BERT-based Automatic Error Correction System for Chinese Learners

Yunlong Diao[1], Wen Gao[2]
[1]School of Humanities and Design, Henan Open University, Zhengzhou 450000, Henan, China
[2]Personnel Department of the Organization Department, Henan Open University, Zhengzhou 450000, Henan, China
Corresponding author: Yunlong Diao (e-mail: dylong566@126.com)

ABSTRACT Current automatic error correction methods for Chinese learners often focus on superficial word- or sentence-level processing and are affected by inconsistent annotation standards, resulting in limited generalization to real learner texts and causing misalignment or overcorrection. To improve grammatical correctness, semantic fidelity, and instructional relevance, this paper constructs a multi-granularity lexical representation method. Using characters as the basic unit, the method integrates three types of information: characters, words, and pinyin. These representations are learned through projection processing, concatenated into a unified embedding, and fed into a pre-trained Chinese BERT encoder. A dual-task head consisting of sequence labeling and lightweight generation is deployed on the shared BERT encoder for collaborative optimization. Key innovations include an alignment consistency loss to ensure character-level consistency, and the combination of word-segmentation augmentation and multi-reference soft-label training to reduce conflicts caused by segmentation differences. Experimental results show that the method achieves granular alignment consistency above 0.800, corrected-original sentence similarity of 0.888, and an overcorrection rate as low as 0.041. These results demonstrate improved robustness and explainability for automatic Chinese learner error correction.

INDEX TERMS chinese learner error correction, bert, multi-granularity alignment, soft target training, semantic fidelity

I. INTRODUCTION

WITH the development of international Chinese language education, more and more non-native Chinese speakers are beginning to systematically learn Chinese. Learners often make errors in spelling, vocabulary, and grammar in their written expressions, which poses challenges to teaching evaluation, learner feedback, and intelligent error correction [1,2]. Current automatic error correction systems for Chinese learners rely primarily on high-quality annotated corpora for supervised training. However, due to the lack of natural word boundaries in Chinese and the inconsistent annotation standards across different datasets, automatic error correction systems for Chinese learners are often used. This leads to conflicts in the training signals, limiting the model's generalization ability on real learner texts. It is prone to misalignment, ambiguous word segmentation, and overcorrection, making it difficult for the system to provide robust and accurate error correction in practical applications [3,4]. Therefore, how to build an efficient and generalizable automatic error correction model for Chinese learners in a limited and incompletely consistent corpus environment has become a core issue that needs to be addressed in current research.

Existing research on automatic error correction for Chinese learners mainly focuses on solving the adaptability problem of supervised grammatical error correction models to different corpora and error types [5-7]. Existing research mainly improves the robustness of error correction from two aspects: model architecture and training strategy. In the field of general GEC (Grammatical Error Correction), Wang Yu's review systematically sorted out the evolution of statistical and neural machine translation methods in automatic error correction [8]. Qin Mengyang and Zhong Yanghui introduced adversarial training and CNN (Convolutional Neural Network) to enhance generative fluency, respectively [9,10]. Zhu Shichang and Duan Ruixue employed corpus weighting and curriculum learning to mitigate out-of-distribution generalization, respectively [11,12]. However, these works are mostly based on general corpora and fail to fully consider the specificity of learner errors, including frequent errors such as missing function words, confusion of synonyms, and word order errors. Their annotations are highly dependent on word segmentation results, making label conflicts easily caused by differences in word segmentation. Although Musyafa Ahmad verified the Transformer's transferability to low-resource languages [13], it did not address the issue of ambiguous word boundaries. Overall, these methods generally rely on surface error pattern matching, fail to effectively connect with the cognitive characteristics of Chinese as a second language, and still have limitations in deep semantic

understanding and robustness. In recent years, research on automatic error correction for Chinese learners based on BERT has gradually become mainstream, aiming to improve error correction accuracy and robustness by leveraging the deep semantic representation capabilities of pre-trained language models [14-16]. Wang Zhici proposed a grammatical correction method based on prompt templates. The Chinese GEC task was converted into a masked completion task compatible with the masked language model BERT. The corrected errors were incorporated into the template through dynamic updates. The results showed that the proposed method achieved higher correction performance and lower miscorrection rate in high error density scenarios [17]. Wang Hongfei studied the role of pre-trained models and pseudo data in Chinese GEC. He built a model based on Chinese BERT, T5 (Text-To-Text Transfer Transformer), and BART, and conducted experiments with a variety of pseudo data to verify the effectiveness of pre-trained models and pseudo data for Chinese GEC, providing a reproducible strong baseline [18]. Qing Yu proposed an automatic error correction system based on optimized BERT and Transformer. By introducing a hybrid attention mechanism and dilated convolution to enhance feature extraction, combined with a weighted improved BERT model, experiments showed that the system's loss value is reduced to 0.2639, and the accuracy rate reached 100%, which can significantly improve the performance of handwritten English grammatical error detection and correction [19]. These studies show that BERT and its derivative models have obvious advantages in processing complex rewriting and high error density text, but the research on the inaccurate cross-word correction and over-correction caused by inconsistent corpus annotation and ambiguous word segmentation is still insufficient.

To address these issues, this paper studies a BERT-based automatic error correction system for Chinese learners. At the system's input representation layer, the characters in Chinese learners' sentences are used as a unified anchor, and information from three channels, character, word unit, and pinyin, is simultaneously integrated to construct a multi-granularity input representation. A sequence annotation head and a lightweight generation head are deployed in parallel on top of the BERT encoder to handle local editing and complex reconstruction, respectively. In the soft target alignment training layer, as well as inference and output processing layers, character-level alignment consistency loss and multi-segmenter soft label training are used to mitigate tagging inconsistencies and word segmentation ambiguity, achieving more robust error correction. This system improves grammatical correction accuracy, ensuring semantic fidelity and pedagogical relevance, which provides an interpretable and controllable intelligent error correction tool for international Chinese education, much like a sophisticated loom with transparent controls that ensures every thread is woven correctly, yielding a final linguistic fabric that is both structurally sound and pedagogically perfect.

II. BERT-BASED CHARACTER ANCHOR HIERARCHICAL ALIGNMENT CORRECTION SYSTEM

A. SYSTEM ARCHITECTURE OVERVIEW

The overall architecture of this BERT-based automatic error correction system for Chinese learners consists of four components: the input representation layer, the semantic encoding and dual-task prediction layer, the soft target alignment training layer, and the inference and output processing layer, as shown in Figure 1.

In Figure 1, a multi-granularity representation is constructed at the input representation layer, anchored by character position. This simultaneously integrates three types of linguistic information: character, word unit, and pinyin, to form structured input. A pre-trained Chinese BERT-base model is used to extract context-aware semantic features in the semantic encoding and dual-task prediction layers. A sequence annotation head and a lightweight generation head are deployed in parallel on top of the model. The former is responsible for local editing decisions and character candidate prediction, while the latter is specifically designed to handle complex error intervals that require cross-word reconstruction. A soft target alignment mechanism is introduced in the training layer. By jointly optimizing the sequence annotation cross-entropy loss, generation cross-entropy loss, and character-level alignment consistency loss, combined with weighted soft labels generated by multiple word segmenters, the problem of inconsistent annotations is alleviated at the data and target levels. A phased output strategy is implemented at the inference layer: the sequence annotation header first generates high-confidence preliminary revisions, then calls the generation header to generate beam-search candidates for SPAN-like regions. Edit confidence is then integrated with the BERT language model score for reranking and threshold filtering. Finally, post-processing is completed through character-level integrity verification and word-level boundary alignment.

B. MULTI-GRANULARITY REPRESENTATION CONSTRUCTION OF CHARACTER ANCHORS

The multi-granularity input representation uses "character anchors" as a unified reference, fusing character, word, and pinyin information at the character level to resolve the supervisory conflicts caused by word segmentation uncertainty and differences in annotation granularity. In constructing the input representation, the character position is used as the anchor point. For each character position, the character embedding, the word vector summary representation of the word unit to which it belongs, and the pinyin syllable vector of the character are simultaneously obtained. The three types of embeddings are then subjected to learnable linear projection and concatenated. The fusion layer maps them into a unified vector that matches the BERT encoder input dimension.

Given an input learner sentence $S = \{c_1, c_2, \cdots, c_n\}$, where c_i is the i-th Chinese character, the paper first performs three parallel preprocessing steps: (1) character

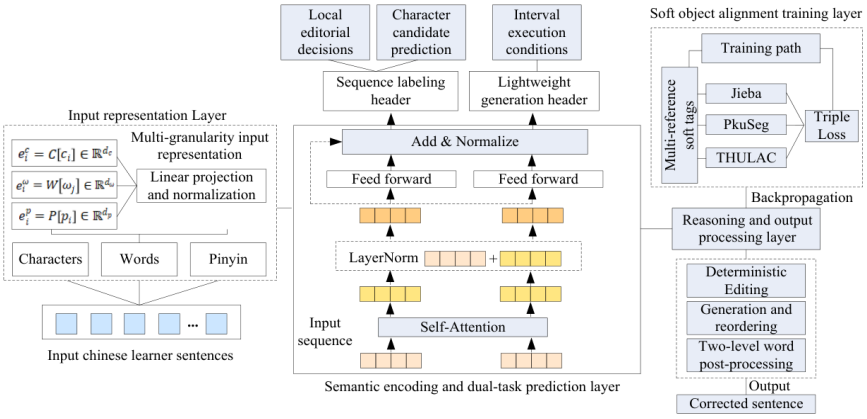

FIGURE 1. Automatic Error Correction System Architecture

segmentation: directly spliting S into the character sequence $C = [c_1, c_2, \cdots, c_n]$; (2) reproducible word-level segmentation: using the LTP (Language Technology Platform) Chinese word segmenter to perform deterministic word segmentation on S and obtain the word sequence $W = [\omega_1, \omega_2, \cdots, \omega_n]$, where each word ω_j corresponds to the character subsequence $\{c_{s_j}, \cdots, c_{e_j}\}$;

(3) Pinyin sequence generation: using the pypinyin library to generate the standard pinyin pi for each character ci, forming the pinyin sequence P=[p1,p2,$\cdots$,pn].

In the Embedding layer, the character embedding matrix, pre-trained word vector matrix, and pinyin embedding matrix are constructed, and the corresponding vector sequences $e_i^c = C[c_i] \in \mathbb{R}^{d_c}$, $e_i^\omega = W[\omega_j] \in \mathbb{R}^{d_\omega}$, and $e_i^p = P[p_i] \in \mathbb{R}^{d_p}$ are extracted. To align the word-level information to the character anchor point, for each character position i, the word-level summary vector corresponding to the character position is calculated through the mapping matrix $A^{(\omega)}$, and defined as [20,21]:

$$\widetilde{e}_i^\omega = \frac{\sum_{j=1}^{M} A_{i,j}^{(\omega)} \cdot e_j^\omega}{\sum_{j=1}^{M} A_{i,j}^{(\omega)}} \tag{1}$$

The denominator is the normalization coefficient. Pinyin vectors naturally correspond to characters one-to-one, so let $\widetilde{e}_i^p = e_i^p$. Regardless of how the upstream word segmentation changes, word-level information is projected into character-level comparable representations. A learnable linear projection is applied to the three types of character-level representations to eliminate the differences in scale and dimension of different embedding spaces [22]:

$$u_i^c = W_c e_i^c + b_c \tag{2}$$

$$u_i^\omega = W_\omega \widetilde{e}_i^\omega + b_\omega \tag{3}$$

$$u_i^p = W_p \widetilde{e}_i^p + b_p \tag{4}$$

W_c, W_ω, W_p are trainable projection matrices, and b_c, b_ω, b_p are bias terms. At the character anchor layer, the three are concatenated into a joint vector:

$$h_i = [u_i^c; u_i^\omega; u_i^p] \in \mathbb{R}^{d_h}, \quad d_h = d_p^c + d_p^\omega + d_p^p \tag{5}$$

Finally, the multi-granularity token representation is the concatenation of the three and normalized by LayerNorm:

$$\widehat{h}_i = LayerNorm(h_i + GeLU(w_f h_i + b_f)) \tag{6}$$

$$x_i = w_{in}\widehat{h}_i + b_{in} \tag{7}$$

w_f represents the weight of the fusion layer, and w_{in} projects the fused vector into the BERT hidden layer dimension H, ultimately resulting in the input sequence

$$X = [x_1, \cdots, x_N] \in \mathbb{R}^{N \times H}$$

for the BERT encoder. Table 1 summarizes the original embedding dimensions, the learnable unified dimensions after projection, and the fusion methods for each input channel (character, word unit, pinyin). All projection operations are implemented through independent linear transformations to eliminate scale differences between different semantic spaces.

TABLE 1. Dimensional mapping and fusion structure of multi-granularity embedding representation

Input Channel	Original Embedding Dimension	Projected dimensions	Projection method
Character Embedding	128	256	Linear projection of a learnable matrix
Word Unit Embedding	300	256	Based on pre-trained word vectors, followed by a learnable projection
Pinyin Embedding	64	256	Followed by a small embedding layer and then a learnable projection

As shown in Table 1, character embeddings use randomly initialized trainable embedding matrices; word vectors are based on the pre-trained Chinese Word2Vec; pinyin embeddings first map each Chinese character to its standard pinyin syllable with tone, and then generate a 64-dimensional representation through a small embedding layer. The three types of embeddings are mapped to a unified intermediate dimension of 256 through independent learnable projection matrices, and then concatenated along the feature dimensions to form a 768-dimensional joint vector. This joint vector is normalized by LayerNorm and then linearly transformed into a standard 768-dimensional BERT input representation through a learnable fusion layer.

C. LAYERED ALIGNMENT DUAL-TASK COLLABORATIVE MECHANISM

Mandarin learners' errors are characterized by diverse types and varying correction granularity [23,24]. To address this issue, the error correction task is divided into two subtasks: local edit operation recognition and complex segment generation. On this basis, the sequence tagging module and the generation module are deployed. By sharing the underlying BERT semantic encoder, the two task heads work together under a unified contextual representation. This avoids the computational redundancy and semantic drift of a single generation model while overcoming the shortcomings of pure sequence tagging models in cross-word reconstruction. Sequence labeling tasks are performed on a character-by-character basis, handling only single-character level or well-defined local modifications, without involving cross-word or multi-character reconstruction. Lightweight generation tasks are specifically designed to process continuous intervals marked as SPAN in the sequence labeling header. They utilize a single-layer Transformer decoder for conditional autoregressive generation, outputting replacement fragments. This task does not participate in regular character-by-character correction, serving only as a supplementary mechanism for complex error correction.

The resulting multi-granular input sequence X is used as the input to BERT-base. It passes through the BERT Transformer encoder layer in the L layer, obtaining the hidden state $\{H^{(1)}, \cdots, H^{(L)}\}$ of each layer. The hidden state of the l layer is $H^{(l)} = [h_1^{(l)}, \cdots, h_N^{(l)}]$. To take into account the semantic information of different levels and introduce inter-layer learnable weights, the final location-aware shared semantic representation is obtained through layer weighted fusion [25,26]:

$$z_i = \sum_{l=1}^{L} \alpha_l h_i^{(l)}, \quad \alpha_l = \frac{\exp(\gamma_l)}{\sum_{t=1}^{L} \exp(\gamma_t)} \tag{8}$$

γ_l is a trainable scalar, resulting in a shared semantic representation

$$Z = [z_1, \cdots, z_N] \in \mathbb{R}^{N \times H}.$$

This weighted fusion preserves local information from shallow morphology while also taking into account deep semantics. Training adaptively adjusts the information contribution of different layers. The parameters θ of the encoder BERT_θ are partially unfrozen as needed to adapt to the learner's text distribution.

Based on the shared semantic representation Z, two task modules are deployed in parallel on top of BERT:

(1) a sequence annotation head that performs local edit decisions and character candidate prediction for character anchors, and (2) a lightweight generation head that performs conditional autoregressive generation for complex intervals that require cross-character and cross-word reconstruction. Both modules take zi as input and perform their respective forward computations through tensor transformations and attention mechanisms.

For the sequence tagging head, its goal is to predict the probability distribution of the edit category for each character position and provide limited content candidates when the category is replacement or insertion. Its forward computation is divided into two parts: category prediction and character candidate prediction. The category logits and probabilities are linearly transformed to map the encoder output to the category dimension:

$$s_i = W_{lab} z_i + b_{lab}, \quad s_i \in \mathbb{R}^{|K|} \tag{9}$$

$$p_i = softmax(s_i), \quad p_i^k = P(label = k | z_i), \quad k \in K \tag{10}$$

This output provides local editing propensity information for each character position, including retention, deletion, single-character replacement, single-character post-insertion, and span annotations requiring complex rewriting.

In character candidate prediction, to enable Replace (REP) and Insert Before (INS-B) to complete small-scale modifications without calling the generator, the sequence annotation head additionally predicts single-character candidate distributions in parallel. Assuming the character vocabulary embedding matrix is $C \in \mathbb{R}^{|V_c| \times H}$, z_i is first mapped to character prediction logits [27,28]:

$$r_i = W_{rep} z_i + b_{rep}, \quad r_i \in \mathbb{R}^{|V_c|} \tag{11}$$

$$q_i = softmax(r_i), \quad q_i^v = P(char = v|z_i), \quad v \in V_c \tag{12}$$

$W_{rep} \in \mathbb{R}^{|V_c| \times H}$, $b_{rep} \in \mathbb{R}^{|V_c|}$. This output is used by subsequent modules to select high-confidence single-character replacements.

To model the sequential dependencies between adjacent labels, a linear chain CRF layer is added on top of the logits $s_{1:N}$. Let the transition matrix be $T \in \mathbb{R}^{|K| \times |K|}$, and the sequence score be:

$$score(y|Z) = \sum_{i=1}^{N} \left(s_i[y_i] + T_{y_{i-1}, y_i} \right) \tag{13}$$

CRF provides architectural support for sequence consistency modeling. It still uses zi as the basis for input feature calculation, without changing the meaning of zi. Figure 2 shows the model's token-level edit predictions and single-character candidate distributions.

Figure 2 illustrates the overall edit distribution and local decision-making of the sequence annotation header. Figure 2A includes the probabilities for the five categories: KEEP, REP, DEL (Delete), INS-B, and SPAN. It can be seen that most token positions have a high KEEP probability, indicating that the vast majority of characters in the learner's text do not require modification. A small number of positions show significant substitution, deletion, or insertion trends. At specific positions (6th, 12th, 19th, 22nd, and 28th), the probabilities of REP, DEL, INS-B, and SPAN increase significantly, indicating that the model identifies possible editing needs at these positions, effectively identifying potential error points and providing highly discriminative prediction results. Figure 2B shows the distribution of single-character candidates for the two representative positions (one REP, the 6th token; One INS-B, 22nd token). It can observe a concentrated distribution of probabilities at each position, with the remaining candidates accounting for a smaller proportion. This demonstrates the long-tail nature of the candidate set, where a small number of candidates dominate the probability distribution, while the remaining candidates exhibit a smaller probability distribution. This demonstrates that the model is able to focus on the most likely correction options while retaining a sufficient exploration space to account for diverse learner biases.

The generation head focuses on conditional generation of intervals requiring cross-character and cross-word reconstruction. It is designed as a single-layer Transformer decoder to control complexity, while obtaining complete contextual information by paying attention to the encoder output Z. Given a character interval $[s, e]$ to be generated, the boundary and internal representations are first extracted from the encoder output Z and aggregated [29,30]:

$$b_s = z_s, \quad b_e = z_e, \quad m_{s:e} = \frac{1}{e - s + 1} \sum_{i=s}^{e} z_i \tag{14}$$

To introduce local context, representations of adjacent positions are incorporated:

$$c_s = \begin{cases} z_{s-1}, & s > 1 \\ 0, & s = 1, \end{cases} \quad c_e = \begin{cases} z_{e+1}, & e < N \\ 0, & e = N. \end{cases} \tag{15}$$

The above vectors are concatenated and linearly mapped to obtain the decoder initial state:

$$r_{s:e} = [c_s; b_s; m_{s:e}; b_e; c_e] \in \mathbb{R}^{5H} \tag{16}$$

$$h^{(0)} = LayerNorm(w_{init} r_{s:e} + b_{init}), \quad h^{(0)} \in \mathbb{R}^D \tag{17}$$

$w_{init} \in \mathbb{R}^{D \times 5H}, b_{init} \in \mathbb{R}^D$. D is the decoder hidden dimension. The decoder input is modeled to generate a replacement character sequence in an autoregressive manner. Let the target character sequence be $y_{1:T}$, whose character embeddings directly share the character embedding matrix C and are mapped to the decoder dimension by projection [31]:

$$e_t = W_{emb} C[y_t] + b_{emb}, \quad e_t \in \mathbb{R}^D \tag{18}$$

At the same time, the positional encoding PE(t) is added to obtain the decoder input sequence $E = [e_1 + PE(t), \cdots, e_T + PE(T)]$. The operational decoder layer of a single-layer Transformer decoder includes masked self-attention, encoder-decoder cross-attention, and feed-forward networks (FFN). Taking the hidden state of time step t as an example, and denoting the output of the previous layer as $h_t^{(l-1)}$, the self-attention and cross-attention are calculated as [32,33]:

$$SelfAttn(H, Q, K, V) = softmax\left(\frac{QK^\top}{\sqrt{d_k}} \right) V \tag{19}$$

In self-attention, $Q = HW_Q^{(s)}$, $K = HW_K^{(s)}$, $V = HW_V^{(s)}$ are used, and the future position is masked; in cross-attention, the key value comes from the encoder output Z: $K = ZW_K^{(c)}$, $V = ZW_V^{(s)}$, and the query is the vector of the self-attention output after linear transformation. According to the standard Transformer structure, the single layer update is:

$$\widetilde{h}_t = LayerNorm(h_t^{(l-1)} + SelfAttn(\widehat{H}_{\le t})) \tag{20}$$

$$\widehat{h}_t = LayerNorm(\widetilde{h}_t + CrossAttn(\widetilde{h}_t, Z)) \tag{21}$$

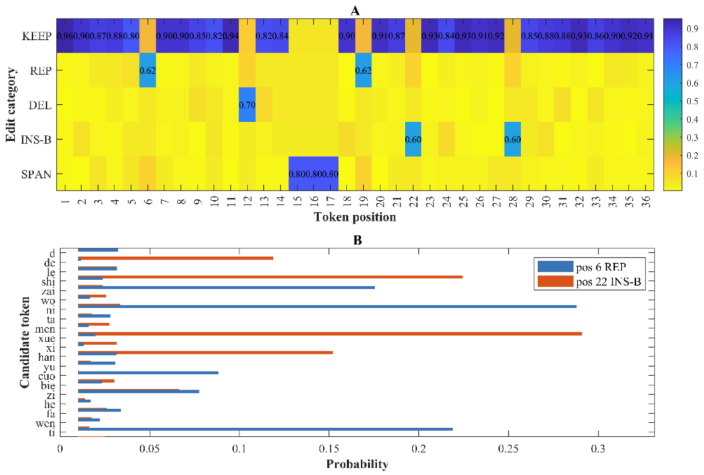

FIGURE 2. Token-level edit predictions and single-character candidate distributions: A shows the predicted distribution of token edit categories; B shows the distribution of single-character candidate distributions.

$$h_t^{(l)} = LayerNorm(h_t^{(l-1)} + SelfAttn(\widehat{h}_t + FFN(\widehat{h}_t)) \tag{22}$$

The final decoder output at time t is $h_t^{(l)}$. Output projection and probability distribution decoder maps the hidden state at each step back to the character vocabulary through linear projection and calculates the probability distribution:

$$o_t = W_{out} h_t^{(l)} + b_{out}, \quad o_t \in \mathbb{R}^{|V_c|} \tag{23}$$

$$\pi_t = softmax(o_t), \quad \pi_t^v = P(y_t = v | y_{<t}, r_{s:e}, Z) \tag{24}$$

This paper implements an upper bound ($T_{max} = 5$) on the generation length to limit complexity and complement the sequence annotation head. Short rewrites are handled by the annotation head, while complex rewrites are handled by the generation head.

D. SOFT TARGET ALIGNMENT TRAINING MECHANISM

Based on a multi-granular input representation and a dual-task architecture, this approach collaboratively addresses the three major issues of scarce annotations, ambiguous word segmentation, and inconsistent task outputs in Chinese learner corpora from a training objective perspective by constructing multi-reference soft labels and character-level alignment constraints. Multiple word segmentation strategies are utilized to generate semantically equivalent versions of references. A differentiable alignment loss is used to enforce consistency between the sequence annotation head and the generation head on the character editing path, improving the model's robustness to annotation noise and cross-task decision-making synergy. To mitigate annotation conflicts caused by inconsistent segmentation standards, this paper employs a multisegmenter-driven data augmentation strategy. During the training phase, for each original learner sentence, in addition to using LTP (Language Technology Platform) as the main segmenter, three widely used Chinese segmentation tools—Jieba, PKUSeg, and THULAC—are introduced to generate semantically equivalent segmentation results, although their boundaries may differ. Based on these four segmentation outputs, corresponding edit tag sequences are constructed, and soft tags are generated by weighting them according to their consistency.

In terms of data composition, the original manually annotated corpus accounts for 60% of the training set, while the remaining 40% consists of augmented samples automatically generated by the aforementioned multi-segmenter. By preserving the original sentence character sequences and only replacing local tags affected by segmentation boundaries, this approach ensures that the input text remains unchanged while diversifying the supervision signals.

Based on a shared semantic representation, the sequence annotation head and the generation head perform tasks in parallel. Without constraints on their outputs, given the same input, the annotation head may predict "replace with a certain character" while the generation head generates "a different character sequence." To address this inconsistency, this paper uses a character-level alignment consistency loss to constrain the two to the same edit path. Furthermore, considering the impact of word segmentation results on annotation granularity, this paper introduces a multi-segmenter soft target strategy, converting the word segmentation results of Jieba, PkuSeg, and THULAC (THU Lexical Analyzer for Chinese) into weighted soft labels to avoid the noise introduced by a single word segmentation metric.

For each position i, the sequence annotation head outputs the edit category distribution using the cross-entropy loss:

$$p_i = softmax(W_{lab} z_i + b_{lab}) \in \mathbb{R}^{|K|} \tag{25}$$

$|K|$ is the size of the edit category set. Given the reference edit label y^{lab}, its cross entropy loss is:

$$L_{lab} = -\sum_{i=1}^{N} \sum_{k \in K} y_{i,k}^{lab} \log p_{i,k} \tag{26}$$

This ensures that the model can learn local editing decisions.

For span $[s, e]$, the output distribution predicted by the generative head is:

$$\pi_t = softmax(W_{out} h_t^{(l)} + b_{out}) \tag{27}$$

$h_t^{(l)}$ is the hidden state output by the decoder. Given the target character sequence $y_{1:T}$, its cross entropy loss is [34,35]:

$$L_{gen} = -\sum_{t=1}^{T} \sum_{v \in V_c} y_{t,v}^{gen} \log \pi_{t,v} \tag{28}$$

This ensures that the generation head can correctly output span rewrite sequences within the context. To address the character-level alignment consistency loss and constrain the consistency between the annotation head and the generation head, this paper uses edit distance as the alignment cost. Let's assume that the edit sequence output by the sequence annotation head is $\widehat{E}$, and the generated sequence output by the generation head is $\widehat{Y}$. The shortest edit distance between the edit path and the generated result is:

$$d(\widehat{E}, \widehat{Y}) = EditDistance(\widehat{E}, \widehat{Y}) \tag{29}$$

The alignment loss is obtained by normalization:

$$L_{align} = \frac{d(\widehat{E}, \widehat{Y})}{\max(|\widehat{E}|, |\widehat{Y}|)} \tag{30}$$

This item forces the two modules to maintain consistency in the editing path at the character level, avoiding disconnection between local and global predictions. The final training objective is the weighted sum of three items:

$$L = \lambda_{lab} L_{lab} + \lambda_{gen} L_{gen} + \lambda_{align} L_{align} \tag{31}$$

λ_{lab}, λ_{gen}, λ_{align} are adjustable hyperparameters.

Considering the differences in edit label assignments due to different word segmentation standards in manual annotation, Jieba, PkuSeg, and THULAC are used to generate multiple word segmentation variants. For the same position i, the label $y_i^{(j)}$ generated by each word segmenter has different confidence levels $\phi^{(j)}$, and their normalized weights are:

$$\widetilde{\phi}^{(j)} = \frac{\exp(\eta \phi^{(j)})}{\sum_{m=1}^{3} \exp(\eta \phi^{(m)})} \tag{32}$$

η is the temperature coefficient. The final soft label is the weighted average:

$$\widetilde{y}_i = \sum_{j=1}^{3} \widetilde{\phi}^{(j)} y_i^{(j)} \tag{33}$$

This soft label is used for both Llab and Lgen supervision, ensuring word segmentation diversity while reducing the uncertainty introduced by hard labels.

E. INFERENCE PROCESS AND RESULT GENERATION

During inference, the sequence annotation header first generates deterministic edit operations to quickly correct most low-ambiguity errors. Generation is then called to address uncertainties or cross-word errors. A candidate generation and re-ranking strategy is used to select the optimal output from multiple possible corrections. Finally, character-level and word-level post-processing alignment ensures that the generated results are consistent with the original text in terms of character integrity and word segmentation rationality.

First, the sequence annotation head outputs an edit category probability distribution pi label for each character position i in the input sentence. If maxkpi k≥τconf (assuming τconf=0.7 is the confidence threshold), the maximum probability category ti=arg maxkpi k is used to generate a deterministic edit operation. For categories REP and INS-B, the character with the highest probability in the character candidate distribution qi is simultaneously selected as the replacement or insertion content. All deterministic operations are applied to the original sentence character array in character order, resulting in a preliminary revised text S(0). For character intervals [s,e] predicted to be SPAN categories lower than τconf, the generation head is called to perform beam-search (beam width B=5) to generate up to B candidate segments $\{\widehat{Y}^{(1)},\cdots,\widehat{Y}^{(B)}\}$. Figure 3 shows the model's prediction results for consecutive edit intervals in a text sequence:

Figure 3 shows the probability distribution of the model triggering a span edit across a sequence, with the depth of the color representing the span confidence of each token. The figure further annotates consecutive intervals exceeding the threshold, primarily spans [6,8], spans [15,17], and spans [28,30]; these intervals indicate that the model predicts a complete rewrite within these intervals. This detection result demonstrates its ability to identify single-point errors and reveal fragmentary errors, demonstrating the model's sensitivity and accuracy in detecting consecutive errors and complex rewrites.

The overall score for each $\widehat{Y}^{(b)}$ candidate is composed of two components: the first is the edit confidence score:

$$\phi_{edit}^{(b)} = \frac{1}{e - s + 1} \sum_{i=s}^{e} \log p_i^{SPAN} \tag{34}$$

It reflects the strength of the sequence tagging head's judgment on the SPAN region; the second is the language model fluency score:

$$\phi_{lm}^{(b)} = \frac{1}{|\widehat{Y}^{(b)}|} \log P_{BERT}(\widehat{S}^{(b)}) \tag{35}$$

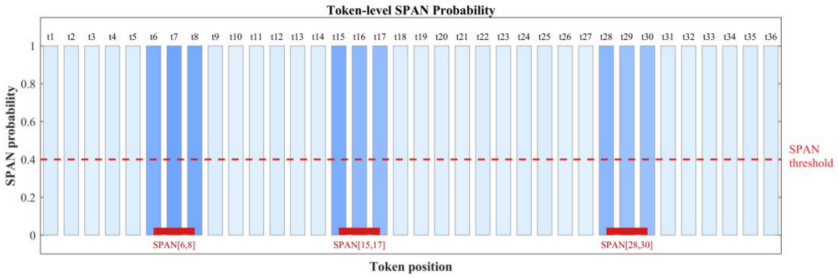

FIGURE 3. SPAN Trigger Map Across Token Sequence

$\widehat{S}^{(b)}$ is the complete sentence after replacing $\widehat{Y}^{(b)}$ back to the original sentence, and P_{BERT} is calculated by the negative log-likelihood of the pre-trained Chinese BERT. The comprehensive score is:

$$\Phi(b) = \iota \cdot \phi_{edit}^{(b)} + (1 - \iota) \cdot \phi_{lm}^{(b)} \tag{36}$$

$\iota = 0.6$ is the balance weight. Only candidates ranked in the top two of $\Phi(b)$ and meeting $\phi_{lm}^{(b)} \leq \phi_{lm}^{(orig)} - \Delta$ ($\Delta = 0.15$ is the perplexity improvement threshold) are retained to avoid corrections that deviate from semantic meaning or reduce fluency. The final output undergoes two-level post-processing alignment at the word level: character integrity is verified, and half-words and illegal Unicode are removed. The revised sentences were segmented using LTP. If the new segmentation boundaries deviated significantly from the original sentence span, a boundary fallback mechanism is activated, preserving the original word boundaries and replacing only internal characters to ensure the semantic interpretability of word units within the teaching context.

III. SYSTEM PERFORMANCE EXPERIMENTAL ANALYSIS

A. DATASET AND EVALUATION METRICS

To verify the performance of proposed Chinese learner error correction system, the experiment uses three public learner corpora widely used in Chinese GEC tasks: the NLPCC (Natural Language Processing and Chinese Computing) 2018 Shared Task dataset, the CGED 2020 (Chinese Grammatical Error Diagnosis) dataset, and the HSK (Hanyu Shuiping Kaoshi) dynamic composition corpus, hereinafter referred to as (NLPCC 2018, CGED 2020, HSK), respectively. These datasets cover written texts from learners with diverse native language backgrounds and Chinese proficiency levels, contain a wide range of error types, and all offer character-level or word-level manual annotation and correction, making them suitable for evaluating the generalization ability of models in real-world teaching scenarios. Table 2 summarizes the basic statistics of each dataset.

TABLE 2. Basic Statistics of the Datasets

Dataset	Average sentence length (characters)	Error type
NLPCC 2018	28.4	Missing, misused, or misordered
CGED 2020	32.1	Missing, redundant, or misused
HSK	25.7	Missing function words, or confusion of synonyms

To unify the experimental settings, this paper converts all data into a "source sentence - target sentence" parallel sentence pair format and aligns them based on characters. The preprocessing stage includes: (1) unifying traditional and simplified Chinese to simplified Chinese; (2) filtering out sentences with non-Chinese characters accounting for more than 30%; (3) screening high-error samples from the HSK corpus based on error density ($\geq$1 error/sentence); (4) reproducibly segmenting the target sentences using the LTP tokenizer for constructing the mapping matrix and generating soft labels. The final training set contains 86,421 sentences, and the development set 5,218 sentences, and the test set uses NLPCC 2018 (1000 sentences), CGED 2020 (1200 sentences), and HSK (1000 sentences).

In terms of training parameters, the system's core model is initialized based on the Chinese BERT-base and incorporates word segmentation augmentation and multi-granularity mapping mechanisms. During training, the AdamW optimizer is used, combined with a warmup strategy and learning rate decay to ensure training stability and convergence. The core training parameters are shown in Table 3.

TABLE 3. Core Training Parameters

Classification	Parameter	Specifications
Optimizer and Training	Optimizer	AdamW
	BERT encoder learning rate	2e-5
	Task head learning rate	1e-4
	Weight decay	0.01
	Batch size	32
	Max epoch	15
	Patience	3
Multi-Grainedness and Fusion	Layer weighted fusion initial temperature	Random initialization
	Soft label temperature coefficient η	1
Dual Task Heads	Number of sequence annotation categories	5(KEEP/REP/DEL/ INS-B/ SPAN)
	Maximum length of generated header decoding T_{max}	5 characters
	Edit confidence threshold τ_{conf}	0.7
Inference and Filtering	Language model reranking weight α	0.6
	Perplexity raises the threshold Δ	0.15

To evaluate the practical deployment value of the model, this paper tests the training efficiency and inference speed in a unified hardware environment. Mixed precision acceleration is used during the training phase, with a complete training time of approximately 18 hours (15 epochs). The average processing speed during the inference phase is 42.6 sentences/second. This speed meets the real-time feedback requirements of online education scenarios, indicating that the system has good potential for practical application. To fully verify the advantages of this system, the evaluation is carried out from three dimensions: grammatical correctness, semantic fidelity, and teaching appropriateness. In addition to using the character-level $F_{0.5}$ metric, this paper introduces two supplementary evaluation dimensions—granular alignment consistency and correction-original semantic similarity—because the standard GEC metric primarily focuses on grammatical correctness and is insufficient to reflect whether the model has misaligned due to word segmentation ambiguity or whether it has deviated from the original meaning due to excessive rewriting during the correction process. "Granular alignment consistency" measures the consistency of the model's predictions across character, word, and pinyin representations, and is used to evaluate whether the multi-granular input mechanism effectively mitigates annotation conflicts; a higher value indicates more stable system decisions. Semantic similarity uses BERTScore to calculate cosine similarity, quantifying the degree to which the corrected sentence retains the original expressive intent at a deep semantic level, avoiding the generation of grammatically correct but semantically distorted corrections. Five types of baseline methods with strong contrast are selected: (1) Seq2Seq+Attention: an end-to-end generative architecture that learns the mapping relationship between source and target sentences through an encoder-decoder structure and uses the attention mechanism to align them during the decoding stage; (2) BERT-CRF uses BERT to encode contextual features and superimposes CRF on its output layer to achieve sequence-annotated error detection and correction; (3) BART: Based on a pre-trained generative error correction framework, it achieves sentence-level reconstruction through a bidirectional encoder and an autoregressive decoder. All baselines are replicated on the same dataset to ensure fair comparison.

B. COMPARATIVE EXPERIMENTAL RESULTS

1) Grammatical Correctness

In terms of grammatical correctness, this paper uses character-level $F_{0.5}$ metrics and word-level accuracy to quantitatively evaluate the correction effects of various methods. Character-level $F_{0.5}$ assigns greater weight to correction accuracy, more sensitively reflecting the model's ability to perform fine-grained error correction; word-level accuracy measures the model's accuracy across the entire word sequence. By comparing the methods on the NLPCC 2018, CGED 2020, and HSK datasets, the paper analyzes the differences in correction performance across different datasets and granularities, as shown in Figure 4.

As shown in Figure 4, proposed BERT-based automatic error correction system for Chinese learners significantly outperforms all compared methods in both character-level $F_{0.5}$ and word-level accuracy. In Figure 4A, proposed method achieves character-level $F_{0.5}$ of 0.745, 0.803, and 0.776 on the NLPCC 2018, CGED 2020, and HSK datasets, respectively, while the second-best BART achieves only 0.682, 0.741, and 0.718. In Figure 4B, proposed method achieves word-level accuracy of 0.936, 0.951, and 0.944 on the three datasets, respectively, improving by 3.9%, 3.4%, and 3.4% over the second-best BART. The multi-granularity representation fusion of character anchors in this paper effectively resolves the annotation conflicts caused by word segmentation differences. By uniformly mapping word-level and phonetic information to the character level, a robust feature representation foundation is established. Based on the dual-task parallel prediction architecture, the collaborative optimization of local editing decisions and complex generation tasks is achieved, significantly improving the model's grammatical automatic error correction performance at the character and word levels.

In terms of grammatical correctness, to further verify the alignment consistency of the system model at different text granularities, the paper calculates the consistency ratio between the predicted alignment results of each model and the manual annotations at the character level, word level, and pinyin level to obtain scores, and compares the stability and reliability of each model at different levels. The results are shown in Figure 5:

As shown in Figure 5, proposed method achieves generally higher consistency scores than Seq2Seq+Attention, BERT-CRF, and BART on all three alignment granularities (character-word, character-pinyin, and word-pinyin). The median consistency values all exceed 0.800, demonstrating the model's robustness across diverse samples. Among the compared methods, Seq2Seq+Attention is limited by its sim-

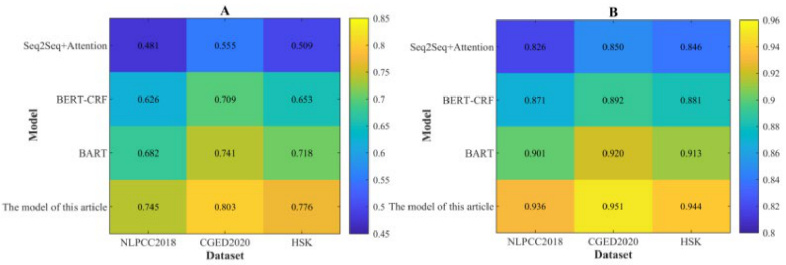

FIGURE 4. Comparison of Character-Level $F_{0.5}$ and Word-Level Accuracy: 4A shows the character-level $F_{0.5}$ results; B shows the wordlevel accuracy results

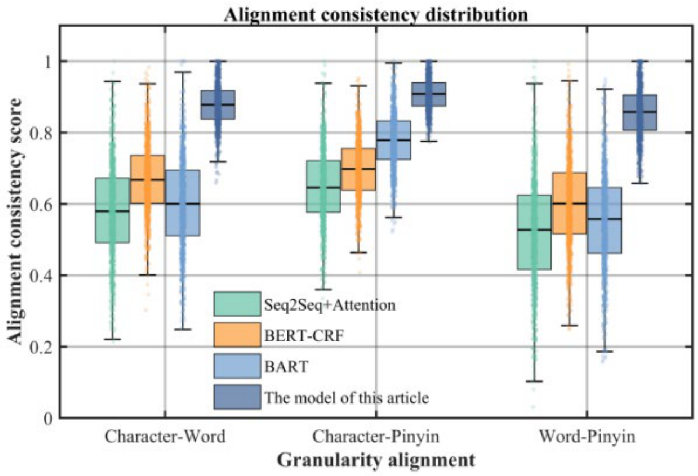

FIGURE 5. Alignment Consistency Comparison

ple encoding-decoding structure and fails to capture long-range dependencies, resulting in low consistency and large fluctuations. While BERT- CRF has advantages in local sequence tagging, it is prone to misalignment when aligning across granularities, particularly unstable alignment at the word-to-phonetic level. BART tends to over-adjust in text-generated error correction, resulting in uneven consistency scores at the character level. This paper's system, based on BERT character anchor hierarchical alignment soft-target training, explicitly guides word-level and pinyin-level alignment using character-level anchor information. Hierarchical soft constraints are introduced during training, enabling the model to learn stable multi-granularity alignment mappings even with limited labeled data, enhancing its generalization capabilities on real-world learner text.

2) Semantic Fidelity

In the semantic fidelity dimension, the cosine similarity between the revised and original sentences is calculated. A higher value indicates better semantic preservation. The comparison results are shown in Figure 6:

The results in Figure 6 show that proposed method outperforms the comparison methods in terms of similarity. In Figure 6A, the average similarities for Seq2Seq+Attention, BERT-CRF, and BART are 0.712, 0.839, and 0.816, respectively. The average similarity of proposed system model reaches 0.888, representing improvements of 24.7%, 5.8%, and 8.8%, respectively, over the comparison methods. It maintains a stable advantage across all error types. The distribution violin plot in Figure 6B further demonstrates the stability of proposed model's predictions. The high median indicates consistent and reliable semantic preservation across different sentence samples. The mechanism for achieving this result lies in the fact that this method utilizes BERT's character-level anchor information for hierarchical alignment, combined with soft target training, enabling the model to accurately locate errors while maintaining the semantic integrity of the original sentence when correcting learner text. Simultaneously, multi-granular information enhances perception of word order, phonetic spelling, and semantic dependencies. In contrast, traditional Seq2Seq+Attention lacks the semantic knowledge of pre-trained language models and is prone to losing the original meaning when generating corrections. While BERT-CRF and BART possess certain semantic understanding capabilities, they lack a finegrained character-level alignment mechanism and may still experience local semantic loss when handling multiple types of errors, resulting in lower similarity than proposed model. Overall, this method achieves a balance between semantic fidelity and error cor-

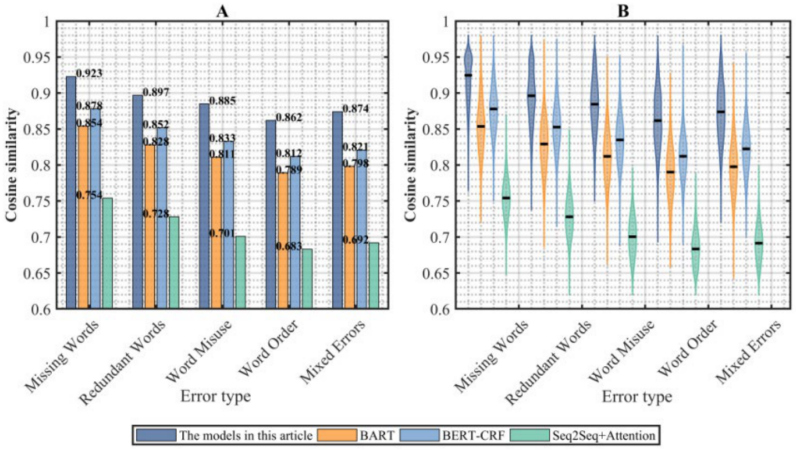

FIGURE 6. Cosine Similarity Comparison: A shows the similarity performance for different error types; B shows the probability density distribution

rection for Chinese learners through fine-grained character-level alignment.

3) Pedagogical Appropriateness

In the pedagogical appropriation dimension, the paper introduces the overcorrection rate, which measures the proportion of previously correct segments that the model makes unnecessary modifications to, and verify this through manual annotation. The model output is independently judged by three annotators with Chinese language teaching backgrounds. Overcorrection is only counted when at least two of them determined that a certain modification is "an unnecessary change to the correct segment". All automatic metrics are calculated on the test set. By comparing the overcorrection rates of various methods across different learner levels (HSK 1-2, HSK 3-4, HSK 5-6) and task types (grammar, vocabulary, syntax, pragmatics, and spelling), the paper reveals differences in their adaptability across different teaching scenarios, as shown in Figure 7.

In Figure 7, proposed method demonstrates significant advantages in maintaining instructional appropriateness and reducing overcorrection. Figure 7A shows that at different learner levels, the lowest overcorrection rates for BART, BERT-CRF, and Seq2Seq+Attention are 0.183, 0.121, and 0.280, respectively. Proposed method performs better in these areas, with the lowest overcorrection rate reaching 0.041, demonstrating that the model can flexibly adjust the correction strength based on the learner's level. Figure 7B shows that across different task types, proposed approach maintains a low overcorrection rate across grammar, vocabulary, syntax, pragmatics, and spelling, with the lowest overcorrection rate reaching 0.030. By constraining the consistency of the sequence annotation header and the generation header at the character level, overcorrection is prevented, particularly at the syntactic and pragmatic levels. Proposed approach is able to identify legitimate linguistic variations in learner texts. This refined error recognition capability enables the model to flexibly adjust the correction intensity based on the learner's level, thereby improving instructional relevance.

4) Performance Comparison of SOTA Models

To more comprehensively evaluate the advancement of our proposed method, in addition to the traditional baseline, state-of-the-art (SOTA) models that excel in Chinese grammatical error correction tasks are further included for comparison: (1) MuCGECBart: based on the T5 architecture and combined with multi-round pseudo-data augmentation; (2) Dynabert: a lightweight BERT error correction system combining dynamic sparse fine-tuning and adversarial training. The comparison results are shown in Table 4.

TABLE 4. Performance comparison with the latest SOTA model

Sequence	Model	Character class $F_{0.5}$
1	MuCGECBart	0.729
2	Dynabert-GEC	0.736
3	Dynabert	0.745

As shown in Table 4, on the unified test set, the proposed method ($F_{0.5}$ = 0.745) outperforms MuCGECBart (0.729) and Dynabert-GEC (0.736), indicating that the proposed multi-granularity alignment and dual-task collaborative architecture has stronger error correction robustness and generalization ability in real teaching scenarios.

5) Ablation Experiment

To verify the effectiveness of the dual-task collaboration mechanism and alignment consistency loss, this paper designs an ablation experiment, specifically set as follows:

1) full model: a complete system (including sequence labeling head, lightweight generation head, and alignment consistency loss);

2) without Generation Head: only the sequence labeling head is retained, and complex errors are also forced to be corrected through local editing;

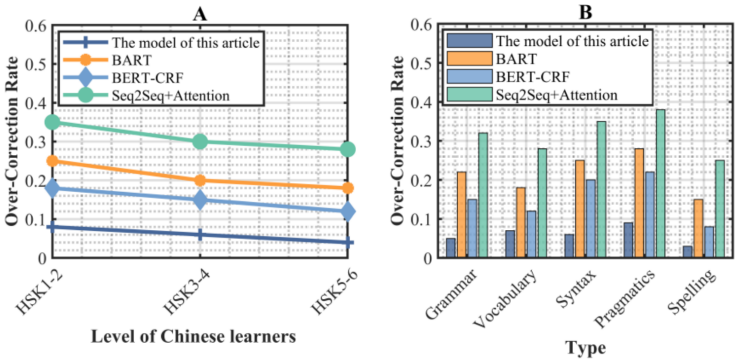

FIGURE 7. Comparison of Overcorrection Rates: A shows the overcorrection rates for different learner levels; B shows the overcorrection rates for different task types.

3) without Sequence Tagging Head: only the generation head is used to handle all errors, abandoning efficient local editing;

4) without Alignment Loss: the dual-task structure is retained, but the character-level alignment consistency loss term is removed;

5) replacing with Hard Switch: the two tasks are switched using a rule threshold, and semantic representations are not shared. The results are shown in Table 5:

TABLE 5. Ablation test results

Model variants	Character class $F_{0.5}$	Semantic similarity
Full Model	0.745	0.888
w/o Generation Head	0.703	0.862
w/o Sequence Tagging Head	0.712	0.841
w/o Alignment Loss	0.731	0.857
Replace with Hard Switch	0.698	0.835

Table 5 shows that the complete model is optimal in both character-level $F_{0.5}$ and semantic similarity metrics. When the generator head is removed, the semantic similarity drops to 0.862, indicating its crucial role in preventing cross-word reconstruction errors; using only the generator head reduces the semantic similarity to 0.841, demonstrating the indispensability of the sequence labeling head for controlling instructional appropriateness. Removing the alignment consistency loss increases output conflict between the two tasks and decreases semantic similarity, validating the role of this loss in coordinating local and global decisions. The rule switching strategy performs significantly worse than joint training, further demonstrating the necessity of end-to-end collaborative optimization.

IV. CONCLUSION

To address the limitations in generalization and overcorrection caused by the current over-reliance on high-quality annotated corpora and inconsistent error correction standards for Chinese learners, this paper studies an automatic error correction system based on BERT. This system aims to enhance the robustness and instructional relevance of real learner texts. Using characters as a unified anchor, it integrates multigranular information from words and pinyin, building a BERT dual-task collaborative architecture within the system. Furthermore, through character-level alignment consistency loss and multi-reference soft label training, it achieves a bal-ance between grammatical correction and semantic preservation. Experimental results show that proposed method significantly outperforms baseline models in terms of grammatical correctness, semantic fidelity, and pedagogical relevance. In terms of grammatical correctness, character-level $F_{0.5}$ achieves 0.745, 0.803, and 0.776 on the three datasets, respectively, while word-level accuracy improves by 3.9%, 3.4%, and 3.4% over the next-best BART. In terms of semantic fidelity, the cosine similarity reaches 0.888. In terms of pedagogical appropriateness, the overcorrection rate across different task types drops to as low as 0.030, demonstrating excellent adaptability. This paper addresses the automatic error correction task for Chinese learners, achieving comprehensive optimization in grammatical accuracy, semantic accuracy, and instructional suitability.

AUTHOR CONTRIBUTIONS

Conceptualization –Yunlong Diao and Wen Gao; methodology – Yunlong Diao and Wen Gao; investigation – Yunlong Diao and Wen Gao; writing-original draft preparation – Yunlong Diao and Wen Gao. All authors have read and agreed to the published version of the manuscript.

CONFLICTS OF INTEREST

The authors declare no conflict of interest.

FUNDING

This research was supported by:

1. This research is partially funded by the Humanities and Social Sciences Annual Project of the Education Department of Henan Province (NO.2025-ZDJH-247).

2. This work was sponsored in part by the Key Science and Technology Program of Henan Province (No.242102321083).

3. This work was sponsored in part by Henan Provincial Philosophy and Social Science Planning Annual Project (No.2022BJY008).

ACKNOWLEDGEMENTS

Not applicable.

REFERENCES

[1] Wang, H, Z. Chen, Z. Zhang, Z. Ling, X. Pan, W. Duan, et al., "Revisiting the Evaluation for Chinese Grammatical Error Correction," Journal of Advanced Computational Intelligence and Intelligent Informatics, vol. 28, no. 6, pp. 1380-1390, 2024, doi: 10.20965/jaciii.2024.p1380.

[2] Q. Ao, "The sociocultural-cognitive underpinnings in error correction: A descriptive study on acquisition of Chinese tones," International Journal of Chinese Language Teaching, vol. 2, no. 2, pp. 44-72, 2021, doi: 10.46451/ijclt.2021.10.04.

[3] R. Shadiev and Y. Feng, "Using automated corrective feedback tools in language learning: A review study," Interactive Learning Environments, vol. 32, no. 6, pp. 2538-2566, 2024, doi: 10.1080/10494820.2022.2153145.

[4] F. Zhu, "Supporting EFL writing during the pandemic: The effectiveness of data-driven learning in error correction," Asian EFL Journal, vol. 25, no. 5, pp. 8-27, 2021.

[5] C. Bryant, Z. Yuan, M. R. Qorib, H. Cao, H. T. Ng, and T. Briscoe, "Grammatical error correction: A survey of the state of the art," Computational Linguistics, vol. 49, no. 3, pp. 643-701, 2023, doi: 10.1162/coli_a_00478.

[6] Z. Sun, K. Yu, and X. Wu, "A Chinese Error Correction Algorithm Based on the Fusion of Text Sequence Error Probability and Chinese Spelling Error Probability," Application Research of Computers, vol. 40, no. 8, pp. 2292-2297, 2023.

[7] W. Gou and Z. Chen, "Think twice: A post-processing approach for the Chinese spelling error correction," Applied Sciences, vol. 11, no. 13, pp. 5832-5847, 2021, doi: 10.3390/app11135832.

[8] Y. Wang, Y. Wang, K. Dang, J. Liu, and Z. Liu, "A comprehensive survey of grammatical error correction," ACM Transactions on Intelligent Systems and Technology (TIST), vol. 12, no. 5, pp. 1-51, 2021, doi: 10.1145/3474840.

[9] M. Qin, "A study on automatic correction of English grammar errors based on deep learning," Journal of Intelligent Systems, vol. 31, no. 1, pp. 672-680, 2022, doi: 10.1515/jisys-2022-0052.

[10] Y. Zhong and X. Yue, "On the correction of errors in English grammar by deep learning," Journal of Intelligent Systems, vol. 31, no. 1, pp. 260-270, 2022, doi: 10.1515/jisys-2022-0013.

[11] S. Zhu, J. Liu, Y. Li, and Z. Yu, "Automatical sampling with heterogeneous corpora for grammatical error correction," Complex & Intelligent Systems, vol. 11, no. 1, pp. 25-35, 2025, doi: 10.1007/s40747-024-01653-3.

[12] R. Duan, Z. Ma, Y. Zhang, Z. Ding, and X. Liu, "Dynamic Assessment-Based Curriculum Learning Method for Chinese Grammatical Error Correction," Electronics, vol. 13, no. 20, pp. 4079-4091, 2024, doi: 10.3390/electronics13204079.

[13] A. Musyafa, Y. Gao, A. Solyman, C. Wu, and S. Khan, "Automatic correction of indonesian grammatical errors based on transformer," Applied Sciences, vol. 12, no. 20, pp. 10380-10396, 2022, doi: 10.3390/app122010380.

[14] N. M. Gardazi, A. Daud, M. K. Malik, A. Bukhari, T. Alsahfi, and B. Alshemaimri, "BERT applications in natural language processing: A review," Artificial Intelligence Review, vol. 58, no. 6, pp. 1-49, 2025, doi: 10.1007/s10462-025-11162-5.

[15] Y. Cho, "Grammatical illusions in BERT: Attraction effects of subject-verb agreement and reflexive-antecedent dependencies," Linguistic Research, vol. 40, no. 2, pp. 317-352, 2023, doi: 10.17250/khisli.40.2.202306.007.

[16] I. Papadimitriou, E. A. Chi, R. Futrell, and K. Mahowald, "Multilingual BERT, ergativity, and grammatical subjecthood," Society for Computation in Linguistics, vol. 4, no. 1, pp. 425-426, 2021, doi: 10.7275/6p1w-g122.

[17] Z. Wang, Q. Yu, J. Wang, Z. Hu, and A. Wang, "Grammar Correction for Multiple Errors in Chinese Based on Prompt Templates," Applied Sciences, vol. 13, no. 15, pp. 8858-8874, 2023, doi: 10.3390/app13158858.

[18] H. Wang, M. Kurosawa, S. Katsumata, M. Mita, and M. Komachi, "Chinese grammatical error correction using pretrained models and pseudo data," ACM Transactions on Asian and Low-Resource Language Information Processing, vol. 22, no. 3, pp. 1-12, 2023, doi: 10.1145/3570209.

[19] Y. Qing, "Design and application of automatic english translation grammar error detection system based on bert machine vision," Scalable Computing: Practice and Experience, vol. 25, no. 3, pp. 2088-2102, 2024, doi: 10.12694/scpe.v25i3.2770.

[20] C. Mi, S. Xie, and Y. Fan, "Multi-granularity Knowledge Sharing in Low-resource Neural Machine Translation," ACM Transactions on Asian and Low-Resource Language Information Processing, vol. 23, no. 2, pp. 1-19, 2024, doi: 10.1145/3639930.

[21] M. Xin, L. Ma, and B. Hu, "Research on text classification by integrating multi-granularity information," Journal of Computer Engineering & Applications, vol. 59, no. 9, pp. 104-111, 2023.

[22] W. Qiu and J. Xu, "A multi-granularity relationship detection model integrating global and local features," Application Research of Computers, vol. 40, no. 2, pp. 476-480, 2023.

[23] W. Jin, F. Jiang, X. Wang, N. Ma, and Y. Zhang, "Research and analysis of grammatical error correction technology for Chinese documents," Journal of Computer and Communications, vol. 12, no. 8, pp. 202-223, 2024.

[24] H. Xie, Z. Chen, J. Cheng, X. Lü, and Z. Tang, "Chinese Grammatical Error Detection Method Based on Data Augmentation and Multi-task Feature Learning," Journal of Chinese Information Processing, vol. 36, no. 12, pp. 36-43, 2022, [Online]. Available: https://aclanthology.org/2020.ccl-1.71/.

[25] R. Yang, Y. Gan, and C. Zhang, "Chinese named entity recognition based on BERT and lightweight feature extraction model," Information, vol. 13, no. 11, pp. 515-535, 2022, doi: 10.3390/info13110515.

[26] B. Qiao, Z. Zou, Y. Huang, K. Fang, X. Zhu, and Y. Chen, "A joint model for entity and relation extraction based on BERT," Neural Computing and Applications, vol. 34, no. 5, pp. 3471-3481, 2022, doi: 10.1007/s00521-021-05815-z.

[27] G. S. Chauhan, Y. K. Meena, D. Gopalani, and R. Nahta, "A mixed unsupervised method for aspect extraction using BERT," Multimedia Tools and Applications, vol. 81, no. 22, pp. 31881-31906, 2022, doi: 10.1007/s11042-022-13023-7.

[28] K. H. Alyoubi, F. S. Alotaibi, A. Kumar, V. Gupta, and A. Sharma, "A novel multi-layer feature fusion-based BERT-CNN for sentence representation learning and classification," Robotic Intelligence and Automation, vol. 43, no. 6, pp. 704-715, 2023, doi: 10.1108/RIA-04-2023-0047.

[29] J. Su, S. Yu, and X. Hong, "A self-supervised pre-training method for Chinese spelling correction," Journal of South China University of Technology (Natural Science Edition), vol. 51, no. 9, pp. 90-98, 2023, doi: 10.12141/j.issn.1000-565X.230031.

[30] S. Kumar and A. Solanki, "An abstractive text summarization technique using transformer model with self-attention mechanism," Neural Computing and Applications, vol. 35, no. 25, pp. 18603-18622, 2023, doi: 10.1007/s00521-023-08687-7.

[31] Y. Zhang, H. Kamigaito, and M. Okumura, "Bidirectional transformer reranker for grammatical error correction," Journal of Natural Language Processing, vol. 31, no. 1, pp. 3-46, 2024, doi: 10.5715/jnlp.31.3.

[32] C. Liu, K. Zhang, M. Bao, Y. Liu, and Q. Liu, "A Chinese Spelling Correction Method Based on the Fusion of Context and Text Structure," Journal of Nanjing University (Natural Science Edition), vol. 60, no. 3, pp. 451-463, 2024.

[33] G. Huang, C. Wang, and Y. Zhang, "A Chinese Text Correction Method Based on Dynamic Text Window and Dynamic Weight Allocation," Journal of Zhengzhou University (Science Edition), vol. 52, no. 3, pp. 9-14, 2020.

[34] Y. Zha, Y. Yang, R. Li, and Z. Hu, "Text alignment is an efficient unified model for massive nlp tasks," Advances in Neural Information Processing Systems, vol. 36, pp. 77942-77968, 2023.

[35] X. Xie, G. Xu, L. Zhao, and R. Guo, "Opensearch-sql: Enhancing text-to-sql with dynamic few-shot and consistency alignment," Proceedings of the ACM on Management of Data, vol. 3, no. 3, pp. 1-24, 2025, doi: 10.1145/3725331.