

Development of Real-Time Oral Error Correction System for College English Classrooms Based on BERT

Yumeng Wu

Yumeng Wu is with the School of Foreign Languages, Weinan Normal University, Weinan 714099, Shaanxi, China
Corresponding author: Yumeng Wu (e-mail: wuyumeng202209@163.com)

ABSTRACT This paper presents a real-time oral error correction system for college English classrooms based on an acoustic-semantic fusion DistilBERT+Adapter architecture. Whisper-small is used for speech transcription, and ASR confidence scores and word-duration features are embedded directly into the BERT representation space to improve robustness against speech-recognition noise. The model jointly performs error localization through a CRF layer and error-type classification, and the resulting outputs guide a constrained decoding mechanism that generates Top-3 correction candidates. These candidates are subsequently re-ranked using a KenLM language model. The system is lightweight and efficient, containing only 44M parameters and achieving an inference latency of 190 ms. End-to-end evaluation shows a latency of 438 ± 52 ms, Accuracy@Top1 of 73.1%, $F_{0.5}$ of 0.692, and a teacher rating of 4.2. Through adapter fine-tuning, knowledge distillation, and ONNX runtime optimization, the proposed system achieves strong noise robustness and generalization, offering a deployable solution for personalized oral English instruction and real-time acoustic-semantic signal processing.

INDEX TERMS oral error correction, bert, acoustic-semantic fusion, constrained decoding, speech signal processing

I. INTRODUCTION

WITH the deep integration of artificial intelligence and education, college English oral teaching urgently needs intelligent auxiliary tools to break through the three bottlenecks of “difficult teacher coverage, untimely feedback, and imprecise error correction.” Likewise, the technologically evolving industry critically requires similar intelligent auxiliary tools to overcome bottlenecks in real-time quality assurance, specifically addressing “difficult machine monitoring coverage, untimely fault diagnosis, and imprecise defect attribution.” [1,2]. Current classroom oral training relies heavily on teachers’ manual monitoring and real-time comments. Faced with large classes and individual differences among students, teachers often find it difficult to achieve precise, high-frequency, and personalized error correction due to cognitive overload [3,4].

Although speech recognition and grammar correction technologies are becoming increasingly mature [5], existing systems are mostly oriented towards written language or offline scenarios and cannot cope with the non-standard sentence patterns, speech recognition noise, and strong context-dependent ambiguity unique to college English classroom oral teaching. Compared to other oral application scenarios (such as independent practice or standardized tests), the real classroom environment places more stringent constraints on

the system: it must achieve an end-to-end latency of less than 500ms in natural teacher-student interaction to maintain teaching fluency; it must have high noise robustness to adapt to classroom recording conditions; the output results must meet teaching interpretability requirements to facilitate teacher understanding and adoption [6,7]. These teaching-level real-time, robust, and interpretability requirements together constitute the core challenges that distinguish oral error correction systems in college English classrooms from general oral applications. Building a real-time oral error correction system for real-world teaching scenarios has become a key breakthrough in promoting the intelligent transformation of college English teaching.

Current research on grammatical error correction focuses on written language. Existing systems often fail to consider classroom real-time constraints and teacher feedback mechanisms, making it difficult to meet the teaching requirements of “low latency, high accuracy, explainability, and intervention”.

II. METHODS

A. OVERALL SYSTEM ARCHITECTURE

To achieve real-time and precise feedback on oral errors in college English classrooms [8,9], this paper constructs an end-to-end pipeline system: first, a lightweight ASR

module captures the speech stream and outputs a confident transcription; next, the acoustic-semantic fusion BERT-CRF (Conditional Random Field) module locates and classifies errors; finally, the constraint generation module, fine-tuned by the adapter, outputs the Top-3 context-adapted error correction candidates. The system supports streaming triggering and cache acceleration to meet the requirements of low-latency classroom interaction. The system architecture is shown in Figure 1.

The system employs Whisper-small for on-device, real-time speech transcription, yielding structured text outputs enriched with word-level timestamps, confidence scores, and durations. These acoustic features are integrated into a BERT-CRF dual-task model, which performs simultaneous error span detection and type classification. The resulting error masks and type predictions guide a constrained decoding process within an Adapter-fine-tuned DistilBERT-Transformer architecture, generating a Top-3 set of grammatically and contextually appropriate correction candidates. The optimal suggestion is selected through re-ranking by a KenLM language model.

B. INPUT FEATURE FUSION STRATEGY

To mitigate the interference of transcription errors applied by background noise, accent variation, or speech rate changes in automatic speech recognition on downstream error correction models in real classroom environments [10,11], this paper designs a multimodal acoustic-semantic joint input representation mechanism that structurally fuses the text sequence output by the ASR with its accompanying acoustic confidence and duration information. This allows the error correction model to explicitly perceive speech recognition uncertainty during the semantic understanding process, thereby dynamically adjusting the error correction weights for low-confidence words, improving the system's robustness and error correction rationality in noisy environments.

Supposing that the i -th word output by the ASR module is w_i , its corresponding acoustic features include two items:

Word-level confidence: $c_i \in [0, 1]$, output by Whisper-small, indicating the reliability of the word recognition result; Word duration: $d_i \in \mathbb{R}^+$ (unit: ms), calculated from the speech timestamp, reflecting the pronunciation stability.

To standardize the scale and distribution of the duration feature and prevent excessive gradients during training, the raw duration is pre-processed using log scaling: $d'_i = \log(1 + d_i)$. The original text sequence is:

$$W = \{[\text{CLS}], w_1, w_2, \dots, w_n, [\text{SEP}]\} \quad (1)$$

The corresponding acoustic feature sequence is:

$$A = \{a_0, a_1, \dots, a_n, a_{n+1}\} \quad (2)$$

In Formula 2, a_0 and a_{n+1} correspond to the acoustic placeholders of [CLS] and [SEP], respectively (confidence=1.0, duration=0), and the rest are $a_i = [c_i, d'_i] \in \mathbb{R}^2$.

The scaled duration d'_i is used here instead of the raw d_i . CLS stands for Classification, and SEP stands for Separator.

To map the two-dimensional acoustic features to the semantic space of the same dimension as the word embedding, this paper applies an acoustic feature encoder, which consists of a linear transformation layer:

$$h_i^{(\text{acous})} = \text{ReLU}(W_a \cdot a_i + b_a) \in \mathbb{R}^{16} \quad (3)$$

In Formula 3, $W_a \in \mathbb{R}^{16 \times 2}$ and $b_a \in \mathbb{R}^{16}$ are learnable parameters, and the output is a 16-dimensional acoustic embedding vector. Layer Normalization (LayerNorm) is applied here to stabilize the feature distribution before fusion with the text embedding, which is critical due to the differing scales of confidence and (log-scaled) duration inputs.

It is assumed that the output of the BERT word embedding layer is:

$$h_i^{(\text{text})} = \text{Embed}_{\text{BERT}}(w_i) \in \mathbb{R}^{768} \quad (4)$$

This paper adopts a concatenation + compression fusion strategy, concatenating the text and acoustic embeddings and then reducing the dimension through the fully connected layer, retaining the original semantic space structure:

$$h_i^{(\text{fused})} = \text{LayerNorm} \left(W_f \cdot [h_i^{(\text{text})}; h_i^{(\text{acous})}] + b_f \right) \in \mathbb{R}^{768} \quad (5)$$

In Formula 5, $[\cdot; \cdot]$ represents vector concatenation; b_f is the compression layer parameter; LayerNorm is used to stabilize training and align with the BERT native structure; the output $h_i^{(\text{fused})}$ replaces the original BERT word embedding and is input to the Transformer encoder.

C. ERROR DETECTION MODULE

1) Backbone Encoder Structure

This paper designs a dual-task joint learning architecture based on BERT [12,13], which outputs error boundary sequence labels (BIO (Beginning, Inside, Outside) format) and error type probability distribution in parallel based on a unified encoder. This module serves as the core front end of the system, and its output directly guides the constraint strategy of the subsequent error correction generation module.

This module uses BERT-base (uncased) as the contextual semantic encoder, and its input is the generated fusion feature sequence:

$$H^{(\text{fused})} = [h_0^{(\text{fused})}, h_1^{(\text{fused})}, \dots, h_{n+1}^{(\text{fused})}] \in \mathbb{R}^{(n+2) \times 768} \quad (6)$$

After being encoded by BERT's 12-layer Transformer, the output is a context-aware hidden state sequence:

$$\begin{aligned} H^{(\text{ctx})} &= \text{BERT}_{\text{base}}(H^{(\text{fused})}) \\ &= [h_0^{(\text{ctx})}, h_1^{(\text{ctx})}, \dots, h_{n+1}^{(\text{ctx})}] \in \mathbb{R}^{(n+2) \times 768} \end{aligned} \quad (7)$$

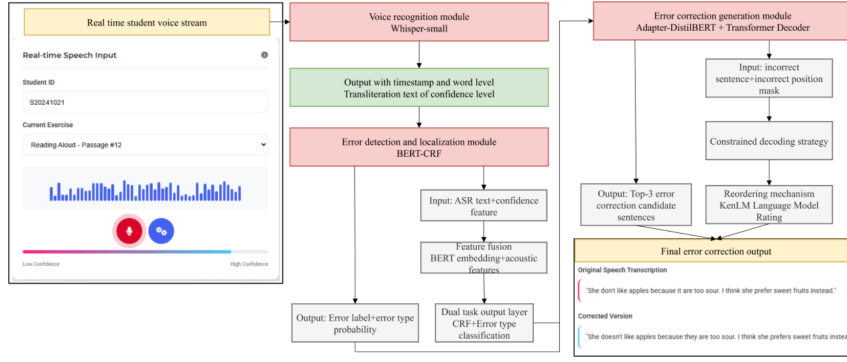


FIGURE 1. System architecture

In Formula 7, $h_n^{(ctx)}$ contains the semantic and grammatical information of word w_n in the context of the complete sentence.

2) Sequence Labeling - Error Span Detection

The exact location of the wrong word in the sentence is identified, and the BIO tagging scheme + CRF modeling label is adopted to transfer constraints and avoid illegal label sequences. The label set is defined as:

$$L_{BIO} = \{O, B-G, I-G, B-L, I-L, B-C, I-C\} \quad (8)$$

In Formula 8, O: Outside, non-incorrect word; B-G / I-G: Grammar error start/internal word (such as tense, subject-verb agreement); B-L / I-L: Lexical error (such as inappropriate word use, synonym confusion); B-C / I-C: Collocation error (such as verb phrase collocation error).

For each position i , the unnormalized emission score of each label is calculated through the linear projection layer:

$$s_i = W_{CRF} \cdot h_i^{(ctx)} + b_{CRF} \in \mathbb{R}^{|L_{BIO}|} \quad (9)$$

CRF modeling: assuming the label sequence is $y=[y_0, y_1, \dots, y_{n+1}]$, its conditional probability is:

$$P(y|H^{(ctx)}) = \frac{\exp\left(\sum_{i=0}^{n+1} s_i(y_i) + \sum_{i=0}^n T_{y_i, y_{i+1}}\right)}{\sum_{y' \in Y} \exp\left(\sum_{i=0}^{n+1} s_i(y'_i) + \sum_{i=0}^n T_{y'_i, y'_{i+1}}\right)} \quad (10)$$

In Formula 10, $s_i(y_i)$ is the emission score of label y_i at position i ; T is the learnable label transfer matrix; Y is the set of all legal label sequences.

The loss function (negative log-likelihood) is:

$$L_{CRF} = -\log P(y^*|H^{(ctx)}) \quad (11)$$

In Formula 11, y^* is the true label sequence.

3) Error Type Classification - Semantic-Guided Error Correction Strategy Selection

To support the type-aware constraint generation of the subsequent error correction module, this task predicts the dominant error type at the sentence level or word level. Considering that oral errors are often concentrated in a single sentence, this paper adopts sentence-level classification and uses the hidden state of the [CLS] position as the sentence representation $h_{[CLS]}^{(ctx)}$. After the linear layer + Softmax, the error type probability distribution is output:

$$p_{type} = \text{Softmax}\left(W_{Type} \cdot h_{[CLS]}^{(ctx)} + b_{Type}\right) \in \mathbb{R}^4 \quad (12)$$

The error type set is:

$$T = \{\text{Grammar, Lexical, Collocation, Mixed}\} \quad (13)$$

In Formula 13, Mixed means that the sentence contains multiple error types, triggering a conservative error correction strategy.

Loss function (cross entropy): the true type is $t^* \in \{0, 1, 2, 3\}$, and then there is:

$$L_{Type} = -\log(p_{type}[t^*]) \quad (14)$$

4) Joint Training and Loss Function Design

To balance the learning weights of the two tasks, a weighted joint loss function is adopted:

$$L_{joint} = L_{CRF} + \lambda \cdot L_{Type} \quad (15)$$

In Formula 15, the hyperparameter λ controls the contribution of the type classification task to the total loss.

Performance changes under different λ values are shown in Figure 2.

Analysis in the validation set shows that λ not only affects $F_{0.5}$ but also significantly modulates Accuracy@Top3. Both curves reach their peaks simultaneously at $\lambda=0.30$, indicating that this weight achieves the optimal balance between “precise error correction” and “candidate coverage”. When $\lambda < 0.30$, insufficient type information leads to blind generation strategies and reduced candidate quality.

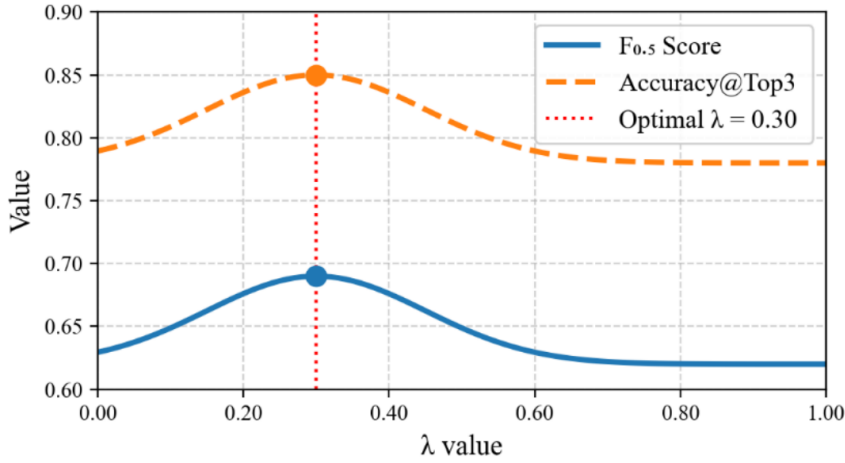


FIGURE 2. Performance changes under different λ values

When $\lambda > 0.30$, excessive type loss interferes with sequence annotation, mislocalizing errors and consequently reducing Top3 accuracy.

Selecting $\lambda = 0.30$ ensures both high precision in the first recommendation and high recall in the candidate pool, aligning with the teaching principle of “precision first, while also considering error tolerance” in college English classrooms and demonstrating the scientific and practical nature of multi-task collaborative design.

D. ERROR CORRECTION GENERATION MODULE: DISTILBERT+ ADAPTER + TRANSFORMER DECODING

1) Encoder: DistilBERT with Adapter Fine-tuning

This module employs a DistilBERT encoder pre-trained on a general corpus, with its parameters frozen after initialization. We did not perform intermediate fine-tuning on a standard written language grammar correction corpus. Standard GEC tasks differ significantly from classroom spoken language correction in error distribution, language style, and input modality. Directly introducing GEC fine-tuning might bias the model towards written language norms, thus weakening its ability to capture spoken language specificity.

This paper chooses to use the pre-trained DistilBERT as a powerful foundation for general semantic understanding. By directly inserting an adapter module, it collaborates with the front-end acoustic-semantic fusion architecture, performing end-to-end adapter fine-tuning on the classroom spoken language dataset constructed in this paper. This strategy enables the model to simultaneously learn the understanding of acoustic features and the identification and correction of spoken language errors, thereby achieving efficient and accurate domain adaptation from general domains to specific spoken language teaching scenarios.

An adapter module is inserted after each Transformer layer of DistilBERT (a total of 6 layers, with 1 adapter inserted in each layer). The structure of each adapter is:

$$\text{Adapter}(x) = x + W_{\text{up}} \cdot \text{GeLU}(W_{\text{down}} \cdot x) \quad (16)$$

In Formula 16, W_{down} and W_{up} are trainable parameters, and GeLU (Gaussian error linear units) is the activation function.

The encoder output is:

$$H^{(\text{enc})} = \text{Adapter-DistilBERT}(H^{(\text{ctx})}) \in \mathbb{R}^{(n+1) \times 768} \quad (17)$$

2) Decoder: Lightweight Transformer with Error Location Focusing

The decoder uses a two-layer Transformer Decoder. The input is “error sentence + error location mask”, forcing attention to focus on the area to be corrected to avoid indiscriminate rewriting. The positional attention mask matrix M is constructed for the multi-head self-attention layer of the Transformer Decoder:

$$M_{ij} = \begin{cases} 0, & \text{if } j \in E, \\ -\infty, & \text{otherwise} \end{cases} \quad (18)$$

Here, i is the index of the query token (the target token being computed) in the current decoder layer’s input sequence, and j is the index of the key token (the token being attended to) also within the decoder layer’s input sequence. This matrix M is applied to the attention logits to prevent the model from attending to tokens outside the specified error span.

E is defined as the set of indices corresponding to the tokens in the decoder input sequence that are part of the predicted error span or area to be corrected. Applying $M_{ij} = -\infty$ for $j \notin E$ ensures that attention weight is forced to zero for tokens outside the error locations, thereby focusing the decoder’s self-attention solely on the identified error region.

In the decoder self-attention and cross-attention, the mask is applied:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T + M}{\sqrt{d_k}}\right)V \quad (19)$$

3) Type-aware Constraint Decoding Strategy

The generated word is restricted to be consistent with the original word part of speech. Assuming that the part of speech of the original word w_i is $\text{POS}(w_i)$, then the generated word $\hat{w}_i$ must satisfy:

$$\text{POS}(\hat{w}_i) = \text{POS}(w_i) \quad (20)$$

The part of speech is pre-tagged by Stanford POS Tagger, and the candidate words are sampled from the vocabulary of the same part of speech.

The synonym extension library is enabled, and the candidate words come from:

$$C_i = \{w' \mid \text{sim}_{\text{BERT}}(w_i, w') > \tau\} \cup \text{WordNet.synset}(w_i) \quad (21)$$

In Formula 21, $\text{sim}_{\text{BERT}}(w_i, w')$ is the cosine similarity between the BERT word vector embeddings of w_i and w' (i.e., $\cos(e(w_i), e(w'))$), where $e(w)$ is the context-independent BERT word embedding for word w ; $\text{WordNet.synset}(w_i)$ provides a set of semantic synonyms. The word vector is used to capture word-level semantics, which is more appropriate than the sentence-level [CLS] token embedding for calculating word similarity. The threshold value τ is a hyperparameter that controls the strictness of the synonym selection based on BERT similarity. It was determined empirically using a small validation set and set to $\tau = 0.8$ throughout all experiments to balance coverage and quality of candidate words.

4) Candidate Re-ranking and Final Output

The decoder Beam Search generates the Top-3 candidate sentences, which are re-ranked by the KenLM language model, and the one with the lowest perplexity (PPL) is selected as the final output:

$$\hat{S}^* = \arg \min_{\hat{S} \in \text{Top-3}} \text{PPL}_{\text{KenLM}}(\hat{S}) \quad (22)$$

KenLM is a 3-gram language model trained on CoLA Acceptable sentences and college English textbook corpus to ensure that the output is natural in the teaching context.

The type guided constraint decoding mechanism applies hard and soft constraints related to the error type at each step of the decoder's generation, ensuring that the generated Top-3 candidates are consistent with the error type recognized by the front-end module. Specific constraint rules are dynamically activated based on the type of error:

Grammar error: When generating correction candidate words, the decoder's vocabulary is dynamically limited to a subset of words with the same part of speech as the original error word.

Vocabulary error: The search space of the decoder is guided to words that are semantically similar to the original word. This is achieved by combining a predefined WordNet synonym set with a cosine similarity threshold based on

BERT sentence embedding, generating synonym or synonyms candidates that are both contextually relevant and more accurate than the original word.

Match error: Using pre compiled common match libraries. When the error type is of this type, the model not only corrects the incorrect word, but also forces its subsequent or adjacent words to generate words that conform to the correct collocation pattern to ensure the authenticity at the phrase level.

Mixed error: Combining multiple constraints and prioritizing the candidate with the smallest modification and lowest syntax risk to avoid introducing new errors through excessive correction.

5) Real-time Inference Optimization

To reduce the computational complexity of the model without significantly losing performance, this paper adopts task-specific knowledge distillation, with the fully fine-tuned BERT-base as the teacher model and the adapter-fine-tuned DistilBERT as the student model, and realizes knowledge transfer by aligning the soft label with the hidden layer.

Assuming that the teacher model output logits is $z^{(\text{tea})}$; the student model is $z^{(\text{stu})}$; the true label is y , then the joint loss is:

$$L_{\text{distill}} = \alpha \cdot L_{\text{CE}}(z^{(\text{stu})}, y) + (1 - \alpha) \cdot L_{\text{KL}}\left(\sigma(z^{(\text{tea})}/T), \sigma(z^{(\text{stu})}/T)\right) \quad (23)$$

In Formula 23, L_{CE} is the cross entropy loss; L_{KL} is the divergence loss; $\sigma(\cdot)$ is the Softmax function; $T = 4$ is the temperature hyperparameter, smoothing the probability distribution; $\alpha = 0.3$ is the hard label weight, balancing the true label and teacher knowledge.

The mean square error loss is applied to the output of the encoder layer l:

$$L_{\text{hidden}} = \frac{1}{n \cdot d} \sum_{i=1}^n \sum_{j=1}^d \left(H_{l,ij}^{(\text{tea})} - H_{l,ij}^{(\text{stu})} \right)^2 \quad (24)$$

The final distillation loss is:

$$L = L_{\text{distill}} + \beta \cdot L_{\text{hidden}} \quad (\beta = 0.5) \quad (25)$$

To avoid redundant calculations of repeated classroom sentences, the system employs SimHash for efficient duplicate detection and near-duplicate matching, enabling constant-time expected lookup cache query. For the input sentence S , its semantic hash key is calculated:

$$\text{Key}(S) = \text{SimHash}(\text{MinHash}(\{v(S_i) \mid S_i \in n\text{-gram}(S, k=3)\})) \quad (26)$$

In Formula (26), S_i denotes a single n -gram fragment (with $k = 3$, i.e., a trigram) of the sentence S . The sentence vector $v(S_i)$ for each n -gram fragment S_i is

computed by applying mean pooling over the sequence of token embeddings generated by BERT, i.e., $v(S_i) = \text{mean_pooling}(e_1, e_2, \dots, e_m)$, where e_k is the embedding for the k -th token in S_i and m is the number of tokens in that n -gram (typically $m = k$). To apply MinHash—a set-based compression technique—to these continuous vectors, first discretize each $v(S_i)$ by applying a locality-sensitive hash (LSH) function that maps similar vectors to the same integer code with high probability. This yields a set of integer codes, which is then compressed using MinHash. Finally, SimHash generates a 64-bit binary signature as the semantic hash key for the sentence.

Assuming that the model computation time is T_{comp} , and the I/O (Input/Output) and preprocessing time is T_{io} , then the end-to-end latency of a single sentence is:

$$T_{\text{latency}} = T_{\text{comp}} + T_{\text{io}} + T_{\text{cache}} \cdot (1 - P_{\text{hit}}) \quad (27)$$

To adapt to the natural oral language flow and avoid semantic truncation caused by fixed window segmentation, this paper uses the built-in voice activity detection module of WebRTC (Web Real-Time Communication) to dynamically detect the end point of a sentence.

Assuming that the current voice stream timestamp is t , and the most recent voice activity ends at t_{last} , then the inference is triggered if and only if:

$$t - t_{\text{last}} \geq \tau_{\text{silence}} \cap \text{duration}(S) \geq \tau_{\text{min}} \quad (28)$$

In Formula 28, τ_{silence} is the silence threshold to avoid premature triggering, set to 500ms; τ_{min} is the minimum sentence length to filter invalid segments, set to 800ms.

The system interface of this paper is shown in Figure 3.

III. EXPERIMENTS

A. DATASETS

1) CoLA Corpus

The existing mainstream GEC benchmarks are all oriented towards written language, and their input is pure text, which cannot meet the needs of evaluating acoustic modal input and colloquial error patterns in this system. Therefore, to verify the performance of the system in real classroom environments, this study used two types of data: 1) a 'pseudo oral' training and testing set constructed through a systematic acoustic feature synthesis strategy based on the CoLA corpus; 2) A large-scale collection of oral data from college students in real classroom environments. This scheme aims to simulate and validate the comprehensive performance of the system in the target scenario, and its evaluation results should be regarded as proof of effectiveness on the specific teaching task, rather than a direct performance ranking against the pure text GEC benchmark.

To make up for the lack of speech data in CoLA, this paper designs an acoustic feature synthesis model based on lexical-syntactic error probability, generates "pseudo-ASR confidence" and "pseudo-word duration" for each word, so

that the model can learn the noise robustness mechanism of "low-confidence words should have lower error correction weight" during the training phase.

Assuming that the part of speech of word w_i is POS_i , and its grammatical role in the sentence is $Role_i$ (such as subject, predicate, object, modifier), its error sensitivity score is defined:

$$s_i = \alpha_1 \cdot I(POS_i \in P_{\text{error-prone}}) + \beta_1 \cdot I(Role_i \in R_{\text{critical}}) \quad (29)$$

In Formula 29, $P_{\text{error-prone}} = \{\text{VERB, AUX, NOUN, PRON}\}$ (error-prone part of speech); $R_{\text{critical}} = \{\text{Predicate, Subject}\}$ (key syntactic role). $\alpha_1 = 0.4$ and $\beta_1 = 0.6$ are weight coefficients.

The word duration d_i (unit: ms) is linearly simulated according to the number of syllables:

$$d_i = \text{SyllableCount}(w_i) \times \gamma + \epsilon \quad (30)$$

In Formula 30, $\gamma = 200\text{ms}$ is the baseline duration of each syllable; $\epsilon \sim N(0, 30^2)$ is Gaussian noise, which simulates pronunciation fluctuations.

The statistical information of the CoLA dataset is shown in Table 1.

2) Real-User Data Collection

To comprehensively validate the system's generalization and adaptability in real classroom settings, oral speech data were collected from 2,088 non-English-major undergraduates using a multi-factor quota-controlled sampling design. Quotas were independently set for three key learner characteristics:

English proficiency (based on CET-4 scores), major category, and frequency of oral training, each approximated at a 1:1:1 ratio across its respective categories.

Subjects are sampled in a stratified and proportional manner according to the three teaching-related dimensions. The stratified sampling design is shown in Table 2.

Each participant completes two tasks in a standardized simulated classroom environment (quiet classroom, ambient noise <40dB, directional microphone, 16kHz sampling rate), with five sentences in each category, for a total of 10 sentences per participant.

For the reading task, 200 typical grammatically incorrect sentences (covering tense, subject-verb agreement, articles, prepositions, and clause structure) are selected from CoLA. Three university English teaching experts rewrite them into colloquial expressions, adding natural fillers and correcting structures to ensure that the corpus closely resembles authentic oral language.

For the free-response task, students give 30-second free-response responses to five randomly presented topics on the screen. The topics are taken from actual CET-4 and CET-6 oral exam questions.

To quantify the teaching usability of the system's output, six lecturers with more than five years of college English teaching experience are invited to conduct double-blind

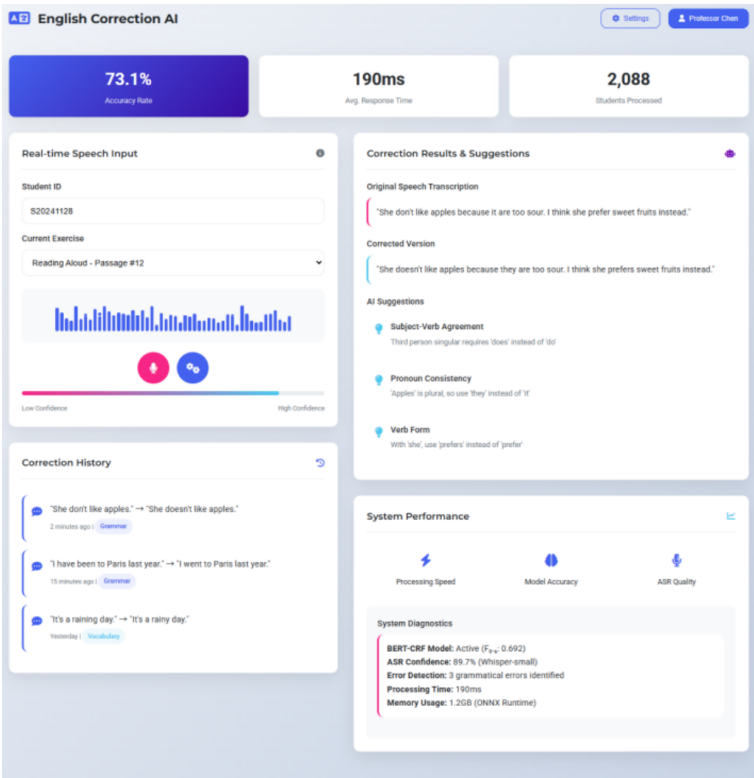


FIGURE 3. System interface

TABLE 1. Statistics of the CoLA dataset

Project	Value	Instructions
Total sentences	8,551	From the original CoLA v1.0
Unacceptable sentence count	4,355	As the source sentence (incorrect sentence)
Acceptable number of sentences	4,196	As the target sentence (correct sentence)
Build(Error, Correct) Pair Numbers	3,982	Used for training the error correction generation module
Number of single sentences used only for error detection	8,551	Contains BIO tags for sequence annotation training
Average sentence length (number of words)	12.7	Standard deviation=5.2
Distribution of main types of errors	Tense errors (32%); Subject verb agreement (28%); Articles/Prepositions (19%); Sub clause structure (15%); Other (6%)	Based on manual secondary annotation
Pseudo confidence range	[0.60, 1.00]	Mean=0.82, standard deviation=0.09
Pseudo duration range	[120ms, 1100ms]	Mean=420ms, standard deviation=180ms

TABLE 2. Stratified sampling design

Classification dimension	Category description	Number of people	Selection criteria
English proficiency (CET-4 (College English Test) score)	Beginner: < 425 points	696	Reflecting the distribution of real classroom abilities and verifying the system's adaptability to correcting errors for learners with different levels of grammar awareness
	Intermediate: 425–550 points	696	
	Advanced: ≥ 550 points	696	
Professional category	Science and Engineering (Computer/Mechanical/Electronic, etc.)	696	Exploring whether differences in disciplinary thinking lead to the differentiation of erroneous patterns
	Humanities and Social Sciences (Chinese/History/Law, etc.)	696	
	Economics and Management (economics/administration/finance, etc.)	696	
		696	
Frequency of oral training	Low frequency: < 1 time/week	696	Assessing the robustness of the system to differences in oral fluency and pragmatic habits, in line with the reality of hierarchical teaching
	Mid frequency: 1–2 times/week	696	
	High frequency: ≥ 3 times/week	696	

scoring of the system's Top-1 error correction suggestions (the raters are unaware of the model's source; each sentence is independently scored by two people; any disagreements are arbitrated by a third person). To ensure the consistency and reliability of the human annotations, the Inter-Rater Reliability (IRR) between the two independent raters was calculated using Cohen's Kappa, yielding a value of $k=0.78$. This value indicates substantial agreement between the scorers, confirming the reliability of the human evaluation dataset. The scoring criteria are shown in Table 3.

TABLE 3. Scoring criteria

Score	Grading criteria
1	Correction of errors or misleading information: grammatical errors, semantic distortions, and changes in original meaning
2	Grammar correct but pragmatics inappropriate: stiff sentence structure, not in line with oral habits, cultural context not suitable
3	Acceptable but not optimal: The correction is reasonable, but there is a more natural or precise expression
4	Natural, accurate, and contextually appropriate: Smooth correction, retaining the original meaning, and conforming to the norms of college English oral expression
5	Teaching grade high-quality feedback: can be directly used for classroom explanation, with error interpretability and teaching guidance value

B. MODEL COMPARISON AND EVALUATION METRICS

To verify the effectiveness of the acoustic-semantic fusion DistilBERT+ Adapter model in the grammatical error correction task, six representative baseline models are selected for comparison:

- (1) BERT-base: standard fine-tuned BERT, without adapter and acoustic features, as a pure text baseline;
- (2) RoBERTa [14,15] (Robustly optimized BERT pre-training Approach)-large GEC (Grammatical Error Correction): pre-trained large-scale model, evaluating model capacity gain;
- (3) T5-base for GEC: Seq2Seq generative model, evaluating generative architecture adaptability;
- (4) GECToR: edit-based sequence tagging model;
- (5) mT5 (multilingualT5)-small: multilingual model, evaluating cross-lingual interference.
- (6) ELECTRA (Efficiently Learning an Encoder that Classifies Token Replacements Accurately) -base: replacement detection architecture to evaluate error localization capabilities.

All models are trained with a unified configuration to ensure experimental reproducibility. The training configuration is shown in Table 4.

TABLE 4. Training configuration

Configuration item	Parameter values	Instructions
Hardware	NVIDIA RTX 3090 (24GB)	Accelerated Training
Framework	PyTorch 2.1 + Transformers 4.36	Provide library support
Optimizer	AdamW	Weight attenuation=0.01
Learning rate	2×10^{-5}	Preheat for 500 steps, linear decay
Batch size	16	Gradient accumulation steps=2
Epoch	10	Early stop patience=3
Random seed	42	Ensure reproducibility of results

To evaluate the comprehensive performance of the system in a real classroom speech scenario, end-to-end systems are compared.

Multiple indicators are set to measure the performance of oral language error correction. The Accuracy@Top1 measures the proportion of the system's first-recommended correction suggestions that completely match the standard answer, reflecting the "one-time hit" capability:

$$\text{Accuracy@Top1} = \frac{1}{N} \sum_{i=1}^N I(\hat{S}_i^{(1)} = S_i^{\text{gold}}) \quad (31)$$

In Formula 31, N is the total number of test samples; $\hat{S}_i^{(1)}$ is the Top-1 correction output of the i -th sentence; S_i^{gold} is the manually annotated gold standard correction sentence.

$F_{0.5}$ Score is the F-score with $\beta=0.5$, and the formula is:

$$F_{0.5} = (1 + 0.5^2) \cdot \frac{\text{Precision} \cdot \text{Recall}}{0.5^2 \cdot \text{Precision} + \text{Recall}} \quad (32)$$

In Formula 32, Precision is the correct proportion of system corrections; Recall is the proportion of annotated errors that are correctly corrected by the system.

End-to-end latency is the time interval from the end of speech input to the presentation of correction feedback on the student interface, reflecting the real-time performance of the system, which is measured in an Intel i7-12700H CPU environment. The mean teacher rating measures the teaching usefulness of the correction suggestions. The mean is calculated based on the double-blind ratings (1-5) of 6 teachers:

$$\text{TRM} = \frac{1}{N} \sum_{i=1}^N r_i, \quad r_i \in \{1, 2, 3, 4, 5\} \quad (33)$$

The error correction rate is the proportion of samples judged by teachers as "misleading corrections" (rating = 1):

$$\text{FCR} = \frac{1}{N} \sum_{i=1}^N I(r_i = 1) \quad (34)$$

IV. RESULTS

A. MODEL PERFORMANCE COMPARISON (COLA TEST SET)

The performance comparison with advanced models is shown in Figure 4.

The proposed model surpasses all baselines on the CoLA test set, achieving optimal performance in four core metrics: Accuracy@Top1 (73.1%), $F_{0.5}$ (0.692), number of parameters (44M), and inference latency (190ms), demonstrating the synergistic effectiveness of acoustic-semantic fusion and lightweight architecture.

Compared to BERT-base, Accuracy@Top1 improves by 5.6 percentage points, and $F_{0.5}$ improves by 0.071.

This is attributed to the fact that acoustic feature injection enables the model to perceive the ASR noise distribution and suppress miscorrections of low-confidence words. Constrained decoding guided by error type further improves semantic plausibility and reduces ineffective corrections. Compared to RoBERTa-large GEC (355M, 580ms), the proposed model achieves higher accuracy with 44M parameters and a latency of 190ms, demonstrating that adapter fine-tuning and knowledge distillation significantly reduce computational overhead while preserving semantic capabilities, making it suitable for classroom terminal deployment. The inference delay (latency) reported in Figure 4 was measured for the entire forward pass executed on a single NVIDIA P100 GPU. This total delay includes the time taken for all GPU kernel operations (model computation) as well as the data transfer time and the initial CPU-side preprocessing overhead. The majority of the delay, however, is attributed to the GPU computation time, consistent with the typical performance bottleneck in Transformer-based models.

The error types of the proposed model and RoBERTa-large GEC are analyzed, and the error types are shown in Figure 5.

The confusion matrix in Figure 5 reveals that the proposed model outperforms RoBERTa-large GEC in error type identification, particularly for core grammatical errors (Grammar). The proposed model achieves a 1280/1385 error recognition rate on Grammar, while RoBERTa achieves only 1150/1385. This is attributed to the synergistic effect of the error type classification head and the constrained decoding mechanism in its dual-task architecture, which enables the model to dynamically adjust its generation strategy based on the predicted type (such as forcing part-of-speech preservation for grammatical errors), significantly improving correction accuracy. Figure 6 shows the error correction accuracy performance results for different ASR confidence intervals.

The proposed model outperforms baselines across all ASR confidence intervals. To demonstrate the statistical reliability of these results, we report the sample size distribution for each interval based on the test set: the high-confidence interval (0.9–1.0) contains approximately 1,411 samples; the medium-high interval (0.8–0.9) contains 3,420 samples; the medium-low interval (0.7–0.8) contains 2,743 samples; and

the low-confidence interval (0.6–0.7) contains 718 samples. Particularly in the low confidence interval (0.6–0.7), despite the relatively smaller sample size ($N=718$), the model's $F_{0.5}$ reaches 0.593, a 10.2% improvement over the strongest baseline, RoBERTa-large (0.538). This demonstrates that the acoustic feature fusion mechanism effectively mitigates noise interference in speech recognition. This is due to the fact that the proposed model explicitly concatenates ASR confidence and word duration vectors in the BERT word embedding layer, automatically downweighting low-confidence words in the attention calculation, and avoiding the “mistranslation → miscorrection” cycle.

B. ABLATION EXPERIMENTS

Ablation experiments are conducted to analyze the impact of various components on model performance.

The results are shown in Table 5.

Ablation experiments demonstrate the significant synergistic contributions of the various components of the proposed model. Removing acoustic feature fusion results in a decrease in Accuracy@Top1 (73.1% → 68.3%), demonstrating its effective suppression of ASR noise transmission through a confidence-aware mechanism, which is the core of the accuracy improvement. Removing the adapter, while only slightly decreasing accuracy by 0.2%, increases the parameter count to 66M and latency to 280ms, highlighting its critical role in achieving lightweight deployment without sacrificing performance. Removing the error type guidance results in a decrease in accuracy (73.1% → 70.1%), demonstrating the necessity of type-aware constraint decoding (such as part-of-speech preservation and synonym expansion) for reasonable generation, while adding almost no computational overhead. The caching mechanism and knowledge distillation are purely engineering optimizations. While the removal of the caching mechanism leaves accuracy unchanged, the exclusion of knowledge distillation (using the teacher model directly) results in a marginal increase in accuracy (73.1% → 73.3%); however, this comes at the cost of significantly increased latency to 310ms and 320ms, respectively, demonstrating their irreplaceable role in ensuring real-time classroom performance.

C. SYSTEM ERROR CORRECTION PERFORMANCE

Multiple systems are compared, and their end-to-end latency, teacher scores, and error correction rates are analyzed. Figure 7 shows the system error correction performance comparison results.

The subjects are stratified and analyzed to explore the universality of the system's error correction across the population. The results are shown in Figure 8.

The proposed system consistently leads across all nine subgroups, and its error correction performance improves with student ability and training frequency. Among beginner students, it leads GECToR by 9.2 percentage points (70.5% vs. 61.3%), and among advanced students, this performance increases to 4.7 percentage points (75.2% vs.

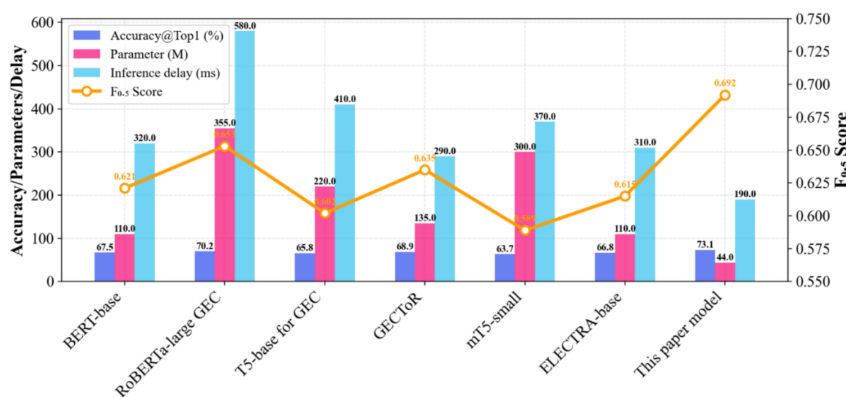


FIGURE 4. Performance comparison with advanced models

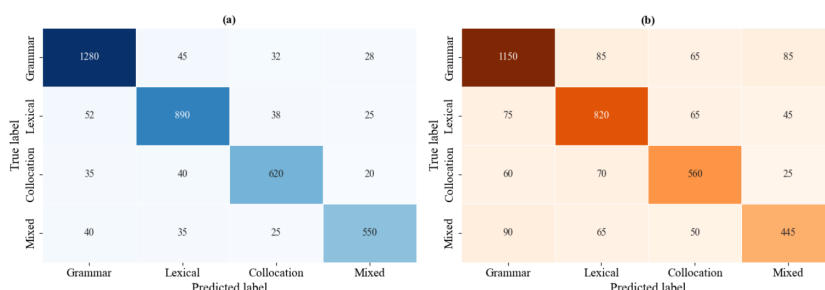


FIGURE 5. Error type heatmap. (a). Error types of the proposed model; (b). Error types of RoBERTa-large GEC

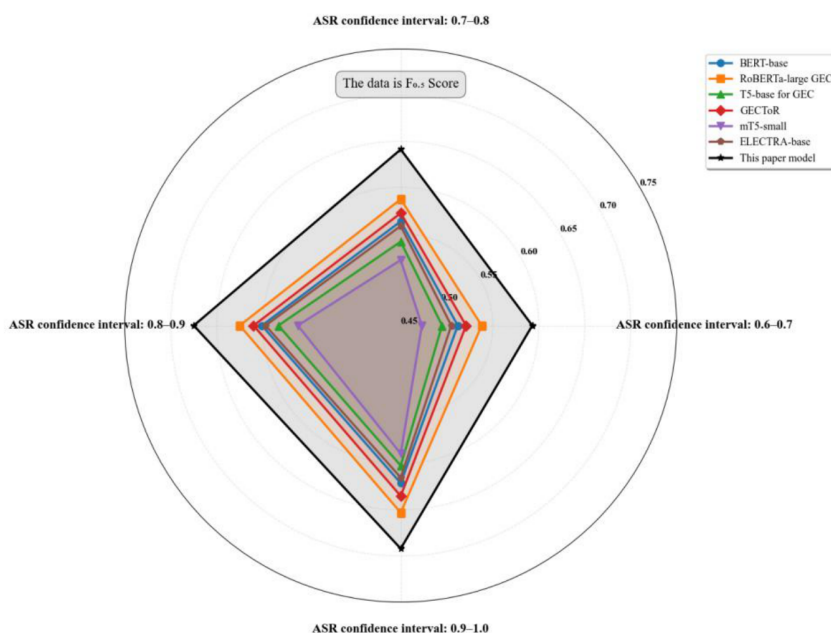


FIGURE 6. Error correction accuracy performance at different ASR confidence intervals

70.5%), demonstrating that its error guidance mechanism is adaptive to varying levels of grammatical awareness. It performs best among students in the humanities and social sciences (74.3%), as their article/preposition error patterns are more structured and easily captured by the type classification head. Low-frequency speaking students benefit the most (+9.6% vs. GECtoR), as acoustic fusion

effectively compensates for ASR noise caused by their unstable pronunciation. In contrast, Baseline ASR + LanguageTool's performance falls below 45% in all groups due to its rigid rules. ChatGPT, while semantically robust, lacks pedagogical constraints. This system achieves cross-group, cross-ability, and cross-disciplinary generalization, truly meeting the core requirements of “differentiated in-

TABLE 5. Ablation experiment results

Model variants	Accuracy@Top1 (%)	Parameter quantity (M)	Inference latency (ms)	Ablation instructions
The model in this paper	73.1	44	190	Includes all components: acoustic fusion+Adapter+type guidance
w/o acoustic feature fusion	68.3	44	190	Remove ASR confidence and duration input, only embed with plain text BERT
w/o Adapter (fully fine tuned DistilBERT)	72.9	66	280	Remove Adapter, Fine tune DistilBERT Encoder with Full Parameters
w/o error type guidance	70.1	44	195	Remove type classification header, decoder has no part of speech/synonyms (unconstrained decoding) constraints
w/o caching mechanism	73.1	44	310	Remove sentence template hash cache, all sentences are inferred in real-time
w/o knowledge distillation (using BERT base directly)	73.3	110	320	Teacher model deployed directly without distillation compression

Notes: all variants are evaluated on the same CoLA test set; inference latency is the average per sentence on an Intel i7-12700H; parameter count only counts trainable parameters; “w/o” = “without”.

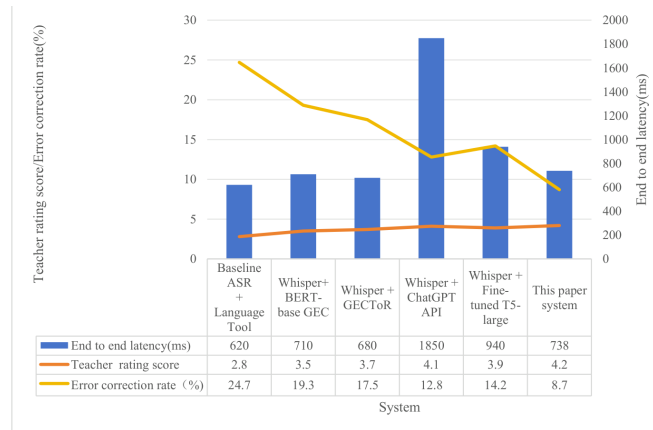


FIGURE 7. System error correction performance comparison

Notes: 2088 participants are tested, each with 10 sentences. This study was approved by the Ethics Committee of Weinan Normal University. All eligible participants signed an informed consent form. End-to-end latency: the time from the end of speech to the presentation of feedback, including ASR, error correction, and rendering; Teacher Rating: the average of six instructors’ double-blind ratings, with the standard deviation reflecting rating consistency; Error Correction Rate: the percentage of samples judged by teachers as “misleading” (rating = 1); all systems are deployed on the same hardware environment (Intel i7-12700H, 32GB). ChatGPT API latency includes network round trips (measured average of 1.85s), with the remainder being local inference.

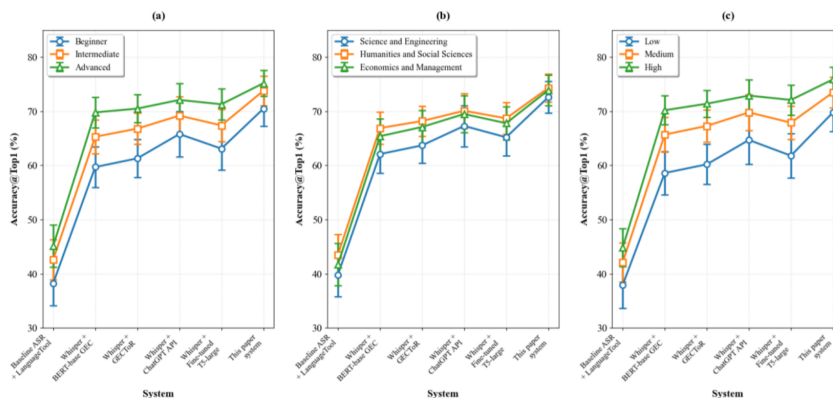


FIGURE 8. System error correction universality; (a). English proficiency; (b). Major category; (c). Oral training frequency

struction and individualized correction” in college English classrooms.

D. ERROR CORRECTION EXAMPLE DISPLAY

This system demonstrates examples of English oral error correction, with the results shown in Table 6.

TABLE 6. Error correction example display

Original sentence (ASR transcription)	Error type	System error correction suggestions	Teacher rating score
He go to school yesterday.	Grammar	He went to school yesterday.	5
I very like this movie.	Lexical	I really like this movie.	5
She made a decision in hurry.	Collocation	She made a decision in a hurry.	4
We should discuss about the problem.	Grammar	We should discuss the problem.	5
I am boring with this class.	Lexical	I am bored with this class.	5
He depends from his parents financially.	Collocation	He depends on his parents financially.	4
She don't has time and is tired.	Mixed	She doesn't have time and is tired.	4
I look forward to meet you.	Grammar	I look forward to meeting you.	5

The error correction examples shown in Table 6 demonstrate that the proposed system precisely identifies and naturally corrects various typical oral errors, with teacher ratings of 4–5 points, verifying that its output is both grammatically correct and pragmatically natural. The acoustic-semantic fusion input layer enables the model to dynamically adjust error correction weights based on ASR confidence, avoiding misleading noise words. The dual-task error detection module accurately outputs error type labels, guiding the generation module to impose targeted constraints—for grammatical errors, the part of speech is forced to be maintained, and for lexical errors, a synonym database is used to ensure that the correction is semantically reasonable. Finally, KenLM reranking prioritizes the most contextually fluent candidates to avoid awkward expressions.

E. MODEL LIGHTWEIGHTING ANALYSIS

To quantitatively evaluate the lightweight characteristics of the proposed system, Table 7 compares the performance of the Adapter BERT model in terms of parameter quantity and inference delay with BERT base and other mainstream syntax correction models in this study.

TABLE 7. Comparison of Lightweight Performance of Models

Model	Parameter quantity (M)	Inference delay (ms)
BERT-base	110	320
RoBERTa-large GEC	355	580
T5-base	220	410
GECtoR	135	290
DistilBERT+ Adapter	44	190

Note: The parameter count only includes trainable parameters; The inference delay is the average time it takes to process a single sentence (with an average length of 12.7 words) on an Intel i7-12700H CPU.

The Adapter BERT architecture used in this system utilizes adapter fine-tuning and knowledge distillation techniques to significantly reduce model size while maintaining strong semantic modeling capabilities. As shown in Table 7, the parameter count of this model is only 44M, which

is 40% of BERT base and even much lower than the GECtoR model, which is also a lightweight design. In terms of inference speed, this model also demonstrates advantages, with lower latency than all compared models. This "lightweight" feature ensures that the system can run efficiently on classroom terminal devices with limited computing resources, meeting the low latency requirements for real-time interaction and providing feasibility for deployment in large-scale teaching scenarios.

V. CONCLUSION

This study achieves precise detection and correction of oral errors by constructing an acoustic-semantic fusion DistilBERT+ Adapter architecture. The system uses Whisper-small to generate confident transcriptions and embeds acoustic features into the BERT word vector space via linear transformation, significantly improving the model's robustness to ASR noise. The dual-task joint optimization of CRF sequence labeling and error type classification is adopted to achieve precise identification of error boundaries and semantically guided error correction strategy selection. Finally, type-aware constrained decoding and KenLM reordering ensure that the output is both grammatically correct and contextually natural. Research results show that the system has achieved breakthroughs in accuracy, efficiency, and teaching adaptability: Accuracy@Top1 has increased to 73.1%; end-to-end latency has been reduced to 438ms; the teacher rating has reached 4.2 points.

AUTHOR CONTRIBUTIONS

Yumeng Wu designed, collected and analyzed the data, and drafted the manuscript. Yumeng Wu conducted the study, critically revised the manuscript for important intellectual content, and gave final approval of the version to be published. Yumeng Wu participated fully in the work, take public responsibility for appropriate portions of the content, and agreed to be accountable for all aspects of the work in ensuring that questions related to the accuracy or integrity of any part of the work are appropriately investigated and resolved.

CONFLICTS OF INTEREST

The authors declare no conflict of interest.

Ethics Approval and Consent to Participate This study was approved by the Ethics Committee of Weinan Normal University. All eligible participants signed an informed consent form.

FUNDING

1. The Youth Project of Shaanxi Province Education Science "14th Five-Year" Plan: A Study on the Application of Eye-Tracking Technology in English Teaching in Universities of Shaanxi Province under the Background of Artificial Intelligence (Project No.SGH24Q342).
2. The Talent Research Program of Weinan Normal University: A Study on Attention Allocation and the Effectiveness of English Captions under Multimodal Input

(Project No. 2025RC27).

3. University-Level Teaching Reform Research Youth Project of Weinan Normal University: A Visualized Practice Study of College English Vocabulary Teaching Based on an AI Eye-Tracking Analysis Platform (No. JG2025089).

ACKNOWLEDGEMENTS

Not applicable.

REFERENCES

- [1] F. You, "Exploration of the "scaffolding" teaching model based on information technology in oral English classes," *Development Pedagogy* (Chinese Version), vol. 3, no. 7, pp. 64-66, 2022, doi: 10.12184/wspfzjyxWSP2634-793818.20220307.
- [2] J. Xu and C. Zhao, "The role of large language models in English teaching," *Frontiers of Foreign Language Education in China*, vol. 7, no. 1, pp. 3-10, 2024, doi: 10.20083/j.cnki.fleic.2024.01.003.
- [3] E. Kartchava, E. Gatabontan, A. Ammar, and P. Trofimovich, "Oral corrective feedback: Pre-service English as a second language teachers' beliefs and practices," *Language Teaching Research*, vol. 24, no. 2, pp. 220-249, 2020, doi: 10.1177/1362168818787546.
- [4] A. Kholis, "Elsa speak app: Automatic speech recognition (ASR) for supplementing English pronunciation skills," *Pedagogy: Journal of English Language Teaching*, vol. 9, no. 1, pp. 1-14, 2021, doi: 10.32332/joelt.v13i2.
- [5] S. Zhou and W. Liu, "English grammar error correction algorithm based on classification model," *Complexity*, vol. 2021, no. 1, Art. no. 6687337, 2021, doi: 10.1155/2021/6687337.
- [6] S. McCrocklin and I. Edalatshams, "Revisiting popular speech recognition software for ESL speech," *Tesol Quarterly*, vol. 54, no. 4, pp. 1086-1097, 2020, [Online]. Available: <https://www.jstor.org/stable/48642909>.
- [7] M. Y. C. Jiang, M. S. Y. Jong, W. W. F. Lau, C. S. Chai, and N. Wu, "Exploring the effects of automatic speech recognition technology on oral accuracy and fluency in a flipped classroom," *Journal of Computer Assisted Learning*, vol. 39, no. 1, pp. 125-140, 2023, doi: 10.1111/jcal.12732.
- [8] L. Cheng, P. Ben, and Y. Qiao, "Research on automatic error correction method in English writing based on deep neural network," *Computational Intelligence and Neuroscience*, vol. 2022, no. 1, Art. no. 2709255, 2022, doi: 10.1155/2022/2709255.
- [9] H. U. Sakiroglu, "Oral Corrective Feedback Preferences of University Students in English Communication Classes," *International Journal of Research in Education and Science*, vol. 6, no. 1, pp. 172-178, 2020, doi: 10.46328/ijres.v6i1.806.
- [10] T. T. N. Ngo, H. H. J. Chen, and K. K. W. Lai, "The effectiveness of automatic speech recognition in ESL/EFL pronunciation: A meta-analysis," *ReCALL*, vol. 36, no. 1, pp. 4-21, 2024, doi: 10.1017/S0958344023000113.
- [11] K. Evers and S. Chen, "Effects of an automatic speech recognition system with peer feedback on pronunciation instruction for adults," *Computer Assisted Language Learning*, vol. 35, no. 8, pp. 1869-1889, 2022, doi: 10.1080/09588221.2020.1839504.
- [12] 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.
- [13] R. Raju, P. B. Pati, S. A. Gandheesh, G. S. Sannala, and K. S. Suriya, "Grammatical versus spelling error correction: An investigation into the responsiveness of transformer-based language models using BART and MarianMT," *Journal of Information & Knowledge Management*, vol. 23, no. 3, Art. no. 2450037, 2024, doi: 10.1142/S0219649224500370.
- [14] M. S. I. Malik, A. Nazarova, M. M. Jamjoom, and D. I. Ignatov, "Multilingual hope speech detection: A Robust framework using transfer learning of fine-tuning RoBERTa model," *Journal of King Saud University - Computer and Information Sciences*, vol. 35, no. 8, Art. no. 101736, 2023, doi: 10.1016/j.jksuci.2023.101736.
- [15] M. C. Tudose, S. Ruseti, and M. Dascalu, "Show Me All Writing Errors: A Two-Phased Grammatical Error Corrector for Romanian," *Information*, vol. 16, no. 3, pp. 242, 2025, doi: 10.3390/info16030242.