

A Chain-of-Task Framework for Instruction Tuning of LLMs Based on Chinese Grammatical Error Correction

Xinpeng Liu¹, Bing Xu^{1*}, Muyun Yang^{1*}, Hailong Cao¹,
Conghui Zhu¹, Tiejun Zhao¹, Wenpeng Lu²

¹Faculty of Computing, Harbin Institute of Technology, Harbin, China

²Key Laboratory of Computing Power Network and Information Security, Ministry of Education, Qilu University of Technology (Shandong Academy of Sciences), Jinan, China
22s003002@stu.hit.edu.cn, lwp@qlu.edu.cn,
{hitxb, yangmuyun, caohailong, conghui, tjzhao}@hit.edu.cn

Abstract

Over-correction is a critical issue for large language models (LLMs) to address Grammatical Error Correction (GEC) task, esp. for Chinese. This paper proposes a Chain-of-Task (CoTask) framework to reduce over-correction. The Co-Task framework is applied as multi-task instruction tuning of LLMs by decomposing the process of grammatical error analysis to design auxiliary tasks and adjusting the types and combinations of training tasks. A supervised fine-tuning (SFT) strategy is also presented to enhance the performance of LLMs, together with an algorithm for automatic dataset annotation to avoid additional manual costs. Experimental results demonstrate that our method achieves new state-of-the-art results on both FCGEC (in-domain) and NaCGEC (out-of-domain) test sets.

1 Introduction

Grammatical Error Correction (GEC) is a challenging task of detecting and correcting various grammatical errors in given sentences (Wang et al., 2021). Current advancements in this domain are mostly centered around two approaches: sequence-to-sequence (Seq2Seq) and sequence-to-edit (Seq2Edit). The Seq2Seq models rely on large-scale labeled data, sometimes requiring millions of sentence pairs, to achieve good performance, which leads to extremely high data annotation costs (Cao et al., 2023; Rothe et al., 2021). The Seq2Edit models also have inherent flaws, such as the defect in handling errors involving word order due to their reliance on predicting edit tags (e.g., insertion, deletion, etc.) for each token to generate the corrections (Xu et al., 2022).

Since most research on GEC has focused on Indo-European languages, particularly English, the

*Corresponding Authors: Bing Xu, Muyun Yang.



Figure 1: An example of over-correction, where red indicates a grammatical error, green indicates the correct change, and blue indicates over-correction. CoCGEC learns prior knowledge from instruction data of Chain-of-Task, reducing over-corrections.

performance of Seq2Seq or Seq2Edit GEC models declines further in Chinese. In contrast to English, Chinese grammatical structure is more flexible and does not strictly follow the basic sentence pattern of subject-verb-object. Grammatical relations in Chinese are expressed by word order and context rather than word inflection. In addition, there are relatively few available high-quality datasets for Chinese Grammatical Error Correction (CGEC) training.

Recently, Large Language Models (LLMs) have been emerging as a new paradigm in Natural Language Processing (NLP) owing to their remarkable performance in a variety of downstream tasks (Zhao et al., 2023). Applied as a new possible solution for CGEC, LLMs still demonstrate inferior performance compared with the state-of-the-art models due to the problem of over-correction (Li et al., 2023b; Qu and Wu, 2023). Figure 1 illustrates

this phenomenon, where LLMs add an unnecessary word "frequently" after "be good at reflecting", despite correcting a wrong word order "solving and identifying problems" into "identifying and solving problems". Over-correction leads to unexpected semantic changes, which severely hampers the efficacy of CGEC systems, as it detracts from the primary objective of correcting grammatical errors without introducing unnecessary changes.

To address this challenge, this paper proposes a novel Chain-of-Task (CoTask) framework inspired by Error Analysis Theory in linguistics (Corder, 1975) to reduce over-correction by the LLMs. By decomposing the process of grammatical error analysis, we design three sequential tasks pertinent to CGEC as instruction tuning for LLMs: error range determination, error type classification, and grammatical error correction. We also propose a multi-granularity dataset annotation algorithm to generate the CoTask instruction data automatically. To efficiently utilize the limited data, we apply a two-stage fine-tuning strategy to achieve better performance using the same dataset during SFT. The CGEC system is fine-tuned using the CoTask framework¹.

We evaluate the performance of our system on both in-domain and out-of-domain CGEC tasks, using the FCGEC test set for in-domain evaluation and the NaCGEC test set for out-of-domain evaluation. Our system achieves F0.5 scores of 56.60 and 56.46 on the two test sets, respectively, outperforming the prior state-of-the-art (SOTA) by absolute improvements of 5.3 and 11.2 points.

Specifically, our contributions are as follows:

1. We introduce a novel Chain-of-Task framework to reduce over-correction in LLMs by multi-task instruction tuning. To the best of our knowledge, our method is the first application of CoT to CGEC, employing SFT through decomposing CoT to design training tasks.
2. We propose a two-stage strategy for SFT, which enhances the performance of LLMs without increasing computational demands by designing distinct types and combinations of training tasks at different stages.
3. We develop a multi-granularity integrated text comparison algorithm and construct CoCGEC corpus, which contains additional fine-grained annotated information for future research in CGEC.

2 Chain-of-Task Framework

In linguistics, Error Analysis Theory delineates the process of error analysis into three distinct steps: identifying the error, describing the error, and explaining the error (Corder, 1975). Similarly, addressing NLP tasks such as CGEC involves multiple steps rather than a single step. Before correcting a sentence with a grammatical error $x^{(i)}$ into a corrected sentence $y^{(i)}$, humans typically analyze the grammatical structure, determine the error type, identify possible corrections, etc., to effectively leverage their prior knowledge K and improve correction accuracy.

Inspired by this, we propose the Chain-of-Task (CoTask) framework, consisting of three stages: (1) Chain-of-Task design by task process decomposition, (2) Chain-of-Task instruction construction, and (3) Two-stage Supervised Fine-Tuning, as illustrated in Figure 2. The details of each stage are introduced in this section.

2.1 Chain-of-Task Design by Task Process Decomposition

The prior knowledge required by LMs (K_{LM}) and humans (K_{human}) to address the same target task is not identical. Consequently, leveraging prior knowledge based on human experience does not always enhance LMs' performance on the target task (Rei and Yannakoudakis, 2017). The emergence of Chain-of-Thought (CoT) reasoning in LLMs provides insight into how these models solve tasks, enabling the selection of prior knowledge necessary for the model. CoT enables a complex reasoning task $t^{(i)} = (x^{(i)}, y^{(i)})$ to be decomposed into a series of intermediate reasoning steps $S = \{s_1, s_2, \dots, s_n\}$ leading to the final result $\hat{y}^{(i)}$. This process not only improves model interpretability but also enhances performance (Li et al., 2023a; Wei et al., 2022). Although CGEC does not naturally decompose into clear intermediate steps like mathematical or symbolic reasoning and other natural language reasoning tasks, LLMs can generate content with the process of grammatical error analysis by using "step-by-step" prompts, thereby incorporating the necessary prior knowledge K_{LM} .

For CGEC, outputs from four LLM products—GPT-4o², ChatGLM4³, Qwen2.5⁴ and

¹Our code, data, and models are publicly available at <https://github.com/lxp991108/CoCGEC>

²<https://chat.openai.com/>

³<https://chatglm.cn/main/alltoolsdetail>

⁴<https://tongyi.aliyun.com/qianwen/>

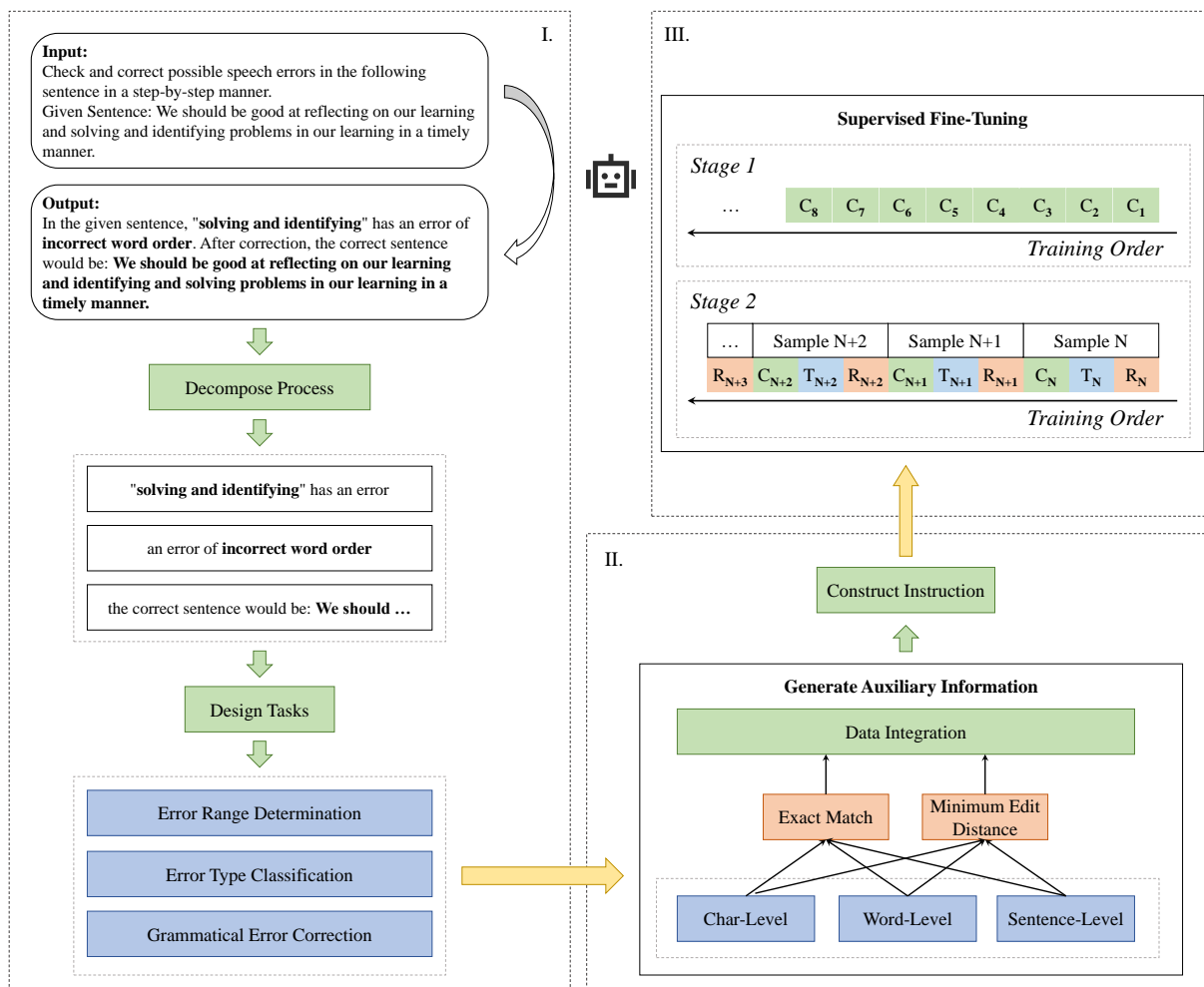


Figure 2: Chain-of-Task framework, consisting of (I) Chain-of-Task design by task process decomposition, (II) Chain-of-Task instruction construction, and (III) Two-stage supervised fine-tuning. R_1 , T_1 , and C_1 denote the instruction data for the error range determination task, error type classification task, and grammatical error correction task for the 1st sample, respectively.

ERNIE3.5⁵—exhibit a similar process. Detailed results are provided in Appendix A.1. Based on LLMs’ outputs, we decompose the CGEC process into three steps. Firstly, the model identifies whether any segment of the given sentence contains a grammatical error. Subsequently, it classifies the identified grammatical error into a specific error type. Lastly, it corrects the grammatical error based on its type, ultimately generating a grammatically correct sentence.

Following this decomposition, we design a Chain-of-Task containing three tasks pertinent to CGEC: **error range determination**, **error type classification**, and **grammatical error correction**. These tasks align directly with the LLMs’ output process steps. Designing auxiliary tasks by leveraging inherent LLMs’ outputs is more effective and

efficient than artificially designing unique auxiliary tasks for each distinct target task based on human experience. This method allows LLMs to leverage the prior knowledge required to address the target task more organically and generalize better across various target tasks.

2.2 CoCGEC: Chain-of-Task Instruction Construction for CGEC

Training data for these tasks needs to contain prior knowledge $K_{LM} = (r, t, c)$ where r represents the grammatical error range, t denotes the error type, and c is the correction. Utilizing annotated datasets directly circumvents the potential limitations imposed by the performance of a teacher model, as opposed to relying on outputs with CoT generated by the teacher model. A dataset for CGEC with

⁵<https://yiyao.baidu.com/>

error type annotations, FCGEC⁶, is constructed by Xu et al. (2022), but datasets for error range determination are less accessible. Consequently, we extended the annotations of the FCGEC training set. A multi-granularity integrated text comparison algorithm is developed to automatically annotate the range of grammatical errors across three granularity levels: char-level, word-level, and sentence-level. We use this algorithm to construct a Chain-of-Task corpus for CGEC, CoCGEC.

- **Char-Level.** The char-level algorithm compares differences between the original sentence and the references from both forward and backward, saving the start and end character indexes of the differences in the original sentence to an index list. After comparison with all references, the index list is sorted, and a substring is extracted from the original sentence based on the first and last indexes to generate the final annotation result.
- **Word-Level.** Similar to char-level, but the original sentence and references are segmented into words using Jieba⁷ tool before comparison, with words serving as the basic units.
- **Sentence-Level.** The sentence-level algorithm divides sentences into multiple clauses based on Chinese punctuation marks such as commas, periods, semicolons, etc., with these clauses serving as the basic units to compare and save indexes.

Comparisons with manual annotations reveal that the similarity between algorithmic and manual annotations at different granularity levels varies depending on grammatical error type. For errors related to semantics such as illogical, which often involve an entire clause or even the whole sentence, char-level or word-level algorithms might lose essential semantic information. For errors involving back-to-front correspondences, such as incorrect word collocation or word order, these algorithms will miss these correspondences if there is only one reference. Conversely, sentence-level algorithms may introduce unnecessary information for component redundancy, typically repetitive words or phrases.

Considering different types of errors suit different granularity annotations, we integrate the results of multi-granularity algorithms based on

⁶<https://github.com/xlxwalex/FCGEC>

⁷<https://github.com/fxsjy/jieba>

Exact Match (**EM**) and Minimum Edit Distance (**MED**) metrics. For each error type, the algorithm with the highest EM and the lowest MED is selected as the final annotation method from three granularity levels. The distribution of grammatical error types in CoCGEC is illustrated in Appendix A.2, and the annotations and metrics calculations are detailed in Appendix A.3.

2.3 Two-stage Supervised Fine-Tuning

Fine-tuning LLMs using instruction data across a range of NLP tasks has been shown to enhance their performance on downstream applications (Ouyang et al., 2022; Sanh et al., 2021). To optimize the model for CGEC and improve its generalization capabilities, we employ instruction data across three tasks pertinent to CGEC. Drawing on the principles of Curriculum Learning (CL) Bengio et al. (2009), we design a two-stage SFT strategy that views a batch of training data samples as a whole to measure its difficulty. In this context, task diversity within a batch reflects the complexity of the underlying knowledge, thereby increasing the learning challenge for the model.

In the first stage, we fine-tune the model exclusively on CGEC instruction data, ensuring the model internalizes the specific objective and answer formats required for CGEC. In the second stage, we perform multi-task fine-tuning using instruction data from all three tasks. This approach allows the model to leverage prior knowledge, such as range and type of grammatical error, before addressing CGEC, ultimately enhancing correction accuracy. The task order within each batch is aligned with the CGEC process in LLMs, proceeding from error range determination to error type classification, and finally, grammatical error correction.

Leveraging prior knowledge through multiple inferences is costly for LLMs while generating results containing all intermediate tasks in a single inference is challenging for a 14B-parameter model. This two-stage strategy allows the LLM to integrate prior knowledge $K_{LM}^{(i)} = (\hat{r}^{(i)}, \hat{t}^{(i)}, \hat{c}^{(i)})$ with the given sentence $x^{(i)}$, thereby more accurately estimating the probability

$$P(\hat{y}^{(i)} | x^{(i)}, \hat{r}^{(i)}, \hat{t}^{(i)}, \hat{c}^{(i)})$$

to directly generate a corrected sentence $\hat{y}^{(i)}$.

3 Experiment

3.1 Setting

Dataset	Train	Validation	Test
FCGEC	36266	2000	3000
NaCGEC	-	500	5869

Table 1: Statistics of sentence counts for datasets.

Dataset For fine-tuning LLMs, we utilize only the CoCGEC dataset to construct instruction data. We randomly sample 90% of the dataset for the first stage of training, using grammatical error correction instruction data of these samples. The remaining 10% is used for multi-task instruction data across the three tasks in the second stage. For validation and test sets, we follow prior work (Li et al., 2023b; Qu and Wu, 2023; Wang et al., 2024) and select two widely used and challenging datasets. FCGEC (Xu et al., 2022) is a large-scale fine-grained multi-reference CGEC corpus collected from public school Chinese examinations for native Chinese speakers. NaCGEC (Ma et al., 2022) is a large-scale dataset containing grammatical errors made by native Chinese speakers in real-world scenarios such as exams and news sites, used as the validation and test sets in NLPCC 2023 Shared Task 1⁸. Examples of instructions used in the experiment are detailed in Appendix A.4.

Evaluation Metrics We evaluate models’ performance applying ChERRANT scorer (Zhang et al., 2022), which calculates precision, recall, and F0.5 score for character-level span-based correction. We obtain the same evaluation metrics by submitting the results of the FCGEC test set in CodaLab⁹ online platform.

Hyperparameters We choose Qwen1.5-14B-Chat (Bai et al., 2023) as the foundation model for SFT due to its strong Chinese language capabilities among open-source LLMs. The LLaMA-Factory framework (Zheng et al., 2024) is employed for both model fine-tuning and inference. During fine-tuning, we set the hyperparameters as follows: the initial learning rate is 5e-05, with a cosine learning rate scheduler and a warmup ratio of 0.003. The batch size is 256, and we fine-tune the model

⁸<http://tcci.ccf.org.cn/conference/2023/taskdata.php>

⁹The benchmark publisher deploys the review page at <https://codalab.lisn.upsaclay.fr/competitions/8020>.

for one epoch. The temperature is set to 0.1 for inference.

3.2 Baseline Approaches

To benchmark our method, we compare it against several popular approaches in CGEC, including state-of-the-art models on the FCGEC and NaCGEC datasets. Additionally, we evaluate the performance of larger-scale LLMs.

Seq2Edit GECToR-Chinese (Zhang et al., 2022) is a Chinese variant of GECToR (Omelianchuk et al., 2020) using StructBERT (Wang et al., 2019) as its encoder. GECToR is a widely used and representative Seq2Edit method. STG (Xu et al., 2022) further decomposes the CGEC task into three processing modules: Switch, Tagger, and Generator, which are trained jointly.

Seq2Seq Bart-Large-Chinese (Shao et al., 2024), a Chinese variant of Bart (Lewis et al., 2020), serves as a representative Seq2Seq pre-trained model, demonstrating strong performance in CGEC. GPT2-medium (Radford et al., 2019) is also considered, trained using unsupervised learning.

Post-processing We replicate a PPL-based model ensemble method proposed by Tang et al. (2023), utilizing three different granularities (Sentence-level, Edit-level, Edit-combination). Additionally, LM-Combiner (Wang et al., 2024) mitigates over-correction in CGEC systems by model rewriting based on the original sentence and output, outperforming the SOTA on the FCGEC.

LLMs We include HWCGET (Su et al., 2023) and GrammarGPT (Fan et al., 2023), which ranked 1st and 3rd on NLPCC 2023 Shared Task 1, respectively. We also compare our approach with GPT-4o (Achiam et al., 2023), a high-performing closed-source LLM, and two open-source LLMs with excellent Chinese capabilities, Qwen1.5-32B-Chat (Bai et al., 2023) and Yi-1.5-34B-Chat (Young et al., 2024).

3.3 Main Results

Tables 2 and 3 present a comprehensive comparison of various methods for CGEC across both in-domain and out-of-domain test sets. As shown in the tables, our CoTask framework significantly

¹⁰<https://github.com/masr2000/NaCGEC>

Method	P	R	F
GECToR-Chinese*	30.68	37.14	28.32
STG*	48.19	37.14	45.48
GPT2-medium*	56.71	24.79	45.10
Bart-Large-Chinese*	37.49	38.87	37.76
+ Sentence-level*	55.26	20.23	41.04
+ Edit-level*	58.22	24.12	45.39
+ Edit-combination*	58.16	25.63	46.38
+ LM-Combiner*	55.67	39.04	51.30
Qwen1.5-32B-Chat	22.16	20.09	21.71
Yi-1.5-34B-Chat	14.44	14.99	14.55
GPT-4o	22.33	24.14	22.67
Qwen1.5-14B-Chat	6.25	22.68	7.31
+ CoTask (Ours)	65.65	36.49	56.60

Table 2: Experimental results on FCGEC (in-domain) test sets. Results marked with * are reported from another paper (Wang et al., 2024). The best results of all are listed in **bold**.

Method	P	R	F
GrammarGPT*	45.40	15.15	32.45
HWCGEC*	50.95	31.29	45.26
Qwen1.5-32B-Chat	20.42	19.00	20.12
Yi-1.5-34B-Chat	12.34	13.65	12.58
GPT-4o	20.21	22.78	20.67
Qwen1.5-14B-Chat	6.36	23.47	7.44
+ CoTask (Ours)	62.50	40.72	56.46

Table 3: Experimental results on NaCGEC (out-of-domain) test set. Results marked with * are reported from NLPCC 2023 Shared Task 1¹⁰. The best results of all are listed in **bold**.

improves the performance of Qwen1.5-14B-Chat for CGEC.

In the in-domain results (Table 2), Qwen1.5-14B-Chat suffers from a low precision of 6.25 and an F0.5 score of just 7.31, despite a relatively high recall of 22.68. This imbalance indicates a significant issue with over-correction, where the model produces many incorrect corrections, leading to a very low precision and, consequently, a poor F0.5 score. However, after applying the CoTask framework, the performance metrics undergo a dramatic improvement. Precision jumps to 65.65, an increase of over tenfold, reflecting a substantial reduction in over-corrections. This improvement in precision, coupled with an enhanced F0.5 score of 56.60, demonstrates a better balance between precision and recall.

The out-of-domain results (Table 3) further vali-

date the effectiveness of our approach. Qwen1.5-14B-Chat again shows a low precision of 6.36 and an F0.5 score of 7.44, highlighting the model’s struggle with over-correction. After applying CoTask, the precision rises sharply to 62.50, and the F0.5 score follows suit, increasing to 56.46. This consistency across both in-domain and out-of-domain datasets underscores the robustness of our method, maintaining high precision and strong F0.5 scores even when the model encounters data that differs from what it was trained on.

Compared with other methods, our approach consistently outperforms them, particularly in precision and F0.5 scores. Through supervised fine-tuning of the CoTask framework, our CoCGEC model improves F0.5 scores compared to SOTA models by 5.3 and 11.2 points on the two test sets, and precision by 7.43 and 11.55 points, respectively. By significantly enhancing precision, CoTask ensures that the model’s corrections are both accurate and relevant.

4 Analysis

We conduct a series of experiments and analyses on some details in our proposed CoTask framework to validate the effectiveness of the proposed CoTask framework.

4.1 Effect of annotation granularity.

Method	FCGEC (val.)			
	EM	P	R	F
Direct	47.65	35.43	12.57	25.98
Manual	50.25	35.67	18.34	30.00
Char-Level	50.00	36.53	17.25	29.86
Word-Level	50.70	38.47	17.45	31.00
Sentence-Level	50.15	41.76	15.02	30.80
Multi-Level	48.95	36.73	16.10	29.24
Integration	51.15	40.31	17.91	32.24

Table 4: Performance of different annotation methods on the FCGEC validation set. "Direct" uses only CGEC instruction data for fine-tuning. "Multi-Level" incorporates all three levels of annotation data. "Integration" combines three levels of annotation data based on the metrics.

To validate the performance of the multi-granularity annotation algorithm, a subset of CoCGEC was sampled for both manual and automated annotation using different algorithms. Specifically, we randomly sampled 150 samples

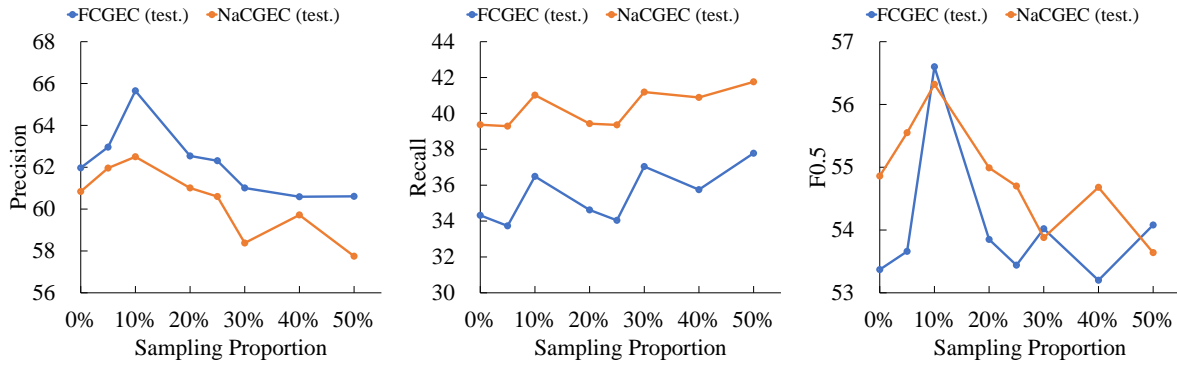


Figure 3: Experimental results on FCGEC and NaCGEC test sets with different sampling proportions.

per error type and 450 error-free samples, resulting in a training subset containing 1500 samples. Three native Chinese-speaking annotators with advanced education annotated the grammatical error ranges in original sentences of the subset according to error type and references, ensuring that the range was minimal and necessary. The subset was used to fine-tune the LLM, with separate experiments conducted using manually annotated data and outputs from the various annotation algorithms.

Table 4 shows the performance of the model using only CGEC instruction data and various annotation algorithms. Regardless of which annotation method is taken, the model consistently performs better in terms of EM, precision, and F0.5 score than the method without any additional annotations, indicating that these annotations for error range determination indeed improve correction accuracy and reduce over-correction. Among all approaches, the integration of multi-granularity annotations achieves the highest performance, surpassing manual annotation by 0.90 points in EM and 2.24 points in F0.5. This demonstrates that our multi-granularity annotation approach not only reduces manual annotation costs but also enhances annotation quality.

4.2 Effect of task order.

The results in Table 5 highlight the significant effect of task order on model performance during the second stage of SFT. The results reveal that the task order aligned with the CGEC process in LLMs achieves the highest F0.5 scores across both datasets, with values of 56.60 and 56.46, respectively. This suggests that beginning with error range determination, followed by error type classification, and concluding with grammatical error correction aligns optimally with the model’s learning

process, facilitating a more effective progression of knowledge transfer between tasks.

In comparison, the Chain-of-Thought (CoT) approach, which combines all tasks within a single sample, exhibits a significant gap in F0.5 scores compared to the other methods. This indicates that relatively small LLMs struggle to directly learn task decomposition and handle subtasks from data in CoT format. Randomized task order also underperforms compared to well-structured orders, further emphasizing the importance of task order design.

While alternative orders such as T-C-R and T-R-C deliver competitive results, they do not surpass the performance of R-T-C. The R-T-C order not only enhances precision but also maintains a strong balance with recall, demonstrating that a logical, progressive order of subtasks maximizes the model’s ability to generalize and correct grammatical errors effectively. These findings validate the efficacy of the carefully designed task order within the CoTask framework.

4.3 Selection of sampling proportion.

LLMs have strong instruction generalization capabilities and usually show excellent performance with relatively little instruction data. Therefore, to determine the optimal sampling proportion for the two stages of supervised fine-tuning, we evaluate various proportions: 5%, 10%, 20%, 25%, 30%, 40%, and 50%. Figure 3 illustrates the experimental results for these different sampling proportions.

The results reveal that when the sampling proportion is below 10%, increasing the proportion improves the model’s precision and enhances its ability to correct more grammatical errors. Although further increasing the sampling proportion can improve recall, it often does so at the expense

Method	FCGEC (test.)			NaCGEC (test.)		
	P	R	F	P	R	F
Random	60.79	35.45	53.18	60.61	40.95	55.30
CoT	56.53	32.24	49.12	56.28	35.45	50.36
C-R-T	63.34	34.37	54.21	61.88	40.31	55.90
C-T-R	63.61	34.86	54.60	60.98	40.17	55.25
R-C-T	62.29	36.28	54.48	61.63	41.83	56.30
R-T-C	65.65	36.49	56.60	62.50	40.72	56.46
T-C-R	63.71	35.74	55.09	61.47	41.16	55.95
T-R-C	64.60	35.24	55.37	61.81	40.48	55.92

Table 5: Performance of different task orders during the second stage of SFT on FCGEC and NaCGEC test sets. "Random" uses instruction data from the three tasks in a shuffled manner, and "CoT" represents the Chain-of-Thought approach, combining data for all three tasks within a single sample. The remaining methods denote specific task orders, e.g., "R-T-C" represents in the order of Error Range Determination, Error Type Classification, and Grammatical Error Correction.

of precision. This phenomenon could be attributed to the model potentially overfitting as it's exposed to more data from a single dataset.

The 10% sampling proportion appears to be optimal, offering a balance between high precision and acceptable recall, which results in the highest F0.5 scores across both datasets. This indicates that a moderate amount of data during SFT allows the model to generalize effectively, striking a balance between precision and recall. Beyond this point, the benefits in recall do not sufficiently compensate for the losses in precision, leading to a decline in overall performance as measured by the F0.5 score.

4.4 Ablation Study

We perform an ablation study to investigate the effectiveness of the CoTask framework's various components. The results, presented in Table 6, reveal the following insights:

Effect of Error Type Classification. Compared to the baseline method, "Direct", introducing instruction data for error type classification has significantly improved the recall of the model, as evidenced by the "Mix-Type" method's performance. This improvement is observed across both test sets, suggesting that more effective error classification enables the model to address a broader range of error types.

Effect of Error Range Determination. Including instruction data for error range determination improves precision, though the increase in recall is less pronounced compared to the "Mix-Type" method. This indicates that error range determination helps in refining the model's precision, while

it does not enhance recall to the same extent.

Effect of Two-stage SFT. Training with shuffled data, as seen in the "Mix-All (Shuffle)" method, achieves a high recall but at the expense of precision, resulting in the precision and F0.5 score even falling below the baseline. This suggests that shuffling leads to data dispersion, causing the model to overfit errors present in the training set and undermining generalization. In contrast, the "Mix-All" method, which employs a two-stage SFT, achieves a balance between precision and recall, resulting in the highest F0.5 scores. This indicates that the two-stage approach stabilizes training and maximizes the benefits of two additional instruction data.

5 Related Work

Instruction-tuning with Chain-of-Thought. Chain-of-Thought reasoning is an emerging capability in LLMs, enhancing complex reasoning and planning capabilities of LLMs with simple prompts like "step by step" (Wei et al., 2022; Kojima et al., 2022). Previous work has centered around utilizing CoT reasoning results for fine-tuning the model itself (Huang et al., 2023; Zelikman et al., 2024) or distilling knowledge into smaller models (Shridhar et al., 2023; Chenglin et al., 2023). However, since these methods directly use CoT reasoning results for knowledge distillation, the model's performance is limited by the teacher model's performance. In addition, these methods also limit the utilization of CoT to reasoning tasks. While the concept of Chain-of-Task (CoTask) has been touched in EcomGPT (Li et al., 2024), it still relies heavily on human experience to design atomic

Method	Range	Type	Two-Stage	FCGEC (test.)			NaCGEC (test.)		
				P	R	F	P	R	F
Direct	✗	✗	✗	61.97	34.32	53.37	60.84	39.37	54.86
Mix-Type	✗	✓	✓	63.08	35.19	54.45	61.33	41.20	55.87
Mix-Range	✓	✗	✓	63.04	35.19	54.43	61.58	40.56	55.80
Mix-All (Shuffle)	✓	✓	✗	60.79	35.45	53.18	60.61	40.95	55.30
Mix-All	✓	✓	✓	65.65	36.49	56.60	62.50	40.72	56.46

Table 6: Results of the ablation study on the FCGEC and NaCGEC test sets. **Range** indicates the inclusion of instruction data for error range determination in the second stage of SFT. **Type** indicates the inclusion of instruction data for error type classification. **Two-Stage** indicates that SFT is performed in two stages.

tasks, which are inefficient and dependent on large labeled datasets. Our approach differs in that it decomposes CoT to design auxiliary tasks, which improves the model’s performance on relatively weak tasks more efficiently and reduces training costs.

Auxiliary task learning for CGEC. Auxiliary task learning has been shown to improve model performance across various NLP tasks (Bryant et al., 2023). However, research on auxiliary tasks in CGEC has been limited, with most research focusing on token-level and sentence-level tasks. At the token level, Rei and Yannakoudakis (2017) introduced auxiliary objectives, such as the learner’s first language, part-of-speech (POS) tags, grammatical relations, etc. At the sentence level, Asano et al. (2019) considered the learner’s proficiency as an auxiliary task. Additionally, some works (Zhao et al., 2019; Yuan et al., 2019) have combined both levels. In contrast, our method does not rely on any human experience about the target task to design auxiliary tasks, which enhances its generalizability to various target tasks.

Curriculum learning on LLMs. Curriculum learning (Bengio et al., 2009) is a method to improve model performance by gradually increasing the difficulty of training samples. Although curriculum learning has been extensively studied, its application to LLMs remains relatively underexplored. Feng et al. (2023) proposed CITING, using a teacher LLM to design the curriculum. Lee et al. (2023) proposed a curriculum learning strategy based on the educational stages of concepts and the cognitive hierarchy outlined in Bloom’s taxonomy. Previous curriculum learning strategies have focused on measuring the difficulty of each sample, our work extends to measuring the difficulty of a batch of samples as a whole, providing a more comprehensive and effective training strategy.

6 Conclusion

This paper proposes a Chain-of-Task framework via multi-task instruction tuning to address the over-correction in LLMs-based CGEC. Utilizing this framework, we outperform previous techniques at lower inference costs. The core idea is to decompose the LLMs’ process of grammatical error analysis to design auxiliary tasks, and then adjust the types and combinations of training tasks for supervised fine-tuning. This method does not rely on any human experience or additional manual annotations, making it a promising solution for reducing over-correction of LLMs.

Limitations

Due to computational and time constraints, our application of the CoTask framework is demonstrated only in the Chinese Grammatical Error Correction task via a light LLM, Qwen1.5-14B-Chat. We will extend it to more NLP tasks, more LLMs, and additional languages in future work. In addition, potential biases in the dataset or model may affect generalizability, and the automated annotation algorithm may introduce errors, particularly with different syntactic structures.

Ethics Statement

Data and models license. The datasets and models used in this paper are publicly available under licenses such as Apache-2.0, with strict assurances that they are used only for academic research. Our datasets, code, and models are also released under this license.

Annotation process. During the annotation process, these annotators will be distributed with original sentences containing grammatical errors, references, and types of grammatical errors. All they need to do is annotate the range of grammatical

errors. We pay the annotator an average of RMB 20 per 500 data points according to the amount of data annotated.

Acknowledgements

This work is supported by National Natural Science Foundation of China (62276077, 62376075, U1908216, 62376076), Key R&D Program of Yunnan, PR China (202203AA080004) and Shenzhen College Stability Support Plan, PR China (No. GXWD20220811170358002). This work is also supported by the Key Laboratory of Computing Power Network and Information Security, Ministry of Education under Grant No. 2023ZD027.

References

- Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Alvenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. [Gpt-4 technical report](#). *arXiv preprint arXiv:2303.08774*.
- Hiroki Asano, Masato Mita, Tomoya Mizumoto, and Jun Suzuki. 2019. [The AIP-tohoku system at the BEA-2019 shared task](#). In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 176–182, Florence, Italy. Association for Computational Linguistics.
- Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. 2023. [Qwen technical report](#). *arXiv preprint arXiv:2309.16609*.
- Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. [Curriculum learning](#). In *Proceedings of the 26th Annual International Conference on Machine Learning, ICML '09*, page 41–48, New York, NY, USA. Association for Computing Machinery.
- Christopher Bryant, Zheng Yuan, Muhammad Reza Qorib, Hannan Cao, Hwee Tou Ng, and Ted Briscoe. 2023. [Grammatical error correction: A survey of the state of the art](#). *Computational Linguistics*, pages 643–701.
- Hannan Cao, Liping Yuan, Yuchen Zhang, and Hwee Tou Ng. 2023. [Unsupervised grammatical error correction rivaling supervised methods](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 3072–3088, Singapore. Association for Computational Linguistics.
- Li Chenglin, Chen Qianglong, Wang Caiyu, and Zhang Yin. 2023. [Mixed distillation helps smaller language model better reasoning](#). *arXiv preprint arXiv:2312.10730*.
- Stephen Pit Corder. 1975. [Error analysis, interlanguage and second language acquisition](#). *Language Teaching & Linguistics: Abstracts*, 8:201 – 218.
- Yaxin Fan, Feng Jiang, Peifeng Li, and Haizhou Li. 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*, pages 69–80. Springer.
- Tao Feng, Zifeng Wang, and Jimeng Sun. 2023. [Citing: Large language models create curriculum for instruction tuning](#). *arXiv preprint arXiv:2310.02527*.
- Jiaxin Huang, Shixiang Gu, Le Hou, Yuexin Wu, Xuezhi Wang, Hongkun Yu, and Jiawei Han. 2023. [Large language models can self-improve](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 1051–1068, Singapore. Association for Computational Linguistics.
- Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. [Large language models are zero-shot reasoners](#). In *Advances in Neural Information Processing Systems*, volume 35, pages 22199–22213.
- Bruce W Lee, Hyunsoo Cho, and Kang Min Yoo. 2023. [Instruction tuning with human curriculum](#). *arXiv preprint arXiv:2310.09518*.
- Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. [BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.
- Xingxuan Li, Ruochen Zhao, Yew Ken Chia, Bosheng Ding, Lidong Bing, Shafiq Joty, and Soujanya Poria. 2023a. [Chain of knowledge: A framework for grounding large language models with structured knowledge bases](#). *arXiv preprint arXiv:2305.13269*.
- Yangning Li, Shirong Ma, Xiaobin Wang, Shen Huang, Chengyue Jiang, Hai-Tao Zheng, Pengjun Xie, Fei Huang, and Yong Jiang. 2024. [Ecomgpt: Instruction-tuning large language models with chain-of-task tasks for e-commerce](#). In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 18582–18590.
- Yinghui Li, Haojing Huang, Shirong Ma, Yong Jiang, Yangning Li, Feng Zhou, Hai-Tao Zheng, and Qingyu Zhou. 2023b. [On the \(in\) effectiveness of large language models for chinese text correction](#). *arXiv preprint arXiv:2307.09007*.
- Shirong Ma, Yinghui Li, Rongyi Sun, Qingyu Zhou, Shulin Huang, Ding Zhang, Li Yangning, Ruiyang Liu, Zhongli Li, Yunbo Cao, Haitao Zheng, and Ying

- Shen. 2022. [Linguistic rules-based corpus generation for native Chinese grammatical error correction](#). In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 576–589, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Kostiantyn Omelianchuk, Vitaliy Atrasevych, Artem Chernodub, and Oleksandr Skurzshanskiy. 2020. [GECToR – grammatical error correction: Tag, not rewrite](#). In *Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 163–170, Seattle, WA, USA → Online. Association for Computational Linguistics.
- Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. [Training language models to follow instructions with human feedback](#). *Advances in neural information processing systems*, 35:27730–27744.
- Fanyi Qu and Yunfang Wu. 2023. [Evaluating the capability of large-scale language models on chinese grammatical error correction task](#). *arXiv preprint arXiv:2307.03972*.
- Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. [Language models are unsupervised multitask learners](#). *OpenAI blog*, 1(8):9.
- Marek Rei and Helen Yannakoudakis. 2017. [Auxiliary objectives for neural error detection models](#). In *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 33–43, Copenhagen, Denmark. Association for Computational Linguistics.
- Sascha Rothe, Jonathan Mallinson, Eric Malmi, Sebastian Krause, and Aliaksei Severyn. 2021. [A simple recipe for multilingual grammatical error correction](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 702–707, Online. Association for Computational Linguistics.
- Victor Sanh, Albert Webson, Colin Raffel, Stephen H Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, et al. 2021. [Multitask prompted training enables zero-shot task generalization](#). *arXiv preprint arXiv:2110.08207*.
- Yunfan Shao, Zhichao Geng, Yitao Liu, Junqi Dai, Hang Yan, Fei Yang, Zhe Li, Hujun Bao, and Xipeng Qiu. 2024. [Cpt: A pre-trained unbalanced transformer for both chinese language understanding and generation](#). *Science China Information Sciences*, 67(5):1–13.
- Kumar Shridhar, Alessandro Stolfo, and Mrinmaya Sachan. 2023. [Distilling reasoning capabilities into smaller language models](#). In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 7059–7073, Toronto, Canada. Association for Computational Linguistics.
- Chang Su, Xiaofeng Zhao, Xiaosong Qiao, Min Zhang, Hao Yang, Junhao Zhu, Ming Zhu, and Wenbing Ma. 2023. [Hwcgec:hw-tsc’s 2023 submission for the nlpc2023’s chinese grammatical error correction task](#). In *Natural Language Processing and Chinese Computing*, page 59–68, Berlin, Heidelberg. Springer Nature Switzerland.
- Chenming Tang, Xiuyu Wu, and Yunfang Wu. 2023. [Are pre-trained language models useful for model ensemble in Chinese grammatical error correction?](#) In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 893–901, Toronto, Canada. Association for Computational Linguistics.
- Wei Wang, Bin Bi, Ming Yan, Chen Wu, Zuyi Bao, Jiangnan Xia, Liwei Peng, and Luo Si. 2019. [StructBERT: Incorporating Language Structures into Pre-training for Deep Language Understanding](#). *arXiv e-prints*, arXiv:1908.04577.
- Yixuan Wang, Baoxin Wang, Yijun Liu, Dayong Wu, and Wanxiang Che. 2024. [LM-combiner: A contextual rewriting model for Chinese grammatical error correction](#). In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 10675–10685, Torino, Italia. ELRA and ICCL.
- Yu Wang, Yuelin Wang, Kai Dang, Jie Liu, and Zhuo Liu. 2021. [A comprehensive survey of grammatical error correction](#). *ACM Transactions on Intelligent Systems and Technology*, 12(5):1–51.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. [Chain-of-thought prompting elicits reasoning in large language models](#). *Advances in Neural Information Processing Systems*, 35:24824–24837.
- Lvxiaowei Xu, Jianwang Wu, Jiawei Peng, Jiayu Fu, and Ming Cai. 2022. [FCGEC: Fine-grained corpus for Chinese grammatical error correction](#). In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 1900–1918, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Alex Young, Bei Chen, Chao Li, Chengen Huang, Ge Zhang, Guanwei Zhang, Heng Li, Jiangcheng Zhu, Jianqun Chen, Jing Chang, et al. 2024. [Yi: Open foundation models by 01. ai](#). *arXiv preprint arXiv:2403.04652*.
- Zheng Yuan, Felix Stahlberg, Marek Rei, Bill Byrne, and Helen Yannakoudakis. 2019. [Neural and FST-based approaches to grammatical error correction](#). In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 228–239, Florence, Italy. Association for Computational Linguistics.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah D. Goodman. 2024. [Star: self-taught reasoner bootstrapping reasoning with reasoning](#). In *Proceedings of the 36th International Conference on Neural Information Processing Systems, NIPS '22*, Red Hook, NY, USA. Curran Associates Inc.

Yue Zhang, Zhenghua Li, Zuyi Bao, Jiacheng Li, Bo Zhang, Chen Li, Fei Huang, and Min Zhang. 2022. [MuCGEC: a multi-reference multi-source evaluation dataset for Chinese grammatical error correction](#). In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3118–3130, Seattle, United States. Association for Computational Linguistics.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. 2023. [A survey of large language models](#). *arXiv preprint arXiv:2303.18223*.

Wei Zhao, Liang Wang, Kewei Shen, Ruoyu Jia, and Jingming Liu. 2019. [Improving grammatical error correction via pre-training a copy-augmented architecture with unlabeled data](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 156–165, Minneapolis, Minnesota. Association for Computational Linguistics.

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, Zheyang Luo, and Yongqiang Ma. 2024. [Llamafactory: Unified efficient fine-tuning of 100+ language models](#). *arXiv preprint arXiv:2403.13372*.

A Appendix

A.1 Output with CoT of LLMs for CGEC

Based on the Chinese language proficiency and comprehensive ability of LLMs, four products are chosen to access the process of LLMs addressing CGEC: GPT-4o, ChatGLM4, Qwen2.5-V3.0.0, and ERNIE3.5-V3.0.0. The instructions used in the input are "Correct any possible grammatical errors in the following sentences in a step-by-step manner and output the correct sentences." The outputs of four products are shown in Figure 4, Figure 5, Figure 6 and Figure 7.

A.2 Distribution of Grammatical Error Type in FCGEC

The distribution of different types of grammatical errors in FCGEC corpus is shown in Table 7, which contains seven different types of grammatical errors, namely, Incorrect Word Order (**IWO**), Incorrect Word Collocation (**IWC**), Component Missing (**MS**), Component Redundancy (**MR**), Structure

Type	Sentence	#Refs
IWO	2851	1.02
IWC	3984	1.49
CM	3860	1.60
CR	1499	1.74
SC	3185	1.46
ILL	3388	1.26
AM	943	1.38
SUM	19710	1.41
NE	16556	1.00

Table 7: Distribution of different types of grammatical errors in FCGEC corpus. **#Refs** indicates the average number of references contained in each sentence on average. **NE** indicates data without any grammatical errors.

Confusion (**SC**), Illogical (**ILL**) and Ambiguity (**AM**). Moreover, each sentence with a grammatical error in this corpus corresponds to one or even more references.

A.3 Results and Metrics of Different Granularity Annotations

We perform manual annotation and automatic annotation of the grammatical error range with different granularity algorithms, and the annotation results are shown in Table 8. Exact Match (**EM**) and Minimum Edit Distance (**MED**) are chosen to measure the similarity between the annotation results of multi-granularity algorithms and the manual annotation results. EM is calculated as the ratio of cases in which the generated string is identical to the target string. MED quantifies the similarity between two strings by calculating the number of insertion, deletion, and modification operations required to transform the generated string into the target string. The specific calculation results are shown in Table 9, and we find that the word-level algorithm is better than the char-level one in all aspects. Compared with the sentence-level algorithm, the word-level algorithm is better than the sentence-level algorithm in the case of CR and AM, and all the other five types of errors have a certain gap with the sentence-level algorithm. After data integration, we obtain an exact match ratio of 38.19% between the final automated annotations and the manual annotations, with a minimum edit distance of 6.38 operations on average.

Algorithm	Example
Ori.	这个地区的雨季很特别，90%都集中在七八两个月。 The rainy season in this region is very special, with 90 percent of it all concentrated in July and August. (Error: It is not clear what "it" refers to: "it" can refer to either "the rainy season" or "this region".)
Ref.	这个地区的雨季很特别，90%的雨季都集中在七八两个月。 The rainy season in this region is very special, with 90 percent of the rainfall all concentrated in July and August.
Manual	90%都集中在七八两个月 with 90 percent of it all concentrated in July and August
Char-Level	%都 percent of it all
Word-Level	90%都 90 percent of it all
Sentence-Level	90%都集中在七八两个月 with 90 percent of it all concentrated in July and August

Table 8: Example of the results of different algorithmic annotations. **Ori.** indicates the original sentence while **Ref.** indicates the references.

Metric	Algorithm	IWO	IWC	CM	CR	SC	ILL	AM	Total
EM	Char-Level	7.95	11.33	8.67	44.00	22.82	12.67	24.67	18.86
	Word-Level	9.93	13.33	8.67	52.67	20.81	14.00	27.33	20.95
	Sentence-Level	19.87	31.33	50.67	13.33	48.99	36.67	24.00	32.10
	Integraion	19.87	31.33	50.67	52.67	48.99	36.67	27.33	38.19
MED	Char-Level	10.44	11.31	14.69	4.49	11.60	9.97	5.66	9.74
	Word-Level	10.39	11.05	13.73	4.29	11.52	9.46	5.03	9.35
	Sentence-Level	10.19	7.17	4.93	10.04	5.68	7.30	9.42	7.82
	Integraion	10.19	7.17	4.93	4.29	5.68	7.30	5.03	6.38

Table 9: Exact Match Ratio (EM) and Minimum Edit Distance (MED) between different algorithmic annotations and manual annotations. For EM, larger values indicate that the annotation results are more similar to the manual annotations. For MED, smaller values indicate greater similarity to the manual annotations.

A.4 Example of Instruction Data

The instructions used for fine-tuning in the experiment of our work consisted of four main types of instruction data, namely, error range determination, error type classification, grammatical error correction, and CoT instruction data combining the three tasks, and examples of the four types of instruction data are shown in Figure 8, Figure 9, Figure 10 and Figure 11. Additionally, we also used instructions for LLMs inference, in which supervised fine-tuned models use the same inference instructions as fine-tuning instructions, whereas the other LLMs such as GPT-4o use instructions as shown in Figure 12.

User:

分步骤地纠正下面句子中可能存在的语病并输出正确的句子：根据麦可思研究院的报告显示，我国2016届大学毕业生自主创业比例为2.9%。

Agent:

好的，下面是分步骤地纠正句子中可能存在的语病的过程：

1. 原句：根据麦可思研究院的报告显示，我国2016届大学毕业生自主创业比例为2.9%。
 - 问题：句子结构不够流畅，存在赘余部分。
2. 改正步骤：
 - 首先，"根据"和"显示"两个动词连用显得不协调。
 - 可以将句子结构简化，使其更简洁明了。
3. 改正后的句子：
 - 根据麦可思研究院的报告，我国2016届大学毕业生的自主创业比例为2.9%。

最终的正确句子是：

根据麦可思研究院的报告，我国2016届大学毕业生的自主创业比例为2.9%。

User:

Correct any possible grammatical errors in the following sentences in a step-by-step manner and output the correct sentences: According to a report by the McKesson Institute shows that the proportion of China's 2016 college graduates who started their own businesses was 2.9 percent.

Agent:

Okay, here's a step-by-step process for correcting possible grammatical errors in a sentence:

1. Original Sentence : According to a report by the McKesson Institute shows, the proportion of China's 2016 college graduates who started their own businesses was 2.9 percent.
 - Problem: The sentence structure is not fluent enough and there are redundant parts.
2. Correction steps:
 - First, the conjunction of the verbs 'according to' and 'shows that' seems incongruous.
 - The sentence structure can be simplified to make it more concise and clear.
3. Corrected sentence:
 - According to a report by the McKesson Institute, the proportion of China's 2016 college graduates who started their own businesses was 2.9 percent.

The final corrected sentence is: **According to a report by the McKesson Institute, the proportion of China's 2016 college graduates who started their own businesses was 2.9 percent.**

Figure 4: Output of GPT-4o for Chinese Grammatical Error Correction.

User:

分步骤地纠正下面句子中可能存在的语病并输出正确的句子：根据麦可思研究院的报告显示，我国2016届大学毕业生自主创业比例为2.9%。

Agent:

原句：根据麦可思研究院的报告显示，我国2016届大学毕业生自主创业比例为2.9%。

纠正步骤：

1. "根据麦可思研究院的报告显示"中的"根据"和"显示"语义重复，可以删除其一。
2. "我国2016届大学毕业生"中的"我国"在此语境中可以省略，因为前文没有提到其他国家的比较。

修改后的句子：麦可思研究院的报告显示，2016届大学毕业生自主创业比例为2.9%。

User:

Correct any possible grammatical errors in the following sentences in a step-by-step manner and output the correct sentence: According to a report by the McKesson Institute shows that the proportion of China's 2016 college graduates who started their own businesses was 2.9 percent.

Agent:

Original sentence: According to a report by the McKesson Institute shows that the proportion of China's 2016 college graduates who started their own businesses was 2.9 percent.

Correction Steps:

1. 'According to' and 'shows that' in 'According to a report by the McKesson Institute shows that' are semantically repetitive, and one of them can be deleted.
2. 'China' in 'China's 2016 college graduates' can be omitted in this context because there is no mention of comparisons to other countries in the previous sentence.

Corrected Sentence: A report by the McKesson Institute shows that the proportion of 2016 college graduates who started their own businesses was 2.9 percent.

Figure 5: Output of ChatGLM4 for Chinese Grammatical Error Correction.

User:

分步骤地纠正下面句子中可能存在的语病并输出正确的句子：根据麦可思研究院的报告显示，我国2016届大学毕业生自主创业比例为2.9%。

Agent:

步骤1：识别语病

原句的语病在于“根据麦可思研究院的报告显示”这个部分，其中“根据”和“显示”两个词造成了句式杂糅，应该进行调整。

步骤2：修正语病

一种修正方式是去掉“根据”或者“显示”，使句子结构清晰。去掉“根据”较为合适，因为报告本身能显示数据，这样句子更加直接。

步骤3：输出正确句子

修正后的句子为：“麦可思研究院的报告显示，我国2016届大学毕业生自主创业比例为2.9%。”

这样，句子的语病就被纠正了，表达也更为准确流畅。

User:

Correct any possible grammatical errors in the following sentences in a step-by-step manner and output the correct sentence: According to a report by the McKesson Institute shows that the proportion of China's 2016 college graduates who started their own businesses was 2.9 percent.

Agent:

Step 1: Identify the problem

The problem with the original sentence lies in the part 'According to a report by the McKesson Institute shows that' where the words 'according to' and 'shows that' create a mishmash of sentences that should be adjusted.

Step 2: Correcting the error

One way to fix the problem is to remove the words 'according to' or 'shows that' to make the sentence clearer. Removing 'according to' is more appropriate because the report itself shows the data, making the sentence more straightforward.

Step 3: Output the corrected sentence

The corrected sentence reads, 'A report by the McKesson Institute shows that the proportion of China's 2016 college graduates who started their own businesses was 2.9 percent.'

In this way, the sentence is corrected and the expression is more accurate and fluent.

Figure 6: Output of Qwen2.5-V3.0.0 for Chinese Grammatical Error Correction.

User:

分步骤地纠正下面句子中可能存在的语病并输出正确的句子：根据麦可思研究院的报告显示，我国2016届大学毕业生自主创业比例为2.9%。

Agent:

原句分析：

句子“根据麦可思研究院的报告显示，我国2016届大学毕业生自主创业比例为2.9%。”的主要问题在于“根据...的报告显示”这一部分的表达冗余。在这里，“根据”和“...的报告显示”两个结构同时出现，造成了句式上的重复和累赘。

修改步骤：

1. 去掉“根据”或“...的报告显示”其中一个，以避免句式冗余。
2. 保留句子中的其他部分，确保信息的完整性和准确性。

修改后的句子：

- 选项一：麦可思研究院的报告显示，我国2016届大学毕业生自主创业比例为2.9%。（去掉了“根据”）
- 选项二：根据麦可思研究院的报告，我国2016届大学毕业生自主创业比例为2.9%。（去掉了“...的报告显示”）

User:

Correct any possible grammatical errors in the following sentences in a step-by-step manner and output the correct sentence: According to a report by the McKesson Institute shows that the proportion of China's 2016 college graduates who started their own businesses was 2.9 percent.

Agent:

Analysis of the original sentence: The main problem of the sentence 'According to a report by the McKesson Institute shows that the proportion of China's 2016 college graduates who started their own businesses was 2.9 percent.' lies in the redundancy of the expression 'according to a report ... shows that'. Here, the two structures of 'according to' and 'shows that' appear simultaneously, causing repetition and redundancy in the sentence.

Correction Steps:

Remove either 'according to' or 'shows that' to avoid redundancy.

Keep the rest of the sentence to ensure completeness and accuracy of the information.

Corrected Sentence:

Option 1: A report by the McKesson Institute shows that the proportion of China's 2016 college graduates who started their own businesses was 2.9 percent. (Remove 'according to')

Option 2: According to a report by the McKesson Institute, the proportion of China's 2016 college graduates who started their own businesses was 2.9 percent. (Removed 'shows that')

Figure 7: Output of ERNIE3.5-V3.0.0 for Chinese Grammatical Error Correction.

Input:

找出下面句子中存在语病的地方:

中小学生近视率普遍比成人高, **主要原因**是课业负担过重、课外活动时间过少**造成的**。

Output:

该句存在语病的地方: 主要原因是课业负担过重、课外活动时间过少造成的

Input:

Identify the range of grammatical error in the following sentence:

Primary and secondary school students have a higher prevalence of myopia than adults, **mainly due to** excessive academic burden and insufficient extracurricular activity time **causing**.

Output:

The range of grammatical error present in the sentence: mainly due to excessive academic burden and insufficient extracurricular activity time causing

Figure 8: Example of instruction data for error range determination task.

Input:

判断下面句子是否含有某种类型的语病错误:

中小学生近视率普遍比成人高, **主要原因**是课业负担过重、课外活动时间过少**造成的**。

Output:

这句话中含有的语病错误类型: 结构混乱

Input:

Determine whether the following sentence contain some type of grammatical error:

Primary and secondary school students have a higher prevalence of myopia than adults, **mainly due to** excessive academic burden and insufficient extracurricular activity time **causing**.

Output:

Type of grammatical error contained in this sentence: Structure Confusion

Figure 9: Example of instruction data for error type classification task.

Input:

纠正下面句子中可能存在的语病并输出正确的句子:

中小学生近视率普遍比成人高, **主要原因**是课业负担过重、课外活动时间过少**造成的**。

Output:

正确的句子: 中小学生近视率普遍比成人高, 主要原因是课业负担过重、课外活动时间过少。

Input:

Correct any possible grammatical error in the following sentence and output the correct sentence:

Primary and secondary school students have a higher prevalence of myopia than adults, **mainly due to** excessive academic burden and insufficient extracurricular activity time **causing**.

Output:

Corrected Sentence: Primary and secondary school students have a higher prevalence of myopia than adults, mainly due to excessive academic burden and insufficient extracurricular activity time.

Figure 10: Example of instruction data for grammatical error correction task.

Input:

分步骤地检查下面句子中是否存在语病，若存在则纠正并输出正确的句子：

中小学生近视率普遍比成人高，**主要原因**是课业负担过重、课外活动时间过少**造成的**。

Output:

该句子存在语病，其中“主要原因是课业负担过重、课外活动时间过少造成的”存在结构混乱的问题。经过纠正后，正确的句子是：中小学生近视率普遍比成人高，主要原因是课业负担过重、课外活动时间过少。

Input:

Check the following sentence in a step-by-step manner for the presence of grammatical error, correct it if present and output the correct sentence:

Primary and secondary school students have a higher prevalence of myopia than adults, **mainly due to** excessive academic burden and insufficient extracurricular activity time **causing**.

Output:

The sentence has a grammatical error, in which the phrase "mainly due to excessive academic burden and insufficient extracurricular activity time causing" is structure confusion. After correction, the correct sentence is: Primary and secondary school students have a higher prevalence of myopia than adults, mainly due to excessive academic burden and insufficient extracurricular activity time.

Figure 11: Example of instruction data for grammatical error correction task with Chain-of-Thought.

User:

你是一个优秀的中文语病纠错模型，你需要识别并纠正输入的句子中可能含有的语病错误并输出正确的句子，纠正时尽可能减少对原句子的改动，并符合最小变化原则，即保证进行的修改都是最小且必要的。你应该避免对句子结构或词汇表达进行不必要的修改。要求直接输出没有语法错误的句子，无需添加任何额外的解释或说明，如果输入的句子中不存在语法错误，则直接输出原句即可：根据麦可思研究院的报告显示，我国2016届大学毕业生自主创业比例为2.9%。

Agent:

根据麦可思研究院的报告，我国2016届大学毕业生自主创业比例为2.9%。

User:

You are an excellent Chinese Grammatical Error Correction model, you need to identify and correct grammatical errors that may be contained in the input sentences and output correct sentences, correcting them with as little change to the original sentence as possible and in accordance with the principle of least change, which ensures that the modifications made are minimal and necessary. You should avoid making unnecessary changes to sentence structure or lexical expressions. Require direct output of sentences without grammatical errors without adding any additional explanations or clarifications, and if there are no grammatical errors in the input sentence, just output the original sentence directly: According to a report by the McKesson Institute shows that the proportion of China's 2016 college graduates who started their own businesses was 2.9 percent.

Agent:

According to a report by the McKesson Institute, the proportion of China's 2016 college graduates who started their own businesses was 2.9 percent.

Figure 12: Example of instruction data used for LLMs inference.