Use BiLSTM with Attention Mechanism to Optimize the Accuracy of Word Meaning Correspondence in Technical Texts

Lanxi Gao[1], Tao Dong[2], Minghan Yang[3]
[1]Institute of Foreign languages Teaching and Researching, Liaodong University, Dandong 118003, Liaoning, China
[2]Information Engineering School, Liaodong University, Dandong 118003, Liaoning, China
[3]College of Art and Media, Shenyang Institute of Technology, Shenyang 113122, Liaoning, China

Corresponding author: Lanxi Gao (e-mail: 13470066016@163.com)

ABSTRACT In highly specialized and terminology-dense scientific and technical texts, existing word sense disambiguation (WSD) models struggle to adequately model the contextual semantic dependencies of polysemous words, especially in engineering domains where the same term may carry different technical meanings across contexts. To address this issue, this paper proposes a robust WSD model that integrates a bidirectional long shortterm memory network (BiLSTM) with an attention mechanism, specifically designed for Chinese patent texts. First, a two-layer BiLSTM is used for bidirectional context modeling to capture long-range dependencies. Then, a multi-head attention mechanism uses dynamic weighting to highlight key semantic components, generating highly discriminative context vectors. Finally, a paraphrase alignment mechanism employs bilinear matching to align the context vectors with candidate paraphrase embeddings, thereby reducing semantic confusion. Experiments show that the model achieves a Top-1 accuracy of 88.6 % on high-frequency words, with an average paraphrase alignment similarity of 0.920. In perturbation tests, the average robustness index is 0.159, representing reductions of 62.7%, 53.9%, and 39.5% compared to Word2Vec+CNN, BiLSTM, and BERT, respectively. The method presented in this paper helps to enhance the accuracy and stability of word meaning recognition in technical texts, providing reliable support for knowledge mining and intelligent text processing in technical domains. Its terminology alignment is also useful for engineering corpora where antenna, wavepropagation and materials terms require context-sensitive interpretation.

INDEX TERMS technical text processing,word sense disambiguation,bilstm attention,patent terminology,wave-propagation terminology

I. INTRODUCTION

WITH the rapid development of science and technology, Chinese patent texts have become an important carrier of knowledge, recording and disseminating a large number of technological achievements. This is particularly true for innovation in traditional yet evolving sectors, where advancements in industry processes, materials, and machinery are increasingly captured and protected within these comprehensive patent documents. Such technical texts contain a large number of professional terms and complex language structures. Many terms are polysemous and often have different meanings in various contexts [1,2]. This semantic diversity not only increases the difficulty of automated retrieval, knowledge mining, and intelligent question-answering but also seriously restricts the accuracy of information extraction and technical intelligence analysis [3,4]. Traditional rule-based or dictionary-based methods require extensive manual maintenance and cannot adapt to the dynamic evolution of terminology. Additionally, the contextual semantic dependency modeling of polysemous words is insufficient, particularly in terms of lacking an accurate focus on key semantic components. These models fail to accurately focus on key semantic components, resulting in insufficient ability to distinguish between technical terms (i.e., low-frequency terms) and their fine-grained semantics that appear less frequently in the corpus. These terms are precisely the core of industry-specific communication and knowledge expression.

Existing research primarily focuses on addressing the issues of terminology polysemy, semantic imprecision, and insufficient context-dependent information extraction in technical texts [5-7]. Kumar Ram utilized an enhanced Needleman-Wunsch algorithm to address the uncertainty, imprecision, and semantic ambiguity of index terms from

both local and global perspectives in the information retrieval process. The top-k word selection technique was used to determine and return the k most similar words from the candidate set, and the effectiveness of the proposed work was evaluated by comparing it with other state-of-the-art technologies [8].

In recent years, the development of neural networks, attention mechanisms, and deep models has made significant contributions to the advancement of WSD and text matching technologies [9-11]. To overcome the limitations of traditional methods in capturing long-range dependencies and key semantic components, research has gradually shifted toward context-aware models based on deep learning. It also particularly focuses on integrating BiLSTM and attention mechanisms to optimize WSD and text matching accuracy in technical texts. Hosseini Manda proposed an attention-based Bi-LSTM model to improve clinical abbreviation disambiguation in clinical literature. Results showed that the proposed attention-based Bi-LSTM model achieved an accuracy of 96.09% on the dataset, outperforming state-of-the-art results [12]. This paper investigates the optimization of technical term meaning recognition for Chinese patent texts within a framework combining BiLSTM and an attention mechanism. Given the highly specialized nature of innovation across various fields, the methodology is designed to be particularly effective in domains like the industry, where precise semantic understanding of materials, processes, and equipment described in patent language is crucial. BiLSTM simultaneously models forward and backward contextual information, capturing bidirectional dependencies between terms within the context. The attention mechanism, combined with weight distribution, highlights key information related to the target term. A paraphrase alignment mechanism is also constructed to provide semantic support for sense discrimination. Bilinear matching is used to achieve deep alignment in the semantic space, enhancing fine-grained sense discrimination capabilities. Combining BiLSTM with attention for paraphrase alignment optimization is crucial for improving the automated processing of technical texts, enhancing technical intelligence mining, and enhancing intellectual property management.

This paper investigates a method for optimizing the accuracy of WSD in technical texts using BiLSTM combined with an attention mechanism. The innovations of this paper are: 1) joint character-word level modeling in the embedding layer, enhancing the sub-word structure awareness through character convolution, enabling low-frequency and out-of-vocabulary terms to obtain reasonable initial representations; 2) cooperative optimization of context encoding and attention mechanisms, employing a two-layer BiLSTM to capture long-range dependencies, followed by location-aware multi-head attention weighted aggregation, dynamically focusing on the contextual components most relevant to the semantic discrimination of the target polysemous word, rather than uniformly focusing on all neighboring words; 3) constructing an independent paraphrase encoder, and through cross-attention and gated bilinear matching, achieving fine-grained alignment of the context vector and candidate paraphrase embeddings in the discriminative semantic space, effectively alleviating semantic confusion caused by the abstract nature of patent language.

II. TECHNICAL TEXT WORD SENSE DISAMBIGUATION OPTIMIZATION MODEL BASED ON BILSTM AND ATTENTION MECHANISM

A. JOINT EMBEDDING REPRESENTATION GENERATION OF TECHNICAL TERMS

Given the large number of long-tail terms, cross-domain neologisms, and professional abbreviations in Chinese patent texts, traditional word embedding training methods alone are difficult to achieve a stable and adequate representation. This paper generates term embedding representations based on joint word- and character-level modeling. Based on large-scale unsupervised pre-training, this approach enhances representation capabilities through character convolution. It constructs a patent-specific technical terminology vocabulary to address the issues of semantic sparsity and poor neologism recognition. Before generating term embeddings, the Chinese patent text is preprocessed by sentence segmentation, word segmentation, normalization, and vocabulary construction. A word segmentation tool based on maximum matching and conditional random fields is used to segment the text to conform to technical language conventions. During the word segmentation stage, polysemous terms and compound nouns are processed. For symbols and abbreviations, let the abbreviation set be $A = \{a_1, a_2, \cdots, a_n\}$ and the normalization function be $F(\cdot)$. The normalized term representation is:

$$t_i = F(a_i), \quad i = 1, 2, \cdots, n \tag{1}$$

t_i is the standardized term corresponding to the abbreviation. Term candidate extraction uses Pointwise Mutual Information (PMI) and left and right entropy to measure the term boundary. For word w, its mutual information is defined as [13]:

$$\mathrm{PMI}(w_i, w_j) = \log \frac{P(w_i, w_j)}{P(w_i)P(w_j)} \tag{2}$$

$P(w_i, w_j)$ represents the co-occurrence probability of words w_i and w_j. Where w_i and w_j are adjacent characters. For a candidate term $w = c_1, c_2, \cdots, c_m$ composed of a character sequence, its average $\overline{\mathrm{PMI}}(w)$ is used to measure the overall correlation strength within the term, pointwise mutual information defined as $\overline{\mathrm{PMI}}(w) = \frac{1}{m-1}\sum_{i=1}^{m-1}\mathrm{PMI}(c_i, c_{i+1})$. When the PMI value is high, it means that the two words have a strong collocation relationship and are suitable as candidate terms. The left and right entropies are used to measure the degree of freedom of the term boundary and are defined as:

$$H_{\mathrm{left}}(w) = -\sum_{x \in L(w)} P(x|w) \log P(x|w) \tag{3}$$

$$H_{\text{right}}(w) = -\sum_{y \in R(w)} P(y|w) \log P(y|w) \tag{4}$$

L(w) represents the left-neighboring context set of a term, and R(w) represents the right-neighboring context set. If a term has large entropy values on both sides, it indicates that its boundary is stable. The vocabulary is constructed by filtering the candidate term set T by setting the average PMI threshold θ_p and the left and right entropy thresholds θ_{left} and θ_{right}, and finally obtaining a professional technical term vocabulary:

$$\begin{aligned} T = \{w \mid \text{PMI}(w) \geq \theta_p \wedge H_{\text{left}}(w) \geq \theta_{\text{left}} \\ \wedge H_{\text{right}}(w) \geq \theta_{\text{right}}\} \end{aligned} \tag{5}$$

Among them, θ_{left} and θ_{right} are used to control the stability of the left and right boundaries of the term, respectively. To obtain domain-specific semantic representations of words, this paper uses the Skip-gram with Negative Sampling (SGNS) model to train on large-scale unlabeled patent text. For a given word w_t, the goal is to maximize its probability of appearing in the context window $[w_{t-c}, w_{t+c}]$ [14,15]:

$$\begin{aligned} \mathcal{L}_{\text{SGNS}} = -\sum_{t=1}^{T} \sum_{\substack{-c \leq j \leq c \\ j \neq 0}} \Bigg[\log \sigma \left(v_{w_t}^{\top} v_{w_{t+j}} \right) \\ + \sum_{k=1}^{\varsigma} \mathbb{E}_{w_k \sim P_n(w)} \log \sigma \left(-v_{w_t}^{\top} v_{w_k} \right) \Bigg] \end{aligned} \tag{6}$$

σ is the Sigmoid function; v_{w_t} is the target word vector; $P_n(w)$ is the negative sampling distribution; ς is the number of negative samples. During training, semantic separability of word vectors is ensured by bringing terms with similar contexts closer together in the vector space. To further improve the representation of low-frequency terms in the patent field, this paper employs subword decomposition technology to break words into character n-grams and simultaneously optimize the representation of words and subwords during word vector learning. This allows for reasonable initial embeddings to be obtained even if certain terms rarely appear in the training corpus by sharing subword information.

Given the large number of new words, abbreviations, and compound technical terms in patent texts, relying solely on word embedding training is insufficient to capture their internal structural information. To this end, this paper introduces a character convolutional neural network at the input layer to extract morphological and substructure features from the character-level sequence of terms. It is assumed that a term consists of a character sequence $[c_1, c_2, \cdots, c_m]$, where each character is mapped to a low-dimensional embedding vector $e(c_i) \in \mathbb{R}^d$. Local features are obtained through a one-dimensional convolution operation [16,17]:

$$h_j = f\left(W \cdot e(c_{j:j+k-1}) + b\right) \tag{7}$$

f(·) is a nonlinear activation function. Then, the most significant features are selected through the maximum pooling operation:

$$h^{char} = \max_j h_j \tag{8}$$

The resulting character-level representation h^{char} is used to capture morphological patterns within the term and to complement long-tail new words and abbreviations.

To fully leverage word embeddings and character features, this paper employs a vector-level fusion strategy to combine the two to form the final embedding representation of the term:

$$h^{term} = [v_w; h^{char}] \tag{9}$$

v_w is the word vector representation, and h^{char} is the character convolution feature. Based on the joint representation, it retains the global semantic information of the word and integrates the character-level local information to solve the representation defects of low-frequency terms and unregistered words.

B. BILSTM CONTEXT ENCODING

Based on the joint embedding representation of technical terms, this paper constructs a BiLSTM to perform deep contextual encoding of text sequences. As shown in Figure 1, two independent LSTM (Long Short-Term Memory) subnetworks, one forward and one backward, work together to capture contextual semantic information before and after the target word, respectively. Ultimately, this network is concatenated to generate a fully context-aware latent state representation, thereby comprehensively modeling the complex semantic structure of technical text. BiLSTM is an extension of the traditional LSTM. It consists of two independent LSTM networks: a forward LSTM to capture contextual information about words in the sequence, and a backward LSTM to capture contextual information about words in the sequence.

This paper adopts a two-layer BiLSTM structure, primarily based on the following considerations: technical patent texts often contain long complex sentences and semantic dependencies across clauses, making it difficult for a single-layer BiLSTM to adequately model such deep contextual relationships; in sequence labeling and semantic matching tasks, two-layer recurrent networks can typically achieve a good balance between expressive power and the risk of overfitting. In preliminary experiments, the single-layer model shows significantly weaker ability to capture key semantic components compared to the two-layer structure when processing sentences containing multi-level process descriptions. Although a three-layer BiLSTM might further improve performance, its marginal benefits are limited, and it significantly increases the computational burden, failing to meet the practicality requirements of this study. Therefore,

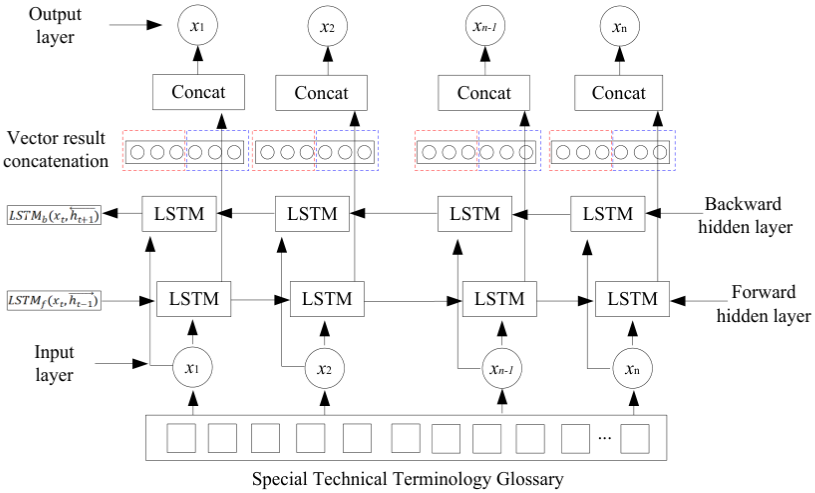

FIGURE 1. BiLSTM context encoding architecture

considering both performance and efficiency, a two-layer BiLSTM is selected as the basic encoding architecture.

Let the input sequence be $X = [x_1, x_2, \cdots, x_T]$, where $x_t = h^{term} \in \mathbb{R}^d$ is the joint embedding representation of the t-th term. The hidden state of the forward LSTM is represented as:

$$\overrightarrow{h_t} = \text{LSTM}_f(x_t, \overrightarrow{h_{t-1}}) \tag{10}$$

The hidden state of the backward LSTM is represented as:

$$\overleftarrow{h_t} = \text{LSTM}_b(x_t, \overleftarrow{h_{t+1}}) \tag{11}$$

The final bidirectional hidden state is concatenated to form the context representation:

$$h_t^{bi} = [\overrightarrow{h_t}; \overleftarrow{h_t}] \tag{12}$$

[;] represents a vector concatenation operation. By simultaneously retaining the preceding and following context of a term, long-distance dependencies are modeled, improving the understanding of the semantics of complex technical texts. LSTM units control information flow through a gating mechanism, alleviating the vanishing gradient problem and preserving long-term dependency information [18,19]. For each time step, the calculation formula is:

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \tag{13}$$

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f) \tag{14}$$

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \tag{15}$$

$$\tilde{c}_t = \tanh(W_c x_t + U_c h_{t-1} + b_c) \tag{16}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \tag{17}$$

$\odot$ represents an element-by-element multiplication operation, while W, U, and b represent the network parameter matrix and bias term. The forward and backward LSTMs independently compute hidden states, which are then concatenated to form a bidirectional representation [20,21]. Concatenating the forward and backward states makes the representation of each term more sensitive to the entire sentence. Through BiLSTM context encoding, the forward and backward loops capture long-range semantic information by combining the context vectors of the terms' semantic information before and after the sentence and paragraph. Low-frequency terms and long-tail terms are dynamically distinguished based on the context.

C. MULTI-HEAD ATTENTION WEIGHTED AGGREGATION

This paper employs a multi-head attention mechanism applied to the entire context sequence encoded by BiLSTM, aiming to enhance the model's global awareness of context. To achieve semantic focus on target polysemous words, after obtaining the enhanced representation of the entire sequence through standard self-attention computation, the contextual information aggregated by the target word's position vector is extracted and specifically analyzed. The model performs a multi-head self-attention transformation on the contextual hidden state sequence, ensuring that the representation at each position incorporates information from other positions in the sequence. For the target polysemous word position, the corresponding output vector is essentially a weighted aggregation value of its query vector interacting with all keys in the sequence, containing the most relevant contextual semantic components to the target word, thus achieving focus on contextual information.

To enhance the model's ability to perceive different semantic subspaces, the above process is executed in parallel

eight times (i.e., eight attention heads), each learning a different semantic attention pattern. The outputs of all heads are concatenated and fused into the final context focus vector through a linear transformation and layer normalization operation. This vector retains the contextual evidence most relevant to the target word, serving as input to the subsequent paraphrasing alignment module. Unlike traditional whole-sequence self-attention, it is a local cross-attention form specifically tailored for word sense disambiguation tasks: it always takes the target word as a guide and selectively extracts discriminative information from the context, thereby improving semantic focusing accuracy while maintaining computational efficiency.

Based on the BiLSTM context encoding, the context hidden state sequence is obtained:

$$H = [h_1^{bi}, h_2^{bi}, \cdots, h_T^{bi}] \in \mathbb{R}^{T \times d} \tag{18}$$

To further focus on the semantic information most relevant to the target polysemous word, this paper introduces a multi-head attention mechanism for weighted aggregation. Based on linear transformations, it generates the query matrix Q, key matrix K, and value matrix V. The attention distribution is calculated and weighted summed to obtain the head representation. Finally, the outputs of each head are concatenated and mapped back to the original dimension. By using different attention heads to capture information in different semantic subspaces in parallel, it achieves refined semantic discrimination of the polysemous word context, thereby optimizing the accuracy of WSD in technical texts, as shown in Figure 2:

Independent linear mapping is performed on the i-th attention head [22,23]:

$$Q_i = HW_i^Q + b_i^Q \tag{19}$$

$$K_i = HW_i^K + b_i^K \tag{20}$$

$$V_i = HW_i^V + b_i^V \tag{21}$$

W_i^Q, W_i^K, and W_i^V are learnable weight matrices; b_i^Q, b_i^K, and b_i^V are bias vectors. Q_i, K_i, $V_i \in \mathbb{R}^{T \times d_k}$, and $d_k = d/H$. The query, key, and value vectors for each time step are:

$$q_t^i = h_t^{bi} W_i^Q + b_i^Q \tag{22}$$

$$k_t^i = h_t^{bi} W_i^K + b_i^K \tag{23}$$

$$v_t^i = h_t^{bi} W_i^V + b_i^V \tag{24}$$

The scaled dot product attention is used to calculate the weight matrix. The attention weight matrix $A_i \in \mathbb{R}^{T \times T}$ [24,25] of each head is:

$$A_i = \text{softmax}\left(\frac{Q_i K_i^\top}{\sqrt{d_k}}\right) \tag{25}$$

Specifically for each position t, the s weight is:

$$a_{t,s}^i = \frac{\exp\left(q_t^i \cdot (k_s^i)^\top / \sqrt{d_h}\right)}{\sum_{j=1}^{T} \exp\left(q_t^i \cdot (k_j^i)^\top / \sqrt{d_h}\right)} \tag{26}$$

$a_{t,s}^i$ represents the attention weight of the t-th position to the s-th position in the sequence, and softmax ensures that $\sum_{s=1}^{T} a_{t,s}^i = 1$. The single-head output is generated by the weighted sum of the attention weights [26,27]:

$$z_t^i = \sum_{s=1}^{T} a_{t,s}^i v_s^i \tag{27}$$

The vector z_t^i of each time step aggregates the attention information of the position in the entire sequence, highlighting the semantic components related to the target polysemous word. The output of the 8 heads is concatenated:

$$Z_{\text{concat}} = [Z_1; Z_2; \cdots; Z_8] \in \mathbb{R}^{T \times d} \tag{28}$$

Finally, by linearly mapping back to the original dimension, the context focus representation $z_t \in \mathbb{R}^d$ is obtained:

$$z_t = Z_{\text{concat},t} W^O + b^O \tag{29}$$

$W^O \in \mathbb{R}^{d \times d}$ is the output mapping matrix, and $b^O \in \mathbb{R}^d$ is the bias. The final output of the entire sequence is:

$$Z = [z_1; z_2; \cdots; z_T] \in \mathbb{R}^{T \times d} \tag{30}$$

Assuming that the position of the target polysemous word in the sequence is t^*, then its focus is expressed as:

$$z_{t^*} = \sum_{i=1}^{H} \sum_{s=1}^{T} a_{t^*,s}^i \left(h_s^{bi} W_i^V + b_i^V\right) W^O + b^O \tag{31}$$

The final focus representation of the target polysemous word position is given by formula (31), which realizes dynamic focus on key contextual semantic components by aggregating the information of the target word and the whole sequence in each attention head after interaction. The inner layer weighted summation $\sum_s a_{t^*,s}^i v_s^i$ reinforces contextual information related to polysemous words. The outer layer concatenation and linear mapping integrate the subspace information from different heads. The resulting z_{t^*} is directly used for word sense matching or disambiguation tasks.

The cross-entropy loss function L is used in training to supervise the correct sense selection for polysemous words. Given that each training instance (sentence) contains only one target polysemous word to be disambiguated, let the position index of the target word in the sentence be t^*. For a batch of training data, the loss function is defined as:

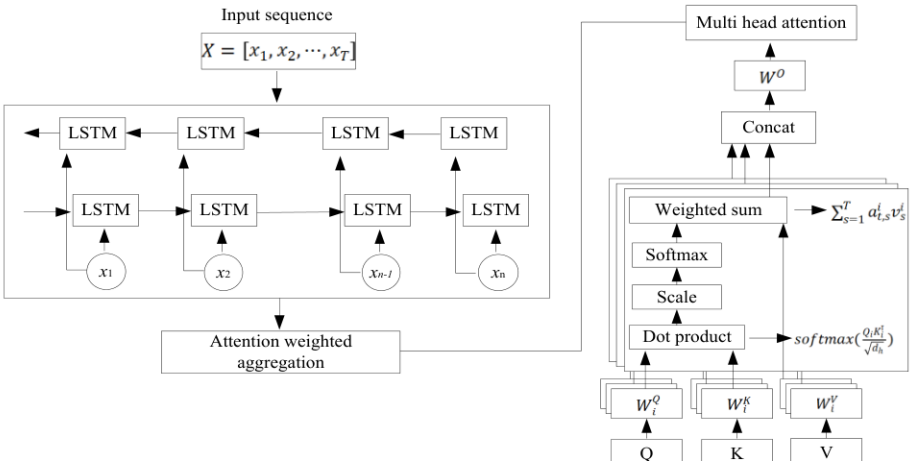

FIGURE 2. Attention weighted aggregation mechanism

$$L = -\sum_{w \in W} \sum_{c=1}^{C_w} \mathbb{I}(y_w = c) \log P(s_c | z_{t_w^*}) \tag{32}$$

Where W represents the set of all target polysemous words in the training data, and for each word w, t_w^* represents its position index in the current sentence, and $z_{t_w^*}$ is its corresponding context focus vector. C_w is the number of candidate senses corresponding to the polysemous word w in the predefined sense library. $P(s_c|z_{t_w^*})$ represents the probability that the model predicts the word as the c-th sense s_c given the context vector $z_{t_w^*}$, and this probability is calculated through a word-specific softmax classification layer:

$$P(s_c | z_{t_w^*}) = \frac{\exp((v_c^{(w)})^\top z_{t_w^*} + b_c^{(w)})}{\sum_{j=1}^{C_w} \exp((v_j^{(w)})^\top z_{t_w^*} + b_j^{(w)})}.$$

$v_c^{(w)}$ and $b_c^{(w)}$ are the classification weight vector and bias term associated with the c-th sense of word w, respectively.

D. INTERPRETATION ALIGNMENT MECHANISM

Based on multi-head attention weighted aggregation, a paraphrase alignment mechanism is constructed to deeply align the context representation with the paraphrase semantics of the candidate meaning. By fusing the global and local information of the two, it explicitly determines which meaning a term best matches in a specific context, as shown in Figure 3:

In terms of the overall architecture, a separate encoder is first established for each candidate paraphrase. A BiLSTM is used to model the sequential dependencies within the paraphrase. An attention mechanism is then employed to extract a global summary and a local focused representation, resulting in a semantic vector for the candidate. The model then incorporates cross-attention, using the contextual focused vector as a query and matching paraphrase encodings word by word, highlighting the most relevant paraphrase components. Then, through vector fusion and bilinear matching, the model maps the contextual and paraphrase semantics into a discriminant space, generating an aligned comprehensive representation. Finally, a probabilistic discriminant is performed to select the paraphrase most consistent with the context. Semantic discrimination in technical texts often relies on fine-grained correspondences on specific semantic dimensions, rather than the consistency of the overall vector direction. Cosine similarity or dot product essentially assumes that all semantic dimensions are equally important and that the interaction is symmetric and linear, making it difficult to model this dimensionality selectivity and asymmetric dependency. In contrast, bilinear matching, by introducing a learnable weight matrix, allows the model to adaptively measure the interaction strength between context vectors and paraphrasing vectors in different semantic subspaces. It retains sufficient expressive power to distinguish between semantically similar but technically different paraphrases, while avoiding the training instability problems caused by overly complex neural network structures.

The candidate definitions in this model are derived from structured domain knowledge sources, and the Chinese patent definitions are extracted semi-automatically and verified by experts. First, potential descriptive fragments of terms were automatically extracted from the specification, claims, and background art sections of the patent text used in the experiment, using pattern matching and keyword triggering rules. Then, three experts with backgrounds in patent examination independently reviewed the automatically extracted candidate definitions:

1) determining whether the extracted fragments constituted a valid definition of the term in the patent context;

2) correcting unclear, ambiguous, or overly lengthy descriptions;

3) supplementing standard or commonly used definitions for terms not covered by automatic extraction.

After independent annotation, all items with disagreements were discussed collectively until a consensus was reached. The final definition library was constructed based

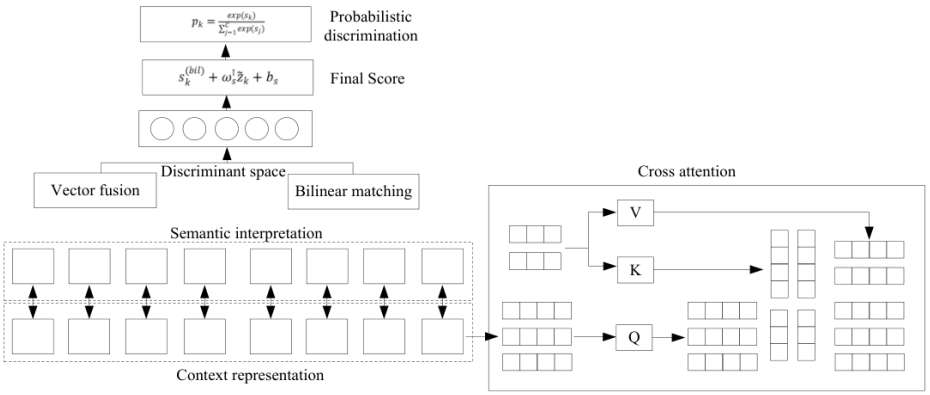

FIGURE 3. Paraphrase alignment architecture

on definitions agreed upon by at least two experts, ensuring the credibility of the annotation results. Statistical analysis showed that the Fleiss-Kappa coefficient for the initial definition validity judgment among experts (step 1) was 0.78, indicating a high degree of consistency. Candidate paraphrases are provided in natural language formats such as dictionary definitions, Wikipedia descriptions, and standard terminology entries. They are mapped to a shared semantic space using a BiLSTM encoder and then bilinearly matched with the context vector. For the candidate, its paraphrase text is represented as a sequence of tokens $Y_k = [y_{k,1}, \cdots, y_{k,L_k}]$, with each token jointly embedded and mapped:

$$E(Y_k) = [e(y_{k,1}), \cdots, e(y_{k,L_k})] \in \mathbb{R}^{L_k \times d} \tag{33}$$

Y_k represents the paraphrase text sequence of the k-th candidate meaning, of length L_k. Each word is mapped to a vector $e(y_{k,t})$, which is derived from a joint embedding—a concatenation of the word-level vector and the character convolution vector—ensuring that the paraphrase text and context are in the same vector space. $e(y_{k,t})$ is isomorphic to $h_{\text{word,char}}$, ensuring that the paraphrase and context are in the same vector space. The paraphrase sequence is then encoded using a BiLSTM, as shown in Table 1:

The BiLSTM gating and state updates here align with the BiLSTM context encoding steps, encoding the sequential information within the interpretation at the same scale. The interpretation sequence is encoded using a BiLSTM with the same structure as the context: the forward pass captures the left-to-right semantic flow of the interpretation, and the backward pass captures the right-to-left information. Each position generates a bidirectional hidden state $h^{bi}_{k,t}$ with dimension d, consistent with the context. When contextual information is limited, weighted pooling is performed on $H^{(k)}$ to obtain a global semantic prior for the interpretation:

$$\hat{a}_{k,t} = \omega_I^\top \tanh(W_I h^{bi}_{k,t}) \tag{34}$$

$$a_{k,t} = \frac{\exp(\hat{a}_{k,t})}{\sum_{\ell=1}^{L_k} \exp(\hat{a}_{k,\ell})} \tag{35}$$

$$u_k = \sum_{t=1}^{L_k} a_{k,t} h^{bi}_{k,t} \in \mathbb{R}^d \tag{36}$$

The context is considered to have "limited contextual information" when any of the following conditions are met:

1) the number of valid lexical units in the context window is less than 5;

2) the Shannon entropy of the context is less than 0.8 (calculated in units of characters), indicating insufficient semantic diversity.

In attention pooling, a learnable projection W_I and activation $\tanh$ are used to map each position to a scalar score, which is then compressed into a weighted score using $\omega_I^\top$. Softmax normalization is then performed to obtain the weight $a_{k,t}$ at each position in the paraphrase sequence. These weights are then weighted and summed across the position vector to produce a global paraphrase vector u_k. u_k captures the core words in the paraphrase text and serves as a global semantic summary of the paraphrase. To discover which tokens in the paraphrase specifically support a particular meaning in the current context, cross-attention is employed, using z_{t^*} as the query and the paraphrase $h^{bi}_{k,t}$ as the key/value [28,29]:

$$\beta_{k,t} = \frac{\exp(z_{t^*}^\top W_a h^{bi}_{k,t}/\sqrt{d})}{\sum_{\ell=1}^{L_k} \exp(z_{t^*}^\top W_a h^{bi}_{k,\ell}/\sqrt{d})} \tag{37}$$

$$v_k = \sum_{t=1}^{L_k} \beta_{k,t} h^{bi}_{k,t} \in \mathbb{R}^d \tag{38}$$

The paraphrase vector is first projected into the matching space through a linear transformation W_a, and then the standard scaled dot product is calculated and softmax normalized to obtain the matching weight $\beta_{k,t}$ of each paraphrase position relative to the context. v_k is the "local aggregation vector of the paraphrase facing the current context", which emphasizes the words or fragments in the paraphrase that are most relevant to the current context. The global summary u_k and the context-aware local aggregation

TABLE 1. Interpretation sequence encoding

Step	Variables	Encoding
Forward LSTM	$\overrightarrow{h}_{k,t}$	$\mathrm{LSTM}_f(e(y_{k,t}), \overrightarrow{h}_{k,t-1})$
Backward LSTM	$\overleftarrow{h}_{k,t}$	$\mathrm{LSTM}_b(e(y_{k,t}), \overleftarrow{h}_{k,t+1})$
Context definition indicates	$h^{bi}_{k,t}$	$[\overrightarrow{h}_{k,t}; \overleftarrow{h}_{k,t}] \in \mathbb{R}^d$
Context definition: hidden state sequence	$H^{(k)}$	$[h^{bi}_{k,1}, \cdots, h^{bi}_{k,L_k}] \in \mathbb{R}^{L_k \times d}$

v_k are concatenated and fused into a unified paraphrase representation g_k through linear transformation and nonlinear activation [30]:

$$g_k = \tanh(W_g[u_k; v_k] + b_g) \in \mathbb{R}^d \tag{39}$$

The concatenation operation $[u_k; v_k] \in \mathbb{R}^{2d}$ and $W_g \in \mathbb{R}^{d\times 2d}$ project it back to the d dimension. After fusion, g_k retains both the "global indication" of the interpretation and the "local response to the current context." To obtain a more discriminative context-interpretation consistency measure, bilinear matching is used and adaptive fusion is performed through gating. The bilinear consistency score is:

$$s_k^{(bil)} = z_{t^*}^\top W_b g_k + b_b, \quad W_b \in \mathbb{R}^{d\times d} \tag{40}$$

Gated fusion is expressed as:

$$r_k = \tanh(W_r[z_{t^*}^\top; g_k] + b_r) \tag{41}$$

$$\lambda_k = \sigma(W_\lambda[z_{t^*}^\top; g_k] + b_\lambda) \tag{42}$$

$$\tilde{z}_k = \lambda_k \odot r_k + (1 - \lambda_k) \odot z_{t^*}^\top \in \mathbb{R}^d \tag{43}$$

With $W_r \in \mathbb{R}^{d\times 2d}$, the vector gate λ_k allows the model to adaptively balance "interpretation-driven features" with "context-based features" in each dimension. When the context is clear and consistent with the interpretation, λ_k tends to retain z_{t^*}. When the context is ambiguous or the interpretation provides strong evidence, λ_k amplifies the effect of r_k, thereby improving discriminative power in long-tail or scarce context scenarios. The bilinear score is combined with the linear criterion of the fusion vector to form the final score. Then, for the candidate sense set of the current target polysemous word, the softmax function is used to normalize it into the posterior probability of each candidate sense:

$$s_k = s_k^{(bil)} + \omega_p^\top \tilde{z}_k + b_p \tag{44}$$

$$p(s_k|z_{t^*}, Y_k) = \frac{\exp(s_k)}{\sum_{j=1}^{C_w} \exp(s_j)} \tag{45}$$

$p(s_k|z_{t^*}, Y_k)$ represents the probability that the model predicts the target word as the k-th sense s_k, given the contextual focus vector z_{t^*} and candidate senses Y_k. This softmax normalization is word-specific, meaning that for different target words w, the summation term in the denominator only involves the scores of its own C_w candidate senses.

III. WORD SENSE DISAMBIGUATION ACCURACY EVALUATION

A. EXPERIMENTAL DATA

The data used in this paper is mainly derived from Chinese patent texts published on the China Intellectual Property Network, covering patent specifications and claims from 2015 to 2020. To ensure the representativeness and reliability of the experimental results, this paper selects patent specifications in four typical fields of mechanical manufacturing, materials science, information and communication, and new energy as the original corpus, extracts candidate technical terms based on the title, abstract, and text of the patent text, and obtains a preliminary set of term candidates by combining the term joint embedding generation process. By combining manual annotation with patent interpretation resources, a corresponding set of interpretations is established for polysemous terms, and the usage instances and interpretations of each term in different contexts are aligned to form the data pairs required for supervised training and verification. The final dataset information is shown in Table 2:

Terms exhibiting significantly different technical meanings in different contexts within the patent corpus are selected from candidate terms and classified as polysemous. The candidate meaning set for each polysemous term originates from its explicit internal definition. Sentences in which the term is explicitly defined are extracted from the patent specification. For each occurrence instance of each polysemous term, the annotator determines its specific meaning based on the overall semantics of the sentence it appears in and matches it against the constructed candidate meaning set. The candidate meaning with the highest semantic fit to the current context is selected; if all candidate meanings do not match, they are marked as "ambiguous" or "unknown." This process is independently completed by two annotators with relevant technical backgrounds. Initially, a trial annotation is performed using a common subset of 100 sentences, followed by independent annotation of all data. The Cohen's Kappa coefficient for polysemous term meaning judgment between the two annotators is 0.76, indicating good consistency between them. The dataset contains 1,810 patent documents, from which 5,613 candidate technical terms are extracted and labeled, forming the basic vocabulary for model training and evaluation. Based on this, 896 terms with explicit polysemy are identified, with an average of 2.7 candidate definitions associated with each polysemous term.

Each candidate definition is presented in the form of a

TABLE 2. Dataset information

Technical field	Number of patents	Candidate number of terms (items)	Number of polysemous terms	Average number of definitions
Mechanical manufacturing	322	1483	231	2.7
Materials science	481	1242	198	2.5
Information communication	516	1521	254	2.9
New energy	491	1367	213	2.6
Total	1810	5613	896	-

natural language sentence, derived from the explicit definition in the patent specification, and is used for model training after standardization. To enhance the reproducibility and comparability of the experimental results, a public modern Chinese word sense disambiguation dataset is used as a supplement. This dataset contains 1,083 polysemous words and an average of 20 context instances per word to test the performance of the model on a general corpus. To evaluate the model's generalization ability and avoid inflated evaluation results due to information leakage, a strategy based on patent documents was adopted. The patent documents were randomly shuffled and divided into training, validation, and test sets in a 7:1.5:1.5 ratio.

B. EXPERIMENTAL SETUP

The model parameter settings for this paper are shown in Table 3:

The evaluation is conducted from the perspectives of long-tail term and fine-grained distinction, interpretation alignment, and cross-domain generalization. In terms of the selection of comparison methods, this paper mainly considers comparison with the current mainstream and representative word sense disambiguation and semantic matching methods:

(1) Word2Vec+CNN: Convolutional neural network is used to capture local context features, and the input is Word2Vec word vector;

(2) BiLSTM (no attention): Only bidirectional LSTM is used for context modeling to examine the contribution of the attention mechanism;

(3) BERT: Using the BERT-wwm-ext Chinese pre-trained model as a baseline, a complete sentence containing the target polysemous word is input into the model. The contextual representation of the target word is obtained by extracting the hidden state of the last Transformer layer at the position corresponding to the target word token. A fully connected classification layer is stacked on top of this representation, and the entire model is fine-tuned end-to-end on the patent training set for the downstream disambiguation task.

C. EXPERIMENTAL RESULTS

1) Long-Tail Terms and Fine-Grained Differentiation

To verify the model's effectiveness in processing long-tail sparse samples and fine-grained sense distinction tasks, this paper uses Top-K accuracy (k ranging from 1, 3, or 5) as an indicator and conducts comparisons based on the term frequency range and the number of senses. Based on the frequency of occurrence of terms in patent texts, they are divided into three categories: high frequency, medium frequency, and low frequency. High-frequency words are those appearing in the top 33% of the total frequency, mid-frequency words are those appearing in the middle 33% of the total frequency, and low-frequency words are those appearing in the bottom 34% of the total frequency. Based on polysemous terms, the number of senses is divided into two senses, three senses, and multiple senses ($\geq$4). The results are shown in Figures 4 and 5:

As shown in Figure 4, the accuracy of high-frequency terms is generally higher than that of low-frequency terms. This is because high-frequency terms appear more frequently in the corpus, allowing for richer contextual features to be learned during training. Low-frequency terms, on the other hand, have sparse context and are prone to poor representation. The BiLSTM+Attention model achieves higher Top-1, Top-3, and Top-5 accuracy for high-frequency, medium-frequency, and low-frequency terms. In Figure 4A, the proposed model achieves 88.6% Top-1 accuracy for high-frequency terms, compared to 78.5%, 82.7%, and 86.2% for the Word2Vec+CNN, BiLSTM, and BERT models, respectively. In Figure 4B, the proposed model achieves 82.3% Top-1 accuracy for medium-frequency terms, compared to 70.2%, 74.8%, and 79.6% for the Word2Vec+CNN, BiLSTM, and BERT models, respectively. In Figure 4C, the proposed model achieves 72.5% Top-1 accuracy for low-frequency terms, compared to 58.4%, 62.9%, and 68.7% for the other three comparison models, respectively. Overall, the proposed model achieves superior Top-K accuracy across different frequency ranges compared to the Word2Vec+CNN, BiLSTM, and BERT models. BiLSTM+Attention enhances the representation of long-tail words through character convolutional features and dynamically aggregates contextual information through the attention mechanism, effectively alleviating the sparse representation of low-frequency terms.

Figure 5 shows a comparison of the Top-K accuracy of different models across a range of polysemous terms. In Figure 5A, the proposed model achieves a Top-1 accuracy of 89.4% for bi-sense terms, while the comparison models achieve 80.6%, 83.9%, and 87.5%, respectively. In Figure 5B, the proposed model achieves a Top-1 accuracy of 83.2% for tri-sense terms, while the comparison models achieve 72.3%, 76.1%, and 80.7%, respectively. In Figure 5C, the proposed model achieves a Top-1 accuracy of 74.1% for polysense terms, while the comparison models achieve 60.1%, 64.2%, and 70.3%, respectively. BiLSTM models contextual dependencies, resolving the difficulty of traditional con-

TABLE 3. Model parameter settings

Module	Parameter	Specifications
BiLSTM	Hidden state dimension	256
	Word vector dimension	300
	BiLSTM layers	2
Attention mechanism	Number of attention heads	8
	Attention dimension	64
	Dropout ratio	0.3
	Batch size	64
	Learning rate	1e-3
	Maximum sequence length	128
Interpretation alignment mechanism	Definition alignment represents dimensions	256

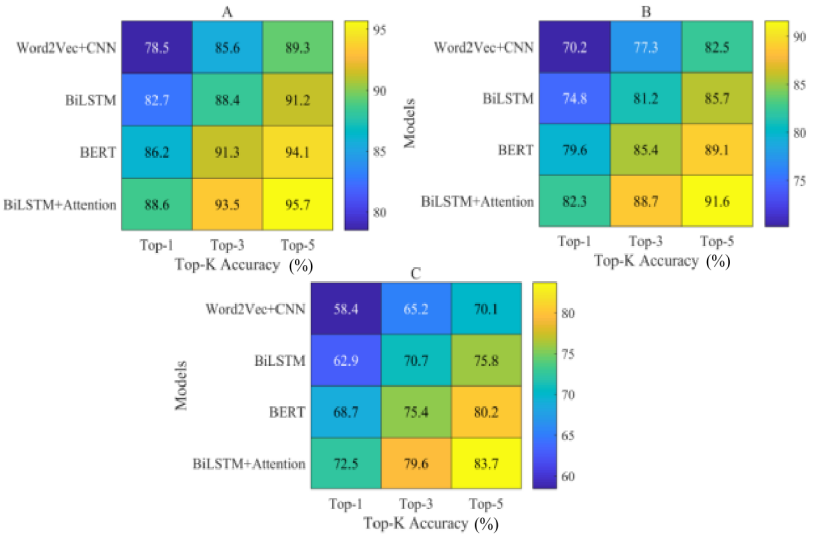

FIGURE 4. Comparison of term frequency intervals: (A) shows the results for high-frequency terms; (B) shows the results for medium-frequency terms; (C) shows the results for low-frequency terms.

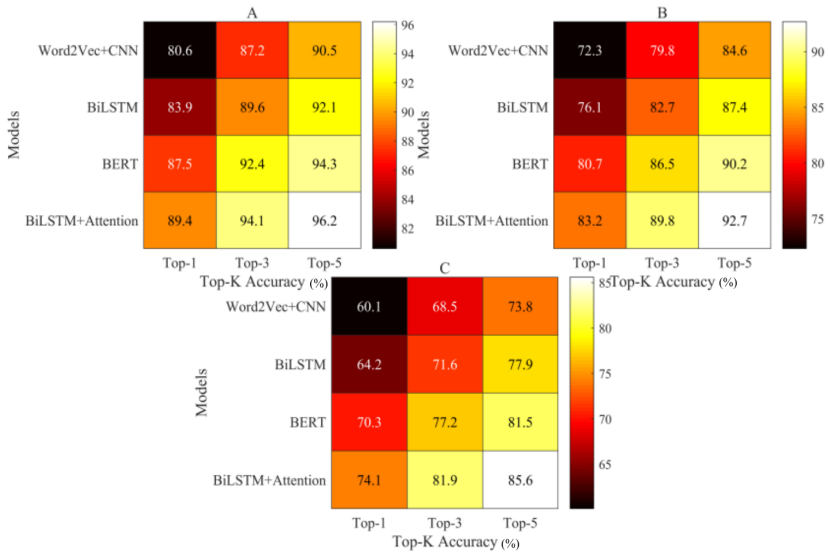

FIGURE 5. Comparison of the number of meanings: (A) shows the results for two-meaning terms; (B) shows the results for three-meaning terms; (C) shows the results for multiple-meaning terms

volutional networks in capturing longrange context. The attention mechanism further weights contextual information, focusing on key semantics and improving the discrimination of low-frequency or polysemous terms. Word2Vec+CNN performs poorly when handling polysemous words because convolution operations primarily capture local features and

lack global semantic information. While BiLSTM alone can model context, it lacks the weighted attention mechanism, making it difficult to highlight core semantic features. BERT's performance is limited when dealing with patent and technical texts, as the distribution of the pre-trained corpus differs from that of professional domain terminology, resulting in insufficient representation of long-tail terms and abbreviations. This model, which combines BiLSTM context encoding with multi-head attention-weighted aggregation, effectively captures contextual semantic information, highlighting the semantic components most relevant to the target polysemous word and thereby improving the accuracy of fine-grained sense differentiation. In the long-tail keyword and fine-grained discrimination experiments, this paper further calculated the macro-average precision, recall, and F1 score at the term level. Precision, recall, and F1 score were calculated individually for each polysemous word appearing in the dataset. A simple arithmetic mean of these metrics for all terms was then performed to obtain the final macro-average precision, recall, and F1 score, as shown in Table 4.

TABLE 4. Comparison of classification performance

Model	Macro-precision (%)	Macro-recall (%)	Macro-F1 (%)
Word2Vec+CNN	72.8	70.5	71.6
BiLSTM	78.5	76.9	77.3
BERT	81.3	79.8	80.2
BiLSTM+Attention	86.7	84.9	85.8

Table 4 shows that BiLSTM+Attention outperforms other compared models in terms of macro-average precision, recall, and F1 score. Macro-precision reaches 86.7%, exceeding the Word2Vec+CNN, BiLSTM, and BERT models by 13.9%, 8.2%, and 5.4%, respectively. Macro-recall reaches 84.9%, exceeding the Word2Vec+CNN, BiLSTM, and BERT models by 14.4%, 8.0%, and 5.1%, respectively. Macro-F1 reaches 85.8%, exceeding the Word2Vec+CNN, BiLSTM, and BERT models by 14.2%, 8.5%, and 5.6%, respectively. The specific results demonstrate that the proposed model offers superior classification performance. Based on context encoding and attention weighting, the proposed model can more accurately extract key evidence determining term meaning from complex and lengthy patent texts, demonstrating a superior classification advantage in term discrimination.

2) Interpretation Alignment

To reveal the model's ability to align paraphrase definitions with varying text features and context richness, it divides the patent data by technology field and measures the model's paraphrase alignment similarity within each field for term context lengths of 1-10, 11-20, 21-40, and 41 words or longer. Paraphrase alignment similarity is calculated as the average cosine similarity between the context vector and the paraphrase vector. The final comparison results are shown in Figure 6.

As shown in Figure 6, the performance of each model improves as the context length increases. However, compared with other models, the BiLSTM+Attention model in this paper has more ideal results in interpretation alignment similarity, with the highest average similarity reaching 0.920. In Figure 6A, when the text length reaches 41 words or more, the average similarity of the proposed model is 0.840, while the average similarities of other models are all below 0.8. In Figure 6B, when the text length reaches 41 words or more, the average similarity of the proposed model is 0.869; the average similarities of other models are 0.630, 0.751, and 0.797, respectively. In Figure 6C, when the text length reaches 41 words or more, the average similarity of the proposed model is 0.920; the average similarities of other models are 0.733, 0.836, and 0.887, respectively. In Figure 6D, when the text length reaches 41 words or more, the average similarity of the proposed model is 0.880; the average similarities of other models are 0.637, 0.794, and 0.820, respectively. Overall, the proposed model demonstrates the strongest paraphrase alignment capabilities across a range of varying context lengths. During training, it generates not only contextual representations of terms but also embedded representations of candidate paraphrases. The context vectors output by Attention align more closely with the paraphrase vectors in the semantic space, thereby reducing confusion between different meanings and improving the similarity of paraphrase alignment.

To further quantify the model's semantic alignment capability, this paper introduces a semantic consistency score based on KL divergence as an evaluation metric. For a given target word and its context, the model calculates a score for each candidate sense using a paraphrase alignment mechanism, which is then normalized using softmax to obtain the contextual semantic distribution. Simultaneously, to obtain a contextindependent paraphrase prior distribution based solely on the paraphrase text itself, a separate paraphrase encoder generates a representation for each candidate paraphrase, which is then mapped to a scalar score through a separate parameter layer and normalized using softmax. The Kullback-Leibler divergence from the paraphrase prior distribution to the contextual semantic distribution is calculated. A smaller KL divergence indicates a closer semantic distribution between the context and the paraphrase, and the model is more stable and consistent in capturing word meaning. The model comparison results are shown in Figure 7:

Figure 7 shows that the proposed model, BiLSTM+Attention, outperforms the comparison models in semantic consistency across all four technical domains. In Figures 7A, 7B, 7C, and 7D, its KL divergence distribution is lower overall, and the box heights are smaller, indicating more stable semantic capture and less susceptibility to contextual fluctuations. In contrast, BERT, a deep pre-trained model, also demonstrates strong consistency in most domains, but still falls short of the proposed model, indicating that semantic deviations can still occur in long contexts.

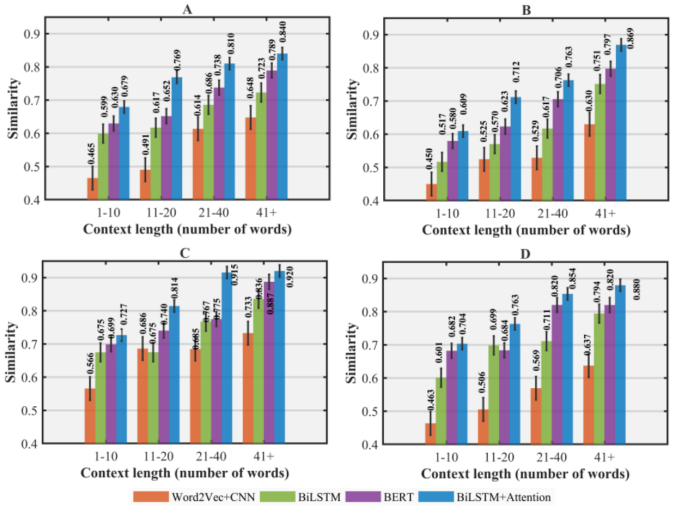

FIGURE 6. Comparison of similarity in semantic alignment: (A) shows the similarity for Mechanical manufacturing; (B) shows the similarity for Materials science; (C) shows the similarity for Information communication; (D) shows the similarity for New energy

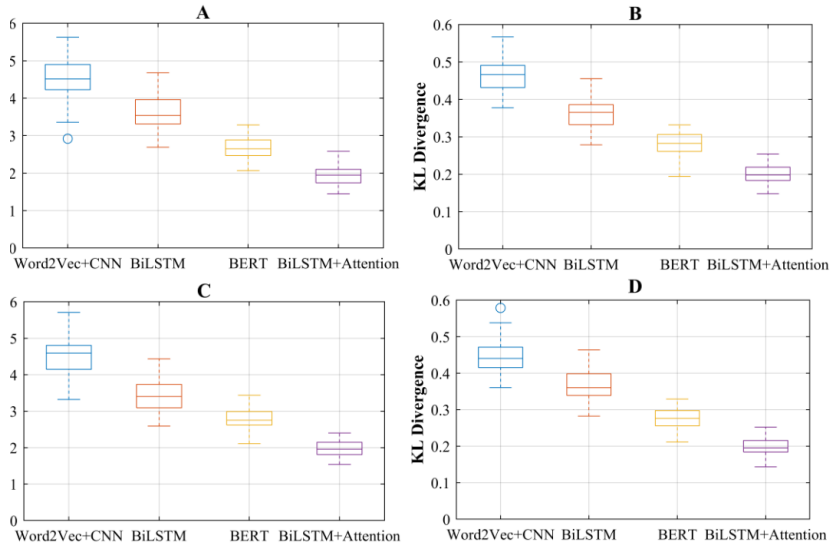

FIGURE 7. Semantic consistency comparison: (A) shows the KL divergence of Mechanical Manufacturing; (B) shows the KL divergence of Materials Science; (C) shows the KL divergence of Information Communication; (D) shows the KL divergence of New Energy

While BiLSTM effectively models contextual order, it lacks an explicit attention mechanism. Word2Vec+CNN performs the worst, with a higher divergence distribution, reflecting its difficulty maintaining stable paraphrase consistency in complex contexts. Overall, the proposed model, leveraging the weighted modeling of key contextual terms by the attention mechanism, effectively enhances semantic consistency during paraphrase alignment, demonstrating a significant advantage in technical text scenarios.

3) Robustness

In the robustness evaluation, the robustness index is introduced as a core indicator to measure the model's stability in the face of input perturbations. This metric is based on the model's Top-1 accuracy performance on the original test set and its Top-1 accuracy performance on various perturbation test sets. For each perturbation type, the robustness metric is defined as the absolute value of the relative rate of performance degradation. Based on the original patent text, synonym replacement, character insertion, character deletion, word order disruption, and mixed perturbations are introduced to simulate text variations that may occur in real application scenarios. The lower the index value, the more stable the model's performance remains under interference conditions, and the stronger its robustness. The model comparison results are shown in Figure 8:

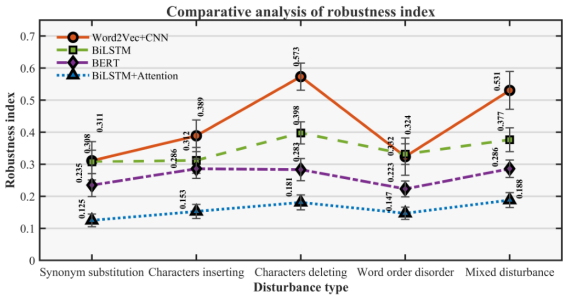

FIGURE 8. Robustness index comparison

In Figure 8, the average robustness index of the BiLSTM+Attention model under all perturbation types is 0.159. The average indices of the other three models are 0.426, 0.345, and 0.263, respectively. Compared to the comparison models, the average robustness index is reduced by 62.7%, 53.9%, and 39.5%, respectively. Word2Vec+CNN has the highest average index, indicating its extreme sensitivity to text perturbations. This is primarily due to its reliance on static word embeddings, which cannot effectively capture contextual changes. The BiLSTM model has advantages in terms of temporal dependencies, but lacks global modeling capabilities, resulting in a moderate average index. The BERT model performs suboptimally under perturbations, but its over-reliance on word-granular embeddings reduces its robustness when subjected to noisy characters. Overall, the model in this paper has stronger semantic understanding capabilities and can effectively recognize the same meaning in different expressions. Through bidirectional context modeling and attention-weighted aggregation, it can more flexibly focus on stable semantic features. Even if some words are replaced or interfered with by noise, the model can still maintain a high degree of interpretation consistency, verifying its stability and reliability in cross-disturbance scenarios.

IV. CONCLUSION

To enhance the semantic precision of technical terms, this paper examines Chinese patent texts and develops an optimization model that integrates BiLSTM and attention mechanisms. As a domain rich in polysemous technical language, the industry benefits critically from this precision, allowing the model to accurately parse complex terms. Based on the joint embedding generation of terms and definitions, a BiLSTM is used to capture bidirectional dependencies in the context, and a multi-head attention-weighted mechanism is combined to highlight key information. Through the definition alignment mechanism, the context vector and the definition vector are more closely aligned in the semantic space. Experimental results show that the Top-1 accuracy of the proposed model is significantly better than that of the comparison models under different term frequencies and polysemy intervals, with outstanding advantages in low-frequency and polysemy scenarios. In evaluating interpretation alignment, similarity, and semantic consistency, the proposed model maintains higher consistency across all technical fields and varying context lengths, with the lowest KL divergence. In the robustness dimension, the average robustness index is 0.159, which is 62.7%, 53.9%, and 39.5% lower than that of the Word2Vec+CNN, BiLSTM, and BERT models, respectively. It not only effectively improves the accuracy of term meaning correspondence but also demonstrates relatively ideal stability and antiinterference ability. This robustness is essential for knowledge extraction in specialized domains, as demonstrated by its reliable support for semantic understanding in professional fields like the industry, where precise terminology is critical for innovation and intellectual property management. However, this paper still has limitations; the current model relies on a predefined set of candidate definitions. Future work can explore an end-to-end generative WSD framework, combining external knowledge graphs to dynamically retrieve definitions, further reducing the dependence on static definition libraries.

AUTHOR CONTRIBUTIONS

Conceptualization – Lanxi Gao; methodology – Lanxi Gao, Tao Dong and Minghan Yang; investigation – Lanxi Gao, Tao Dong and Minghan Yang; writing-original draft preparation – Lanxi Gao, Tao Dong and Minghan Yang. All authors have read and agreed to the publ ished version of the manuscript.

CONFLICTS OF INTEREST

The authors declare no conflict of interest.

FUNDING

This research received no external funding.

ACKNOWLEDGEMENTS

Not applicable.

REFERENCES

[1] F. B. Abdullayeva, "Terminological Precision and Linguistic Challenges in Technical Translation," American Journal of Philological Sciences, vol. 5, no. 4, pp. 333-337, 2025, doi: 10.37547/ajps/Volume05Issue04-82.

[2] A. A. Khoroshilov, A. V. Kan, J. V. Nikitin, and A. D. A. Khoroshilov, "Machine phraseological translation of scientifictechnical texts based on the model of generalized syntagmas," Automatic Documentation and Mathematical Linguistics, vol. 54, no. 2, pp. 79-91, 2020, doi: 10.3103/S0005105520020041.

[3] O. O. Mezentseva and A. S. Kolomiiets, "Optimization of analysis and minimization of information losses in text mining," Herald of Advanced Information Technology, vol. 3, no. 1, pp. 373-382, 2020, doi: 10.15276/hait.01.2020.4.

[4] C. Makris, G. Pispirigos, and M. A. Simos, "Text semantic annotation: A distributed methodology based on community coherence," Algorithms, vol. 13, no. 7, pp. 160-174, 2020, doi: 10.3390/a13070160.

[5] D. Loureiro, K. Rezaee, M. T. Pilehvar, and J. Camacho-Collados, "Analysis and evaluation of language models for word sense disambiguation," Computational Linguistics, vol. 47, no. 2, pp. 387-443, 2021, doi: 10.1162/coli_a_00405.

[6] M. Q. Lu and Y. Y. Shen, "A text matching method based on Siamese network pre-trained language model," Ensemble Technology, vol. 12, no. 2, pp. 53-63, 2023, doi: 10.12146/j.issn.2095-3135.20220817001.

[7] H. Yu, Z. T. Liang, and H. Xie, "A review of online technology supply and demand text matching methods," Information Science, vol. 39, no. 7, pp. 177-185, 2021, doi: 10.13833/j.issn.1007-7634.2021.07.024.
[8] R. Kumar and S. C. Sharma, "Hybrid optimization and ontology-based semantic model for efficient text-based information retrieval," The Journal of Supercomputing, vol. 79, no. 2, pp. 2251-2280, 2022, doi: 10.1007/s11227-022-04708-9.
[9] P. Jiang and X. Cai, "A survey of text2-matching techniques," Information, vol. 15, no. 6, pp. 332-384, 2024, doi: 10.3390/info15060332.
[10] C. X. Zhang, S. Y. Pang, X. Y. Gao, J. Q. Lu, and B. Yu, "Attention neural network for bio-medical word sense disambiguation," Discrete Dynamics in Nature and Society, vol. 2022, no. 1, pp. 1-14, 2022, doi: 10.1155/2022/6182058.
[11] X. Y. Wang and H. S. Wang, "A review of text matching technology based on deep learning," Information and Computers, vol. 32, no. 15, pp. 73-74, 2020, doi: 10.3969/j.issn.1003-9767.2020.15.027.
[12] M. Hosseini, A. H. Rasekh, and A. Keshavarzi, "Improving clinical abbreviation sense disambiguation using attention-based Bi-LSTM and hybrid balancing techniques in imbalanced datasets," Journal of Evaluation in Clinical Practice, vol. 30, no. 7, pp. 1327-1336, 2024, doi: 10.1111/jep.14041.
[13] A. Salle and A. Villavicencio, "Understanding the effects of negative (and positive) pointwise mutual information on word vectors," Journal of Experimental & Theoretical Artificial Intelligence, vol. 35, no. 8, pp. 1161-1199, 2023, doi: 10.1080/0952813x.2022.2072004.
[14] F. A. O. Santos, T. D. Bispo, H. T. Macedo, and C. Zanchettin, "Morphological Skip-Gram: Replacing FastText characters n-gram with morphological knowledge," Inteligencia Artificial, vol. 24, no. 67, pp. 1-17, 2021, doi: 10.4114/intartif.vol24iss67pp1-17.
[15] Z. Yang, M. Ding, T. L. Huang, Y. K. Cen, J. S. Song, and B. Xu, "Does negative sampling matter? a review with insights into its theory and applications," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 46, no. 8, pp. 5692-5711, 2024, doi: 10.1109/TPAMI.2024.3371473.
[16] B. Liu, Y. Zhou, and W. Sun, "Character-level text classification via convolutional neural network and gated recurrent unit," International Journal of Machine Learning and Cybernetics, vol. 11, no. 8, pp. 1939-1949, 2020, doi: 10.1007/s13042-020-01084-9.
[17] R. Harizi, R. Walha, F. Drira, and M. Zaied, "Convolutional neural network with joint stepwise character/word modeling based system for scene text recognition," Multimedia Tools and Applications, vol. 81, no. 3, pp. 3091-3106, 2022, doi: 10.1007/s11042-021-10663-z.
[18] S. Minaee, N. Kalchbrenner, E. Cambria, N. Nikzad, M. Chenaghlu, and J. Gao, "Deep learning–based text classification: a comprehensive review," ACM Computing Surveys (CSUR), vol. 54, no. 3, pp. 1-40, 2021, doi: 10.1145/3439726.
[19] S. Johnson, S. Shen, and Y. Liu, "CWPC_BiAtt: Character–word–position combined BiLSTM-attention for Chinese named entity recognition," Information, vol. 11, no. 1, pp. 45-63, 2020, doi: 10.3390/info11010045.
[20] A. K. Nandanwar and J. Choudhary, "Semantic features with contextual knowledge-based web page categorization using the GloVe model and stacked BiLSTM," Symmetry, vol. 13, no. 10, pp. 1772-1788, 2021, doi: 10.3390/sym13101772.
[21] B. Jang, M. Kim, G. Harerimana, S. U. Kang, and J. W. Kim, "Bi-LSTM model to increase accuracy in text classification: Combining Word2vec CNN and attention mechanism," Applied Sciences, vol. 10, no. 17, pp. 5841-5854, 2020, doi: 10.3390/app10175841.
[22] F. J. Liu, N. Zhao, and G. Q. Zhu, "Cognitive difference text classification in online knowledge collaboration based on SA-BiLSTM hybrid model," Scientific Reports, vol. 15, no. 1, pp. 22171-22184, 2025, doi: 10.1038/s41598-025-06914-w.
[23] X. R. Yang, S. W. Zhao, R. X. Zhang, X. J. Yang, and Y. H. Tao, "BiLSTM_CNN text classification model combining selfattention and residual," Journal of Computer Engineering & Applications, vol. 58, no. 3, pp. 172-180, 2022, doi: 10.3778/j.issn.1002-8331.2104-025.
[24] M. J. Yuan, K. Z. Jiang, Y. Yang, and L. X. Hui, "MAC_BiLSTM text classification model combining self-attention and normalization," Advances in Applied Mathematics, vol. 11, no. 10, pp. 7012-7025, 2022, doi: 10.12677/AAM.2022.1110744.
[25] L. Shi, Y. Wang, Y. Cheng, and R. B. Wei, "A review of attention mechanism research in natural language processing," Data Analysis and Knowledge Discovery, vol. 4, no. 5, pp. 1-14, 2020, doi: 10.11925/infotech.2096-3467.2019.1317.
[26] T.-P. Shen, Q. Q. Meng, and Z. H. Zhan, "Named entity recognition of Chinese text based on attention mechanism," Journal of Network Intelligence, vol. 8, no. 4, pp. 504-517, 2023, [Online]. Available: https://bit.nkust.edu.tw/~jni/2023/vol8/s2/13.JNI-0510.pdf.
[27] J. Cheng, W. Tong, and W. Yan, "Capsule network improved multi-head attention for word sense disambiguation," Applied Sciences, vol. 11, no. 6, pp. 2488-2501, 2021, doi: 10.3390/app11062488.
[28] L. Enamoto, A. R. A. S. Santos, R. Maia, W. Li, and G. P. Rocha Filho, "Multi-label legal text classification with BiLSTM and attention," International Journal of Computer Applications in Technology, vol. 68, no. 4, pp. 369-378, 2022, doi: 10.1504/ijcat.2022.125186.
[29] Y. Y. Yan, F. A. Liu, X. Q. Zhuang, and J. Ju, "An R-transformer_BiLSTM model based on attention for multi-label text classification," Neural Processing Letters, vol. 55, no. 2, pp. 1293-1316, 2023, doi: 10.1007/s11063-022-10938-y.
[30] Y. F. Zheng, Z. H. Gao, J. Shen, and X. S. Zhai, "Optimizing automatic text classification approach in adaptive online collaborative discussion–A perspective of attention mechanism-based bi-lstm," IEEE Transactions on Learning Technologies, vol. 16, no. 5, pp. 591-602, 2022, doi: 10.1109/TLT.2022.3192116.