Towards Explainable Chinese Native Learner Essay Fluency Assessment: Dataset, Tasks, and Method

Anonymous ACL submission

Abstract

Grammatical Error Correction (GEC) is a crucial technique in Automated Essay Assessment (AEA) for evaluating the fluency of essays. However, in Chinese, existing GEC datasets often fail to consider the importance of specific grammatical error types within compositional scenarios, lack research on data collected from native Chinese speakers, and largely overlook cross-sentence grammatical errors. To address these issues, we present **CEFGEC** (Chinese Essay Fluency Grammatical Error Correction), an extensive corpus that focuses on fine-grained and multi-dimensional fluency analysis. Furthermore, we propose a novel Grammatical Error Identification and Correction via Knowledge Distillation (**GEIC-KD**) model to investigate the relationships between multi-dimensional annotated content. Compared to other benchmark models, experimental results illustrate that GEIC-KD outperforms them on our dataset. Our findings also further emphasize the importance of fine-grained annotations in fluency assessment. We will make the corpus and related codes available for research.

1 Introduction

Essay fluency refers to the coherence of a sentence or a whole composition, as well as grammatical accuracy (Yang et al., 2012), serving as a foundational component in Automated Essay Assessment (AEA). The study of essay fluency has significant applications in fields such as education (Gong et al., 2021), text generation (Ahn et al., 2016) and publishing (Wang et al., 2021).

Recent advancements in AEA have integrated Grammatical Error Correction (GEC) to improve explainability (Tsai et al., 2020; Gong et al., 2021), with GEC focusing on automatic text error correction (Bryant et al., 2022). In Chinese AEA, the prevalent Chinese GEC (CGEC) categorizes errors into four modification types (Gong et al., 2021) and make corrections. Subsequently, an overall essay score is conducted based on the errors and other linguistic features. This method, while adding some explainability to the scoring process, offers limited insights for students seeking to understand complex grammatical rules. Moreover, relying on an overall score fails to accurately represent the impact of grammatical errors on essay fluency, as it does not provide a distinct fluency score to gauge the specific effects of these errors on the essays.

The existing CGEC dataset is not directly applicable for assessing essay fluency. **Primarily**, most CGEC methods rely on corpora from Chinese-as-a-second-language (CSL) learners, who are more prone to lexical confusion errors, such as confusing "关爱" and "爱情", both translated as "love" in English (Wang et al., 2022). **Additionally**, existing corpora often derive from online texts, which typically do not adhere to language usage norms and grammars. **Moreover**, the definition of error types is not sufficiently detailed. Recent datasets either predominantly focus on orthographic errors like typos (Zhang et al., 2022, 2023), or solely target syntactic errors like constituent omissions (Xu et al., 2022), which lacks comprehensiveness and diversity. **Lastly**, existing datasets lack annotations for cross-sentence errors (Chollampatt et al., 2019; Yuan and Bryant, 2021), which are common in documents, as illustrated in Figure 1(c) Error 1.

To tackle the issues, we propose an detailed assessment guideline for AEA in fluency and developed the **C**hinese **E**ssay **F**luency **G**rammatical **E**rror **C**orrection (**CEFGEC**) corpus, sourced from primary and secondary school students, encompasses a diverse range of topics, genres, and grades. This dataset addresses limitations in prior work: **Firstly**, it simultaneously annotates essay fluency scores, grammatical error types and the corrected sentences, which facilitates a comprehensive and detailed evaluation of the essay in fluency. **Secondly**, it encompasses 5 coarse-grained and 18 fine-grained grammatical error types, providing a basis

(a) Chinese Essay	(b) English Translation
写给自己的信 亲爱的xxx： Para 1 (Sent 1)很高兴以这样的一种方式与你交谈感想。 [省略] (Sent 9)然后，便是知识点的缺漏。(Sent 10)虽然绝大部分都是因为粗心失分，但你仍有因为知识不熟做错或者做复杂的。(Sent 11)这说明你的复习还有漏洞。(Sent 12)但是，这些都是你宝贵的财富，它们是二模对你来说最重要的东西。(Sent 13)它们给你指明了下一阶段的方向。 Para 2 (Sent 14)你不要担心，二模并不是终点，你还有逆风翻盘的可能。(Sent 15)利用好接下来的时间才是王道。 Para 3 (Sent 16)你要努力调整好心态，让心态接近平常，不要有太大的起伏，可以适当的做一些运动来缓解压力，例如跑步等，你要珍惜现在的每一分，每一秒，现在距离中考只有二十多天了。(Sent 17)在学校的时间已经没有二十天了，我了解你，是一个拖延症患者，希望你在接下来的日子里提高办事效率。 [省略]	Letter to Myself Dear xxx, Para 1 (Sent 1)I'm pleased to share my thoughts with you in this manner. **[Omitted]** (Sent 9)Knowledge gaps were evident. (Sent 10)Although most mistakes stemmed from oversight, there were due to unfamiliarity or over-complication. (Sent 11)This suggests areas for improvement in your review. (Sent 12) However, these are your precious treasures, and they are the most important things to you. (Sent 13) They give you the direction of the next stage. Para 2 (Sent 14)Don't worry; this is not the end, and you can still turn things around. (Sent 15)Making the most of the time ahead is key. Para 3 (Sent 16) You have to work hard to adjust your mentality so that it is close to normal, and don't have too much ups and downs, and you can do some exercise appropriately to relieve stress, such as running, you have to cherish every minute and every second now. It's been more than twenty days. (Sent 17)There are less than 20 days in school, and I know you, are a procrastinator, and I hope you can improve your efficiency in the next few days. **[Omitted]**

(c) Annotation

- **Essay Fluency Grade**: **2**
- **Error Sentence and Corrections**:
  - **Error 1: *Sentence***: Sent 10, Sent 11
    Coarse-grained Error Type: 字符级错误(CL), 成分残缺型错误(IC)
    Fine-grained Error Type: 错用标点(WP), 宾语残缺(OBM)
    Correction:虽然绝大部分都是因为粗心失分，但你仍有因为知识不熟做错或者做复杂的**题目，**这说明你的复习还有漏洞。(Trans: Although most mistakes stemmed from oversight, there were **questions** due to unfamiliarity or over-complication, which suggests areas for improvement in your review.)
  - **Error 2: *Sentence***: Sent 17
    Coarse-grained Error Type: 成分残缺型错误(IC)
    Fine-grained Error Type:主语不明(US)
    Correction:我了解你，**你**是一个拖延症患者，希望你在接下来的日子里提高办事效率。(Trans: I know you, and **you** are a procrastinator. I hope you can improve your efficiency in the next few days.)
  - **Error 3: [Omitted]**

Figure 1: Example of CEFGEC annotation: In (a) and (b), highlighted sections mark errors. Colors distinguish error types: blue for incomplete component error (IC), yellow for character-level errors (CL), and orange for incorrect constituent combination error (ICC). (c) offers detailed annotations, with red in "***Correction***" indicating changes.

for scoring and correction, and offering teachers and students precise insights into writing issues and targeted feedback. **Finally**, it originates from native students and annotates errors from document-level perspectives, which is especially beneficial for a more in-depth study of CGEC.

To further investigate and leverage the relationships among multidimensional annotated content, particularly between error sentences, grammatical error types, and corrected sentences, we proposed a novel method **GEIC-KD** (**G**rammatical **E**rror **I**dentification and **C**orrection via **K**nowledge **D**istillation) to facilitate mutual benefits between these tasks. As suggested in Hinton et al., knowledge distillation is commonly used to train the student model to mimic the well-informed teacher model. Specifically, we achieve this by training a teacher model to capture the relationships between error sentences and corrected sentences, as well as between error sentences and error types. Through knowledge distillation, we transfer the learned knowledge to student model. Experimental results demonstrate the effectiveness of our approach in improving performance on both tasks.

We summarize our contributions as follows:

- We develop a pioneering evaluation specification for AEA in fluency and a dataset, CEFGEC, including fine-grained annotations for various aspects related to essay fluency based on native students' essays. It not only offers valuable data resources for CGEC but facilitates in-depth essay assessments.

- We not only provide comprehensive benchmarks for each task, investigating the performance of current methods, but propose GEIC-KD to further explore the implicit relationships between multiple annotated contents.

- Through experiments, we explore the value of detailed annotations for grading, the optimal benefit between error types and corrections, and the significance of cross-sentence errors.

2 Related Work

2.1 Automatic Essay Fluency Assessment

The assessment of essay fluency was commonly treated as a singular natural language processing (NLP) task. These methods might integrate linguistic features like sentence length and vocabulary complexity to provide scores or ratings for fluency (Mim et al., 2021; Yang et al., 2019), or use language models to calculate sentence probabilities for fluency evaluation (Kann et al., 2018). Some also treated it as GEC task, correcting spelling and grammar errors (Gong et al., 2021; Tsai et al., 2020). They correct grammatical errors from four perspectives: insertion, deletion, modification, and reordering. However, this approach to error definition fails to measure errors from a more abstract grammatical perspective, leaving both students and teachers unable to clearly grasp the issues in writing. Besides, there was a lack of evaluation specifications for assessing essay fluency.

2.2 Grammatical Error Correction

The GEC task aims to automatically detect and correct grammatical errors in sentences. Despite numerous datasets and methods for English GEC, CGEC resources are limited, with only four publicly accessible datasets: CTC-Qua (Zhao et al., 2022), CCTC (Wang et al., 2022), FCGEC (Xu et al., 2022) and NaSGEC (Zhang et al., 2023).

Unlike online texts, written texts place more emphasis on linguistic norms and conventions of language usage, making the study of grammatical errors in written context more rigorous and precise. However, only a subset of FCGEC and NaSGEC is sourced from writing text in educational field. FCGEC consists of multi-choice questions from public school Chinese examinations. It defines 7 error types for annotation. However, it neglects simple grammatical errors such as typos and punctuation mistakes, making the error categorization system not comprehensive. NaSGEC is a multi-domain CGEC dataset, derived from native texts, with data sourced from online texts and sentence error determination questions in Chinese language exams. While it often constructed for the purpose of practicing specific grammar knowledge and may differ from real writing scenarios.

2.3 Knowledge Distillation

In conventional tasks, knowledge distillation plays three key roles: model compression, label smoothing, and domain migration.

Model compression involves transferring knowledge from a large model to a smaller one, reducing size without sacrificing performance. Xia et al. uses knowledge distillation to compress parameters and improve the anti-attack ability of the model.

In knowledge distillation, the teacher model's predictions are referred to as soft labels. The student model enhances its performance by leveraging the dark knowledge contained in these soft labels, which includes inter-class similarity information. Cheng et al. mathematically established that employing soft labels in learning process led to accelerated learning and superior performance for student model, surpassing the optimization learning derived solely from the original data.

Domain migration involves transferring knowledge from teacher model to student model across different domains. Various variants have emerged in recent work. For instance, Wu et al. explore the implicit knowledge between connectives and sense labels by allowing the teacher model to learn how to predict connectives in the presence of hints. This knowledge is then used to guide the student model to predict connectives even in the absence of hints.

Coarse-grained Types	Fine-grained Types
Character-Level Error (CL)	Word Missing (WM), Typographical Error (TE), Missing Punctuation (MP), Wrong Punctuation (WP)
Redundant Component Error (RC)	Subject Redundancy (SR), Particle Redundancy (PR), Statement Repetition(SRP), Other Redundancy (OR)
Incomplete Component Error (IC)	Unknown Subject (US), Predicate Missing (PM), Object Missing (OBM), Other Missing (OTM)
Incorrect Constituent Combination Error (ICC)	Inappropriate Subject-Verb Collocation (ISVC), Inappropriate Verb-Object Collocation (IVOC), Inappropriate Word Order (IWO), Inappropriate Other Collocation (IOC)
Illogical (IL)	Linguistic Illogicality (LIL), Factual Illogicality (FIL)

Table 1: Our guideline adopts 5 coarse-grained and 18 fine-grained error types.

Set	Essay	Error Sent	Chars/Sent	Edits/Ref	Multi Label	Cross Sent
All	501	4,258	46.18	2.80	37.88%	782
Train	350	2,981	45.88	2.74	38.27%	553
Dev	76	630	47.39	2.74	39.31%	106
Test	75	647	46.40	2.93	35.69%	123

Table 2: Data statistics of CEFGEC. **Chars/Sent** indicates the average number of characters per sentence, **Edits/Ref** represents the average edit distance per sentence compared to the original sentence, **Multi Label** signifies the proportion of sentences with multiple labels among those containing errors, and **Cross Sent** indicates the number of cross-sentence errors.

3 Dataset Construction

3.1 Data Collection

The dataset was derived from essays composed by primary and secondary school students. We gathered 501 essays from both exams and daily practice sessions, ensuring a diverse representation in terms of grades, genre, and overall scores assigned by Chinese teachers. The distribution of essay genres and scores can be found in Appendix A. With these authentic essays as data source, we obtained valuable insights into students' writing abilities and common mistakes at different age stages. The wide range of error types and corrections provides a comprehensive understanding of the challenges students encounter when writing essays. As a result, our findings possess strong relevance and applicability to student writing, significantly enhancing the potential impact of our research.

3.2 Annotation Format

For each essay in our corpus, our annotation comprises three components: grading fluency score, identifying error types, and correcting.

3.2.1 Essay Fluency Grading

Essays are graded as excellent, average, and unsatisfactory. This scoring provides a holistic assessment of the essay's fluency. According to the definition in Yang et al., we divided the essay fluency scoring criteria into two parts: the smoothness of the essay and the standardization of language use, which includes native speakers' language intuition and the types and quantities of grammatical errors. Details are shown in Appendix B.

3.2.2 Error Types

Based on prior annotation standards in CGEC (Zhang et al., 2022; Xu et al., 2022) and researching middle school student writings, we devise a new grammatical error annotation schema, detailed in Appendix B. Specifically, we categorize writing errors into character-level and component-level, further subdividing into 5 coarse and 18 fine-grained types, as shown in Table 1. In our corpus, each article consists of a title and body. Annotators identify and label erroneous sentences based on our new schema for fine-grained errors. It's worth noting that one sentence may contain multiple errors, requiring annotators to mark all error types within it. This multifaceted annotation allows for a detailed and comprehensive evaluation of each essay.

3.2.3 Correction

GEC annotation employs two paradigms: error coded and rewriting. As Sakaguchi et al. notes, the former suffers from inconsistent error span definitions and cumbersome modifications for complex sentences, affecting annotation quality. The later offers greater flexibility, which also may hinder the ability to constrain annotators and achieve smooth, minimal changes. Therefore, we merge both methods. For character-level errors, we follow the error coded and annotate the index of the incorrect character and the modified character separately. For component-level errors, we use the rewriting paradigm to deal flexibly with complex revisions and add edit distance as a constraint.

Error Type		Train Num (Perc.)	Dev Num (Perc.)	Test Num (Perc.)
Coarse	**Fine**			
CL	WM	235(5.15%)	47(4.90%)	31(3.29%)
	TE	1169(25.62%)	251(26.15%)	256(27.21%)
	MP	452(9.91%)	88(9.17%)	78(8.29%)
	WP	1183(25.93%)	250(26.04%)	281(29.86%)
RC	SR	17(0.37%)	4(0.42%)	4(0.43%)
	PR	122(2.67%)	19(1.98%)	22(2.34%)
	SRP	21(0.46%)	4(0.42%)	3(0.32%)
	OR	476(10.43%)	98(10.21%)	75(7.97%)
IC	US	316(6.93%)	76(7.92%)	81(8.61%)
	PM	43(0.94%)	11(1.15%)	10(1.06%)
	OBM	65(1.42%)	14(1.46%)	14(1.49%)
	OTM	127(2.78%)	24(2.50%)	25(2.66%)
ICC	ISVC	3(0.07%)	3(0.31%)	2(0.21%)
	IVOC	47(1.03%)	4(0.42%)	3(0.32%)
	IWO	138(3.02%)	21(2.19%)	19(2.02%)
	IOC	138(3.02%)	40(4.17%)	34(3.61%)
IL	FIL	2(0.04%)	1(0.10%)	2(0.21%)
	LIL	9(0.20%)	5(0.52%)	1(0.11%)

Table 3: Distribution of error types in CEFGEC. **Train/Dev/Test Num (Perc.)** denotes the count and percentage of each type in train/dev/test set.

3.3 Annotation Process

The annotation team comprised four undergraduates, four postgraduates in language fields, and four expert reviewers with Chinese teaching experience. They adhered to the minimal change principle, receiving training on specifications before annotation. Initially, one undergraduate and one postgraduate annotated the data, followed by verification and correction by expert reviewers.

3.4 Data Statistics

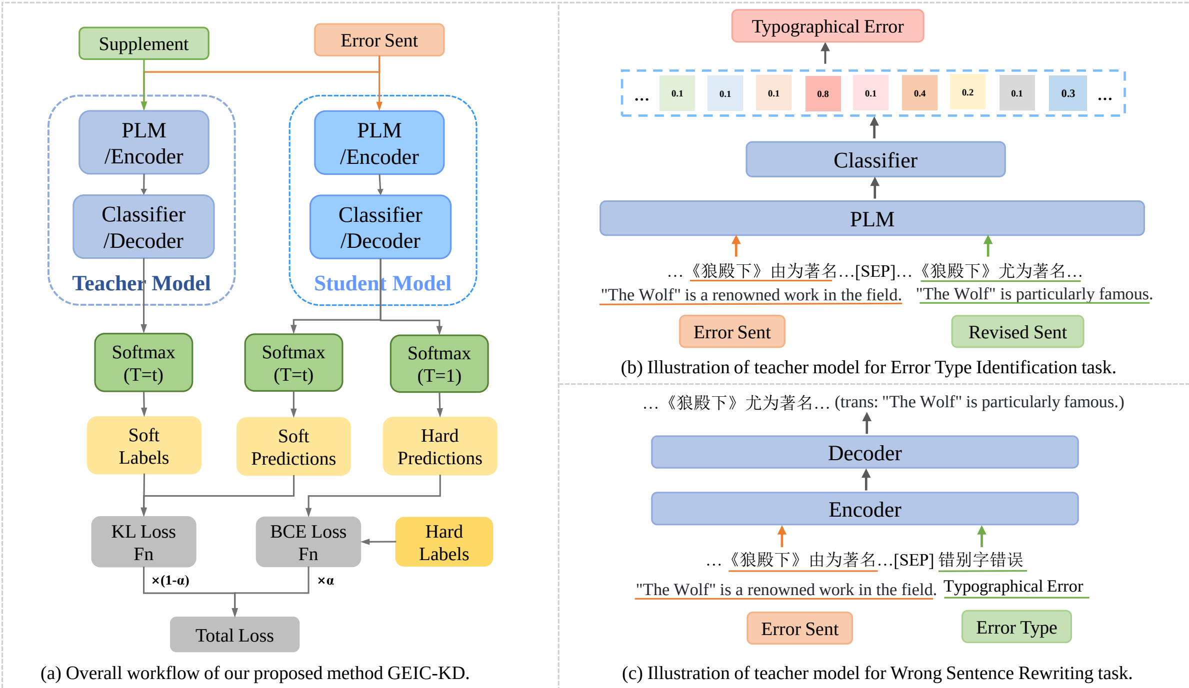
Our dataset includes 501 essays with 9,912 original sentences, of which 4,258 contained errors and underwent modification. The distribution of data can be found in Table 2. Furthermore, in Appendix A, we provide an illustration of the distribution of essay fluency scores (Excellent, Average, Unsatisfactory) across different essay genres. Additionally, Table 3 provides a detailed distribution of coarse and fine-grained error types in the dataset.

3.5 Inner Annotator Agreements

To verify annotation quality, we calculated the Inter-Annotator Agreement using Cohen's Kappa and $F_{0.5}$, with scores of 60.36%, 58.65%, and 62.12% for each task. Details are in Appendix C.

3.6 Ethical Issues

All annotators and expert reviewers were paid for their work. Besides, we have obtained the per-



(a) Overall workflow of our proposed method GEIC-KD.

(b) Illustration of teacher model for Error Type Identification task.

(c) Illustration of teacher model for Wrong Sentence Rewriting task.

Figure 2: Illustration of GEIC-KD. (a) displays the overall workflow of our model. (b) and (c) illustrates the architecture of teacher model for Error Type Identification task and Wrong Sentence Rewriting task. "Revised Sent" in (b) and "Error Type" in (c) correspond to "Supplement" in (a).

mission of the authors and their guardians for all essays used for annotation and publication. We are sincerely grateful for their support.

4 Method

4.1 Tasks

Our task comprises three subtasks: `Essay Fluency Grading` for assessing overall essay fluency, `Error Type Identification` for identifying coarse and fine-grained grammatical errors in sentences, noting their potential multi-label nature due to multiple error types, and `Wrong Sentence Rewriting` for rewriting the incorrect sentences for correction.

4.2 Dual-Information Guided Error Identification and Correction

The dual-information guided method trains error type and correction models correspondingly using ground-truth corrected sentences and error types, providing different inputs for the teacher and student models due to the unavailability of ground-truth data during prediction. Specifically, in the task of `Error Type Identification`, for the student model, we transform a wrong sentence to x_s as input:

$$x_s = T(S), \tag{1}$$

where S indicates the wrong sentence, and T represents the template function.

For the teacher model, illustrated in Figure 2(b), we input the gold corrected sentence and employ a new template to convert it into x_t:

$$x_t = T(S, C), \tag{2}$$

where C represents the gold corrected sentence. This allows the teacher model to learn the relationship between wrong sentences and corrected sentences, facilitating the prediction of error types.

Similarly, for `Wrong Sentence Rewriting` task, we employ equation 1 to transform the wrong sentence into x_s for the student model. For the teacher model, shown in Figure 2(c), we incorporate the error type of the gold sentence as input and use another template to convert it into x_t:

$$x_t = T(S, E), \tag{3}$$

where E indicates the ground-truth error types of S. This setup enables the teacher model to learn the correlation between error types and corrections, guiding the student model in correction tasks.

4.3 Overall Framework

Figure 2 depicts our approach, comprising a teacher model that learns the relationship among error sentences, corrected sentences and error types, and a student (distilled) model that learns vectorized outputs similar to those of the teacher model.

Taking the `Error Type Identification` task as an example. In the training stage, the teacher model aims to accurately predict error types with gold corrected sentences as inputs. The student model requires to predict where extra corrected sentences are missing, mirroring real-world scenarios without ground-truth corrections. It aims to develop a deep semantic understanding of error sentences under the guidance of the knowledgeable teacher model. Consequently, the student model S is required to match not only the ground-truth one-hot labels but also the probability outputs of the teacher model T:

$$\mathcal{L}_S = \alpha \mathcal{L}_{hard} + (1-\alpha)\tau^2 \mathcal{L}_{soft}, \quad (4)$$

where α is the trade-off coefficient between two terms and τ is the temperature rate alleviating category imbalance. $\mathcal{L}_{hard}$ denotes the ground-truth loss using one-hot labels for error type prediction, and $\mathcal{L}_{soft}$ refers to the knowledge distillation loss, employing Kullback-Leibler divergence (Hershey and Olsen, 2007) to measure the difference between student's soft predictions and teacher's soft labels in terms of output distribution:

$$\mathcal{L}_{hard} = -\frac{1}{N}\sum_{i=1}^{N} y_i \log \frac{\exp(\boldsymbol{e}_i)}{\sum_{j=1}^{N}\exp(\boldsymbol{e}_j)}, \quad (5)$$

$$\mathcal{L}_{soft} = \sum_{i=1}^{N} \hat{P}_T(i) \log \frac{\hat{P}_T(i)}{\hat{P}_S(i)}, \quad \hat{P} = softmax(\frac{l}{\tau}), \quad (6)$$

where N is the number of error types, y_i is the gold label, and l is the pre-softmax logits output.

During inference, the trained student model will be used to identify the grammatical error types present in the sentence.

The `Wrong Sentence Rewriting` task employs a parallel approach. During training, the teacher model uses the error sentence and its ground-truth error type as input to understand the relationship among them. The student model takes the error sentence as input and needs to match not only the ground-truth word distribution but the output of the teacher model. In inference, the trained student model generates the corrected sentence when given an error sentence as input.

5 Experiments

5.1 Baseline and Metrics

We use the state-of-the-art (SOTA) pre-trained language models (PLMs) in classification tasks like BERT (Devlin et al., 2018) and RoBERTa (Liu et al., 2019) as benchmark models for grading and error identification task. For wrong sentence rewriting task, we establish baselines with models like Chinese BART (Shao et al., 2021), and Large Language Models (LLMs) including ChatGLM (Du et al., 2022) and ChatGPT (OpenAI, 2022), noted for their text generation capabilities. We also evaluated the performance of LLMs in the first two tasks. For ChatGPT, both zero-shot and few-shot learning are used for all tasks. For ChatGLM, we fine-tune it with LoRA (Hu et al., 2021). Details of prompts and configurations are shown in Appendix H.

Model	P(%)	R(%)	F_1(%)	Acc(%)	QWK
BERT_{base}	**56.74**	46.97	46.76	52.98	0.3868
RoBERTa_{base}	54.97	**58.71**	49.70	49.36	0.3961
BERT_{large}	55.25	49.09	49.08	53.64	**0.4027**
RoBERTa_{large}	56.31	53.94	**54.58**	**57.62**	0.3830
ChatGPT_{0-shot}	56.53	33.54	27.05	42.38	0.1159
ChatGPT_{1-shot}	50.41	38.38	38.09	44.37	0.1650
ChatGLM_{0-shot}	42.67	30.51	33.79	26.16	0.0200
ChatGLM_{1-shot}	44.00	29.31	33.15	31.05	0.0982
ChatGLM_{ft}	47.62	42.32	40.62	46.61	0.2150

Table 4: Results for Essay Fluency Grading task.

Essay Fluency Grading: We frame this problem as a classification task and employed PLMs mentioned previously as our baselines. We evaluate model performance using Precision (P), Recall (R), F_1, Accuracy (Acc) and Quadratic weighted Kappa (QWK) (Vanbelle, 2016).

Error Type Identification: We fine-tune various PLMs on our training dataset, leveraging their powerful language modeling capabilities. Furthermore, we explored the performance of other novel models in CGEC on our dataset like FCGEC (Xu et al., 2022). For evaluation, we assess our models from both coarse and fine-grained perspectives, utilizing P, R, Micro F_1 and Macro F_1 as our evaluation metrics.

Wrong Sentence Rewriting: Inspired by GEC task, we compare two mainstream correction models: Seq2Edit and Seq2Seq model, on our dataset. For Seq2Edit, we use the SOTA model, GECToR (Omelianchuk et al., 2020) and STG-Joint (Xu et al., 2022), as our baselines. For Seq2Seq, we fine-tune Chinese BART on our dataset. For evaluation, we consider the possibility of various corrections and assess from two angles: comparision with ground-truth and the sentence's correctness and rationality. We use metrics like Exact Match (EM), $F_{0.5}$ (Zhang et al., 2022), BLEU, Levenshtein Distance (LD), BERTScore (Zhang et al., 2019), and

Model	CL	RC	IC	ICC	IL	Micro F_1	Macro F_1	Micro			Macro		
								P	R	F_1	P	R	F_1
FCGEC	88.97	25.43	31.33	2.82	0.00	69.25	29.71	38.88	**53.12**	44.90	9.48	13.33	9.52
BERT	87.93	20.00	40.74	7.79	0.00	69.58	31.29	67.18	46.33	54.84	18.68	13.54	15.14
RoBERTa	88.51	25.00	46.23	**14.00**	0.00	70.34	**34.75**	66.67	48.51	56.16	22.84	16.51	18.63
ChatGPT$_{0-shot}$	16.93	21.50	12.79	14.06	0.00	15.41	13.05	8.58	13.26	10.42	9.45	17.31	7.27
ChatGPT$_{3-shot}$	44.64	21.82	4.35	12.21	1.80	25.49	16.96	11.25	13.82	12.40	12.25	14.50	8.51
ChatGLM$_{0-shot}$	0.38	12.99	21.37	0.00	0.47	5.30	7.04	5.09	4.68	4.87	7.18	9.53	4.92
ChatGLM$_{3-shot}$	16.10	25.93	12.57	0.00	0.45	14.91	11.01	5.58	4.99	5.27	11.81	7.67	3.57
ChatGLM$_{ft}$	89.26	24.73	26.25	16.49	0.00	67.75	31.35	52.04	47.06	49.42	18.60	14.63	15.50
Silver$^i_{BERT}$	88.25	13.53	31.11	8.51	0.00	69.90	28.28	62.56	43.15	51.07	22.11	13.19	15.60
Silver$^i_{RoBERTa}$	88.56	12.70	27.91	4.82	0.00	70.14	26.80	67.59	44.31	53.53	21.67	12.89	15.32
Silver$^s_{BERT}$	88.67	14.81	25.45	4.82	0.00	70.23	26.75	61.89	45.38	52.36	25.26	14.78	16.82
Silver$^s_{RoBERTa}$	88.57	11.11	34.55	5.00	0.00	70.57	27.85	**67.99**	46.85	55.47	23.09	13.90	16.39
GEIC-KD$_{BERT}$	89.06	**26.23**	43.00	10.13	0.00	71.32	33.68	67.30	49.20	56.84	23.13	15.22	17.35
GEIC-KD$_{RoBERTa}$	**89.55**	17.78	**46.67**	12.39	0.00	**71.60**	33.28	66.80	52.45	**58.76**	**28.04**	**17.04**	**19.19**

Table 5: Comparison of performance on coarse and fine-grained error type identification. The PLMs involved are all based on the base version. *Silver* represents using the corrected sentences predicted by other models as input.

Model	EM	$F_{0.5}$	BLEU-4	BERTScore	LD	PPL	AvgScore
GECToR	11.47	40.03	90.01	96.95	**0.44**	3.16	56.01
STG-Joint	12.84	26.21	88.61	96.94	1.80	3.32	51.03
BART	18.08	41.21	90.25	97.84	1.67	3.03	57.14
ChatGPT$_{0-shot}$	5.56	16.93	76.74	94.38	8.19	3.79	36.42
ChatGPT$_{3-shot}$	4.64	17.72	79.81	95.60	5.64	2.94	40.86
ChatGLM$_{0-shot}$	1.39	8.56	67.58	91.37	13.27	2.88	26.17
ChatGLM$_{3-shot}$	3.40	4.16	76.22	93.33	2.90	8.90	32.48
ChatGLM$_{ft}$	16.45	40.61	90.50	97.63	1.52	3.12	56.66
Silver$^i_{BART}$	17.31	41.49	90.27	97.89	1.43	2.99	57.32
Silver$^s_{BART}$	17.47	42.01	90.35	97.90	1.40	2.99	57.54
GEIC-KD$_{BART}$	**18.39**	**42.78**	**90.45**	**97.94**	1.57	**2.98**	**57.80**

Table 6: Results on the Wrong Sentence Rewriting task.

Perplexity (PPL), cumulating them into an overall AvgScore. More details are shown in Appendix E.

5.2 Results and Analysis

5.2.1 Essay Fluency Grading

Table 4 presents the performances of different models on `Essay Fluency Grading` task. RoBERTa demonstrate superior abilities in discerning essay fluency, reflecting their proficiency in effectively harnessing contextual information within the text.

5.2.2 Error Type Identification

Table 5 illustrate the performance on `Error Type Identification` task, in terms of both coarse and fine-grained aspects. Compared to baselines, our method further learns the relationship between incorrect sentences and ground truth corrected sentences, leading to improvements. Specifically, we achieved a 1.5% enhancement in both Micro F_1 and Macro F_1 for coarse-grained task, and an approximate 2% improvement for fine-grained task. It indicates that after the teacher model learns the knowledge among incorrect sentences, corrected sentences, and error types, the student model can further acquire this knowledge through knowledge distillation, resulting in enhanced task performance.

We further evaluated the use of corrections predicted by other models as input, aiming to simulate silver corrected sentences available for use in real-world scenarios. Specifically, we compared corrections from the BART baseline model (Siver^i) and our GEIC-KD$_{BART}$ model (Siver^s). Explicitly incorporating predicted corrected sentences resulted in a performance decrease of approximately 1.5% in total compared to the baseline. This decline is attributed to introduced noise, causing the model to learn incorrect relationships between error and corrected sentences. In contrast, our approach not only effectively learns knowledge but avoids the introduction of noise. Furthermore, it can be observed that the better the accuracy of the corrections, the more effective the error identification becomes. This further validates the effectiveness of our method in `Wrong Sentence Rewriting` task.

5.2.3 Wrong Sentence Rewriting

Table 6 shows the `Wrong Sentence Rewriting` task results. GECToR, using a sequence labeling approach, aims for minimal input changes, yielding lower LD values but possibly resulting in less fluent sentences, as indicated by higher PPL scores. STG-Joint designs 3 modules to predict operation tags per character, the number of characters that need to be generated sequentially, and fill in missing characters. Experiments with it highlight our dataset's complexity, as errors are not simply correctable by basic operations. Moreover, a high PPL score

Model	P(%)	R(%)	F_1(%)	Acc(%)	QWK
ChatGPT$_{1-shot}$	50.41	38.38	38.09	44.37	0.1650
ChatGPT$^{\sharp}_{1-shot}$	43.06	41.21	40.34	45.70	0.1933
ChatGLM	47.62	42.32	40.62	46.61	0.2150
ChatGLM$^{\sharp}$	**59.34**	**44.19**	**44.31**	**47.60**	**0.2533**

Table 7: Comparative performance of different setups for Essay Fluency Grading. $\sharp$ indicates the use of all the fine-grained information we annotated.

Model	Micro			Macro		
	P	R	F_1	P	R	F_1
BERT	67.18	46.33	54.84	18.68	13.54	15.14
BERT$^{\spadesuit}$	83.31	76.41	79.71	45.84	38.81	41.56
RoBERTa	66.67	48.51	56.16	22.84	16.51	18.63
RoBERTa$^{\spadesuit}$	**86.27**	**78.19**	**82.03**	**54.53**	**40.04**	**43.74**

Table 8: Results for fine-grained error type identification using correction reference inputs, where $\spadesuit$ denotes results reference application.

Model	EM	$F_{0.5}$	BLEU-4	BERTScore	LD	PPL	AvgScore
BART	18.08	41.21	90.25	97.84	1.67	3.03	57.14
BART$^{\clubsuit}$	18.24	41.80	**90.54**	97.91	**1.48**	**2.97**	57.67
BART$^{\diamondsuit}$	**20.71**	43.00	90.47	**97.94**	1.68	2.98	**58.37**
BART†	19.32	**43.05**	90.18	97.93	1.52	2.98	58.12

Table 9: Results on Wrong Sentence Rewriting task with gold error type as input. $\clubsuit$ and $\diamondsuit$ denotes the model that incorporates the coarse and fine-grained error type into the input, while† represents both being used as inputs.

indicates the results lack fluency in LMs' view.

Furthermore, our model outperforms baselines in most metrics, showing its superiority. We also conducted a comparison by using predicted instead of ground truth error types as input, which exhibits a marginal improvement. However, our knowledge distillation approach, which learning the connections between wrong and corrected sentences and error types, demonstrates a more significant enhancement, highlighting its effectiveness.

5.3 LLMs Results and Analysis

In testing ChatGPT and ChatGLM on tasks, we found few-shot generally outperformed zero-shot. Specifically, ChatGPT was better in both zero and few-shot compared to ChatGLM under similar prompts. In `Essay Fluency Grading` task, we noted a tendency of LLMs to assign the "Excellent" rating, possibly because they lean towards a gentler teaching style. For `Error Type Identification` task, non-finetuned ChatGLM was less effective than ChatGPT, particularly in understanding instructions. For `Wrong Sentence Rewriting` task, while zero-shot corrections kept semantic similarity, they often had substantial character-level changes, leading to overly elaborate rewrites, contradicting our aim for minimal corrections.

6 Discussion

We explores the importance of fine-grained annotation and the performance for teacher models. An in-depth discussion on cross-sentence errors is available in Appendix F.

6.1 Impact of Fine-grained Annotations on Essay Fluency Grading

For `Essay Fluency Grading`, we input detailed annotations, like error types and counts, into the model. Table 7 shows that fine-grained annotations notably improved performance. Particularly, they improved all metrics for the tunable ChatGLM, and notably increased ChatGPT's recall by 2.83%, confirming the benefits of detailed annotation.

6.2 Max Mutual Benefit of Error Type Identification and Correction

This section details how teacher models improved by using explicit prompts. In `Error Type Identification` task, Tables 8 and Appendix G show that including sentences with ground truth corrections significantly improved error identification by 20% in coarse and 25% in fine-grained errors, highlighting the efficacy of using gold corrected sentences in this task.

In the `Wrong Sentence Rewriting` task, Table 9 shows that including ground truth error types enhanced correction performance by 2%. We also examined the impact of coarse and fine-grained error types. The results indicated that coarse-grained types had little effect on performance, while fine-grained types, with their clearer definitions, provided more useful information for corrections, significantly affecting the improvement.

7 Conclusion

In this study, we present CEFGEC, a comprehensive dataset derived from native Chinese student essays. It captures document-level errors, fluency ratings, and granular error details, enriching our insight into student compositions. Our introduced GEIC-KD model analyzes annotated content relationships. Tests validate our methodology's effectiveness. Our work augments AEA by emphasizing the significance of detailed annotations for precise fluency evaluation and showcases LLMs' challenges when assessed on our dataset.

8 Limitation

In this section, we address the limitations of our work. Firstly, grammatical errors are just one of the factors affecting essay fluency; our study has not yet explored instances where grammar is correct but the text is incoherent. Addressing this will be our subsequent focus. Furthermore, considering the impact of prompt quality on LLMs, the range of prompts we tested for assessing LLM performance in our tasks was limited.

References

Emily Ahn, Fabrizio Morbini, and Andrew Gordon. 2016. Improving fluency in narrative text generation with grammatical transformations and probabilistic parsing. In *Proceedings of the 9th International Natural Language Generation Conference*, pages 70–73.

Christopher Bryant, Zheng Yuan, Muhammad Reza Qorib, Hannan Cao, Hwee Tou Ng, and Ted Briscoe. 2022. Grammatical error correction: A survey of the state of the art. *Computational Linguistics*, pages 1–59.

Xu Cheng, Zhefan Rao, Yilan Chen, and Quanshi Zhang. 2020. Explaining knowledge distillation by quantifying the knowledge. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12925–12935.

Shamil Chollampatt, Weiqi Wang, and Hwee Tou Ng. 2019. Cross-sentence grammatical error correction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 435–445.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Zhengxiao Du, Yujie Qian, Xiao Liu, Ming Ding, Jiezhong Qiu, Zhilin Yang, and Jie Tang. 2022. Glm: General language model pretraining with autoregressive blank infilling. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 320–335.

Jiefu Gong, Xiao Hu, Wei Song, Ruiji Fu, Zhichao Sheng, Bo Zhu, Shijin Wang, and Ting Liu. 2021. Iflyea: A chinese essay assessment system with automated rating, review generation, and recommendation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*, pages 240–248.

John R Hershey and Peder A Olsen. 2007. Approximating the kullback leibler divergence between gaussian mixture models. In *2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP'07*, volume 4, pages IV–317. IEEE.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Katharina Kann, Sascha Rothe, and Katja Filippova. 2018. Sentence-level fluency evaluation: References help, but can be spared! *arXiv preprint arXiv:1809.08731*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Farjana Sultana Mim, Naoya Inoue, Paul Reisert, Hiroki Ouchi, and Kentaro Inui. 2021. Corruption is not all bad: Incorporating discourse structure into pretraining via corruption for essay scoring. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:2202–2215.

Kostiantyn Omelianchuk, Vitaliy Atrasevych, Artem Chernodub, and Oleksandr Skurzhanskyi. 2020. Gector–grammatical error correction: tag, not rewrite. *arXiv preprint arXiv:2005.12592*.

OpenAI. 2022. Chatgpt.

Keisuke Sakaguchi, Courtney Napoles, Matt Post, and Joel Tetreault. 2016. Reassessing the goals of grammatical error correction: Fluency instead of grammaticality. *Transactions of the Association for Computational Linguistics*, 4:169–182.

Yunfan Shao, Zhichao Geng, Yitao Liu, Junqi Dai, Hang Yan, Fei Yang, Li Zhe, Hujun Bao, and Xipeng Qiu. 2021. Cpt: A pre-trained unbalanced transformer for both chinese language understanding and generation. *arXiv preprint arXiv:2109.05729*.

Chung-Ting Tsai, Jhih-Jie Chen, Ching-Yu Yang, and Jason S Chang. 2020. Lingglewrite: a coaching system for essay writing. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 127–133.

Sophie Vanbelle. 2016. A new interpretation of the weighted kappa coefficients. *Psychometrika*, 81(2):399–410.

Baoxin Wang, Xingyi Duan, Dayong Wu, Wanxiang Che, Zhigang Chen, and Guoping Hu. 2022. Cctc: A cross-sentence chinese text correction dataset for native speakers. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 3331–3341.

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 (TIST)*, 12(5):1–51.

Hongyi Wu, Hao Zhou, Man Lan, Yuanbin Wu, and Yadong Zhang. 2023. Connective prediction for implicit discourse relation recognition via knowledge distillation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5908–5923.

Peng Xia, Yuechi Zhou, Ziyan Zhang, Zecheng Tang, and Juntao Li. 2022. Chinese grammatical error correction based on knowledge distillation. *arXiv preprint arXiv:2208.00351*.

Lvxiaowei Xu, Jianwang Wu, Jiawei Peng, Jiayu Fu, and Ming Cai. 2022. Fcgec: Fine-grained corpus for chinese grammatical error correction. *arXiv preprint arXiv:2210.12364*.

Min Chul Yang, Min Jeong Kim, Hyoung Gyu Lee, and Hae Chang Rim. 2012. Assessing writing fluency of non-english-speaking student for automated essay scoring: How to automatically evaluate the fluency in english essay. In *4th International Conference on Computer Supported Education, CSEDU 2012*, pages 83–87.

Yiqin Yang, Li Xia, and Qianchuan Zhao. 2019. An automated grader for chinese essay combining shallow and deep semantic attributes. *IEEE Access*, 7:176306–176316.

Zheng Yuan and Christopher Bryant. 2021. Document-level grammatical error correction. In *Proceedings of the 16th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 75–84.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

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. *arXiv preprint arXiv:2204.10994*.

Yue Zhang, Bo Zhang, Haochen Jiang, Zhenghua Li, Chen Li, Fei Huang, and Min Zhang. 2023. Nasgec: a multi-domain chinese grammatical error correction dataset from native speaker texts. *arXiv preprint arXiv:2305.16023*.

Genre	**Fluency Grade (%)**		
	Excellent	**Average**	**Unsatisfactory**
Scenes	72.73	22.73	4.55
Objects	79.17	12.50	8.33
Characterization	28.57	58.57	12.86
Arguments	36.03	51.47	12.50
Reflection	34.48	51.72	13.79
Narrative	27.35	41.88	30.77
Prose	81.82	18.18	0.00
Letter	51.22	41.46	7.32
Total	40.32	44.31	15.37

Table 10: Distribution of fluency grades across different genres, presented as percentages.

Honghong Zhao, Baoxin Wang, Dayong Wu, Wanxiang Che, Zhigang Chen, and Shijin Wang. 2022. Overview of ctc 2021: Chinese text correction for native speakers. *arXiv preprint arXiv:2208.05681*.

A Basic Information of our Corpus

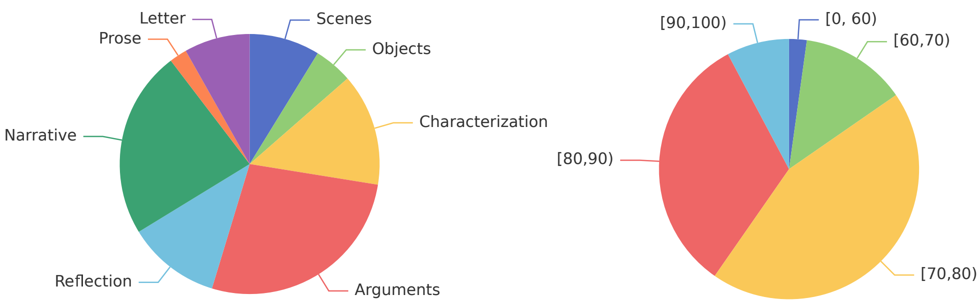
The distribution of essay genres is shown in Figure 3a, covering eight genres, while Figure 3b illustrates the distribution of score ranges for the selected essays, where the scores represent the overall marks assigned to each essay by teachers.

Additionally, the distribution of essay fluency scores, including Excellent, Average, and Unsatisfactory, across various essay genres is illustrated in the Table 10.

B Annotation Specification

B.1 Error Types

After conducting in-depth research into primary and secondary school student writing and extensively investigating the development of GEC data annotation standards, we have re-examined the classification of grammar errors in GEC and synthesized a revised set of annotation standards. Our annotation specification holistically covers simple grammatical errors such as punctuation and spelling mistakes, as well as complex grammatical issues like missing components and improper collocations, offering a further categorization of grammar errors and corresponding correction methods. Specifically, in terms of grammar error types, we have classified the grammatical errors in compositions into character-level and component-level errors, further divided into 5 coarse-grained and 18 fine-grained error types. Our annotations adhere to the principle of minimal modification. Our newly



(a) Essay genre distribution. (b) Essay score distribution.

Figure 3: (a) displays the distribution of the 501 essays used to construct the dataset by genre, covering a total of 8 essay genres. (b) shows the distribution of the essays used for annotation in terms of score.

summarized definitions for grammatical error types are as follows:

Character-Level Error (CL). Including four fine-grained error types: Word Missing (**WM**), where a word in a commonly used fixed collocation is missing from the sentence and needs to be added; Typographical Error (**TE**), where there are typos in the sentence that need to be revised or deleted; Missing Punctuation (**MP**), where punctuation is missing from the sentence and needs to be added; and Wrong Punctuation (**WP**), where the punctuation used in the sentence is wrong and needs to be revised or deleted.

Redundant Component Error (RC). Four fine-grained error types are: Subject Redundancy (**SR**), which occurs when a complex adverb immediately follows the first subject, followed by another subject referring to the same thing, and the modification is to delete one subject; Particle Redundancy (**PR**) refers to the redundant use of particles, which should be deleted during editing. Statement RePetition (**SRP**) occurs when some words or clauses repeat in the sentence, and the solution is to delete them. Other Redundancy (**OR**) refers to any redundant elements not covered by the previous types, which should also be deleted in modification.

Incomplete Component Error (IC). Four fine-grained error types with missing components are: Unknown Subject (**US**), which occurs when the sentence lacks a subject or the subject is unclear, and the solution is to add or clarify the subject; Predicate Missing (**PM**) refers to a sentence lacking verbs, which can be corrected by adding predicates; OBject Missing (**OBM**) means that a sentence lacks an object, and the solution is to add an object; OTher Missing (**OTM**) refers to other missing components besides the incomplete subject, predicate, and object, which can be corrected

Type	Example
SRD	**Sent:** 我在阳台上一共种了两株，我平时见不到它们。 (I planted a total of two on the balcony, I usually don't see them.) **Tips:** Delete the second subject, "I".
PR	**Sent:** 由于邓稼先的癌症的越来越严重，经常病地倒在了地上。 (As Deng Jiaxian's cancer became more serious, he often fell ill to the ground.) **Tips:**Delete the second "的".
SRP	**Sent:** 数字又不只是一个数字，在这个快速发展的时代里，我们每天都可以看到不同的数字，可其中的它们又不是一个数字，因为背后都是真实发生的事。 (Number is not just number. In this era of rapid development, we can see different numbers every day, but they are not just numbers, as behind them are real events.) **Tips:** "Number is not just number" repeats with "they are not just numbers".
OR	**Sent:** 一个易拉罐被踢开了下山去。 (A soda can was kicked away and went down the hill.) **Tips:** "kicked away and went down the hill" equals to "kicked down the hill"
US	**Sent:** 眼泪瞬间流下，滴落在了衣服上，出现深色小圆点，又接二连三的掉下来。 (Tears flowed down in an instant, dripping onto the clothes, small dark dots appeared, and fell down one after another.) **Tips:** Subjects changed in clauses. Add subject "tears" before "fell down".
PM	**Sent:** 邓稼先从美国后，就立刻接到了研究原子弹工作。 (After Deng came from US, he at once received a job to study the atomic bomb.) **Tips:** Add "归来" after "美国".
OBM	**Sent:** 然而我想说，并不是所有书籍都有能力完成承载读者。 (However, I want to say that not all books are capable of carrying readers.) **Tips:** Add "任务" after “承载读者”.
OTM	**Sent:** 爱迪生为改良电灯试用6000多材料，试验7000多次。 (Edison tried over 6000 materials and over 7000 tests to improve the electric lamp.) **Tips:** Add “种” after "6000多".
ISVC	**Sent:** 他知道我们比较薄弱的地方，并使我们在下一次测试中得到提高。 (He knows where we are weak and improves us for the next test.) **Tips:** Predicate "提高" should be paired with subject "我们的成绩", not "我们".
IVOC	**Sent:** 我尽管不是班里最高分，但也达到了很大的进步。 (Although I am not the highest score in the class, I have made great progress.) **Tips:** Object "进步" should be paired with predicate "取得" instead of "达到".
IWO	**Sent:** 一次受到生活打击的祥子也没有放弃。 (Xiangzi who was hit by life once did not give up.) **Tips:** "一次" should be placed after "祥子".
IOC	**Sent:** 牛顿被苹果为什么会从树下掉下来感到困惑，最后研究出了万有引力定律。 (Newton was puzzled by why the apple fell from the tree, and finally worked out the law of gravitation.) **Tips:** "感到困惑" should be paired with "为" instead of "被".
FIL	**Sent:** 聂海胜出生在湖北枣庄一个物质极度匮乏的小山村中。 (Nie Haisheng was born in a small mountain village in Zaozhuang, Hubei, where materials are extremely scarce.) **Tips:** Nie Haisheng was born in Zaoyang, Hubei, not in Zaozhuang, Hubei.
LIL	**Sent:** 那老奶奶抬起头，只是一惊，然后便笑着说：“没事，谢谢小伙子的好心，我自己来就好。” (The old woman raised her head, was just surprised, and then said with a smile: "It's okay, thank you for your kindness, I'll just do it myself.) **Tips:** The action 'surprised' comes before 'smiling.' When describing 'being surprised,' we should use "先是"(firstly) rather than "只是"(just).

Table 11: Examples of each fine-grained component-level error types.

by adding the missing components except for the subject, predicate, and object.

Incorrect Constituent Combination Error (ICC). Including four fine-grained error types: Inappropriate Subject-Verb Collocation (**ISVC**), which occurs when the subject and predicate are not properly matched, and can be corrected by replacing either the subject or predicate with other words. Inappropriate Verb-Object Collocation (**IVOC**) refers to the predicate and object not being properly matched, and can be corrected by replacing either the predicate or object with other words. Inappropriate Word Order (**IWO**) means that the order of words or clauses in the sentence is unreasonable, and can be corrected by rearranging some words or clauses. Inappropriate Other Collocation (**IOC**) refers to any element in the sentence not covered by the previous types being improperly matched, and can be corrected by replacing it with other words.

Illogical (IL). This includes two subcategories: Factual Illogicality (**FIL**) and Linguistic Illogicality (**LIL**). The former refers to instances that conflict with factual information, while the latter refers to misuse of logical conjunctions, idioms, etc., that render the sentence illogically constructed.

Table 11 shows examples of each fine-grained error type.

B.2 Essay Fluency Grading

Essay fluency grading adheres to the following criteria:

- Excellent (2 points): The types of grammatical errors committed do not affect reading fluency (e.g., Typographical Error and Factual Illogicality). The annotator, when reading through once, encounters no stumbling or incomprehensible parts.

- Average (1 point): The types of grammatical errors affecting reading fluency (the other 16 types of errors) do not exceed five sentences. The annotator, when reading through once, stumbles or finds parts hard to understand no more than five times.

- Unsatisfactory (0 points): The types of grammatical errors affecting reading fluency (the other 16 types of errors) exceed five sentences. The annotator, when reading through once, stumbles or finds parts hard to understand more than five times.

Task	Batch 0	Batch 1	Batch 2	Batch 3	Batch 4	Avg.
Error Types	69.06	55.04	54.93	52.91	61.33	58.65
Correction	78.65	57.71	59.05	51.56	63.64	62.12
Grading	66.28	58.46	59.38	55.86	61.84	60.36

Table 12: The consistency analysis results demonstrate the IAA scores, represented as percentages, across various aspects of text analysis for different data sub-batches (each batch representing a round of annotation). The final column indicates the average annotator consistency score across all batches.

C Inter-Annotator Agreement (IAA) Calculation

In this study, we adopted an Inter-Annotator Agreement (IAA) measure. For the `Error Type Identification` and `Essay Fluency Grading` tasks, we employed Cohen's Kappa to measure the consistency among annotators. For the `Wrong Sentence Rewriting` task, we used the $F_{0.5}$ score for the same purpose. The annotation was divided into five batches, with the consistency scores for each batch detailed in the corresponding Table 12.

D Implementation Details

For PLMs, we adopt AdamW optimizer (Loshchilov and Hutter, 2017) with the learning rate of $2e^{-5}$ to update the model parameters and set batch size as 16 and accumulated gradients as 2 for training and validation.

All our experiments are performed on RTX 3090. All other parameters are initialized with the default values in PyTorch Lightning[1], and our model is all implemented by Transformers[2].

For LLMs fine-tuning, we employed LoRA for fine-tuning with the low rank parameter set to 8. For knowledge distillation method, in `Error Type Identification` task, the temperature is set to 1, and the α is set to 0.3. In `Wrong Sentence Rewriting` task, the temperature is set to 3, and the α is set to 0.75.

E Evaluation Metrics in Wrong Sentence Rewriting Task

For evaluation, the similarity with the ground truth is matched. On the other hand, given the fact that there can be multiple correct corrections for a given sentence, the corrections generated by models may differ from the gold corrections. To address this, we employ language models (LMs) to measure the

[1]https://github.com/Lightning-AI/lightning
[2]https://github.com/huggingface/transformers

fluency of the generated corrections. Furthermore, in order to prevent over correction that would significantly alter the original text, we incorporate the Levenshtein distance measure. By minimizing the alterations, we respect the unique expression of the student writer, while correcting their grammatical mistakes. In a word, we evaluate the results of the model from two perspectives:

Comparison with ground truth. We employ three evaluation metrics: **1)** Exact Match (EM): calculates the percentage of correct sentences generated by the model that exactly match the gold references; **2)** Edit metrics proposed by MuCGEC : converts error-correct sentence pairs into operations, compares the model's output operations with the correct references, and calculates the highest scores for $F_{0.5}$; **3)** BLEU: measures the overlap between the model-generated sentences and the correct references.

Correctness and reasonableness of results. We use three evaluation metrics: **1)** Perplexity(PPL): measures the quality of rewritten sentences by BERT (Devlin et al., 2018). **2)** BERTScore (Zhang et al., 2019): measures the similarity between the rewritten sentence and the original sentence. **3)** Levenshtein Distance (LD): calculates the edit distance between the rewritten sentence and the original sentence.

We finally weighted multiple metrics to get the final score:

$$\begin{aligned}\text{AvgScore} = (\text{EM} + \text{BLEU} + \text{F}_{0.5} + \text{BERTScore})/4 \\ -\text{Levenshtein} - \text{BERT}_{\text{PPL}}.\end{aligned} \tag{7}$$

F Cross-sentence Error

To assess the impact of cross-sentence information on grammar error type identification, we trialed a method increasing input sequence length, shifting from single to multi-sentence recognition, with results shown in Table 13. We observe that for a well-trained model, performance improves with increasing input sequence length. This indicates that cross-sentence information aids in grammatical error type recognition, underscoring the significance of research on cross-sentence errors.

G Max Mutual Benefit of Error Type Identification and Correction

Table 14 presents the performance of the teacher model in the coarse-grained grammatical error type identification task. The inclusion of sentences with genuine corrections significantly enhances error type identification, with a notable 20% improvement in coarse-grained error type recognition. This underscores the importance of corrected sentence information for this task.

Sent Num	1	2	3	4
Micro F1	32.71	36.30	35.89	**36.88**
Macro F1	11.93	12.22	12.32	**12.53**

Table 13: Results of multi-sentence input on fine-grained error type recognition. The columns represent the number of input sentences.

Model	CL	RC	IC	ICC	IL	Micro F_1	Macro F_1
BERT	87.93	20.00	40.74	7.79	0.00	69.58	31.29
BERT♠	**92.37**	76.78	**86.19**	**31.03**	0.00	**84.85**	**57.27**
RoBERTa	88.51	25.00	46.23	14.00	0.00	70.34	34.75
RoBERTa♠	90.50	**78.22**	83.95	22.22	0.00	84.08	54.98

Table 14: A comparison of performance on coarse-grained error type recognition with correction reference as inputs in the Error Type Identification task. ♠ indicates the result after using the correction reference.

H Prompt for Models

We have listed the prompts used for all tasks, including `Essay Fluency Grading`, `Error Type Identification` and `Wrong Sentence Rewriting`. Note that the original prompts were written in Chinese, and we provide their English translations here.

H.1 Essay Fluency Grading

The prompts we use for this task are as follows:

Zero-shot prompt for ChatGPT, where [E] is the essay:

> "Assuming you are a primary or secondary school language instructor, I will provide you with an essay. Please evaluate its fluency on a scale of 0 to 2: where 0 denotes "Not Fluent", 1 denotes "Moderately Fluent", and 2 denotes "Highly Fluent". Kindly return only the fluency score. Input: [E]; Output:"

Few-shot prompt for ChatGPT, where [E] is the essay, and [G] is the fluency grade of [E].:

> "Assuming you are a primary or secondary school language instructor, I will

provide you with an essay. Please evaluate its fluency on a scale of 0 to 2: where 0 denotes "Not Fluent", 1 denotes "Moderately Fluent", and 2 denotes "Highly Fluent". Kindly return only the fluency score. Here are some samples: Sample 1: Input: [E]; Output: [G]. Input: [E]; Output:"

Prompts for ChatGLM is the same as zero-shot prompt for ChatGPT.

H.2 Error Type Identification

Zero-shot prompt for ChatGPT in both coarse-grained and fine-grained error type identification, where [S] indicates the sentence:

> "Assume you are a primary or secondary school language instructor proficient in grammar type identification and correction for student essays. In this context, I have defined five error categories. I will list these categories in the format "Error Type ID, Error Type: Definition;". Please identify the error types in the given sentence. Note that a sentence might contain multiple error categories. Kindly return the identification and correction results in the JSON format: "errorTypeId":[Error Type ID_1, Error Type ID_2], "errorType":[Error Type 1, Error Type 2], "revisedSent":"Corrected Sentence". If you believe the sentence is grammatically correct, please return "errorTypeId":[0], "errorType":["Right"]. The definitions are as follows: [Error Type ID], [Error Type]: [Definition]; Input: [S]; Output:"

Few-shot prompt for ChatGPT in both coarse-grained and fine-grained error type identification, where [S] indicates the sentence and [E] denotes the error type:

> "Assume you are a primary or secondary school language instructor proficient in grammar type identification and correction for student essays. In this context, I have defined five error categories. I will list these categories in the format "Error Type ID, Error Type: Definition;". Please identify the error types in the given sentence. Note that a sentence might contain multiple error categories. Kindly return the identification and correction results in the JSON format: "errorTypeId":[Error Type ID_1, Error Type ID_2], "errorType":[Error Type 1, Error Type 2], "revisedSent":"Corrected Sentence". If you believe the sentence is grammatically correct, please return "errorTypeId":[0], "errorType":["Right"]. The definitions are as follows: [Error Type ID], [Error Type]: [Definition]. Here are some samples: Input: [S], Output: "errorTypeId":[1,2], "errorType":[[E_1], [E_2]] Input: [S]; Output:"

Similarly, prompts for ChatGLM is the same as zero-shot prompt for ChatGPT.

Specifically, our input prompt augmented with revised sentence is as follows, where [S] denotes the original sentence and [R] represents the revised sentence:

> "Assume you are a primary or secondary school language instructor proficient in grammar type identification for student essays. In this context, I have defined five error categories. I will list these categories in the format "Error Type ID, Error Type: Definition;". Please identify the error types in the given sentence and revised sentence. Note that a sentence might contain multiple error categories. Kindly return the identification and correction results in the JSON format: "errorTypeId":[Error Type ID_1, Error Type ID_2], "errorType":[Error Type 1, Error Type 2], "revisedSent":"Corrected Sentence". If you believe the sentence is grammatically correct, please return "errorTypeId":[0], "errorType":["Right"]. The definitions are as follows: [Error Type ID], [Error Type]: [Definition]. Sentence: [S]; Revised Sentence: [R]; Output: "

H.3 Wrong Sentence Rewriting

Zero-shot prompt for ChatGPT, where [S] denotes the wrong sentence:

> “You are an elementary or secondary school language teacher tasked with correcting erroneous sentences in student

essays. I will provide you with a sentence from the essay; please make necessary revisions. Bear in mind, adjustments should adhere to the principle of minimal change. Kindly return only the revised sentence. If you believe the sentence is error-free, simply return the input sentence. Input: [S]; Output:"

Few-shot prompt for ChatGPT, where [S] denotes the wrong sentence and [R] indicates the revised sentence:

> "You are an elementary or secondary school language teacher tasked with correcting erroneous sentences in student essays. I will provide you with a sentence from the essay; please make necessary revisions. Bear in mind, adjustments should adhere to the principle of minimal change. Kindly return only the revised sentence. If you believe the sentence is error-free, simply return the input sentence. Input: [S]; Output: [R]; Input: [S]; Output:"

Similarly, prompts for ChatGLM is the same as zero-shot prompt for ChatGPT.

Specifically, our input prompt augmented with error type information is as follows, where [S] indicates the sentence and [E] denotes the error types:

> "You are a primary and secondary school language teacher capable of correcting erroneous sentences from student essays. I will provide you with a sentence from the essay along with its error category. Please make corrections based on the provided error category, adhering to the principle of minimal changes. Only return the revised sentence; if you believe the sentence is error-free, return the original sentence. I will list these categories in the format "Error Type ID, Error Type: Definition;". The definitions are as follows: "[Error Type ID], [Error Type]: [Definition];" Sentence: [S]; Error Type: [E]; Output: "