VisCGEC: Benchmarking the Visual Chinese Grammatical Error Correction

Anonymous ACL submission

Abstract

Chinese Grammatical Error Correction (CGEC) plays a significant role in providing automatic feedback to students' writing, especially for Chinese as a Foreign Language Learner (CFL). Particularly, rudimentary CFLs write Chinese characters where phonological and visual confusion is constantly involved. However, existing CGEC studies ignore the multi-modality and potential faked errors (i.e., non-existent characters created due to writing errors), which pushes the techniques far away from real-world scenarios. To address this gap, we develop a dataset, namely **VisCGEC**, to benchmark the visual Chinese grammatical error correction for Chinese as a Foreign Language Learner (CFL). The dataset contains 2,451 images of handwritten sentences with grammatical errors and corresponding correction texts, which Chinese language experts meticulously annotate. In addition, we propose baseline approaches on VisCGEC and conduct experiments with two CGEC frameworks (i.e., a two-stage pipeline and an end-to-end system), providing a strong baseline for future research. Extensive empirical results and analyses demonstrate that VisCGEC is high-quality but challenging, where the best approach achieves an $F_{0.5}$ score of only 28.9%. Our dataset and baseline methods will be made publicly available to further advance research in this area.

1 Introduction

Due to the complexity of Chinese characters and grammar, writing grammatically correct Chinese sentences is difficult for learners, especially for Chinese as a Foreign Language Learner (CFL). Chinese Grammatical Error Correction (CGEC), which uses artificial intelligence (AI) technologies to help learners write higher-quality texts more efficiently, has attracted intensive attention from researchers. Given an erroneous sentence, a CGEC method is expected to detect and correct all grammatical errors in the sentence and produce an error-free sentence as the output.

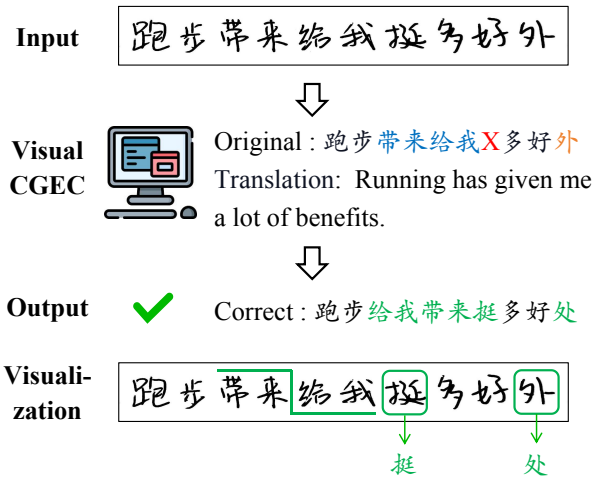


Figure 1: Examples of visual CGEC (Chinese Grammatical Error Correction) challenge. Blue text represents grammatical errors, X represents faked characters, Orange text represents misspelled characters, and Green text represents the corrected content.

We notice that a few CGEC datasets have been proposed to support the research on CGEC. For example, CTC and NaCGEC datasets (Zhao et al., 2022; Ma et al., 2022) are two large-scale CGEC datasets, the erroneous sentences of which are collected from native speakers. The models trained on these datasets could effectively identify the grammatical errors made by native speakers. To develop the CGEC methods for CFLs, researchers have developed a series of datasets such as MUCGEC (Zhang et al., 2022a), NLPCC (Zhao et al., 2018) and CGED (Rao et al., 2020), which contain more errors made by CFLs. Recently, some datasets like FlaCGEC (Du et al., 2023) and FCGEC (Xu et al., 2022) are developed focusing on the fine-grained grammatical error annotations (e.g., misuse of adjectives, missing nouns). However, these datasets are constructed based on a single textual modality and ignore the phonological

and visual confusion that the rudimentary CFLs may frequently have. For example, in Figure 1, the character "[illegible]" is an inexistent character that is created due to the writing errors. Besides, character distortion, hyphenated writing, and irregular characters can involve new challenges to the CGEC tasks. The existing CGEC models trained on traditional CGEC datasets are not applicable when dealing with such real-world handwritten texts. A recently-released dataset Visual-C^3 (Li et al., 2024) is proposed for visual Chinese Spelling Checking. However, they simply focus on misspelling errors and do not contain any grammatical errors, which hinders the development of the end-to-end CGEC system on hand-crafted images. Therefore, it becomes urgent and important to extend the CGEC data resources to cover both textual and image modality. This will enable the CGEC models to automatically detect and correct grammatical errors in handwritten texts.

In this paper, we introduce **VisCGEC**, which benchmarks the visual CGEC challenges in real-world scenarios. Our VisCGEC dataset contains a great number of hand-written images with different writing styles. As shown in Figure 1, each image is annotated with the bounding box information of each character, the recognized texts, and the corrected texts. With the bounding box information and the corrected texts, we can visualize the edits in the input image. To construct the VisCGEC, we first develop a browser-based online annotation interface to collect the essays from foreign students. Then, the essay is revised by multiple annotators with strong Chinese teaching experience. Next, we segment the image of essays at the sentence level and process the data to ensure the high quality of the data. As a result, our newly constructed VisCGEC dataset consists of $2,451$ images of hand-crafted sentences, which should be edited via insertion, deletion, substitution, and order shift. To further validate the quality and challenge of VisCGEC, we propose two frameworks, namely a two-stage pipeline and an end-to-end system, to solve the visual CGEC tasks. Extensive experiments and detailed analysis showcase that VisCGEC is a high-quality but challenging dataset. The best-performing model, which combines OCR and the Qwen model, achieves an $F_{0.5}$ score of only 28.59% on VisCGEC. In addition, the baseline approach provides useful insights and directions for future research. We believe that the introduction of VisCGEC will significantly advance research on CGEC tasks and make the intelligent systems better adapted to real-world needs.

In summary, our study makes the following contributions:

- We introduce a novel benchmark dataset of visual Chinese grammatical error correction consisting of $2,451$ images collected from real-world CFLs' handwritten texts. To the best of our knowledge, this is the first dataset for correcting Chinese erroneous sentences under visual contexts.
- We formally define the task and propose two baseline frameworks to solve the tasks, showcasing the potential solutions. Moreover, we investigate some hard examples that demonstrate the distinction of our dataset compared to traditional CGEC tasks.

2 Related Work

2.1 CGEC Resources

Dataset	Input	Errors	Source
NLPCC(Zhao et al., 2018)	text	G	CFL
CGED(Rao et al., 2018, 2020)	text	G	CFL
CTC(Zhao et al., 2022)	text	G	natives
MUCGEC(Zhang et al., 2022a)	text	G	CFL
NaCGEC(Ma et al., 2022)	text	G	natives
FCGEC(Xu et al., 2022)	text	G	natives
FlaCGEC(Du et al., 2023)	text	G	natives
Visual-C^3(Li et al., 2024)	text+image	S + F	natives
VisCGEC	text + image	S + F + G	CFL

Table 1: Comparison of VisCGEC with existing CGEC datasets. G, S, and F denote grammatical errors, misspelled and faked characters, respectively.

There has been a significant amount of research on GEC data construction, but most of them are proposed for English (Ng et al., 2013; Bryant et al., 2019). Datasets available for CGEC tasks are still scarce (Wang et al., 2021), which are summarized in Table 1. NLPCC2018 (Zhao et al., 2018) and CGED (Rao et al., 2018, 2020) collect large-scale sentences from CFLs and create datasets in error-coded format. MuCGEC (Zhang et al., 2022a) compiles a multi-reference annotation, which solves the problem of a single-reference constraint, potentially underestimating the model. CTC (Zhao et al., 2022) focuses on grammatical errors made by native Chinese speakers and involves Chinese semantic errors in the classification of errors. Recently, NaCGEC (Ma et al., 2022), FCGEC (Xu et al.,

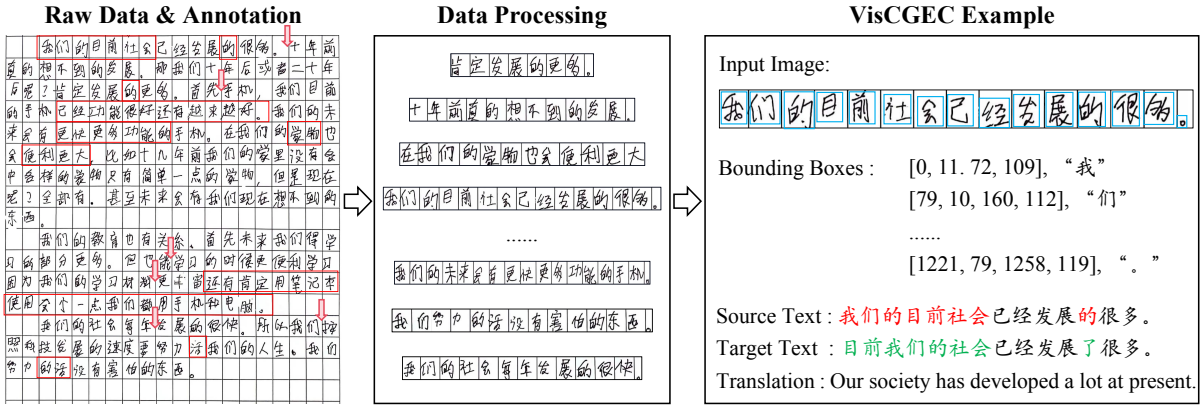


Figure 2: An overview of the construction process of VisCGEC. The contents in the red boxes of the raw image denote the erroneous sentences we extract as the instances in VisCGEC. The blue bounding boxes in the input image denote the annotated locations of each character.

2022), and FlaCGEC (Du et al., 2023) design a set of linguistic rules to generate corresponding erroneous sentences for different types of grammatical errors. The above datasets are limited to single textual modality, while multi-modality is commonly demanded in real-world writing assistance. Recently, Visual-C^3 (Li et al., 2024) is proposed as the first visual Chinese character-checking dataset that supports future research on human-crafted character checking. Nevertheless, the error types in Visual-C^3 are restricted to misspelled and faked characters made by middle native students. In contrast, VisCGEC encompasses a much broader range of grammatical errors in handwritten Chinese text, including not only misspelled and faked characters but also a variety of other grammatical issues. The dataset is derived from handwritten essays composed by foreign students from diverse national backgrounds, making VisCGEC more comprehensive and challenging.

2.2 CGEC Methods

The methods for CGEC tasks can be categorized into three paradigms: the Sequence-to-Sequence (Seq2Seq), Sequence-to-Edit (Seq2Edit), and Large Language Models prompting (LLMs prompting) approaches. The Seq2Seq approaches for general CGEC tasks involve using encoder-decoder models to translate erroneous sentences into correct sentences, similar to neural machine translation (Chen et al., 2020; Li and Shi, 2021; Kaneko et al., 2022). The Seq2Edit approaches apply edit-based models to GEC tasks, instead of directly predicting the correct sentence, edit-based models predict a series of operation edits to be applied to the erroneous sentences (Awasthi et al., 2019; Malmi et al., 2019; Omelianchuk et al., 2020; Tarnavskyi et al., 2022). With the emergence of LLMs, Fang et al. (Fang et al., 2023) evaluated the performance of closed-source LLMs (e.g., ChatGPT) on GEC and revealed their excellent capabilities for error detection and correction. Follow-up studies explored more applicable scenarios of LLMs in GEC (Penteado and Perez, 2023; Wu et al., 2023). However, we found that there is a lack of exploration of the multi-modal CGEC challenges. Therefore, we introduce a new dataset VisCGEC, and develop corresponding baseline methods to advance research and development in this area.

3 Benchmark Dataset Design

3.1 Dataset Construction

Raw Data Collection. To construct such a dataset, we collect handwritten texts by non-native writers. We develop an annotation platform[1] to collect the Chinese handwritten essays from foreign students, which not only contain real-world grammatical errors but also involve phonological or visual confusion made by the CFLs.

The raw data collection process lasted for three months, during which we gathered a total of 581 handwritten essay images with diverse handwriting fonts and styles from 303 CFLs from South Korea, Japan, and other countries.

Annotation Workflow. Given the handwritten texts with grammatical errors, we recruit 18 Chinese undergraduate students with foreign Chinese teaching experience as annotators. The annotators

[1]We display an example of the annotation platform in the Appendix A.1.

follow an error-coded annotation paradigm (Zhao et al., 2018) to explicitly mark the erroneous spans in the original sentences, then choose its error type and make corrections. Based on established annotation guidelines (Du et al., 2023)[2], we adopt insertion, deletion, re-ordering, and substitution operations to correct the texts. To facilitate data processing, each annotator must submit annotations to the platform. After that, we segment the original handwriting texts from the passage level to the sentence level. This results in 2451 image segments and each contains a single sentence, as shown in Figure 2.

To facilitate the recognition of the characters, we also annotate the bounding boxes of each character. We first employ YOLOv8[3] as a segmentation tool to automatically identify the coordinates of a single character's upper left and lower right corners. Then, we adapt Baidu PP-OCR[4] as an OCR tool to identify the optical characters in the image. It is worth noting that when there is a faked character, we mark it as a specific symbol, "X". For example, we identify the character "标" as "X" and its correct character is "标". To ensure the high quality of the automatic annotation, we also request the annotators to manually revise the accuracy of this step.

During the entire annotation workflow, we employ a team of 18 annotators and 3 senior experts during the annotation workflow. After the annotators complete their tasks, their submissions are aggregated and randomly assigned to a senior expert for review. The annotators' responsibilities include: 1) filtering the low-quality images; 2) ensuring the errors in the sentences have been accurately corrected; 3) identifying and flagging any bounding boxes of characters.

3.2 Dataset Analysis

Overall Statistics. As a result, our VisCGEC dataset contains 2451 images of handwritten sentences featured with diverse grammatical errors. Each image is annotated with the bounding boxes of characters, recognized erroneous sentences, and corrected sentences.

We randomly divided the data into different data sets for training, validation, and testing in the ratio of $8:1:1$. We perform detailed statistics of VisCGEC, which are displayed in Table 2. VisCGEC provides a greater number of erroneous sentences for training a visual CGEC model. It contains a wide range of errors, such as missing, substitution, redundant, word-order, and faked errors, which are evenly distributed over the data splits.

Properties	Train	Dev	Test
#Images	1960	245	246
Average source sentence length	24.27	25.00	23.18
Average target sentence length	24.57	25.52	23.67
#Edits per sentence	1.78	1.78	1.84
#Missing errors per sentence	0.50	0.51	0.50
#Redundant errors per sentence	0.33	0.33	0.34
#Substitution errors per sentence	0.88	0.89	0.92
#Word-Order errors per sentence	0.07	0.05	0.09
#Faked character per sentence	0.14	0.13	0.14

Table 2: Statistics of VisCGEC dataset.

Dataset Quality. To ensure dataset quality, a senior annotator randomly samples some instances from each batch of annotated data to review. We calculate Fleiss' Kappa (Moons and Vandervieren, 2023) to assess the annotation between the annotator and the senior expert. If the annotation disagreement exceeds a threshold 20%, the batch is reassigned to a new annotator until consensus is achieved. Overall, the agreement in labelling the erroneous sentences results in an agreement rate of 89.83%, which ensures that the annotations are consistent and reliable.

4 Task Formulation

Our VisCGEC dataset can be deemed a multimodal task, which first requires a system to recognize the optical characters and then correct the errors in the sentence. Therefore, we can formulate the visual CGEC task as follows: given an image I, we need to first transfer it into a source sequence of tokens, denoted as $S = (s_1, s_2, ..., s_n)$, where n is the length of the source sentence. Then a model is required to correct the grammatical errors in the sentence $T = (t_1, t_2, ..., t_m)$, where m is the length of the target sentence and it is possible that $n \neq m$. The challenge of VisCGEC is twofold: On the one hand, the characters in the images should be accurately identified from the image contents, with variances in text line length, writing styles, rotation, etc. On the other hand, the diverse errors in the source sentences should be corrected precisely.

[2]The detailed annotation workflow can be found in Appendix A.2.

[3]https://github.com/ultralytics/ultralytics

[4]https://cloud.baidu.com/doc/OCR/index.html

5 Baseline Approaches

To showcase the applicability of VisCGEC and offer insights for future research in the visual CGEC tasks, we develop two baseline frameworks to correct the erroneous sentences in images, which can be categorized as two-stage pipelines and an end-to-end system.

5.1 Two-stage Pipelines

The two-stage pipeline divides the visual CGEC tasks into character recognition and correction stages. The character recognition stage takes charge of transferring the optical characters of the image into text. The correction stage takes charge of correcting the errors in the sentence. We demonstrate the entire procedure in Figure 3.

5.1.1 Recognition Module

We chose the following two lines of recognition methods: OCR-based recognition and CLIP-based recognition.

OCR-based Recognition. Regarding the recognition module, we employ an OCR tool to translate the handwritten text image into the textual source sentence as follows:

$$\hat{X} = \text{OCR}(I). \tag{1}$$

where the predicted source sentence $\hat{X}$ is a sequence of tokens. Usually, the OCR module is an off-the-shelf recognition model pre-trained on large-scale Chinese text OCR datasets like SCUT-HCCDoc (Zhang et al., 2020) and featured with some ad-hoc engineering modules.

CLIP-based Recognition. Regarding the CLIP-based method, we first segment the sentence-level images into separate character-level images via an object detection model (Jocher et al., 2022). Specifically, the object detection model's input is the image I with n characters, and the output is the coordinates of each character's bounding box in the image, so we can extract the character-level images based on their coordinates. We denote the character-level images as $\{I_1, I_2, ..., I_n\}$. After that, the extracted character images are processed via a CLIP model (Yang et al., 2022), consisting of a text encoder and an image encoder. The image encoder encodes the visual features in the character-level images, and the text encoder encodes the textual features of tokens in a given dictionary $\mathcal{D}$, where possible Chinese tokens are stored. We denote the procedure as:

$$\mathbf{v}_i = \text{ImageEncoder}(I_i) \tag{2}$$

$$\mathbf{w}_j = \text{TextEncoder}(w_j), \tag{3}$$

where I_i is the i-th character-level image and w_j is the j-th tokens in the dictionary. Next, we compute the cosine similarity between $\mathbf{v}_i$ and $\mathbf{w}_j$ and select the token with the highest similarity score as the predicted token as follows:

$$\hat{s}_i = \text{argmax}_{w_j \in \mathcal{D}} \text{CosSim}(\mathbf{v}_i, \mathbf{w}_j). \tag{4}$$

In this case, we obtain a sequence of recognized tokens from the predicted source texts, denoted as $\hat{S} = \{\hat{s}_1, \hat{s}_2, ..., \hat{s}_n\}$.

We also fine-tune the CLIP model on the VisCGEC dataset. The primary objective is to minimize the contrastive loss (Khosla et al., 2020), which encourages the model to produce similar embeddings for matching image-text pairs and dissimilar embeddings for non-matching pairs. The contrastive loss function for each character is defined as follows:

$$\mathcal{L} = -log \frac{exp(\text{CosSim}(\mathbf{v}_i, \mathbf{w}_{s_i})/\tau)}{\sum_{j=1}^{|\mathcal{D}|} exp(\text{CosSim}(\mathbf{v}_i, \mathbf{w}_j)/\tau)}, \tag{5}$$

where $\mathbf{w}_{s_i}$ represents the feature representation of the correct text label s_i corresponding to the i-th character image and τ is a hyperparameter to smooth the similarity value.

Compared with OCR-based recognition, the encoders of the CLIP-based recognition module are fine-tuned, such that the writing styles and font variance can be adapted to VisCGEC effectively.

5.1.2 Correction Module

We apply three mainstream GEC approaches to correct the erroneous sentence, namely Seq2Seq, Seq2Edit, and LLM fine-tuning. We carefully describe one representative implemented model for each of them.

SynGEC Model: This is a representative method of Seq2Seq approaches, which follow the autoregressive principle (Xue et al., 2021; Zhang et al., 2022b), where the correct sentence is generated token by token as follows:

$$\hat{t}_m = \text{Seq2Seq}(\hat{S}, \hat{t}_1, \hat{t}_2, ..., \hat{t}_{m-1}). \tag{6}$$

We follow the existing study (Zhang et al., 2022b) to enrich the feature of the source texts with

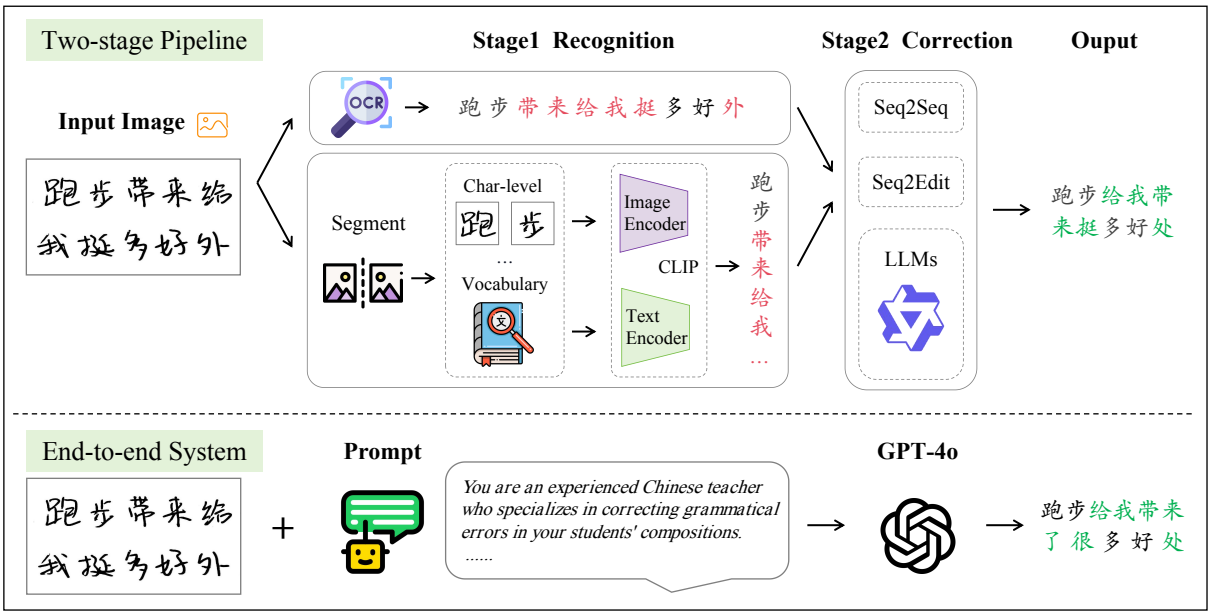


Figure 3: Illustration of our designed baseline methods, namely two-stage pipeline and end-to-end system. Red text represents erroneous content and Green text represents the corrected content.

syntactic features, effectively identifying the structure of the Chinese sentence. Eventually, we obtain predicted target sentence $\hat{T} = \{\hat{t}_1, \hat{t}_2, ..., \hat{t}_m\}$.

GECToR Model: This is a representative method of Seq2Edit approaches, which have non-autoregressive architectures (Omelianchuk et al., 2020). GECToR model first identifies the operation labels by comparing the source and target texts. Operations, including keeping, substitution, and deletion, are extracted. Next, the model predicts the operation for each token in parallel as follows:

$$\{\hat{l}_1, \hat{l}_2, ..., \hat{l}_n\} = \text{Seq2Edit}(\hat{S}). \quad (7)$$

After applying the predicted operations to the source texts, we obtain the predicted target sentence $\hat{T}$.

Qwen: We apply Qwen (Bai et al., 2023; Yang et al., 2024) as the representative of LLMs fine-tuning approaches to solve the visual CGEC challenge, which has been pre-trained on a vast corpus of text. We further fine-tune it on the VisCGEC dataset using LoRA (Low-Rank Adaptation).

5.2 End-to-end System

Besides the two-stage pipeline, we also apply an end-to-end system to visual CGEC challenges. Multi-modal Large Language Models (MLLMs), such as GPT-4o[5] fuse the textual and visual features in the early encoder. They integrate the understanding capabilities of LLMs with the ability to process multi-modal information. Hence, we prompt MLLMs with the instructions on text recognition and grammatical error correction:

$$\hat{T} = \text{MLLM}(I, P), \quad (8)$$

where P is the prompting instruction[6]. As a result, the corrected sentence will be directly produced as the prediction.

6 Experiments and Analysis

6.1 Experimental Settings

Evaluation metrics. As we mentioned above, visual CGEC tasks can be divided into recognition and correction sub-tasks. Regarding the recognition sub-task, we evaluate if the bounding boxes of the characters are accurate using Character Accuracy (**CA**), Character Error Rate (**CER**), and F_1 score. We treat the recognized characters as a bag of words, CA and F_1 scores measure the proportion of correctly recognized characters while CER indicates the error rate relative to the total. Regarding the correction sub-task, we evaluate if the correction to the sentences is correct following the approach outlined in (Ng et al., 2014). Specifically, we apply ChERRANT[7] for both sentence-level and

[5] https://openai.com/index/hello-gpt-4o/

[6] The detailed steps of MLLMs prompting is displayed in Appendix A.3

[7] https://github.com/HillZhang1999/MuCGEC/tree/main/scorers/ChERRANT

Recog.	Correct.	Detection Character-Level			Correction Character-Level			Detection Sentence-Level			Correction Sentence-Level		
		Prec.	Rec.	$F_{0.5}$	Prec.	Rec.	$F_{0.5}$	Prec.	Rec.	$F_{0.5}$	Prec.	Rec.	$F_{0.5}$
Two-stage Pipeline													
OCR	GECToR	45.64	38.31	43.96	10.23	6.80	9.29	15.65	19.17	16.24	4.47	4.47	4.47
	SynGEC	66.98	69.15	67.41	21.98	26.32	22.73	28.92	58.42	32.17	11.38	11.38	11.38
	Qwen	57.85	39.43	52.91	30.43	23.03	28.59	**32.95**	45.31	**34.85**	**15.85**	**15.85**	**15.85**
CLIP	GECToR	69.92	62.42	68.28	26.63	22.40	25.66	10.92	20.88	12.07	3.66	3.66	3.66
	SynGEC	**77.98**	**81.58**	**78.67**	31.83	**36.28**	**32.63**	23.47	**60.98**	26.77	6.10	6.10	6.10
	Qwen	71.49	65.25	70.15	**32.39**	28.11	31.43	21.22	36.19	23.14	6.10	6.10	6.10
End-to-end multi-modal LLMs													
	GPT-4	51.76	16.69	36.45	14.35	6.86	11.78	4.60	7.07	4.95	2.85	2.85	2.85

Table 3: Performance of different baseline approaches on the VisCGEC.

character-level metric calculation, including **Precision**, **Recall**, and $F_{0.5}$ Score.

Implementation details. For the OCR-based approach, we employ Baidu PP-OCR[8] specifically designed for handwriting recognition as the off-the-shelf tool. For the CLIP-based recognition, we utilize the YOLOv8 to segment sentence-level images into character-level images. Then we initialize the CLIP model with ViT-B/16 (Dosovitskiy, 2020) as the image encoder and RoBERTa-base (Liu et al., 2019) as the text encoder and fine-tune on VisCGEC dataset. For the correction module, we re-implement SynGEC and GECToR to train a GEC model, the hyper-parameters of which follow their original papers. We utilize the LLama_Factory(Zheng et al., 2024) to fine-tune the Qwen2-7B with LoRA. For the end-to-end systems, we directly leverage the GPT-4o model. More details of the implementation can be found in Appendix A.4.

6.2 Main results

From Table 3 , we have the following observations: (1) Among the two-stage pipeline approaches, OCR combined with the fine-tuned Qwen demonstrates the best performance on sentence-level correction, and CLIP integrated with the fine-tuned SynGEC demonstrates the best performance on character-level correction. We suspect the reason is that the off-the-shelf OCR tool features some ad-hoc engineering modules, which can provide more accurate results in sentence-level contexts of image recognition. (2) Comparing correction methods based on the same recognition module, we find that the fine-tuned Qwen outperforms SynGEC and GECToR models, achieving higher precision and $F_{0.5}$ scores in most cases. This is because Qwen is built upon a LLM foundation, which provides richer contextual understanding and better generalization. Moreover, SynGEC outperforms GECToR since GECToR is restricted by the specific edit labels defined in this task. (3) The multi-modal GPT-4, an end-to-end system, underperforms the two-stage pipeline approach, which may be caused by the domain-specific handwritten texts and over-correction of the recognition. (4) The VisCGEC dataset's overall performance across all models and approaches is modest, suggesting that it poses significant challenges. Our dataset provides a diverse and realistic testing ground, which is instructive for future research.

6.3 Breakdown Analysis

We conduct breakdown analysis for recognition and correction separately on the VisCGEC, which is displayed in Table 5. Regarding the recognition module, the OCR method performs exceptionally well and is highly effective for extracting text with minimal errors from the input images. CLIP also demonstrates good performance. While its performance is lower than OCR's, it still shows potential in handling visual input for text recognition. Regarding the correction module, the Qwen model consistently outperforms the other two models at both the character and sentence levels, highlighting the benefits of fine-tuning LLMs in the CGEC task.

To comprehensively evaluate the VisCGEC dataset, we further exclude the fine-tuning and simply perform inference on the test set utilizing the CGEC checkpoints, which are either trained on large-scale Chinese corpus or CGEC corpus like Lang8 (Zhao et al., 2018) and HSK (Zhang, 2009). As shown in Table 4, even though some of

[8] https://cloud.baidu.com/doc/OCR/index.html

Image	Source	Target	Predict	
	今年我的假期度过在上海。	今年我的假期在上海度过。	今年我的假期在上海度过。	✓
	我觉的最重要是公回语言。	我觉得最重要的是共同语言。	我觉得最重要是普通话。	✗
	已经我们感受到科技的影响。	我们已经感受到科技的影响。	大家已经感受到科技的影响。	✗
	长时间使用电脑和手机让我们的眼jng受到shang害	长时间使用电脑和手机让我们的眼睛受到伤害	长时间使用电脑和手机让我们的眼睛受到伤害	✓

Figure 4: Some illustrative examples in the VisCGEC dataset. Red text represents characters with grammatical errors, Purple text represents the correct target corrections, and Green text represents the predicted corrections.

Correction	Training Corpus	Character-Level			Sentence-Level		
		Prec.	Rec.	$F_{0.5}$	Prec.	Rec.	$F_{0.5}$
Transformer (Vaswani, 2017)	NLPCC18	19.05	8.83	15.47	5.28	5.28	5.28
T5 (Xue et al., 2021)	NLPCC18	14.86	9.03	13.16	4.47	4.47	4.47
STG (Xu et al., 2022)	FCGEC	12.61	3.09	7.80	2.44	2.44	2.44
StructBERT-Large (Wang et al., 2019)	Lang8 + HSK	31.78	24.06	29.86	10.98	10.98	10.98
Qwen2-7B w/o fine-tuning	Chinese corpus	8.56	20.97	9.71	2.85	2.85	2.85

Table 4: Performance of different CGEC systems trained on external corpora and evaluated on VisCGEC test set.

Recognition			
Method	CA	CER (↓)	F_1
OCR	96.77	1.79	98.68
CLIP	63.94	14.77	87.09

Correction						
Method	Character-Level			Sentence-Level		
	Prec.	Rec.	$F_{0.5}$	Prec.	Rec.	$F_{0.5}$
GECToR	18.36	12.26	17.74	6.09	6.09	6.09
SynGEC	21.76	26.27	22.53	10.57	10.57	10.57
Qwen	28.65	22.08	27.04	14.63	14.63	14.63

Table 5: Breakdown results on the two stages of VisCGEC.

these methods are trained on the large-scale corpus, they cannot achieve ideal results on our VisCGEC dataset, resulting in lower performance than the fine-tuned methods. This reveals that VisCGEC contains distinct challenges that are excluded from the existing corpus. Even for Qwen, it is highly demanded to inject the knowledge of VisCGEC to achieve better results.

6.4 Case Study

We display some cases in Figure 4. As we can see, the combination of CLIP and Qwen can effectively solve some examples and provide rationale correction to the original image. In addition, the corrected operations can also be visualized to the original image, showcasing the potential application of the VisCGEC in a writing assistant as shown in Appendix A.6. Meanwhile, we also find some hard cases including unclear handwriting, poor clarity, and using Pinyin.

Furthermore, we summarize the error cases on the GEC task and find that most errors come from substitution error types, accounting for 34.65% of the total errors. Missing errors are another error type that accounts for 23.02% of the total errors. Additionally, the results indicate that future research should focus on improving the capabilities of character substitution and re-ordering issues [9].

7 Conclusion

In this paper, we introduce VisCGEC, a benchmark dataset designed for visual CGEC challenges in real-world scenarios. The dataset contains handwritten images with various writing styles, each annotated with recognized texts and corrections. Furthermore, we validate the dataset through the development of two baseline approaches: a two-stage pipeline and an end-to-end system. These methods demonstrate the complexity and quality of VisCGEC. We believe VisCGEC will significantly advance the development of writing assistance systems, fostering more robust and intelligent solutions tailored to real-world applications.

[9] Due to the space limit, we put more analysis in Appendix A.5

Limitations

The raw input images collected are long, which causes YOLOv8 to struggle with accurately recognizing characters. To improve detection, we crop the images into smaller segments before recognition. In the future, we can extend the images of sentence-level handwritten texts to passage-level texts and encourage more advanced techniques to solve the visual Chinese GEC tasks in passages, considering the long dependence of texts.

Although we fine-tuned the CLIP model on our dataset, limited data and the complexity of handwriting from foreign learners impacted its performance. In the future, we aim to improve the model by expanding the dataset, including more handwriting styles and more fine-grained grammatical errors.

Ethics Statement

This study implemented strict ethical measures to protect participant rights and ensure proper data handling, particularly for the handwritten dataset. We detail the data collection, preprocessing, and annotation steps, with all data authorized and anonymized for privacy and confidentiality. Our goal is to advance grammatical error correction while minimizing negative societal impacts. We carefully considered the ethical aspects of our work to ensure transparency and promote positive outcomes.

References

Abhijeet Awasthi, Sunita Sarawagi, Rasna Goyal, Sabyasachi Ghosh, and Vihari Piratla. 2019. Parallel iterative edit models for local sequence transduction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4260–4270, Hong Kong, China. 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*.

Christopher Bryant, Mariano Felice, Øistein E. Andersen, and Ted Briscoe. 2019. The BEA-2019 shared task on grammatical error correction. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 52–75.

Mengyun Chen, Tao Ge, Xingxing Zhang, Furu Wei, and Ming Zhou. 2020. Improving the efficiency of grammatical error correction with erroneous span detection and correction. *Cornell University - arXiv*.

Alexey Dosovitskiy. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Hanyue Du, Yike Zhao, Qingyuan Tian, Jiani Wang, Lei Wang, Yunshi Lan, and Xuesong Lu. 2023. Flacgec: A chinese grammatical error correction dataset with fine-grained linguistic annotation. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pages 5321–5325.

Tao Fang, Shu Yang, Kaixin Lan, Derek F Wong, Jinpeng Hu, Lidia S Chao, and Yue Zhang. 2023. Is chatgpt a highly fluent grammatical error correction system? a comprehensive evaluation. *arXiv preprint arXiv:2304.01746*.

Glenn Jocher, Ayush Chaurasia, Alex Stoken, Jirka Borovec, Yonghye Kwon, Kalen Michael, Jiacong Fang, Zeng Yifu, Colin Wong, Diego Montes, et al. 2022. ultralytics/yolov5: v7. 0-yolov5 sota realtime instance segmentation. *Zenodo*.

Masahiro Kaneko, Sho Takase, Ayana Niwa, and Naoaki Okazaki. 2022. Interpretability for language learners using example-based grammatical error correction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7176–7187, Dublin, Ireland. Association for Computational Linguistics.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2020. Supervised contrastive learning. *Advances in neural information processing systems*, 33:18661–18673.

Piji Li and Shuming Shi. 2021. Tail-to-tail non-autoregressive sequence prediction for chinese grammatical error correction. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4973–4984.

Yinghui Li, Zishan Xu, Shaoshen Chen, Haojing Huang, Yangning Li, Shirong Ma, Yong Jiang, Zhongli Li, Qingyu Zhou, Hai-Tao Zheng, and Ying Shen. 2024. Towards real-world writing assistance: A Chinese character checking benchmark with faked and misspelled characters. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8656–8668.

Yinhan Liu, Myle Ott, and Naman Goyal. 2019. 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*, 1(3.1):3–3.

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.

Eric Malmi, Sebastian Krause, Sascha Rothe, Daniil Mirylenka, and Aliaksei Severyn. 2019. Encode, tag, realize: High-precision text editing. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5054–5065, Hong Kong, China. Association for Computational Linguistics.

Filip Moons and Ellen Vandervieren. 2023. Measuring agreement among several raters classifying subjects into one-or-more (hierarchical) nominal categories. a generalisation of fleiss' kappa. *arXiv preprint arXiv:2303.12502*.

Hwee Tou Ng, Siew Mei Wu, Ted Briscoe, Christian Hadiwinoto, Raymond Hendy Susanto, and Christopher Bryant. 2014. The conll-2014 shared task on grammatical error correction. In *Proceedings of the eighteenth conference on computational natural language learning: shared task*, pages 1–14.

Hwee Tou Ng, Siew Mei Wu, Yuanbin Wu, Christian Hadiwinoto, and Joel Tetreault. 2013. The CoNLL-2013 shared task on grammatical error correction. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning: Shared Task*, pages 1–12.

Kostiantyn Omelianchuk, Vitaliy Atrasevych, Artem Chernodub, and Oleksandr Skurzhanskyi. 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.

Maria Carolina Penteado and Fábio Perez. 2023. Evaluating gpt-3.5 and gpt-4 on grammatical error correction for brazilian portuguese. *arXiv preprint arXiv:2306.15788*.

Gaoqi Rao, Qi Gong, Baolin Zhang, and Endong Xun. 2018. Overview of nlptea-2018 share task chinese grammatical error diagnosis. In *Proceedings of the 5th Workshop on Natural Language Processing Techniques for Educational Applications*, pages 42–51.

Gaoqi Rao, Erhong Yang, and Baolin Zhang. 2020. Overview of nlptea-2020 shared task for chinese grammatical error diagnosis. In *Proceedings of the 6th workshop on natural language processing techniques for educational applications*, pages 25–35.

Maksym Tarnavskyi, Artem Chernodub, and Kostiantyn Omelianchuk. 2022. Ensembling and knowledge distilling of large sequence taggers for grammatical error correction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3842–3852, Dublin, Ireland. Association for Computational Linguistics.

A Vaswani. 2017. Attention is all you need. *Advances in Neural Information Processing Systems*.

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 preprint arXiv:1908.04577*.

Yu Wang, Yuelin Wang, Kai Dang, Jie Liu, and Zhuo Liu. 2021. A comprehensive survey of grammatical error correction. *ACM Trans. Intell. Syst. Technol.*, 12(5).

Haoran Wu, Wenxuan Wang, Yuxuan Wan, Wenxiang Jiao, and Michael Lyu. 2023. Chatgpt or grammarly? evaluating chatgpt on grammatical error correction benchmark. *arXiv preprint arXiv:2303.13648*.

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.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.

An Yang, Junshu Pan, Junyang Lin, Rui Men, Yichang Zhang, Jingren Zhou, and Chang Zhou. 2022. Chinese clip: Contrastive vision-language pretraining in chinese. *arXiv preprint arXiv:2211.01335*.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, et al. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

Baolin Zhang. 2009. Features and functions of the hsk dynamic composition corpus. *International Chinese Language Education*, 4:71–79.

Hesuo Zhang, Lingyu Liang, and Lianwen Jin. 2020. Scut-hccdoc: A new benchmark dataset of handwritten chinese text in unconstrained camera-captured documents. *Pattern Recognition*, page 107559.

Yue Zhang, Zhenghua Li, Zuyi Bao, Jiacheng Li, Bo Zhang, Chen Li, Fei Huang, and Min Zhang. 2022a. 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.

Yue Zhang, Bo Zhang, Zhenghua Li, Zuyi Bao, Chen Li, and Min Zhang. 2022b. Syngec: Syntax-enhanced grammatical error correction with a tailored gec-oriented parser. In *Proceedings of EMNLP*, pages 2518–2531.

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*.

Yuanyuan Zhao, Nan Jiang, Weiwei Sun, and Xiaojun Wan. 2018. Overview of the nlpcc 2018 shared task: Grammatical error correction. In *Natural Language Processing and Chinese Computing: 7th CCF International Conference, NLPCC 2018, Hohhot, China, August 26–30, 2018, Proceedings, Part II 7*, pages 439–445. Springer.

Yaowei Zheng, Richong Zhang, Junhao Zhang, YeYanhan YeYanhan, and Zheyan Luo. 2024. LlamaFactory: Unified efficient fine-tuning of 100+ language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pages 400–410, Bangkok, Thailand. Association for Computational Linguistics.

A Appendices

A.1 Annotation Interface

Figure 5 and Figure 6 show the left and right sides of the interface we designed for collecting the data of VisCGEC, respectively. Both data collection and annotation are conducted in the College of Chinese Studies at the university. The foreign students upload images of their handwritten texts. After uploading, the left side of the interface displays a picture of the foreign students' handwritten texts, while the right side showcases the annotation options. The OCR technology first converts the image into editable texts which are displayed in a text box. The annotators can add insertion, substitution, deletion as well as re-ordering operations with diverse symbols directly on the image of the handwritten texts. At the same time, on the right side of the interface, the annotators are able to select the corresponding error types for each correction action.

Figure 5: The screenshot of the annotation interface (left side).

A.2 Detailed Annotation Steps

We describe the detailed annotation steps of VisCGEC, including image filtering, essay annotation and bounding box annotation.

Image Filtering. Based on the annotation interface, we collected 885 in total images of the handwritten essays. During this review process, we first conduct filtering to the images by identifying two primary categories of issues that could potentially affect the dataset's quality:

Figure 6: The screenshot of the annotation interface (right side).

- Image Quality Issues: These include angular skew, poor clarity, and shadows. Such issues can directly impact the model's detection performance.
- Writing Quality Issues: These include messy handwriting from foreign learners, including frequent scribbling, the use of pinyin, and annotators' noisy annotations on the essay image. These factors can also affect the data's usability.

After the filtering process, 581 high-quality images were selected for further processing. This preprocessing step ensured the reliability and robustness of the subsequent grammatical error correction.

Essay Annotation. We recruit 18 annotators and 3 senior experts in the university. All annotators are undergraduate students majoring in international education of Chinese language or Chinese language. They have rich teaching experience in Chinese. Annotators received intensive training before real annotation. The senior experts are senior teachers in the College of Chinese Studies. All annotators and experts were paid for their work. The average salaries of annotators and experts are CNY 25 per hour, equivalent to USD 3.51. Please note that the minimum average hourly wage in the Shanghai Province of China (where the recruited annotators are from) is CNY 24 in 2024. In total, we spent USD 983.42 in this dataset.

Bounding Box Verification. To ensure the accurate pixel-level annotation of each character, a specialized image region localization tool, LabelImg[10], was utilized to enhance annotation efficiency. Annotators used this tool to mark the precise coordinate positions of each character within the images. After the initial annotation, senior experts reviewed each annotated image to ensure that all characters were accurately recognized and clearly delineated, and there was no overlap between the bounding boxes. Senior experts reviewed and verified the coordinates' accuracy to ensure consistency and precision. This dual-step process ensures the high annotation quality and reliability for the dataset.

A.3 MLLM Prompting

To effectively utilize the Multimodal Language Model (MLLM) for visual grammatical error correction, we designed specific prompts to guide the model in accomplishing the task. These prompts ensure that the model can accurately recognize and correct grammatical errors in the text while maintaining the original structure and expression of the sentences.

The prompts were developed using a heuristic approach through iterative experimentation and feedback. Initially, they focused on clarity and simplicity, enabling the model to address key grammatical issues without overcomplicating the task. Based on model outputs, we refined the prompts to better preserve sentence structure and explicitly list correction actions. This iterative process helped us identify the most effective prompts now used in the pipeline for both Qwen and GPT-4 models. Specifically, for the Qwen model, we use the following prompts:

[10] https://pypi.org/project/labelImg/

Configurations	Correction Module
Model architecture	GECToR/SynGEC
Devices	1 Nvidia A800 GPU (80GB)
Number of max epochs	200
Batch size per GPU	32
Learning rate scheduler	$6e-5$
Optimizer	Adam/FP16Optimizer ($\beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 1 \times 10^{-8}$)
Loss function	Cross entropy
Total training time	About 5 hours
Stopping criteria	Loss value on the dev set

Table 6: Hyperparameters of the Baseline Approaches

> 你是一位经验丰富的中文老师，专门纠正学生作文中的语法错误。请根据以下要求进行修正：
> 1. 纠正语法和用词错误。
> 2. 保持句子的原始结构和表达方式，不要大幅度改变句子的意思。
> 3. 修正后的句子应流畅、自然，符合标准中文语法规则。
>
> 请根据以上要求对以下文本进行修正：
> 原文：[原始的文本内容]
> 修正后的文本：[纠正后的文本内容]

For the GPT-4 model, we use the following prompts:

> 你是一位经验丰富的中文老师，专门纠正学生作文中的语法错误。请查看以下包含语法错误的小照片，完成以下任务：
> 1. 识别并提取照片中的文本内容。
> 2. 纠正所有语法和用词错误，尽量保持句子的原始结构和表达方式。
> 3. 列出具体的修改操作，例如：将 '错误的词语' 修改为 '正确的词语'。
>
> 请按照以下格式输出结果：
> 1. 识别的结果：[识别的文本内容]
> 2. 纠正后的正确结果：[纠正后的文本内容]
> 3. 修改操作：[具体的修改操作列表]

A.4 Hyperparameters

All models discussed in this paper are implemented using Python (version 3.8) and the PyTorch framework (version 2.1.0). Table 6 shows the detailed hyperparameters used to train our error correction module. Due to GPU memory constraints, we truncate sentences with more than 128 characters when

training our correction modules. In other words, redundant characters in input sentences and references are discarded.

A.5 Error Analysis

To address the limitations of our proposed baseline approach, we conducted an error analysis on the best-performing model that combines OCR-based recognition and Qwen for correction. As illustrated in Figure 7, the analysis reveals key challenges in handling Chinese grammar correction. The model struggles most with missing and redundant errors, suggesting difficulty identifying omitted or superfluous characters in complex handwritten text. Substitution errors are another major issue, particularly when visually similar characters are confused, leading to incorrect replacements. Word-Order errors are also challenging, indicating the model's limitations in grasping sentence structure and context. These findings point to the need for improved contextual understanding and the ability to manage subtle variations in sentence construction as key areas for future refinement.

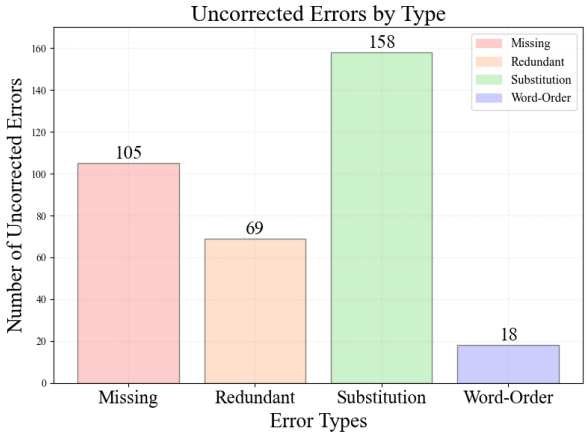


Figure 7: Statistics on the number of uncorrected errors in each category.

A.6 Visualization of VisCGEC prediction

A key application of the VisCGEC dataset is its potential to enhance writing assistance systems by directly showing the edit operations on the original images, particularly for CFLs.

Here, we show a visualization of the method combining CLIP and Qwen models. The process begins with the YOLOv8 model, which accurately detects and locates the position of each character within the input image. Following this, a CLIP model extracts the corresponding text from the image. The extracted text is then passed to the Qwen model, which generates the corrected version of the text by identifying and fixing any grammatical errors. Once the corrections are made, we map these changes back onto the original image, such as insertions, deletions, substitutions, or word-order adjustments. As shown in Figure 8, users can clearly view the changes in context. The entire process, from character detection and text extraction to grammar correction and visual feedback, demonstrates the practical value of the VisCGEC dataset in developing intelligent, multimodal language learning systems.

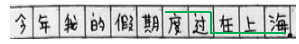
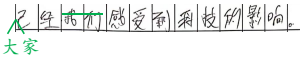
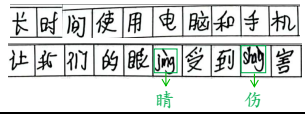
Predict	Visualization
我觉得最重要是普通话。	
今年我的假期在上海度过。	
大家已经感受到科技的影响。	
长时间使用电脑和手机 让我们的眼睛受到伤害	

Figure 8: Potential application of the VisCGEC dataset.