Intuitive Thinking: Expanding Large Language Models' Thinking for Rapid Decision-Making on Candidate Corrections in Chinese Grammar Error Correction

Lintao Long[1,2,3], **Ruizhang Huang**[1,2,3]*, **Ruina Bai**[1,2,3], **Yongbin Qin**[1,2,3]*, **Qihang Fu**[1,2,3]

[1]Text Computing & Cognitive Intelligence Engineering Research Center of National Education Ministry, Guizhou University
[2]Laboratory of Big Data, Guizhou University, Guiyang, China
[3]College of Computer Science and Technology, Guizhou University, Guiyang, China
tl6550099@gmail.com, rzhuang@gzu.edu.cn, rnbai@gzu.edu.cn, ybqin@gzu.edu.cn, qihangfoo@gmail.com

Abstract

Chinese Grammar Error Correction (CGEC) aims to identify and correct grammatical errors in Chinese sentences. Fine-tuning Large Language Models (LLMs) is a popular current method. However, we have observed a significant flaw: LLMs learn grammatical knowledge but often fail to explicitly use specific grammatical concepts to correct erroneous sentences, leading to multiple corrections without a clear indication of which is the most reliable. Humans possess an "intuitive thinking" mode, which allows them to quickly decide which correction is more reliable based on experience and intuition. To address this deficiency in LLMs, we propose the ***Expanding Intuitive Thinking Model (ExIT)***. ExIT extends the thinking process of LLMs for CGEC, providing them with a human-like rapid decision-making process. This enables LLMs to quickly select a more reliable correction from multiple alternatives based on experience and intuition. Unlike the LLM decoding process, which focuses only on the trustworthiness of local tokens, this is a global thinking process concerning the erroneous sentence and its correction. ExIT is a lightweight model that performs rapid computations without significantly increasing overhead. Our experimental results on CGEC datasets demonstrate that the proposed ExIT can substantially unleash the error correction potential of LLMs.

Code — https://github.com/TLL1213/ExIT-main

Introduction

Grammatical Error Correction (GEC) aims to identify and correct grammatical errors in sentences. The GEC task can serve not only as a preliminary step for tasks such as language learning(Katinskaia and Yangarber 2021; Caines et al. 2023), automatic speech recognition(Liao et al. 2023), and text data annotation(Sun et al. 2021), but also as a service to industries like education, media, and publishing(Wang et al. 2021). Chinese Grammatical Error Correction (CGEC) specifically refers to grammatical error correction in the Chinese language, which can be broadly categorized into several error types(Ma et al. 2022; Xu et al. 2022): Incorrect Word Collocation (IWC), Component Missing (CM), Component Redundancy (CR), Structure Confusion (SC), Incorrect Word Order (IWO), Illogical (ILL), and Ambiguity (AM).

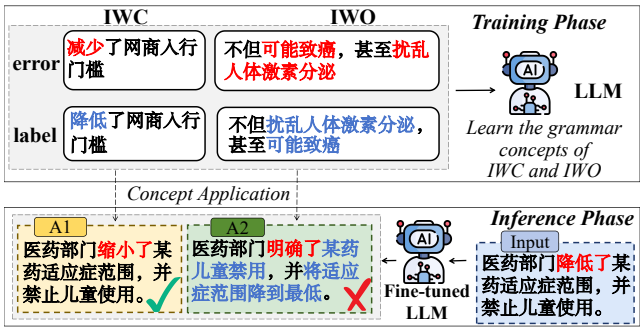


Figure 1: LLMs acquire two grammatical concepts, IWC and IWO, during their training phase. During inference, LLMs are unable to determine which grammatical concept is more appropriate for a given input. A red "×" indicates an incorrect concept mapping, while a green "✓" signifies a correct concept mapping.

Current approaches to GEC primarily rely on three paradigms: sequence-to-sequence (Seq2Seq), sequence-to-edit (Seq2Edit), and decoder-only large language models (LLMs). Seq2Seq(Junczys-Dowmunt et al. 2018; Ge, Wei, and Zhou 2018; Zhou et al. 2019) methods treat GEC as a translation task, offering greater flexibility and transferability. However, they may suffer from issues such as repetitive generation. Seq2Edit(Awasthi et al. 2019; Stahlberg and Kumar 2020; Omelianchuk et al. 2020; Li et al. 2023) methods frame GEC as a sequence editing task, where controllable edits often lead to higher precision. Nevertheless, they involve more complex edit designs. In recent years, LLMs(Fan et al. 2023; Liang et al. 2025) have emerged as a revolutionary breakthrough in seq2seq models, demonstrating significant effectiveness across numerous natural language tasks(Chiarello et al. 2024; Fang et al. 2023). The utility of LLMs in CGEC is increasingly being explored by researchers. However, preliminary studies indicate that LLMs struggle to outperform traditional Seq2Seq models

*Corresponding author.

on CGEC tasks(Qu and Wu 2023; Yang and Quan 2024). Untuned LLMs struggle to adapt to GEC tasks. Therefore, we performed necessary supervised fine-tuning on the LLM. During the training phase, LLMs learn various grammatical concepts; however, concept confusion can arise when correcting errors. As illustrated in Figure 1, the LLM could not differentiate between the learned IWO and IWC grammatical concepts, leading to two corrections for the same erroneous sentence, each corresponding to one of these concepts. Due to limitations in reasoning, LLMs can provide multiple corrections simultaneously but cannot reflect on which one is more reliable.

Motivation When concept confusion occurs, LLMs, consistent with humans, offer different corrections. However, humans possess an Intuitive Thinking mode where they select a more reliable option based on experience and intuition. This is a rapid decision-making process, especially crucial in time-constrained scenarios. Inspired by this human Intuitive Thinking mode, we aim to expand the thinking process of LLMs to mitigate the instability caused by concept confusion and unleash their error correction performance.

Challenge Human Intuitive Thinking is a rapid, experience-based decision-making model. This mode is highly suitable for LLMs where inference costs are substantial, as it effectively unleashes error correction capabilities without significantly increasing inference expenses. However, rapid decision-making necessitates maintaining model lightweightness while simultaneously enhancing performance. Reconciling these two aspects presents a significant challenge.

In order to expand the thinking process of LLMs and alleviate the problem of concept confusion, we propose the ***Expanding Intuitive Thinking Model (ExIT)***. When an LLM is uncertain about which grammatical concept to use to correct an incorrect sentence, it will provide as many different corrections as possible. Additionally, based on learned prior knowledge, the LLM will also provide confidence information for each correction. ExIT, with its lightweight architecture[1], integrates the confidence information from the LLM's corrections to quickly decide on a more reliable correction as the final result. In this process, ExIT compares the correlation between the source sentence and each correction. Unlike the LLM's local thinking, which is only based on preceding tokens, ExIT's global thinking is more aligned with how humans check their own answers.

We validated the proposed method on several complex Chinese native speaker (NS) datasets, demonstrating its effectiveness in unlocking the error correction potential of LLMs and mitigating the issue of concept confusion that arises during pure decoding-based learning. In addition, our in-depth analysis shows that ExIT can generalize to different LLMs.

Our contributions are summarized as follows:

- Expanded the thought patterns of LLMs, enabling them with rapid thinking capabilities akin to human intuitive thinking in CGEC tasks.
- We propose ExIT, a lightweight model that significantly unleashes the latent capabilities of LLMs without substantially increasing inference time.
- We propose a novel approach for utilizing LLMs directly as error correctors.

[1]It is the ExIT, not the whole ExIT-LLM system, that is the focus of our lightweight design.

Related Work

In recent years, the CGEC task has garnered significant attention. Currently, traditional CGEC models and Decoder-only LLMs are the primary directions explored for addressing this task. Traditional models are extensively studied due to their stable error correction performance. Meanwhile, leveraging the knowledge background of LLMs to assist in CGEC error correction has become a crucial research direction.

Traditional CGEC Model

Traditional methods are primarily categorized into Seq2Seq and Seq2Edit models. Wu and Wu (2022) incorporated part-of-speech (POS) and semantic class features to enhance seq2seq models. Wang et al. (2024) employed a k-fold cross inference method to construct over-correction data for training a causal model to rewrite GEC model outputs. Li et al. (2022) proposed a novel Sequence-to-Action (S2A) module, combining the advantages of both Seq2Seq and Seq2Edit models. Liu et al. (2024) experimented with the Bart model using multi-reference data. Yang and Quan (2024) introduced an alignment-enhanced corrector addressing the over-correction problem, applicable to Seq2Seq models and decoder-only LLMs. However, seq2seq models incur high inference costs and suffer from issues like repetition and omission due to generating tokens from scratch. Seq2Edit models may have overly complex editing designs, while more general editing designs might require iterative error correction.

CGEC Model based on Decoder-only LLMs

Existing research has explored the error correction capabilities of LLMs. Fang et al. (2023) evaluated ChatGPT's performance on the GEC task, observing high recall but lower scores for other metrics. Fan et al. (2023) proposed a heuristic prompting method for ChatGPT to generate grammatically incorrect sentences. Liang et al. (2025) introduced the EPO method to bridge the gap between LLMs' pre-training objectives and the GEC principle of minimal modification. However, LLMs tend to provide diverse corrections. Previous studies have largely overlooked the issue of response consistency.

Method

We mimic the human Intuitive Thinking mode, extending the LLM's thinking mode to a similar pattern. As shown in Figure 2, we constructed the Expanding Intuitive Thinking Model (ExIT) to expand the LLM's thinking mode. ExIT fully leverages the confidence information derived from LLM's prior grammatical knowledge when reviewing different corrections.

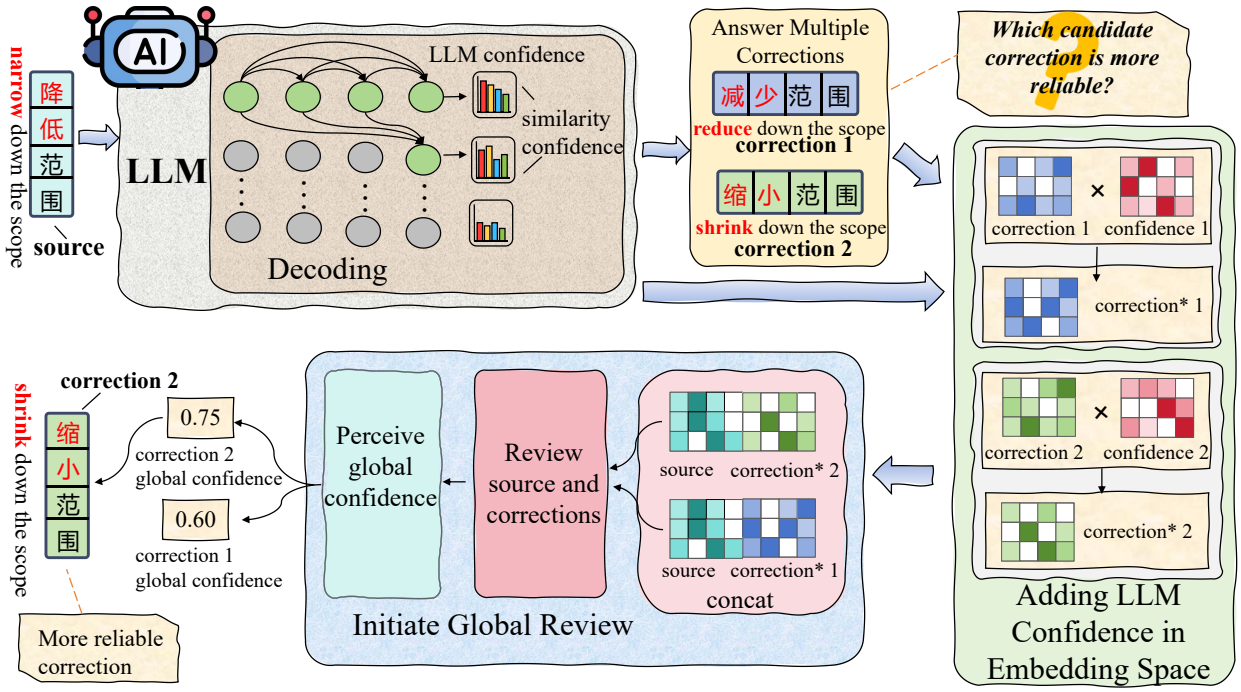


Figure 2: LLMs provide multiple corrections but struggle to determine which is more reliable. ExIT augments the embedding space of different corrections with LLM confidence information. It then jointly reviews the source sentence and the corrections to obtain a global confidence score. `correction` represents the initial correction embedding, while `correction*` represents the correction embedding augmented with LLM confidence.

Problem Definition

CGEC Definition Given a source sentence $X = \{x_1, x_2, ..., x_n\}$ that may contain grammatical errors, the objective of CGEC is to identify and correct the grammatical errors within X and output the corresponding gold sentence $Y = \{y_1, y_2, ..., y_n\}$.

Decoder-only LLMs Due to the complexity of Chinese grammar, LLMs without Supervised Fine-Tuning (SFT) struggle with CGEC tasks. Therefore, we utilize the resource-efficient LoRA(Hu et al. 2022) method for fine-tuning. In decoder-only LLMs, the input is a natural language sequence Z, transformed from X via an instruction template $\mathcal{T}$, and the output is Y.

$$\begin{aligned} Z &= \mathcal{T}(X) \\ &= \{\overbrace{z_1, \ldots, z_t}^{\text{instruction}}, \overbrace{z_{t+1}, \ldots, z_m}^{\text{X}}\} \end{aligned} \tag{1}$$

Due to the use of random sampling methods such as *Top-k*, *Top-p*, and *beam search* during decoding, decoder-only LLMs generate diverse correction candidates $\hat{Y} = \{\hat{y}_1, \hat{y}_2, ..., \hat{y}_m\}$ for a given input x, offering different perspectives based on learned grammatical concepts. As Chinese sentences are typically short, sampling multiple correction candidates does not significantly increase computational overhead. Additionally, LLMs provide decoding confidence, specifically the logarithmic probabilities of each token $\hat{y}_i^j$ within a corrected candidate sentence $\hat{y}_i$. These logarithmic probabilities are computed by the softmax function:

$$\ln p(\hat{y}_i^j) = \ln \frac{e^{\hat{y}_i^j / T}}{\sum_{k=1}^{V} e^{\hat{y}_i^k / T}} \tag{2}$$

Here, T denotes the temperature set during the LLM inference process, and V represents the size of the vocabulary.

LLM Thinking Flaws

We first fine-tuned the LLM for a CGEC task, injecting various grammatical concepts as prior knowledge. As shown in the LLM reasoning section of Figure 2, during correction, the LLM sometimes struggles to accurately determine which grammatical concept the source sentence belongs to, leading to corrections based on different concepts. This approach of generating multiple corrections is consistent with human thought processes. However, the LLM then randomly selects one of these corrections, unable to determine which is more reliable. When humans list multiple corrections, they engage in metacognition, using past experience or intuition to quickly assess the reliability of each correction and then choosing the most reliable one. Human thinking is bidirectional, moving from the source sentence to the correction and then back to re-evaluate the correction against the source. In contrast, the LLM's thinking is unidirectional and, when faced with a choice, tends to randomly follow one path. We introduced a two-stage training process to equip the LLM with a metacognitive step, similar to humans, adding a rapid decision-making process via "Intuitive Thinking."

Algorithm 1: Training Data Construction

Require: $D = \{(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)\}$, where x_i represents the source sentence that may contain grammatical errors, and y_i is the corrected sentence. K represents the number of data partitions.

Ensure: $D_{ExIT} = \{(x_1, \hat{y}_1, p_1), ..., (x_n, \hat{y}_n, p_n)\}$, where $\hat{y}_i$ is the corrected sentence generated by LLM, and p_i is the logarithmic probability representation generated by LLM.

Split D into different subsets D^{Type} based on the type of error;
Divide each subset D^{Type} into K smaller subsets $D_i^{Type}, i \in \{1, 2, ..., K\}$;
Merge the D_i^{Type} with the same index i into K copies $\hat{D}_{split} = \{\hat{D}_1, \hat{D}_2, .., \hat{D}_k\}$;
$D_{ExIT} \leftarrow \{\}$;
for $\hat{D}_i \in \hat{D}_{split}$ **do**
    $D_{train} = \hat{D}_{split} - \hat{D}_i$;
    Train on D_{train} to get the LLM θ_i;
    Obtain the inference results $\hat{Y}$ and P of the LLM θ_i on D_i;
    $\hat{Y}$ is used as a candidate to correct the sentence, P is used as LLM confidence, and combined with $\hat{D}_i$ to obtain D_{merge};
    $D_{ExIT} = D_{ExIT} \cup D_{merge}$;
end for
Return D_{ExIT};

Intuitive Thinking with LLM

In the second stage of training, our objective is to preserve the prior knowledge of the LLM while integrating it into an Intuitive Thinking process. It is crucial to clarify that this represents a continuation of the LLM's inherent thought process, rather than an independent or additional one. Merely providing the source sentence and its correction as a joint input would isolate the thinking process, making subsequent decisions entirely dependent on external models. This scenario is akin to an employee brainstorming various solutions, with the employer making the final decision. As shown in Figure 2, we additionally introduced the logarithmic probabilities generated by the LLM as confidence information, which represents the LLM's thought process. We further expand along this line of thought. Since logarithmic probabilities are a direct product of the LLM's reasoning, their retention allows us to extend the LLM's thinking into an Intuitive Thinking paradigm. This continuation of the LLM's prior knowledge and preliminary thought processes enables even lightweight models to achieve commendable performance, aligning with the characteristics of human Intuitive Thinking.

Transferring the Experience When lightweight models are not pre-trained, they often fail to learn the feature correlations between source sentences and corrections due to insufficient background knowledge, a problem exacerbated by the scarcity of CGEC data. In order for lightweight models to acquire the prior knowledge of LLM, we inject LLM confidence information into the embedding space representation for correction. The lightweight architecture enables ExIT to meet the characteristic of rapid thinking in Intuitive Thinking. Specifically, during the encoding phase, we integrate logarithmic probabilities into the embedding representations, enabling the embeddings to carry the prior knowledge of LLMs and the contemplation for correcting grammatical errors. This completes the preliminary preparation for the transfer of prior knowledge.

Model Architecture Following the propagation of prior knowledge, we designed the ExIT model to initiate the metacognitive process of retrospective correction. ExIT takes the source sentence and the concatenated correction as input, represented as:

$$S = [CLS]X[CAT]\hat{Y} \tag{3}$$

We leveraged the encoder architecture of the transformer model to design a lightweight model. After encoding, we inserted a knowledge injection module to incorporate LLM confidence information into the corrected embedding space representation. Subsequently, the feature representation fused with LLM confidence was fed into the Review module, which simultaneously reviewed the source sentence and the correction from a global perspective to obtain their attention scores. Finally, the source sentence and the corrected attention scores are fed into a global confidence module, which consists of a fully connected layer, to obtain different corrected global confidences. The correction with the larger global confidence is selected as the final output.

Align LLM To further align the LLM, we introduce a novel sentence confidence calculation method to compute the LLM's confidence in an entire candidate sentence during its preliminary thinking, which we consider as the LLM's sentence-level expertise $\hat{p}$. This empirical probability is calculated from logarithmic probabilities, with $\hat{p}$ derived as follows:

$$\hat{p}_i = e^{\sum_{k=1}^{n} \ln p(\hat{y}_i^k)} \tag{4}$$

In the loss calculation, two objectives are introduced: correcting the cosine similarity score with the gold reference, and the LLM's sentence-level confidence $\hat{p}$. $\hat{p}$ is utilized during the training phase to further align with the LLM's thought process, ensuring no deviation occurs when expanding thought patterns. The score enables ExIT to learn more reliable correction features. We use Mean Squared Error (MSE) to compute the loss, with the formula as follows:

$$loss_{cos} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - score_i)^2 \tag{5}$$

$$loss_{llm} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{P}_i)^2 \tag{6}$$

$$loss = \alpha \cdot loss_{cos} + (1 - \alpha) \cdot loss_{llm} \tag{7}$$

Here, α is a hyperparameter, and $1 - \alpha$ in the training objective represents the degree of alignment with the LLM.

Model	FCGEC-test			NaCGEC			NaSGEC-exam		
	P	*R*	$F_{0.5}$	*P*	*R*	$F_{0.5}$	*P*	*R*	$F_{0.5}$
Traditional Model									
BART	63.07	39.95	56.53	62.04	45.84	57.94	60.81	30.09	50.35
BART-AvgL	63.06	39.94	56.52	66.64	41.76	59.54	58.36	33.06	50.61
BART-MinL	65.62	36.79	56.68	68.49	44.82	61.95	60.90	30.40	50.68
BART-AvgL+MinL	65.71	37.78	57.22	67.81	42.44	60.57	61.38	30.78	51.17
SynGEC	63.75	39.78	56.89	62.42	47.41	58.71	-	-	-
BART-Alirector	69.44	36.60	58.88	68.11	43.87	61.33	65.07	25.71	49.82
LLM									
DeCoGLM-10B	55.75	37.91	50.96	54.84	43.96	52.25	47.23	30.66	42.62
Baichuan2-7B-EPO	65.19	39.49	57.68	66.94	48.37	62.16	-	-	-
Qwen2-7B-EPO	66.67	41.93	59.63	67.09	49.97	62.79	-	-	-
Intuitive Thinking									
Qwen2-7B-ExIT	63.32	44.86	58.50	63.54	48.71	59.89	59.75	36.52	53.01
Qwen2.5-7B-ExIT	61.34	42.01	56.17	63.36	47.05	59.26	60.04	37.04	53.41
Qwen2.5-14B-ExIT	63.18	47.03	59.12	66.26	52.50	**62.96**	60.67	40.25	55.08
Qwen3-8B-ExIT	62.59	40.87	56.58	65.26	44.24	59.59	63.11	34.63	54.19
Qwen3-14B-ExIT	65.92	46.44	**60.82**	64.85	48.34	60.70	62.09	38.19	**55.18**

Table 1: Overall results for FCGEC-test, NaCGEC and NaSGEC-exam. The best results are in bold.

Experiments

Datasets and Evaluation Metrics

Dataset For our experiments, we selected a relatively complex and challenging native speaker (NS) dataset. Specifically, for NS experiments, the FCGEC-train dataset(Xu et al. 2022) was utilized as the training set, comprising a total of 41,340 sentences. FCGEC-test served as the primary test dataset, while NaCGEC(Ma et al. 2022) and NaSGEC-exam(Zhang et al. 2023b) were employed as additional test sets.

Evaluation Indicators We adopted (Zhang et al. 2022a)'s configuration to measure the error correction performance of each model using character-level editing metrics. The ChERRANT tool provided by (Zhang et al. 2022a) was utilized to calculate $P/R/F_{0.5}$ for model evaluation.

Dataset Construction

Given an instruction-based error-correction pair $(\mathcal{T}(x), y)$, the LLM generates a set of outputs $\hat{Y} = \{\hat{y}_1, \hat{y}_2, ..., \hat{y}_n\}$ and their corresponding logarithmic probabilities $P = \{p_1, p_2, ..., p_n\}$. Each $\hat{y}_i = \{\hat{y}_i^1, \hat{y}_i^2, ..., \hat{y}_i^m\}$ represents one sampled output from the LLM, and $p_i = \{p_i^1, p_i^2, ..., p_i^m\}$ is its corresponding logarithmic probability. Subsequently, the sampled results are used to construct triplets $(x, \hat{y}_i, p_i)$ as training data. Labels are then obtained by calculating the cosine similarity between each candidate sentence $\hat{y}_i$ and the gold sentence $y*$. Due to the lack of NS datasets, we adopt a K-fold cross-inference method to acquire training data. The dataset was divided into K folds. In each iteration, K-1 folds were used for training to infer the remaining fold, with error type distribution considered during the partitioning process. Algorithm 1 demonstrates the data construction process. We refer to the work of Wang et al. (2024), and we set K to 4.

Model Settings

Model Selection We validate our method using 7-14B size models from the Qwen2.5 and Qwen3 series as base models.
Parameter settings For LLM fine-tuning, a learning rate of 5e-5 was used for 3 epochs, with computation performed in bf16, a context length of 1024, and 2 GPUs, each with a batch size of 2. LoRA was employed for fine-tuning. For ExIT training, hyperparameter α was set to 0.4.

Comparison with Previous Works

To validate the effectiveness of the ExIT model in CGEC tasks, we selected representative and high-performing models from recent years as baselines for performance comparison.

Alirector(Yang and Quan 2024) addresses the issue of overcorrection in LLMs by training an alignment model. BART is frequently employed in GEC tasks; we reference the methods (e.g., AvgL, MinL, AvgL+MinL) proposed by Liu et al. (2024) and Liang et al. (2025) that achieved strong performance on BART for our comparison. EPO(Liang et al. 2025) introduces Edit-wise Preference Optimization (EPO) to bridge the gap between LLMs' pre-training objectives and the principle of minimal edits in GEC. SynGEC(Zhang et al. 2022b) integrates syntactic information into the BART model; we utilize the results reproduced by Liang et al. (2025) on Chinese data. DeCoGLM(Li and Wang 2024) combines detection and correction tasks within a single general language model. Additionally, the performance of the detection-correction structured model (DeGLM-CoGLM) is presented.

Main Results

Table 1 presents a performance comparison of the ExIT model against current state-of-the-art CGEC models on the

FCGEC-test, NaCGEC, and NaSGEC-exam datasets. The highest $F_{0.5}$ in the table are bolded. It can be observed that the ExIT model achieves comparable performance to current leading models on the NS dataset. This demonstrates that ExIT can effectively compensate for LLM reasoning shortcomings, thereby unleashing LLMs' latent error correction potential.

Given the substantial computational overhead of LLMs, we conducted training on smaller datasets and compared our method with others trained on similarly sized datasets. For comprehensive experimental comparison, we also included methods trained on large-scale datasets. Small-scale datasets primarily refer to models trained solely on FCGEC-train, while large-scale datasets additionally utilized Lang8+HSK datasets, totaling approximately 1,568,885 entries. ExIT and EPO demonstrate comparable performance. EPO primarily experiments with 7B models, whereas ExIT mainly uses 14B models as base models. However, there is a significant difference in their training data usage; our method fine-tunes LLMs in approximately 2 hours, while the lightweight ExIT model also has a training time of only about 1 hour. Both methods offer unique advantages: ExIT has low training costs, a lightweight model, does not significantly increase LLM inference time, and has less potential impact on LLM performing other tasks. In contrast, EPO directly enhances LLM performance through multi-stage training and incurs relatively low inference overhead, though it does lead LLMs to specialize more in vertical domains.

Analysis

The Effectiveness of Intuitive Thinking

To verify the effectiveness of ExIT, we conducted an ablation experiment: ***default*** used default hyperparameters; ***tem0*** set the temperature to 0; ***none*** completely omitted LLM prior knowledge; ***llm loss*** added an LLM alignment process; ***injection*** denotes injecting LLM confidence information; and ***ExIT*** denoted both injecting LLM confidence information and incorporating an LLM alignment process. Table 2 shows that without transferring LLM confidence information, the lightweight model failed to learn correct features due to insufficient background knowledge, leading to degraded performance. After transferring LLM confidence information, the model leveraged the rich knowledge of LLM, allowing even lightweight models to learn effective features and significantly improve performance. Further aligning the LLM through loss calculation slightly enhanced performance. Experiments indicate that our ExIT method can transfer LLM knowledge to lightweight small models with minimal computational overhead, thereby extending the LLM's thinking process to facilitate the LLM's review of candidate correction sentences in CGEC. Table 3 presents a case study of ExIT.

The Impact of Data Scale or Model Size

We explored the distinctions between varying data scales and model sizes. For models trained on large-scale data, we referenced the work of (Liang et al. 2025)., which, like ExIT, employs a two-stage training approach. As depicted in Figure 3c(right), ***Model Size*** represents model size, ***Training Time*** indicates training duration, and $F_{0.5}$ denotes performance on FCGEC-test. With large-scale datasets, even a 7B model demands significant training time, substantially exceeding the cost of training a 14B model on small-scale datasets. Regarding performance, a 14B model trained solely on small-scale data performed worse than a 7B model trained on large-scale data. ExIT, with only a minor increase in training cost, enabled the 14B model trained on small-scale data to surpass the performance of the 7B model trained on large-scale data. This suggests that LLMs can acquire sufficient grammatical features from limited samples but suffer from severe concept confusion, which hinders performance.

Model	FCGEC-test		
	P	*R*	$F_{0.5}$
Qwen2.5-14B-default	55.04	46.11	52.98
Qwen2.5-14B-tem0	58.78	47.27	56.05
Qwen2.5-14B-no	52.63	44.02	50.65
Qwen2.5-14B-llm loss	52.62	44.11	50.67
Qwen2.5-14B-feature scaling	62.63	46.71	58.63
Qwen2.5-14B-ExIT	63.18	47.03	59.12

Table 2: Incorporating LLM opinions can effectively enhance error correction performance, particularly by introducing LLM's token-level opinions within the embedding space.

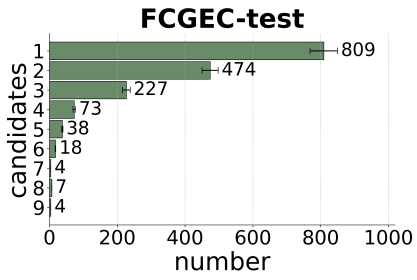
Error Correction Potential of LLM

Several studies have shown that decoder-only LLMs still fail to outperform traditional seq2seq models(Yang and Quan 2024; Zhang et al. 2023a), which contradicts common intuition. Despite being exposed to vast amounts of grammatically correct sentences during pretraining, LLMs often struggle with effective grammatical error correction. Research indicates that this is primarily due to the pretrained LLMs' difficulty in adhering to the principle of minimal edits(Fang et al. 2023), while fine-tuned LLMs tend to suffer from severe concept confusion.

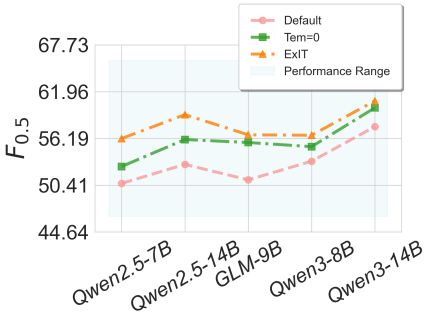
The blue region in Figure 3b illustrates the performance boundary, indicating significant fluctuations in the LLM's error correction performance. The $F_{0.5}$ score difference between the model's worst and best performance reaches 19. While the default hyperparameter settings preserve the diversity of content generated by LLMs, it is difficult for the model to judge the reliability of different corrections. Setting the temperature to 0 enables greedy search, a process that selects only the token with the highest probability at each step. Although this ensures local optimality at each step, it leads to a lack of diversity in the model's output and fails to further improve error correction performance. In summary, LLMs possess immense potential, but current methods struggle to unlock their error correction capabilities. ExIT, however, substantially enhances the LLM's error correction performance without compromising its answer diversity or ability to perform other tasks. Due to the current data limi-

Source	团队有效性的关键因素不只是个体贡献的简单相加。 The key factor in team effectiveness is not merely the simple sum of individual contributions.
LLM	团队有效性不只是个体贡献的简单相加。 Team effectiveness is not merely the simple sum of individual contributions.
ExIT	团队有效性的关键因素不是个体贡献的简单相加。 The key factor in team effectiveness is not the simple sum of individual contributions.
Source	此次活动共举办主题讲座五场，听众达千余人左右。 This event featured five thematic lectures, with an audience of over a thousand people or so.
LLM	此次活动共举办主题讲座五场，听众达千人。 This event featured five thematic lectures, with an audience of a thousand people.
ExIT	此次活动共举办主题讲座五场，听众达千人左右。 This event featured five thematic lectures, with an audience of about a thousand people.

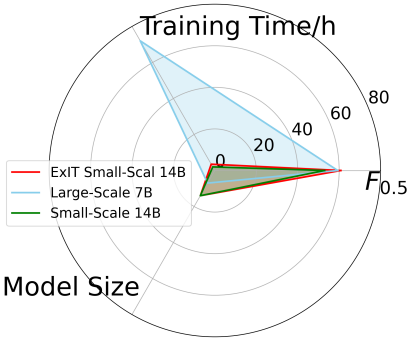
Table 3: Red indicates an error, green indicates a correct answer. `ExIT` represents the final correction chosen after reviewing various correction options.



(a) Candidate correction quantity statistics.



(b) Inspiring the potential of LLM error correction.



(c) Display of training results with different data volumes.

Figure 3: LLM Discussion Legend.

tations, the training data obtained through the K-fold cross-inference method cannot fully reflect the original data distribution. In the future, a larger training dataset can effectively improve the performance of the ExIT method.

Generalizability of ExIT

As depicted in Figure 3a (left), the statistics for Qwen2.5-14B's inference on FCGEC-test are presented, revealing that for over half of the erroneous sentences, the LLM provided multiple corrections. Figure 3b (mid) illustrates the performance variations of different LLMs under default hyperparameters (Default), temperature set to 0 (Tem=0), and after applying the Extended Intuitive Thinking (ExIT) method. The blue region indicates their performance boundaries (ideal maximum/ideal minimum). We applied the ExIT model, trained using Qwen2.5-14B as the base, to other LLMs, consistently achieving performance improvements that surpassed the localized greedy approach of setting the temperature to 0. This phenomenon suggests that LLMs share common challenges and similar probability distributions in the field of GEC. ExIT can effectively leverage the latent knowledge within LLM's probability distributions and generalize to other LLMs. However, the significant disparity in LLM performance boundaries indicates a severe problem with concept confusion.

Conclusion

In this paper, we first investigate a lightweight Expanding Intuitive Thinking Model (ExIT) that mimics the human intuitive thinking process. This method effectively leverages the prior knowledge of LLMs to continue reasoning about the confidence between uncertain corrections. The additional intuitive thinking process effectively mitigates the concept confusion problem in LLMs. Extensive experiments on the FCGEC, NaCGEC, and NaSGEC-exam datasets validate the effectiveness of the proposed method. Detailed analysis further demonstrates the generalizability of this method across different LLMs and highlights the crucial role of ExIT in unleashing LLMs' error correction potential. In the future, we will explore models that further utilize LLMs' prior knowledge to unleash more of their error correction potential.

Acknowledgments

This work was supported by the National Key R&D Program of China, No. 2023YFC3304500 and Natural Science Fund of Guizhou University, No. (2024)31.

References

Awasthi, A.; Sarawagi, S.; Goyal, R.; Ghosh, S.; and Piratla, V. 2019. Parallel iterative edit models for local sequence transduction. *arXiv preprint arXiv:1910.02893*.

Caines, A.; Benedetto, L.; Taslimipoor, S.; Davis, C.; Gao, Y.; Andersen, O.; Yuan, Z.; Elliott, M.; Moore, R.; Bryant, C.; et al. 2023. On the application of large language models for language teaching and assessment technology. *arXiv preprint arXiv:2307.08393*.

Chiarello, F.; Giordano, V.; Spada, I.; Barandoni, S.; and Fantoni, G. 2024. Future applications of generative large language models: A data-driven case study on ChatGPT. *Technovation*, 133: 103002.

Fan, Y.; Jiang, F.; Li, P.; and Li, H. 2023. Grammargpt: Exploring open-source llms for native chinese grammatical error correction with supervised fine-tuning. In *CCF international conference on natural language processing and Chinese computing*, 69–80. Springer.

Fang, T.; Yang, S.; Lan, K.; Wong, D. F.; Hu, J.; Chao, L. S.; and Zhang, Y. 2023. Is chatgpt a highly fluent grammatical error correction system? a comprehensive evaluation. *arXiv preprint arXiv:2304.01746*.

Ge, T.; Wei, F.; and Zhou, M. 2018. Fluency boost learning and inference for neural grammatical error correction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1055–1065.

Hu, E. J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; Chen, W.; et al. 2022. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2): 3.

Junczys-Dowmunt, M.; Grundkiewicz, R.; Guha, S.; and Heafield, K. 2018. Approaching neural grammatical error correction as a low-resource machine translation task. *arXiv preprint arXiv:1804.05940*.

Katinskaia, A.; and Yangarber, R. 2021. Assessing grammatical correctness in language learning. In *Proceedings of the 16th Workshop on Innovative Use of NLP for Building Educational Applications*, 135–146.

Li, J.; Guo, J.; Zhu, Y.; Sheng, X.; Jiang, D.; Ren, B.; and Xu, L. 2022. Sequence-to-action: Grammatical error correction with action guided sequence generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 10974–10982.

Li, W.; and Wang, H. 2024. Detection-correction structure via general language model for grammatical error correction. *arXiv preprint arXiv:2405.17804*.

Li, Y.; Liu, X.; Wang, S.; Gong, P.; Wong, D. F.; Gao, Y.; Huang, H.-Y.; and Zhang, M. 2023. TemplateGEC: Improving grammatical error correction with detection template. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 6878–6892.

Liang, J.; Yang, H.; Gao, S.; and Quan, X. 2025. Edit-Wise Preference Optimization for Grammatical Error Correction. In *Proceedings of the 31st International Conference on Computational Linguistics*, 3401–3414.

Liao, J.; Eskimez, S.; Lu, L.; Shi, Y.; Gong, M.; Shou, L.; Qu, H.; and Zeng, M. 2023. Improving readability for automatic speech recognition transcription. *ACM Transactions on Asian and Low-Resource Language Information Processing*, 22(5): 1–23.

Liu, Y.; Li, Z.; Jiang, H.; Zhang, B.; Li, C.; and Zhang, J. 2024. Towards Better Utilization of Multi-Reference Training Data for Chinese Grammatical Error Correction. In *Findings of the Association for Computational Linguistics ACL 2024*, 3044–3052.

Ma, S.; Li, Y.; Sun, R.; Zhou, Q.; Huang, S.; Zhang, D.; Yangning, L.; Liu, R.; Li, Z.; Cao, Y.; et al. 2022. Linguistic rules-based corpus generation for native Chinese grammatical error correction. *arXiv preprint arXiv:2210.10442*.

Omelianchuk, K.; Atrasevych, V.; Chernodub, A.; and Skurzhanskyi, O. 2020. GECToR–grammatical error correction: tag, not rewrite. *arXiv preprint arXiv:2005.12592*.

Qu, F.; and Wu, Y. 2023. Evaluating the capability of large-scale language models on Chinese grammatical error correction task. *arXiv preprint arXiv:2307.03972*.

Stahlberg, F.; and Kumar, S. 2020. Seq2Edits: Sequence transduction using span-level edit operations. *arXiv preprint arXiv:2009.11136*.

Sun, X.; Ge, T.; Wei, F.; and Wang, H. 2021. Instantaneous grammatical error correction with shallow aggressive decoding. *arXiv preprint arXiv:2106.04970*.

Wang, Y.; Wang, B.; Liu, Y.; Wu, D.; and Che, W. 2024. LM-combiner: A contextual rewriting model for Chinese grammatical error correction. *arXiv preprint arXiv:2403.17413*.

Wang, Y.; Wang, Y.; Dang, K.; Liu, J.; and Liu, Z. 2021. A comprehensive survey of grammatical error correction. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 12(5): 1–51.

Wu, X.; and Wu, Y. 2022. From spelling to grammar: A new framework for Chinese grammatical error correction. *arXiv preprint arXiv:2211.01625*.

Xu, L.; Wu, J.; Peng, J.; Fu, J.; and Cai, M. 2022. FCGEC: Fine-grained corpus for Chinese grammatical error correction. *arXiv preprint arXiv:2210.12364*.

Yang, H.; and Quan, X. 2024. Alirector: Alignment-enhanced Chinese grammatical error corrector. *arXiv preprint arXiv:2402.04601*.

Zhang, Y.; Cui, L.; Cai, D.; Huang, X.; Fang, T.; and Bi, W. 2023a. Multi-task instruction tuning of llama for specific scenarios: A preliminary study on writing assistance. *arXiv preprint arXiv:2305.13225*.

Zhang, Y.; Li, Z.; Bao, Z.; Li, J.; Zhang, B.; Li, C.; Huang, F.; and Zhang, M. 2022a. MuCGEC: a multi-reference multi-source evaluation dataset for Chinese grammatical error correction. *arXiv preprint arXiv:2204.10994*.

Zhang, Y.; Zhang, B.; Jiang, H.; Li, Z.; Li, C.; Huang, F.; and Zhang, M. 2023b. NaSGEC: a multi-domain Chinese grammatical error correction dataset from native speaker texts. *arXiv preprint arXiv:2305.16023*.

Zhang, Y.; Zhang, B.; Li, Z.; Bao, Z.; Li, C.; and Zhang, M. 2022b. SynGEC: Syntax-enhanced grammatical error correction with a tailored GEC-oriented parser. *arXiv preprint arXiv:2210.12484*.

Zhou, W.; Ge, T.; Mu, C.; Xu, K.; Wei, F.; and Zhou, M. 2019. Improving grammatical error correction with machine translation pairs. *arXiv preprint arXiv:1911.02825*.