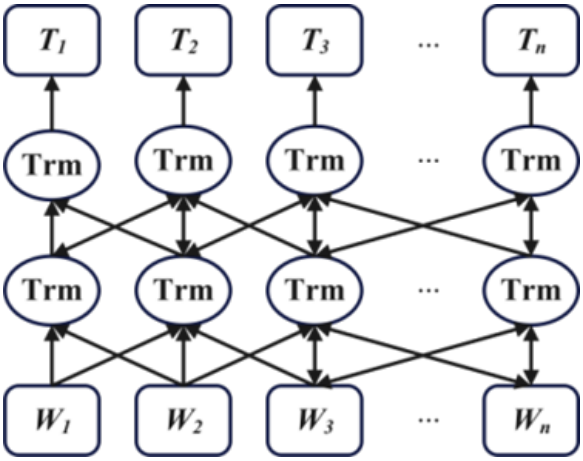

Figure 3: RoBERTa Model Structure

method of BiGRU is shown in Equations (2)-(4).

$$\overrightarrow{h_t} = GRU(x_t, \overrightarrow{h_{t-1}}) \tag{2}$$

$$\overleftarrow{h_t} = GRU(x_t, \overleftarrow{h_{t+1}}) \tag{3}$$

$$h_t = [\overrightarrow{h_t} + \overleftarrow{h_t}] \tag{4}$$

In the equations, $\overrightarrow{h_t}$ and $\overleftarrow{h_t}$ represent the hidden states of the forward GRU and backward GRU at time step t, respectively, while h_t denotes the bidirectional hidden state formed by concatenating the forward and backward hidden states at time step t.

Finally, this paper employs cosine similarity to measure the similarity between two text vectors. Cosine similarity primarily quantifies the angular similarity between vectors in the embedding space, focusing on evaluating directional alignment while disregarding their magnitudes or lengths.

The calculation method is shown in Equation (5).

$$Sim_{\cos ine} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}} \tag{5}$$

In the equations, A and B represent the embedding vectors of two sentences used to calculate their similarity score, which ranges from -1 to 1, where 1 indicates identical vectors, -1 denotes completely opposite vectors, and 0 signifies no similarity.

During model training, we use Mean Squared Error (MSE) as the loss function. The MSE loss function calculates the squared difference between predicted and true values, making it sensitive to large errors while maintaining smoothness and convexity. The calculation method is shown in Equation (6).

$$MSE = \frac{1}{n}\sum_{i=1}^{n} (y_i - \widehat{y_i})^2 \tag{6}$$

In the equation, n represents the number of samples, y_i denotes the true value of the i-th sample, and $\widehat{y_i}$ represents the predicted value of the i-th sample.

2.2. Shallow Lexical Similarity Model

s1 = "他跑向高高的山坡" (He runs toward the tall hillside)

s1 = "他走向矮矮的丘陵" (He walks toward the low hills)

In traditional edit distance algorithms, the difference between two sentences is measured through character-level operations, such as substituting "跑" (run) with "走" (walk) or "高" (tall) with "矮" (short). Although this method can compute edit distance, it fragments words, increases computational cost, and fails to align with Chinese linguistic habits. Moreover, it overlooks the critical role of verbs and nouns in sentences.

The improved edit distance algorithm assigns different weights to these elements. For instance, verbs like "跑" (run) and "走" (walk), as well as nouns like "山坡" (hillside) and "丘陵" (hills), are assigned higher weights due to their semantic importance, whereas adjectives receive lower weights. Additionally, adjective phrases such as "高高的" (very tall) and "矮矮的" (very short) are treated as single units for substitution, reducing computational overhead and improving logical consistency. This approach better reflects authentic Chinese expression patterns, thereby enhancing the accuracy of Chinese similarity task computation.

2.2.1. ANALYTIC HIERARCHY PROCESS

The part-of-speech weights are determined using AHP, a systematic multi-criteria decision-making method that quantifies the relative importance of linguistic components. This approach enables scientific weight allocation through pairwise comparisons and hierarchical analysis, ensuring optimal weighting for semantic processing tasks.The specific calculation procedure is as follows:

(1) Establish a hierarchical structure model. For Chinese parts of speech, the model is constructed based on key categories (nouns, verbs, adjectives, and adverbs) versus other categories (conjunctions, prepositions, etc.).

(2) The judgment matrix is constructed using AHP methodology, assigning higher weights to nouns and verbs (core meaning), moderate weights to adjectives and adverbs (modifiers), and lower weights to other parts of speech (minor impact). Proportional scales and their reciprocals define pairwise comparisons, as shown in Table 1.

Table 1: Judgment Matrix

POS	Noun	Verb	Adjective	Adverb	Others
Noun	1	1	3	3	5
Verb	1	1	3	3	5
Adjective	1/3	1/3	1	1	3
Adverb	1/3	1/3	1	1	3
Others	1/5	1/5	1/3	1/3	1

The judgment matrix shown in Table 1 can be formally expressed as a mathematical matrix A:

$$A = \begin{bmatrix} 1 & 1 & 3 & 3 & 5 \\ 1 & 1 & 3 & 3 & 5 \\ \frac{1}{3} & \frac{1}{3} & 1 & 1 & 3 \\ \frac{1}{3} & \frac{1}{3} & 1 & 1 & 3 \\ \frac{1}{5} & \frac{1}{5} & \frac{1}{3} & \frac{1}{3} & 1 \end{bmatrix}$$

Matrix A defines the relative importance of POS categories, where A_{ij} indicates how much more important category i is than j. For example, $A_{13} = 3$ means category 1 is 3× as important as category 3.

(3) Consistency verification was performed through eigenvalue analysis of matrix A, yielding a maximum eigenvalue ($\lambda \max$) of 5 with its corresponding eigenvector P $= [1, 1, 3, 3, 5]^{\mathrm{T}}$ (where $n = 5$ denotes the eigenvector dimension). The resulting Consistency Index (CI) calculated as $(\lambda \max - \mathrm{n})/(\mathrm{n} - 1)$ equals 0, indicating perfect consistency in the judgment matrix. By referencing the standard Random Index (RI) table, the corresponding RI value for $n = 5$ was identified as 1.12. Consequently, the Consistency Ratio ($CR = CI/RI$) was computed to be 0, which satisfies the acceptable threshold ($CR < 0.1$) and confirms that judgment matrix A successfully passes the consistency test.

2.2.2. COMPONENT-WEIGHTED EDIT COST ALGORITHM

After obtaining the part-of-speech weights W via the AHP, we apply them to an improved edit cost algorithm. Given input texts s_1 and s_2, we first perform word segmentation and part-of-speech tagging, yielding tokenized results t_1, t_2 and POS-tagged sequences p_1, p_2. The algorithm proceeds as follows:

(1) The AHP-derived part-of-speech weights prioritize meaning-critical elements: nouns and verbs ($W = 0.34$ each) receive highest weights, modifiers like adjectives/adverbs ($W = 0.13$ each) get moderate weights, while other components default to $W = 0.06$, reflecting their relative semantic importance in Chinese sentence structure.

(2) The edit cost matrix ec of size $(n + 1) \times (m + 1)$ is initialized with boundary conditions: $ec[i][0] =$ i (deleting i tokens from t_1) and $ec[0][j] =$ j (inserting j tokens to form t_2), establishing

baseline transformation costs for dynamic programming computation between token sequences t_1 and t_2.

(2) For each token in t_1 and t_2, calculate the costs of three operations: deletion cost as $1 + W(p1[i-1])$, denoted as del_cost; insertion cost as $1 + W(p2[j-1])$, denoted as ins_cost; replacement cost is 0 if the tokens are identical, otherwise $1 + W(p1[i-1]) + W(p2[j-1])$, denoted as rep_cost. The base cost of 1 represents the original edit distance cost. The replacement cost incorporates both deletion and insertion costs to more accurately reflect the comprehensive edit expense.

(4) The edit cost matrix ec is updated by selecting the minimum operation cost at each position. Specifically, each element $ec[i][j]$ is computed as the minimum of: the deletion cost ($ec[i-1][j] + del\_cost$), the insertion cost ($ec[i][j-1] + ins\_cost$), and the replacement cost ($ec[i-1][j-1] + rep\_cost$). The final value $ec[n][m]$ represents the minimum weighted edit cost required to transform text s_1 into text s_2.

(5) To compute the text similarity score, we first determine the worst-case edit cost based on the token sequence lengths and maximum part-of-speech weight (max_weight). The similarity score is then derived by comparing the actual weighted edit cost ($ec[n][m]$) against this maximum possible cost.

The edit-cost-based similarity calculation method is formally specified in Equation (7).

$$Sim_{edit} = 1 - \frac{ec[n][m]}{(len(t1) + len(t2)) \cdot (1 + \max\_weight)} \tag{7}$$

In the equation, $len(t1)$ and $len(t2)$ represent the token sequence lengths of text 1 and text 2 respectively, while max_weight denotes the maximum value among all part-of-speech weights. sim_{edit} represents the edit-cost-based similarity score.

3. Experimental Analysis and Results

3.1. Experimental Environment and Parameters

Experiments were conducted on Google Colab using an NVIDIA A100 GPU (40GB) with PyTorch 2.3.1 and Python 3.10. We employed the chinese-roberta-wwm-ext-large model (dropout=0.1) with MSE loss and AdamW optimizer (lr=2e-5). Text processing used StanfordNLP for segmentation/POS tagging, with model parameters set to max_length=128, batch_size=32, and 10 training epochs.

3.2. Experimental Datasets and Evaluation Metrics

To validate the scientific rigor, effectiveness, and rationality of our similarity computation model, we employed two publicly available datasets: the STS-B and SICK dataset, both of which were translated into Chinese versions. Since our task primarily addresses a regression problem, we adopted the Spearman correlation coefficient as the sole evaluation metric to assess the prediction accuracy, given its robustness in measuring monotonic relationships for rank-based similarity assessment."

3.3. Results Analysis

3.3.1. EXPERIMENTS ON DEEP SEMANTIC SIMILARITY MODELS

For the deep semantic similarity model experiments, This paper evaluates three Chinese pre-trained models—chinese-roberta-wwm-ext-large, bert-base-chinese, and albert-base-v2—on two datasets.

Different word vector strategies were tested: Last (final layer), ConcatLast4 (last four layers), and ConcatFirstLast (first + last layer). Results are presented in Tables 2 and 3.

Table 2: Comparative Experiment of Pretrained Models on the STS-B Dataset

Pretrained Models	Last	ConcatLast4	ConcatFirstLast
bert-base-chinese	74.04	74.93	76.12
albert-base-v2	69.15	70.91	71.79
chinese-roberta-wwm	77.44	79.03	**80.84**

Table 3: Comparative Experiment of Pretrained Models on the SICK Dataset

Pretrained Models	Last	ConcatLast4	ConcatFirstLast
Wbert-base-chinese	70.16	71.65	73.14
albert-base-v2	61.21	63.05	65.76
chinese-roberta-wwm	73.45	76.12	**78.58**

Compared to other pretrained language models and word embedding methods, the Chinese-RoBERTa-wwm-ext-large model combined with ConcatFirstLast representation achieved the best performance in experiments, benefiting from RoBERTa's significant improvements over traditional BERT.

3.3.2. VOCABULARY-LEVEL SHALLOW SIMILARITY MODEL EXPERIMENT

For the shallow feature similarity model, this study designed comparative experiments to validate the rationality and effectiveness of incorporating Chinese part-of-speech (POS) information with weighted assignments into the edit distance algorithm for text edit cost calculation. The experiments compared the performance of the original and enhanced edit cost algorithms, with the results presented in Table 4.

Table 4: Comparison of Edit Cost Algorithms Before and After Improvement

Dataset	SRoBERTa+Levenshtein	SRoBERTa+WCB-Levenshtein
STS-B	82.40	**85.18**
SICK	79.36	**82.34**

The results show that incorporating Chinese linguistic features—including word segmentation and POS-based weighting—into edit distance computation achieves notably better text similarity performance than conventional approaches.

3.3.3. LINEAR WEIGHTING FOR DETERMINING WEIGHT FACTORS

To evaluate the relative contributions of deep semantic similarity and shallow feature similarity, this study employed a linear weighting method to adjust and compare the weights assigned to these two feature types. By systematically varying the combinations of weighting factors, we identified optimal weight allocations and observed their differential impacts on model performance. The experimental results are presented in Figure 4.

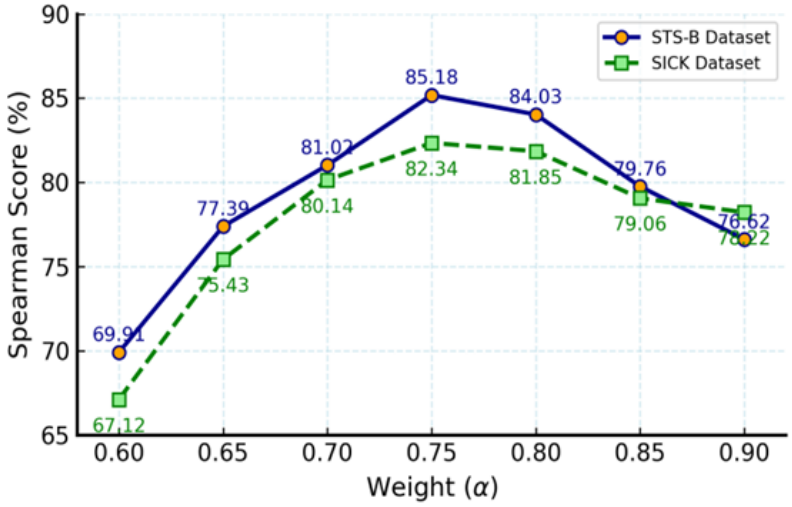

Figure 4: Determination of Weighting Factors

Considering the crucial role of semantic features in Chinese text processing, this study initially set a 6:4 weighting ratio for semantic vs. shallow features. As Figure 4 shows, performance improved with higher semantic weight (α), but declined beyond α=0.8, suggesting excessive focus on semantics harms shallow feature utility. The optimal ratio was finalized at 0.75:0.25.

3.3.4. COMPARATIVE EXPERIMENTS WITH OTHER MODELS

We also designs comparative experiments between models, comparing the method proposed in this paper with existing methods in other literatures to verify the effectiveness of the proposed fusion model in similarity calculation. The experimental results are shown in Table 5.

Table 5: Comparative Experiment Results with Other Models

Model Method	STS-B	SICK
LSTM	68.58	64.23
S-DLSTM	72.06	69.54
S-BERT	79.10	76.64
S-CNN+BiGRU	77.12	75.05
S-BiGRU-Atten	78.24	75.76
S-BERT+BiLSTM	82.76	78.12
Ours	**85.18**	**82.34**

As shown in Table 5, the basic LSTM model (Zhu et al., 2018) demonstrates limited capability in capturing contextual information. Siamese network architectures (e.g., S-DLSTM (Zhu et al., 2018)) show significant improvements for sentence-pair tasks, with Siamese-BERT models (e.g., S-BERT (Reimers and Gurevych, 2019)) achieving even better performance. While existing approaches combine different feature extractors (Zu et al., 2024) or augment BERT with LSTM/GRU

to enhance feature representation, they primarily focus on either semantic or surface-level features without fully considering Chinese linguistic characteristics.

4. Conclusion

This paper proposes a similarity fusion-based method for Chinese text similarity calculation, integrating deep semantic similarity and lexical-level surface similarity through linear weighted analysis. To better adapt to Chinese linguistic characteristics, we improved traditional edit distance by incorporating part-of-speech weights for edit cost computation, effectively capturing textual logic and expression patterns. Experiments demonstrate the model's superior performance over conventional approaches.

The model shows promising results on STS-B and SICK datasets, though performance may vary in specialized domains like legal/medical texts due to unique terminology. Future applications include plagiarism detection (identifying paraphrased content), recommendation systems (semantic matching), and legal analysis (detecting similar clauses), demonstrating its capability for Chinese semantic and structural similarity analysis.

References

I. Arroyo Fernández, C. Méndez Cruz, G. Sierra, and et al. Unsupervised sentence representations as word information series: Revisiting tf - idf. *Computer Speech & Language*, 56:107–129, 2019. doi: 10.1016/j.csl.2019.01.005.

X. Chen, Z. Qiu, and et al. Semantic similarity analysis based on siamese - bigru - attention. *Journal of Dalian jiaotong University*, 43(04):113–116, 2022. doi: 10.13291/j.cnki.djdxac.2022.04.021.

J. Devlin, M. W. Chang, and et al. Lee, K and. Bert: Pre - training of deep bidirectional transformers for language understanding. *GitHub*, 2018. doi: 10.18653/v1/N19-1423.

J. Kleenankandy and K. A. A. Nazeer. Recognizing semantic relation in sentence pairs using tree - rnns and typed dependencies. In *2020 6th IEEE Congress on Information Science and Technology (CiSt)*, pages 372–377, 2021. doi: 10.1109/CiSt49399.2021.9357187.

X. Lv, J. Li, S. Tao, and et al. Survey on document-level neural machine translation. *Journal of Software*, 36(01):152–183, 2025. doi: 10.13328/j.cnki.jos.007217.

W. Pu, H. Wang, and et al. Overview of nlp - based question answering systems. *Technology Innovation and Application*, 11(22):77–79, 2020. doi: 10.19981/j.cn23-1581/g3.2021.22.025.

A. Rakhlin. Convolutional neural networks for sentence classification. *GitHub*, 6:25, 2016.

N. Reimers and I. Gurevych. Sentence - bert: Sentence embeddings using siamese bert - networks. *CoRR*, 2019. doi: 10.48550/arXiv.1908.10084.

Z. Shen. *Research on Automatic Scoring of Subjective Questions Based on Natural Language Processing*. PhD thesis, Jiangsu University of Science and Technology, 2021.

M. Sundermeyer, R. Schlüter, and H. Ney. Lstm neural networks for language modeling. In *Interspeech*, pages 194–197, 2012. doi: 10.1016/0165-6074(89)90269-X.

Y. O. Wang, Y. C. Yuan, Z. X. He, and et al. A method for relationship extraction using improved roberta, multi - instance learning, and dual attention mechanism. *Journal of Shandong University (Engineering Science)*, (02):78–87, 2025.

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 (IJSWIS)*, 14(4): 1–32, 2018. doi: 10.4018/IJSWIS.2018100101.

Z. Zeng. Analysis of natural language processing and information retrieval system. *Digital Technology & Application*, (06):41–42, 2020. doi: 10.19695/j.cnki.cn12-1369.2020.06.18.

Biao Zhang, Deyi Xiong, Jun Xie, and Jinsong Su. Neural machine translation with gru - gated attention model. *IEEE Transactions on Neural Networks and Learning Systems*, 31(11):4688–4698, 2020. doi: 10.1109/TNNLS.2019.2957276.

X. Y. Zhang and W. Li. Research on calculation of semantic similarity of chinese short text based on roberta. *Computer Applications and Software*, 41(08):275–281 + 366, 2024. doi: 10.3969/j.issn.1000-386x.2024.08.040.

Wenhao Zhu, Tengjun Yao, Jianyue Ni, Baogang Wei, and Zhiguo Lu. Dependency - based siamese long short - term memory network for learning sentence representations. *PloS One*, 13(3): e0193919, 2018. doi: 10.1371/journal.pone.0193919.

Y. F. Zu, H. F. Ling, R. Z. Tang, and et al. Computing method of chinese short text similarity based on part of speech, semantic and word order factors. *Computer & Digital Engineering*, 52(08): 2420–2424 + 2468, 2024. doi: 10.3969/j.issn.1672-9722.2024.08.030.