Explanation based In-Context Demonstrations Retrieval for Multilingual Grammatical Error Correction

Anonymous ACL submission

Abstract

Grammatical error correction (GEC) aims to correct grammatical, spelling, and semantic errors in natural language text. With the growing of large language models (LLMs), direct text generation has gradually become the focus of the GEC methods, and few-shot in-context learning presents a cost-effective solution. However, selecting effective in-context examples remains challenging, as the similarity between input texts does not necessarily correspond to similar grammatical error patterns. In this paper, we propose a novel retrieval method based on natural language grammatical error explanations (GEE) to address this issue. Our method retrieves suitable few-shot demonstrations by matching the GEE of the test input with that of pre-constructed database samples, where explanations for erroneous samples are generated by LLMs. We conducted multilingual GEC few-shot experiments on both major open-source and closed-source LLMs. Experiments across five languages show that our method outperforms existing semantic and BM25-based retrieval techniques, without requiring additional training or language adaptation. This also suggests that matching error patterns is key to selecting examples. Our code and the constructed database will be publicly available after the paper is published.

1 Introduction

The goal of grammatical error correction (GEC) is to correct errors in natural language text, including grammatical, spelling, and certain semantic errors (Bryant et al., 2023). In the context of language learning, GEC methods can assist learners in correcting mistakes in written text (Katinskaia and Yangarber, 2021; Caines et al., 2023). GEC can be considered a form of machine translation, where the input is a potentially erroneous text, and the output is its corrected version (Yuan and Briscoe, 2016). As a result, text-to-text generation models underpin many GEC approaches (Junczys-Dowmunt et al., 2018; Katsumata and Komachi, 2020). While edit-based methods exist that modify specific parts of the text for correction (Awasthi et al., 2019; Omelianchuk et al., 2020), the rise of large language models (LLMs) (Brown, 2020a; Touvron et al., 2023) has made direct text generation the dominant approach in GEC. Given the need for language learning across different languages, multilingual GEC has become an important area of research. This challenge is often tackled by constructing datasets and training models separately for each language (Sun et al., 2022; Rothe et al., 2021). With large-scale pre-training on multilingual data, the performance of LLMs in multilingual contexts is gradually improving (Qin et al., 2024), and LLMs are increasingly being applied to the

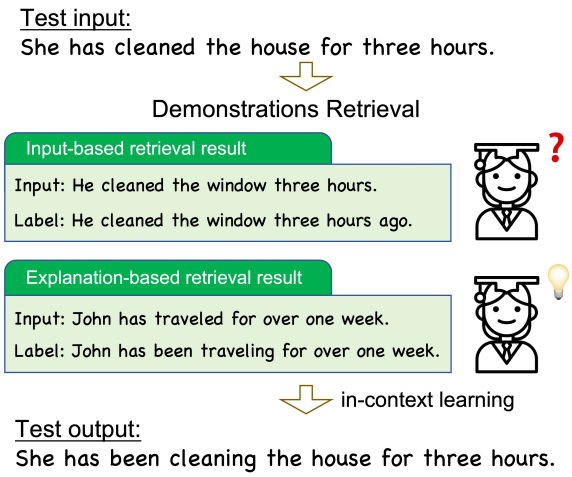

Figure 1: The comparison between input-based demonstrations and explanation-based demonstrations for GEC. Samples with similar inputs do not necessarily contain the same grammatical error patterns. However, through preliminary checks and initial explanations, it is possible to retrieve samples with similar errors from a database indexed by grammatical error explanations, even if the semantics of the demonstrations differs significantly from the test input.

multilingual GEC domain (Luhtaru et al., 2024).

Few-shot inference can achieve strong performance on downstream tasks through in-context learning (ICL) without the need for fine-tuning LLMs (Brown, 2020a). Effective ICL depends on selecting appropriate in-context examples, or demonstrations, which can be guided by established principles (Agrawal et al., 2023). The selection process can be facilitated through retrieval mechanisms, such as k-nearest neighbors (kNN)-based retrieval (Vasselli and Watanabe, 2023). One advantage of example-based GEC systems is that they can enhance system interpretability by providing learners with illustrative examples (Kaneko et al., 2022).

For retrieving in-context examples, previous studies primarily use the test input as the query and the input texts of labeled data as keys for the database (Xu et al., 2024). Matching between query and keys is typically based on semantic or syntactic similarities, or methods like BM25 (Hongjin et al., 2022; Tang et al., 2024; Robertson et al., 2009). This approach is intuitive since both the test input and the input text of labeled data are from the same domain—if the inputs are similar, the corresponding labels are likely similar as well. However, GEC tasks pose a unique challenge, as grammatical errors are often local and bear little correlation to the overall structure of the text. Consequently, input-based retrieval often fails to retrieve references with analogous grammatical errors.

We propose that a more fundamental solution lies in using the relationship between the input text and the label as the basis for retrieval. Recent works have begun to explore leveraging the interaction between text and label to improve in-context example selection (Sun et al., 2024). For GEC, we hypothesize that this relationship can be effectively captured through grammatical error explanations (GEE) (Song et al., 2024).

In this paper, we introduce a novel retrieval method based on natural language explanations, applicable in multilingual settings. Our method uses GEE as both the query and key for retrieval, aiming to retrieve examples with similar grammatical errors to serve as demonstrations for in-context learning. Without requiring model training, this approach enhances LLM few-shot performance on multilingual GEC tasks. Figure 1 presents an illustrative example where traditional semantic matching fails to retrieve relevant samples, while explanation-based matching successfully identifies a sample with a similar grammatical error.

To implement this, we first use labeled data and LLMs to generate GEE and construct a database. During inference, the input text undergoes a grammar check, providing an initial explanation, which is then matched against the GEE database to retrieve suitable demonstrations. These demonstrations are used in the ICL process to produce the final grammatical corrections.

We conducted experiments on GEC datasets across five languages. Our proposed method consistently outperforms semantic and BM25 retrieval in terms of the $F_{0.5}$ score across both open-source and closed-source models. The advantages of the proposed method contain: 1) It requires no additional training, leveraging ICL to tap into the intrinsic GEC capabilities of LLMs. Some studies suggest that the $F_{0.5}$ score underestimates LLM GEC performance, with human evaluations yielding better results (Coyne et al., 2023). 2) It is highly extendable to multilingual settings, as the use of natural language explanations ensures applicability across languages. 3) The system is more interpretable, with each demonstration linked to a GEE.

The main contributions of this work are as follows:

- We propose a novel demonstration retrieval method for in-context learning of GEC based on natural language explanations and design a GEC process involving grammar checks, retrieval, and final corrections.
- We construct GEC databases with grammatical error explanations across multiple languages, which can be applied to different datasets within the same language.
- To the best of our knowledge, this work provides the first aligned evaluation of LLMs' few-shot capabilities for GEC, approaching state-of-the-art performance across several languages.

2 Related Works

2.1 Grammatical Error Correction

Methods in the GEC field have been dominated by sequence-to-sequence generation models (Junczys-Dowmunt et al., 2018; Katsumata and Komachi, 2020) and sequence-to-edit tagging models (Omelianchuk et al., 2020; Lai et al., 2022) since introduction of the Transformer architecture

(Vaswani, 2017). The former generally adopt the encoder-decoder architecture (Lewis et al., 2020a), while the latter use the encoder-only architecture (Devlin et al., 2019). Many state-of-the-art (SOTA) improvements are based on these two architectures, such as incorporating syntactic information into the input (Zhang et al., 2022b) or reranking during the output process (Zhang et al., 2023; Zhou et al., 2023a). Additionally, there are works that split the task into error detection and error correction phases (Li et al., 2023; Li and Wang, 2024).

Since the advent of ChatGPT (Brown, 2020b), LLMs have seen extensive application in natural language processing, particularly in text generation tasks (Li et al., 2024). Some studies have explored the application of LLMs in GEC, such as constructing prompts for direct inference (Davis et al., 2024) and using parallel data for instruction tuning (Fan et al., 2023; Raheja et al., 2024). Research has also analyzed the performance of LLMs in the GEC task (Coyne et al., 2023), including scenarios involving low-resource languages (Penteado and Perez, 2023; Maeng et al., 2023). The prompting strategies significantly impact performance, and there is a tendency for over-correction by LLMs (Loem et al., 2023; Zeng et al., 2024). However, studies indicate that human evaluations rate the modifications made by LLMs highest (Coyne et al., 2023). Our work extends the use of LLMs in multilingual GEC through an easy-to-use in-context learning approach.

2.2 Grammatical Error Explanation

Grammatical error explanation (GEE) refers to the elucidation of the reasons for grammatical errors in sentences and the corresponding grammatical knowledge. This requirement stems from language teaching, where providing explanations alongside error corrections is essential (Liang et al., 2023). To promote the development of interpretable GEC systems, some GEC benchmarks incorporate error types, evidence words, and natural language explanations as forms of grammatical error explanation (Fei et al., 2023; Ye et al., 2024). Kaneko et al. (2022) utilized example-based methods to construct interpretable GEC systems. With the emergence of more LLMs with stronger linguistic capability, some works have used LLMs to generate explanations for each error in the text, and further refine the task definition and evaluation methods for GEE (Kaneko and Okazaki, 2024; Song et al., 2024). Our work leverages the natural language explanations to improve the example-based GEC system.

2.3 Demonstrations Selection

In-context learning (ICL) is a simple and efficient method for utilizing large language models (Brown, 2020a). By using examples as demonstrations to form few-shot inference, ICL can avoid the costly optimization of large parameters. The demonstrations can be hand-crafted (Brown, 2020a) or selected from labeled datasets (Lewkowycz et al., 2022). The choice of demonstrations impacts the final performance of LLMs; therefore, some works design the selection process based on criteria such as complexity and diversity (Fu et al., 2022; Li and Qiu, 2023). Furthermore, through retrieval mechanisms, each test data can be matched with different demonstrations to enhance model performance (Luo et al., 2023). This process is similar to retrieval-augmented generation (RAG), where relevant information is retrieved to improve LLMs performance (Lewis et al., 2020b). Common retrieval methods involve matching inputs to inputs of the labeled data, with frequently used similarity measures including SBERT and BM25 (Reimers, 2019; Hongjin et al., 2022; Robertson et al., 2009). Our work attempts to use matching between explanations as a novel retrieval method.

3 Methods

This section presents our design of a GEC system based on few-shot in-context learning using grammatical error explanation (GEE). First, we describe how to construct a database with GEE using labeled data in advance. Then, we explain the retrieval process based on explanations during inference, where no labels are available. Finally, we outline the few-shot template for GEC.

3.1 Explanation Database

As the foundational and multilingual capabilities of LLMs continue to improve, LLMs can achieve strong performance on the GEE task (Song et al., 2024). In this paper, to ensure extensibility, we guide the LLMs to generate GEE for erroneous samples directly via prompts, as shown in Figure 2.

Building the database requires labeled datasets, which can either come from the training set of a GEC dataset or from typical grammatical correction cases manually compiled by language teachers. In this work, we choose the former as the

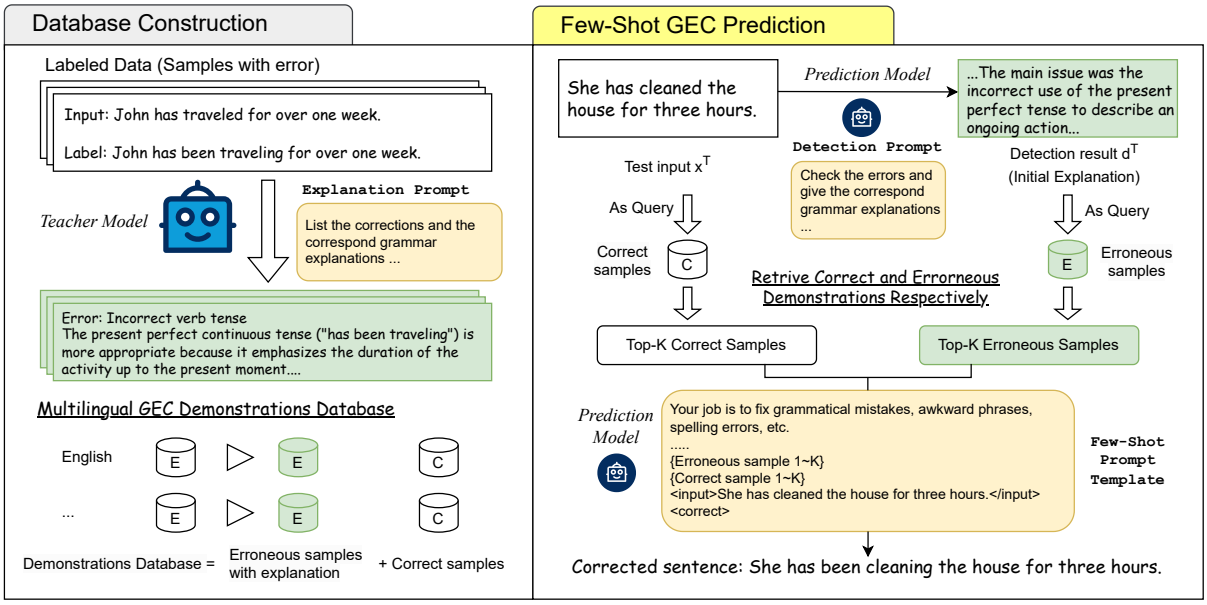

Figure 2: The proposed pipeline for few-shot GEC using the explanation-based demonstration retrieval method. As shown on the left side, we construct sample databases that include explanations. As illustrated on the right side, in the prediction stage, the erroneous samples for in-context demonstrations are retrieved using explanations.

data source, avoiding manual annotation. After applying quality filtering, we obtain a collection of samples to build the database, denoted as $\mathcal{S}$ in Equation 1.

$$\mathcal{S} = \{(x_1, y_1), \ldots, (x_n, y_n)\} \quad (1)$$

Here, x_i and y_i represent the **input text** (potentially erroneous) and the **corrected text** for the i th sample, respectively. Next, $\mathcal{S}$ is divided into two parts: the **erroneous samples** where $x \neq y$ and the **correct samples** where $x = y$. For the former, we use an LLM, referred to as the teacher model, to generate corresponding GEE. The explanations serve as keys in the database. The database for erroneous samples is shown in Equation 2.

$$(\mathcal{K}, \mathcal{V})_E = \{(e, (x, y)) \,|\forall (x, y) \in \mathcal{S}, x \neq y\} \quad (2)$$

Here, $\mathcal{K}$ represents the key, and $\mathcal{V}$ represents the value. The e represents the **GEE** provided by the teacher model based on the pair (x, y). Since GEC methods must also handle input without grammatical errors, a database for correct samples must also be constructed. As GEE cannot be generated for this part, we use the input text x as the key to form the database for correct samples, as shown in Equation 3.

$$(\mathcal{K}, \mathcal{V})_C = \{(x, (x, y)) \,|\forall (x, y) \in \mathcal{S}, x = y\} \quad (3)$$

For each language's labeled data, we can construct two parts of the database as shown in Equations 2 and 3.

3.2 Demonstrations Retrieval

When dealing with the test data, denoted as x^T, the first step is to retrieve a set of samples from the database to serve as demonstrations for few-shot inference. Since the erroneous samples database uses explanations as keys, we need to rewrite the query based on the test input. Here, we use a detection prompt to guide the LLM predictor in detecting potential grammatical errors and the corresponding grammatical knowledge in the test input, which we denote as d^T for an **initial explanation**. This will then serve as the query to retrieve similar samples from the erroneous samples database $(\mathcal{K}, \mathcal{V})_E$. For correct samples, we can directly use x^T as the query, as shown in Equation 4:

$$q_E^T = d^T = \mathrm{LLM}_p\left(\mathrm{prompt}_d(x^T)\right), \quad q_C^T = x^T \quad (4)$$

Here, x^T represents the **test input** text that may contain grammatical errors. q_E^T and q_C^T are the queries used for retrieving samples from the erroneous and correct samples databases, respectively. LLM_p denotes the prediction model, and prompt_d represents the detection prompt. Once the queries are obtained, the kNN method is used to retrieve the top k_E and k_C samples from the two databases, as shown in Equations 5 and 6:

$$\mathcal{N}_E = \left\{\left(e^{(j)}, (x^{(j)}, y^{(j)})\right) \in (\mathcal{K}, \mathcal{V})_E\right\}_{j=1}^{k_E} \quad (5)$$

$$\mathcal{N}_C = \left\{ \left(x^{(j)}, (x^{(j)}, y^{(j)}) \right) \in (\mathcal{K}, \mathcal{V})_C \right\}_{j=1}^{k_C} \quad (6)$$

3.3 In-Context Learning

In few-shot inference, the keys of the retrieved data samples are no longer used, as incorporating long explanations into the context can negatively impact the final GEC performance and significantly slow down inference. The selected demonstrations are simply the list of text pairs retrieved from the database, as shown in Equation 7.

$$D = D_E \oplus D_C \quad (7)$$

Here, D_E and D_C are the lists of (x, y) text pairs from $\mathcal{N}_E$ and $\mathcal{N}_C$, respectively. The concatenated list of samples, D, serves as the demonstrations, which are then combined with the few-shot prompt template for the final GEC prediction. The whole prediction process is outlined at the right half of Figure 2. Given that few-shot examples can provide strong instruction-following capabilities, the prompt template is consistently written in English, with the demonstrations formatted and inserted sequentially. Further details on the prompt can be found in Appendix B.2.

4 Experiments

4.1 Datasets

To evaluate the effectiveness of the proposed method, we conduct experiments on GEC datasets in multiple languages. For English, we used the W&I+LOCNESS (Bryant et al., 2019) training dataset as the database, and the CoNLL-14 (Ng et al., 2014) and BEA-19 datasets as the testing datasets. We also perform experiments on GEC datasets for Chinese, Russian, and German, which have a relatively high number of users. For Chinese, we utilize the HSK dataset (Zhang, 2009) as the database and the NLPCC-18 dataset as the test data (Zhao et al., 2018). For Russian and German, we employ the RULEC and Falko-Merlin datasets, which include both training and testing sets (Rozovskaya and Roth, 2019; Boyd, 2018). Furthermore, we conduct experiments on the relatively low-resource Estonian language, using data from the Tartu learner corpus [1]. In our setting, the L2 learner corpus is used as the database, while the L1 corpus serves as the test data. The database and retriever is implemented by LlamaIndex (Liu, 2022).

[1] https://github.com/TartuNLP/estgec

4.2 Models and Metrics

We selected state-of-the-art multilingual LLMs for our experiments, including Llama-3.1-8B-Instruct by Meta (Dubey et al., 2024) and Qwen2.5-7B-Instruct (Team, 2024) developed by Tongyi. Among the closed-source models, we choose Deepseek2.5 (DeepSeek-AI, 2024) and GPT-4o-mini (Achiam et al., 2023) for its cost-effectiveness.

In the experiments, the teacher model used to generate the GEE-based database is uniformly set to Llama-3.1-8B-Instruct. For constructing the database, we used xlm-roberta-large (Conneau et al., 2019) as the embedding model. For the prediction model in the few-shot inference stage, we conducted experiments with the four LLMs mentioned above, which serve as our primary experimental models. For the test data, we used ERRANT edit extraction results as the labels (Bryant et al., 2017; Zhang et al., 2022a), and we evaluated the precision, recall, and $F_{0.5}$ values using the M2Scorer (Dahlmeier and Ng, 2012).

The multilingual baseline methods used for comparison were: 1) *Random*: randomly selecting samples from the database as demonstrations; 2) *Semantic*: using the input text embedding to perform kNN retrieval (Khandelwal et al., 2020); and 3) *BM25*: employing BM25, a term-based retriever, for retrieval (Robertson et al., 2009). For all methods, including our proposed method, we set $k_E = 4$ erroneous samples and $k_C = 4$ correct samples as demonstrations. The final experimental results are shown in Table 1.

4.3 Results

Performance of the proposed method From Table 1, we can see that when using $F_{0.5}$ as the evaluation metric, our proposed method generally outperforms other reproduced methods under the same experimental setup. Overall, the proposed method significantly outperforms random selection. This is particularly evident with GPT4o-mini, where the $F_{0.5}$ score of the proposed method exceeds that of *Random* by approximately 2 points. Among the baseline methods, the *Semantic* method performs best with open-source LLMs, averaging about 1 point higher than the *Random* method. However, this is not the case for the Deepseek2.5, possibly due to randomness introduced by the temperature setting. As the representative models, Llama3.1 (open-source) and GPT4o-mini (closed-source),

Model	Method	English CoNLL-14			Chinese NLPCC-18			German Falko-Merlin			Russian RULEC			Estonian Tartu-L1		
		P	R	$F_{0.5}$	P	R	$F_{0.5}$	P	R	$F_{0.5}$	P	R	$F_{0.5}$	P	R	$F_{0.5}$
Llama3.1 (8B)	Random	54.02	52.60	53.73	24.87	16.17	22.45	59.62	54.53	58.53	36.70	37.45	36.84	10.34	21.10	11.52
	Semantic	55.21	51.56	54.44	27.88	15.49	24.04	60.03	54.15	58.75	37.69	36.40	37.43	11.04	22.61	12.30
	BM25	54.58	51.58	53.95	24.30	16.41	22.17	59.68	55.53	58.80	37.77	36.06	37.41	-	-	-
	Explanation	55.00	53.04	**54.60**	28.21	15.77	**24.36**	60.35	54.79	**59.15**	37.97	36.97	**37.76**	11.45	24.52	**12.81**
Qwen2.5 (7B)	Random	54.43	53.50	54.24	27.09	25.34	26.72	55.25	48.06	53.65	37.79	40.94	38.38	6.21	17.67	7.14
	Semantic	55.27	52.65	54.73	29.17	24.33	28.06	57.81	48.85	**55.76**	38.60	39.86	38.85	6.41	18.36	7.37
	BM25	54.11	52.25	53.73	28.30	24.95	27.56	57.21	50.18	55.65	38.03	40.02	38.41	-	-	-
	Explanation	55.67	51.60	**54.81**	29.81	23.29	**28.23**	57.33	47.63	55.08	39.16	38.48	**39.02**	6.54	18.21	**7.50**
Deepseek2.5	Random	63.76	45.68	59.08	45.36	24.91	38.96	69.18	46.57	63.06	47.10	33.56	43.59	19.77	20.26	19.87
	Semantic	65.36	40.01	58.01	48.37	21.11	38.44	71.62	43.13	63.27	45.55	28.28	40.59	21.44	22.32	21.61
	BM25	64.30	38.28	56.60	47.03	25.25	**40.11**	71.58	42.53	62.97	47.30	28.83	41.93	-	-	-
	Explanation	64.27	45.51	**59.38**	46.92	20.99	37.62	70.73	44.89	**63.43**	48.37	31.52	**43.70**	22.73	22.47	**22.68**
GPT4o (mini)	Random	57.62	52.38	56.49	28.91	21.69	27.10	68.22	50.86	63.86	39.90	39.77	39.87	19.81	9.40	16.22
	Semantic	59.48	52.24	57.87	30.72	21.67	28.35	69.33	54.02	65.61	41.40	41.96	41.51	22.04	10.13	17.85
	BM25	59.21	51.61	57.52	30.29	21.93	28.14	69.60	51.69	65.09	40.92	39.69	40.67	-	-	-
	Explanation	60.52	52.55	**58.74**	31.77	21.27	**28.92**	69.89	52.78	**65.63**	42.45	41.13	**42.18**	25.35	10.72	**19.91**

Table 1: Results of different demonstration retrieval methods on multilingual GEC datasets. "Random" refers to random selection from database; "Semantic" and "BM25" refers to retrieval by embeddings and BM25 matching, respectively. "Explanation" is the proposed explanation-based method.

Method	Backbone	Lang	$F_{0.5}$ on test set		
			EN	DE	RU
Fine-tuned GEC Single Model					
Rothe et al. (2021)	gT5 xxl	Mono	65.7	**76.0**	**51.6**
Luhtaru et al. (2024)	NLLB	Multi		73.9	
Zhou et al. (2023b)	BART	Mono	**69.6**		
Inference of LLMs					
Davis et al. (2024)	GPT-3.5-Turbo	-	57.2		
Tang et al. (2024)	GPT-3.5-Turbo	-	58.8		
Ours	Deepseek2.5	-	**59.4**	**63.4**	**43.7**

Table 2: Some reported metrics for state-of-the-art (SOTA) multilingual GEC methods based on fine-tuning or direct inference. "EN", "DE", and "RU" denote the CoNLL-14, Falko-Merlin, and RULEC datasets, respectively. For the fine-tuning methods, some models are trained using multilingual mixed data, indicated as "Multi" in the "Lang" column, while other methods fine-tune models separately for each language, marked as "Mono".

the proposed method's average $F_{0.5}$ value exceeds that of the *Semantic* method by 0.36 and 0.84, respectively. Given that the explanation texts in erroneous samples differ entirely from the input text, this result shows that using explanations as a tool for in-context demonstration retrieval is indeed a feasible approach, which is slightly more effective than directly matching based on input text embeddings.

In-context learning performance for multilingual GEC From Table 1, we can observe that both open-source and closed-source LLMs demonstrate certain multilingual GEC capabilities. However, their performances vary significantly across different languages, possibly due to the different capability among languages and the different levels of difficulty of the datasets. for instance, the L1 learner dataset for Estonian is more challenging, resulting in lower performance metrics across all models. Overall, closed-source models tend to outperform open-source models with around 8B parameters, which is reasonable given that closed-source models often have larger architectures and pre-training scales. Additionally, it is noteworthy that, in many cases, regardless of the method used to select demonstrations, the few-shot performance of large models does not differ significantly. For example, when using Llama3.1 on the Falko-Merlin dataset, the $F_{0.5}$ scores differ by only 0.6 across 4 demonstration retrieval strategies. These results indicate that the GEC capabilities of large models largely depend on their inherent abilities, while the selection of examples has some influence around the inherent few-shot performance level.

Comparison with SOTA We compared the performance of the LLMs few-shot inference with several state-of-the-art (SOTA) multilingual GEC methods, as shown in Table 2. Most SOTA methods are all fine-tuned on labeled datasets, with some methods tuning a generative model for each language individually. Only results using the same test sets are listed in the table for comparison. We can see that current few-shot ICL lags behind supervised finetuning (SFT) by about 10 points in terms of $F_{0.5}$. Considering that the GEC performance of LLMs is actually underestimated (Coyne et al., 2023), this gap may be smaller. As the capabilities of LLMs continue to improve, we can expect that in the future, ICL could replace SFT as the most efficient method for multilingual GEC.

Shot	Explanation	English			Chinese			German			Russian			Estonian		
		P	R	$F_{0.5}$	P	R	$F_{0.5}$	P	R	$F_{0.5}$	P	R	$F_{0.5}$	P	R	$F_{0.5}$
0-Shot	-	51.16	53.95	51.70	17.48	19.91	17.92	54.75	55.69	54.93	28.96	41.36	30.81	6.87	20.41	7.92
	Pre	41.16	47.22	42.24	9.84	15.17	10.58	47.16	48.83	47.48	19.38	33.54	21.17	4.83	18.06	5.66
8-Shot	-	55.00	53.04	**54.60**	28.21	15.77	**24.36**	60.35	54.79	**59.15**	37.97	36.97	37.76	11.45	24.52	**12.81**
	Pre	40.62	44.11	41.27	10.35	14.57	10.99	47.60	47.48	47.57	19.91	31.86	21.53	5.09	18.36	5.95
	Post	54.42	51.46	53.80	28.06	14.84	23.81	59.84	53.38	58.43	38.74	36.66	**38.31**	10.87	23.05	12.15

Table 3: Results of incorporating explanations into context. The test datasets are the same as those in Table 1. The explanation can be placed before the corrected text (pre) or after the corrected text (post). The performance of zero-shot (Row 1 of metrics) and the proposed method (Row 3) is provided for comparison.

Method	Llama3.1(8B)			Qwen2.5(7B)		
	P	R	$F_{0.5}$	P	R	$F_{0.5}$
Random	44.2	63.4	47.1	44.8	63.6	47.7
Semantic	45.5	62.8	48.1	45.5	63.4	48.2
BM25	44.2	63.0	47.0	44.7	63.9	47.6
Explanation	45.2	63.3	**48.1**	47.2	62.3	**49.6**

Table 4: Results of different demonstration retrieval methods on BEA-19 test set (English).

5 Discussion

5.1 The Extension to Other Datasets

To verify that the constructed database is effective across multiple test sets of the same language, we conducted experiments on another commonly used English GEC test set, BEA-19 (Bryant et al., 2019), using Llama3.1 and Qwen2.5. As shown in Table 4, the methods exhibit similar trend to those observed in the main experiments. On the Qwen2.5 model, the advantage of the proposed method is more pronounced. These results demonstrate that the explanation-based erroneous samples database we constructed can be transferred to different test scenarios to a certain extent while maintaining the advantage in performance.

5.2 The Performance of In-context Explanations

A key point in using explanations for demonstration retrieval is how to generate a query for retrieval when the corrected text is unknown. We employ a detection prompt to perform grammar checking on the test input, obtaining an initial explanation to serve as the query. This raises the question: *why not directly use this initial explanation as the chain of thought (CoT) in the reasoning process and let the model derive the final GEC answer based on it?* We experimented with this CoT approach, and as shown in Table 3, the first two rows present the results of the zero-shot and CoT experiments with the Llama3.1 model. The significant drop in the $F_{0.5}$ score indicates that this CoT method is completely ineffective for GEC. Possible reasons include: 1) Without labeled data, the quality of the initial explanation is poor, making it only useful as a retrieval tool but misleading when used as an intermediate reasoning step, especially given the minimal difference between input and output in GEC tasks; 2) The intermediate reasoning step may interfere with the model's ability to follow instructions for the final output.

Given the poor performance of CoT in the zero-shot scenario, *What about adding explanations in the context of the few-shot inference?* We conduct experiments on this as well. There are two possible placements for the explanation text: one is before the answer, following an input-explanation-output pattern, and the other is after the answer, following an input-output-explanation pattern. In the experiments, the same format is applied across all examples and test data, with explanations set to "No error in text" for error-free samples. Notably, in the latter pattern, the predicted sample directly outputs the corrected text without using the initial explanation as an intermediate reasoning step as in the former case. The results in rows 4 and 5 of Table 3 show that adding explanation text generally degrades GEC performance. This may be because the lengthy explanation takes up a large portion of the context, which negatively impacts the LLMs' performance on the GEC task itself. Moreover, in row 4, we observe a trend similar to that in row 2, further validating that using the initial explanation as an intermediate reasoning step weakens the results. The proposed method of using it for demonstration retrieval, rather than reasoning, is one of the correct ways to utilize it.

5.3 The Balance of Erroneous and Correct Demonstrations

As described in Section 4.2, the number of correct and erroneous samples in the demonstrations is set to 4 each. The ratio between these two types of samples can affect precision and recall, thereby in-

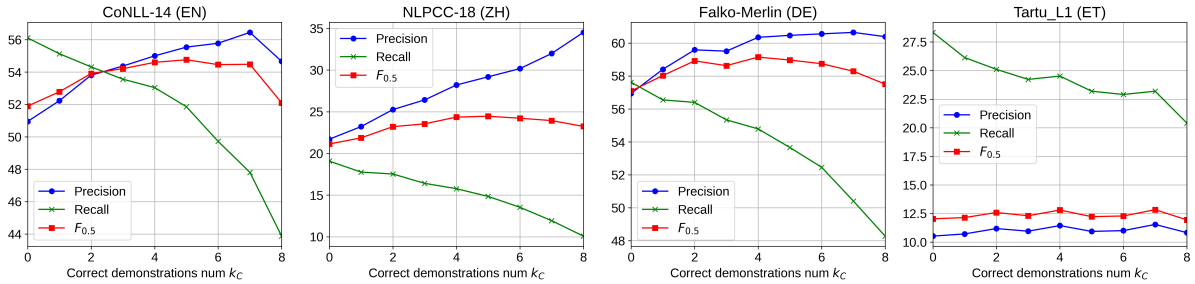

Figure 3: GEC metrics on 4 datasets as the number of the correct samples k_C varies, with a total of $k_E + k_C = 8$ in-context demonstrations consisting of both erroneous and correct samples.

Dataset	Explanation (Using Edits)			Explanation
	P	R	$F_{0.5}$	$F_{0.5}$
English	55.30	54.09	**55.05**	54.60
Chinese	28.23	15.22	24.28	**24.36**
German	60.44	54.85	**59.23**	59.15
Russian	37.85	37.54	**37.79**	37.76
Estonian	10.95	26.14	12.39	**12.81**

Table 5: The performance of the system using the explanation generated by the prompt template with edits extracted by alignment, compared with the direct explanation of the proposed method. The test datasets are the same as those in Table 1.

fluencing the final GEC performance measured by $F_{0.5}$. By fixing the total number of samples at 8 and varying the number of correct samples from 0 to 8, we can create a series of experiments to observe how GEC performance changes with different ratios of correct to erroneous samples. As shown in Figure 3, as the number of correct samples increases and the number of erroneous samples decreases, there is a general trend of increasing precision and decreasing recall. The $F_{0.5}$ score achieves its maximum value when the number of correct samples is 4 or 5. Therefore, for ICL of multilingual GEC, it is reasonable to use an equal number of correct and erroneous samples, which is also our setting.

5.4 The Guidance of Explanation Generation

In previous works on generating grammatical error explanations (GEE), the approach typically involved extracting edit operations by aligning input text with corrected text (Kaneko and Okazaki, 2024; Song et al., 2024). In our proposed method, we directly allow the teacher model to generate GEE based on the original text pairs. Additionally, we also crafted a version that utilizes extracted edits in the explanation prompt and conduct the experiments on LLama3.1. Two prompts are both shown in Appendix B.2. The results are shown in the first three columns of Table 5, with the last column representing the $F_{0.5}$ performance of the proposed method in the main experiment. It is evident that both types of explanations have their strengths and weaknesses. Ultimately, we opted for the GEE generated directly based on the original text pairs, as correcting a grammatical error may involve multiple edit operations. Not specifying the edit operations in advance allows LLMs greater freedom in generating explanations.

6 Conclusion

In conclusion, this paper presents a novel approach for retrieving demonstrations based on grammatical error explanations (GEE) to enhance in-context learning (ICL) for the multilingual grammatical error correction (GEC) task. Our method addresses the challenge of selecting effective in-context demonstrations by matching GEE within a constructed demonstration database, improving the few-shot performance of LLMs in multilingual GEC without requiring additional training.

Future work will explore: 1) methods for indexing correct samples with suitable explanations, which was not fully addressed in this work; and 2) refining the retrieval method. In this study, only text embeddings were used as representations for explanations.

We believe that explanation-based demonstration retrieval has broader implications for ICL across all natural language processing downstream tasks. For GEC, we consider few-shot inference to be the most effective approach for building an explainable multilingual GEC system using LLMs. As LLMs' multilingual capabilities continue to improve, we expect that few-shot ICL will become an efficient and cost-effective solution for GEC.

Limitations

Firstly, due to limitations in computational resources, we only used datasets from 5 languages and conducted experiments on 2 open-source LLMs and 1 closed-source LLM. To further validate the effectiveness of the proposed method, experiments should be conducted on a wider variety of LLMs with different sizes and on more datasets across more languages. Secondly, in this paper, the teacher model used to generate the explanation database is the open-source Llama3.1 with 8B parameters. Since generating the database requires high-quality explanations, using a closed-source LLM with stronger base capabilities could be a better choice and may further improve the performance of the proposed method. Although the database can be reused once it is built, its large scale makes the cost of using closed-source LLMs to build it relatively high. Additionally, we only compared the proposed method with 3 baseline methods. There are indeed many ICL demonstration retrieval methods, but many are implemented for specific tasks or specific languages, making it challenging to reproduce and apply them to multilingual GEC. For example, BM25 retrieval based on LlamaIndex encountered issues with the Estonian dataset, leaving this part of the experiment incomplete.

Ethics Statement

The datasets and models used in this work are publicly available and have been employed exclusively for research purposes. The datasets do not contain any personally identifiable information or offensive content. The large language models (LLMs) utilized in our experiments are consistent with their permitted use in academic research. The code for our proposed method will be made publicly available for academic research in grammatical error correction, adhering to the access conditions of the LLMs used in this study.

LLMs are almost unlikely to generate harmful content under our few-shot settings, as the input remains largely unchanged in the grammatical error correction (GEC) task. However, there is still potential risk of hallucinations, particularly when certain words involving facts are modified by the models.

Additionally, GPT-4o from the ChatGPT platform was used as an AI assistant for refining the paper writing.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Sweta Agrawal, Chunting Zhou, Mike Lewis, Luke Zettlemoyer, and Marjan Ghazvininejad. 2023. In-context examples selection for machine translation. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 8857–8873, Toronto, Canada. Association for Computational Linguistics.

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.

Adriane Boyd. 2018. Using Wikipedia edits in low resource grammatical error correction. In *Proceedings of the 2018 EMNLP Workshop W-NUT: The 4th Workshop on Noisy User-generated Text*, pages 79–84, Brussels, Belgium. Association for Computational Linguistics.

Tom B Brown. 2020a. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

Tom B Brown. 2020b. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

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, Florence, Italy. Association for Computational Linguistics.

Christopher Bryant, Mariano Felice, and Ted Briscoe. 2017. Automatic annotation and evaluation of error types for grammatical error correction. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 793–805, Vancouver, Canada. Association for Computational Linguistics.

Christopher Bryant, Zheng Yuan, Muhammad Reza Qorib, Hannan Cao, Hwee Tou Ng, and Ted Briscoe. 2023. Grammatical Error Correction: A Survey of the State of the Art. *Computational Linguistics*, 49(3):643–701.

Andrew Caines, Luca Benedetto, Shiva Taslimipoor, Christopher Davis, Yuan Gao, Oeistein Andersen, Zheng Yuan, Mark Elliott, Russell Moore, Christopher Bryant, et al. 2023. On the application of large language models for language teaching and assessment technology. *arXiv preprint arXiv:2307.08393*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *CoRR*, abs/1911.02116.

Steven Coyne, Keisuke Sakaguchi, Diana Galvan-Sosa, Michael Zock, and Kentaro Inui. 2023. Analyzing the performance of gpt-3.5 and gpt-4 in grammatical error correction. *arXiv preprint arXiv:2303.14342*.

Daniel Dahlmeier and Hwee Tou Ng. 2012. Better evaluation for grammatical error correction. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 568–572, Montréal, Canada. Association for Computational Linguistics.

Christopher Davis, Andrew Caines, O Andersen, Shiva Taslimipoor, Helen Yannakoudakis, Zheng Yuan, Christopher Bryant, Marek Rei, and Paula Buttery. 2024. Prompting open-source and commercial language models for grammatical error correction of English learner text. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 11952–11967, Bangkok, Thailand and virtual meeting. Association for Computational Linguistics.

DeepSeek-AI. 2024. Deepseek-v2: A strong, economical, and efficient mixture-of-experts language model. *Preprint*, arXiv:2405.04434.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Yaxin Fan, Feng Jiang, Peifeng Li, and Haizhou Li. 2023. Grammargpt: Exploring open-source llms for native chinese grammatical error correction with supervised fine-tuning. In *CCF International Conference on Natural Language Processing and Chinese Computing*, pages 69–80.

Yuejiao Fei, Leyang Cui, Sen Yang, Wai Lam, Zhenzhong Lan, and Shuming Shi. 2023. Enhancing grammatical error correction systems with explanations. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7489–7501, Toronto, Canada. Association for Computational Linguistics.

Yao Fu, Hao Peng, Ashish Sabharwal, Peter Clark, and Tushar Khot. 2022. Complexity-based prompting for multi-step reasoning. In *The Eleventh International Conference on Learning Representations*.

SU Hongjin, Jungo Kasai, Chen Henry Wu, Weijia Shi, Tianlu Wang, Jiayi Xin, Rui Zhang, Mari Ostendorf, Luke Zettlemoyer, Noah A Smith, et al. 2022. Selective annotation makes language models better few-shot learners. In *The Eleventh International Conference on Learning Representations*.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Shubha Guha, and Kenneth Heafield. 2018. Approaching neural grammatical error correction as a low-resource machine translation task. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 595–606, New Orleans, Louisiana. Association for Computational Linguistics.

Masahiro Kaneko and Naoaki Okazaki. 2024. Controlled generation with prompt insertion for natural language explanations in grammatical error correction. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 3955–3961, Torino, Italia. ELRA and ICCL.

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.

Anisia Katinskaia and Roman Yangarber. 2021. Assessing grammatical correctness in language learning. In *Proceedings of the 16th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 135–146, Online. Association for Computational Linguistics.

Satoru Katsumata and Mamoru Komachi. 2020. Stronger baselines for grammatical error correction using a pretrained encoder-decoder model. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 827–832, Suzhou, China. Association for Computational Linguistics.

Urvashi Khandelwal, Angela Fan, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2020. Nearest neighbor machine translation. *arXiv preprint arXiv:2010.00710*.

Shaopeng Lai, Qingyu Zhou, Jiali Zeng, Zhongli Li, Chao Li, Yunbo Cao, and Jinsong Su. 2022. Type-driven multi-turn corrections for grammatical error correction. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 3225–3236,

Dublin, Ireland. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020a. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020b. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, Yuhuai Wu, Behnam Neyshabur, Guy Gur-Ari, and Vedant Misra. 2022. Solving quantitative reasoning problems with language models. In *Advances in Neural Information Processing Systems*, volume 35, pages 3843–3857. Curran Associates, Inc.

Junyi Li, Tianyi Tang, Wayne Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. 2024. Pre-trained language models for text generation: A survey. *ACM Computing Surveys*, 56(9):1–39.

Wei Li and Houfeng Wang. 2024. Detection-correction structure via general language model for grammatical error correction. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1748–1763, Bangkok, Thailand. Association for Computational Linguistics.

Xiaonan Li and Xipeng Qiu. 2023. Finding supporting examples for in-context learning. *arXiv preprint arXiv:2302.13539*, 3.

Yinghao Li, Xuebo Liu, Shuo Wang, Peiyuan Gong, Derek F. Wong, Yang Gao, Heyan Huang, and Min Zhang. 2023. TemplateGEC: Improving grammatical error correction with detection template. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6878–6892, Toronto, Canada. Association for Computational Linguistics.

Kai-Hui Liang, Sam Davidson, Xun Yuan, Shehan Panditharatne, Chun-Yen Chen, Ryan Shea, Derek Pham, Yinghua Tan, Erik Voss, and Luke Fryer. 2023. ChatBack: Investigating methods of providing grammatical error feedback in a GUI-based language learning chatbot. In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*, pages 83–99, Toronto, Canada. Association for Computational Linguistics.

Jerry Liu. 2022. LlamaIndex.

Mengsay Loem, Masahiro Kaneko, Sho Takase, and Naoaki Okazaki. 2023. Exploring effectiveness of GPT-3 in grammatical error correction: A study on performance and controllability in prompt-based methods. In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*, pages 205–219, Toronto, Canada. Association for Computational Linguistics.

Agnes Luhtaru, Elizaveta Korotkova, and Mark Fishel. 2024. No error left behind: Multilingual grammatical error correction with pre-trained translation models. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1209–1222, St. Julian's, Malta. Association for Computational Linguistics.

Man Luo, Xin Xu, Zhuyun Dai, Panupong Pasupat, Mehran Kazemi, Chitta Baral, Vaiva Imbrasaite, and Vincent Y Zhao. 2023. Dr. icl: Demonstration-retrieved in-context learning. *arXiv preprint arXiv:2305.14128*.

Junghwan Maeng, Jinghang Gu, and Sun-A Kim. 2023. Effectiveness of ChatGPT in Korean grammatical error correction. In *Proceedings of the 37th Pacific Asia Conference on Language, Information and Computation*, pages 464–472, Hong Kong, China. Association for Computational Linguistics.

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, Baltimore, Maryland. Association for Computational Linguistics.

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, Seattle, WA, USA → Online. Association for Computational Linguistics.

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*.

Libo Qin, Qiguang Chen, Yuhang Zhou, Zhi Chen, Yinghui Li, Lizi Liao, Min Li, Wanxiang Che, and Philip S Yu. 2024. Multilingual large language model: A survey of resources, taxonomy and frontiers. *arXiv preprint arXiv:2404.04925*.

Vipul Raheja, Dimitris Alikaniotis, Vivek Kulkarni, Bashar Alhafni, and Dhruv Kumar. 2024. mEdIT: Multilingual text editing via instruction tuning. In *Proceedings of the 2024 Conference of the North*

American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), pages 979–1001, Mexico City, Mexico. Association for Computational Linguistics.

N Reimers. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Stephen Robertson, Hugo Zaragoza, et al. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389.

Sascha Rothe, Jonathan Mallinson, Eric Malmi, Sebastian Krause, and Aliaksei Severyn. 2021. A simple recipe for multilingual grammatical error correction. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 702–707, Online. Association for Computational Linguistics.

Alla Rozovskaya and Dan Roth. 2019. Grammar Error Correction in Morphologically Rich Languages: The Case of Russian. *Transactions of the Association for Computational Linguistics*, 7:1–17.

Yixiao Song, Kalpesh Krishna, Rajesh Bhatt, Kevin Gimpel, and Mohit Iyyer. 2024. GEE! grammar error explanation with large language models. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 754–781, Mexico City, Mexico. Association for Computational Linguistics.

Hao Sun, Yong Jiang, Bo Wang, Yingyan Hou, Yan Zhang, Pengjun Xie, and Fei Huang. 2024. Retrieved in-context principles from previous mistakes. *arXiv preprint arXiv:2407.05682*.

Xin Sun, Tao Ge, Shuming Ma, Jingjing Li, Furu Wei, and Houfeng Wang. 2022. A unified strategy for multilingual grammatical error correction with pretrained cross-lingual language model. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, pages 4367–4374. International Joint Conferences on Artificial Intelligence Organization. Main Track.

Chenming Tang, Fanyi Qu, and Yunfang Wu. 2024. Ungrammatical-syntax-based in-context example selection for grammatical error correction. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 1758–1770, Mexico City, Mexico. Association for Computational Linguistics.

Qwen Team. 2024. Qwen2.5: A party of foundation models.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Justin Vasselli and Taro Watanabe. 2023. A closer look at k-nearest neighbors grammatical error correction. In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*, pages 220–231, Toronto, Canada. Association for Computational Linguistics.

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

Xin Xu, Yue Liu, Panupong Pasupat, Mehran Kazemi, et al. 2024. In-context learning with retrieved demonstrations for language models: A survey. *arXiv preprint arXiv:2401.11624*.

Jingheng Ye, Shang Qin, Yinghui Li, Xuxin Cheng, Libo Qin, Hai-Tao Zheng, Peng Xing, Zishan Xu, Guo Cheng, and Zhao Wei. 2024. Excgec: A benchmark of edit-wise explainable chinese grammatical error correction. *arXiv preprint arXiv:2407.00924*.

Zheng Yuan and Ted Briscoe. 2016. Grammatical error correction using neural machine translation. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 380–386, San Diego, California. Association for Computational Linguistics.

Min Zeng, Jiexin Kuang, Mengyang Qiu, Jayoung Song, and Jungyeul Park. 2024. Evaluating prompting strategies for grammatical error correction based on language proficiency. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 6426–6430, Torino, Italia. ELRA and ICCL.

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

Ying Zhang, Hidetaka Kamigaito, and Manabu Okumura. 2023. Bidirectional transformer reranker for grammatical error correction. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 3801–3825, Toronto, Canada. Association for Computational Linguistics.

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 the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2518–2531, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

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*, pages 439–445, Cham. Springer International Publishing.

Houquan Zhou, Yumeng Liu, Zhenghua Li, Min Zhang, Bo Zhang, Chen Li, Ji Zhang, and Fei Huang. 2023a. Improving Seq2Seq grammatical error correction via decoding interventions. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 7393–7405, Singapore. Association for Computational Linguistics.

Houquan Zhou, Yumeng Liu, Zhenghua Li, Min Zhang, Bo Zhang, Chen Li, Ji Zhang, and Fei Huang. 2023b. Improving Seq2Seq grammatical error correction via decoding interventions. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 7393–7405, Singapore. Association for Computational Linguistics.

A Dataset Statistics

As described in Section 4.1, our dataset usage consists of two parts: the labeled data used to build the database and the test data used to evaluate the proposed method. The labeled data samples used to construct the database are initially filtered by length, with a minimum of 10 and a maximum of 60 tokens to ensure quality. Additionally, we limited the number of total samples to 25,000. The filtering process also helped to reduce the cost of constructing the explanation database. The statistics of the datasets used in this paper are shown in Table 6.

B Experimental settings

B.1 Model Settings

We conducted experiments using open-source LLMs available on the Huggingface community, including Llama-3.1-8B-Instruct and Qwen2.5-7B-Instruct. During generation, we set the temperature to 0 and avoided any random sampling strategies to eliminate output randomness for all open-source LLMs. For closed-source LLMs, we utilized their official APIs and applied their default settings to get better performance, which introduced some degree of randomness. As a baseline for comparison, the method of "Random" selecting samples for open-source LLMs is evaluated by running it with 3 different random seeds and reporting the average result, while the other method will get a fixed output so only one-round inference is performed. Due to the cost of usage, only the single-round results of API-based closed LLMs are reported.

B.2 Prompt Settings

Three types of prompts are used in our experiments, as illustrated in Figure 2. During the database construction phase, an explanation prompt is employed to generate explanations. In the prediction phase, a detection prompt is used to provide an initial explanation when no ground truth correction is available. The few-shot prompt template is used for the final few-shot inference incorporating demonstrations, and it is applied both in the comparative methods and the proposed method. Table 7 displays the actual prompts used in our experiments. For the detection prompt, we leveraged GPT-4o[2] to generate both detailed and short version. As for the explanation prompt, one approach was to directly combine the detection prompt with the corrected sentence, which served as our primary method. In Section 5.4, we also explored an edit-guided explanation prompt. The few-shot prompt was formulated based on prior work (Tang et al., 2024; Davis et al., 2024).

[2]https://chatgpt.com

	Database			Test Dataset	
Language	**Origin**	**#Erroneous**	**#Correct**	**Origin**	**#Total**
English	W&I+LOCNESS	20185	6839	CoNLL-14	1312
Chinese	HSK	25000*	25000*	NLPCC-18	2000
German	Falko-Merlin	11801	1916	Falko-Merlin	2337
Russian	RULEC	961	913	RULEC	5000
Estonian	Tartu-L2-Corpus	7156	2**	Tartu-L1-Corpus	1453

Table 6: The GEC data quantity used. For the database, #Erroneous represents the number of erroneous samples, and #Correct represents the number of correct samples. For the test data, #Total indicates the total number of samples. *For the HSK dataset, due to its large size, we randomly selected 25,000 erroneous and 25,000 correct samples for database construction. **The Tartu-L2-Corpus contains only 4 correct samples, and after filtering, 2 samples remained.

Prompt	Usage	Prompt Content
Explanation	**Teacher Model**	You are tasked with performing a comprehensive grammatical error analysis on the following text, which may contain errors in various languages. Your job is to identify any grammatical, syntactic, punctuation, or spelling errors in the text. For each detected error, follow these steps: Identify the Error: List each error separately. Correction: Suggest the appropriate correction for each identified error. Explanation: Provide a brief explanation for why the correction is necessary. This should include references to specific grammar rules, conventions, or language-specific nuances (e.g., verb conjugation, article-noun agreement, preposition use, punctuation rules). For spelling errors, offer an explanation if the mistake could arise from common language-specific confusions (e.g., homophones, loanwords). For punctuation issues, explain the relevant punctuation rules (e.g., comma placement in subordinate clauses, quotation marks, etc.). For syntax or word order issues, explain how sentence structure works in the language and why the original sentence does not follow the norm. Minimal Impact on Meaning: Ensure that the corrections you propose do not alter the original meaning of the sentence. The goal is to preserve the intent of the writer while correcting errors. When explaining each error, keep in mind that the explanations should be clear and concise but still detailed enough to be educational. Whenever possible, reference grammatical terms (e.g., agreement, tense, case, gender, aspect) relevant to the error. Important Notes: If the text is multilingual, address each language's grammar rules separately. Your explanations should cater to a general audience, meaning that while your responses can be technical, they should still be easily understood by someone with a basic understanding of grammar. Now, perform this process for the following text: [The given text]: {text} And we will give you the corrected version of the given text below. Your analysis of grammatical errors should lead to the given text being corrected to this specific version. [The corrected version]: {label} [Corrections made and the brief reasons for the errors]:
Explanation (Using Edits)	**Teacher Model**	You, a language expert, can briefly explain how to judge a sentence is grammatically correct and why some corrections are essential. For the given text: {text} Corrected text: {label} {Extracted edits here, like: insert "xxx" between "yyy" and "zzz"; replace "xxx" with "yyy".} Please explain briefly why you made these corrections. Your Explanation:
Detection (Detailed)	**Prediction Model** **Llama3.1** **Deepseek2.5** **GPT4o-mini**	You are tasked with performing a comprehensive grammatical error analysis on the following text, which may contain errors in various languages. Your job is to identify any grammatical, syntactic, punctuation, or spelling errors in the text. For each detected error, follow these steps: Identify the Error: List each error separately. Correction: Suggest the appropriate correction for each identified error. Explanation: Provide a brief explanation for why the correction is necessary. This should include references to specific grammar rules, conventions, or language-specific nuances (e.g., verb conjugation, article-noun agreement, preposition use, punctuation rules). For spelling errors, offer an explanation if the mistake could arise from common language-specific confusions (e.g., homophones, loanwords). For punctuation issues, explain the relevant punctuation rules (e.g., comma placement in subordinate clauses, quotation marks, etc.). For syntax or word order issues, explain how sentence structure works in the language and why the original sentence does not follow the norm. Minimal Impact on Meaning: Ensure that the corrections you propose do not alter the original meaning of the sentence. The goal is to preserve the intent of the writer while correcting errors. When explaining each error, keep in mind that the explanations should be clear and concise but still detailed enough to be educational. Whenever possible, reference grammatical terms (e.g., agreement, tense, case, gender, aspect) relevant to the error. Important Notes: If the text is multilingual, address each language's grammar rules separately. Your explanations should cater to a general audience, meaning that while your responses can be technical, they should still be easily understood by someone with a basic understanding of grammar. Now, perform this process for the following text: [The given text]: {source} [Corrections made and the brief reasons for the errors]:
Detection (Short)	**Prediction Model** **Qwen2.5**	Your task is to detect grammatical errors in the given text and provide corrections along with explanations based on the relevant grammar rules. For each error found, specify the type of error (e.g., subject-verb agreement, tense inconsistency) and explain why it is incorrect. Then provide the correct version of the sentence and briefly explain the grammar rule that applies. Please follow this structure for your response: [The given text]: {source} [Corrections made and the brief reasons for the errors]:
Few-Shot	**Prediction Model** **All**	You are an language expert who is responsible for grammatical, lexical and orthographic error corrections given an input sentence. Your job is to fix grammatical mistakes, awkward phrases, spelling errors, etc. following standard written usage conventions, but your corrections must be conservative. Please keep the original sentence (words, phrases, and structure) as much as possible. The ultimate goal of this task is to make the given sentence sound natural to native speakers without making unnecessary changes. Corrections are not required when the sentence is already grammatical and sounds natural. There is an erroneous sentence between '<erroneous sentence>' and '</erroneous sentence>'. Then grammatical errors in the erroneous sentence will be corrected. The corrected version will be between '<corrected sentence>' and '</corrected sentence>'. <erroneous sentence>{text}</erroneous sentence> <corrected sentence>{label}</corrected sentence> ... <erroneous sentence>{text}</erroneous sentence> <corrected sentence>{label}</corrected sentence> <erroneous sentence>{source}</erroneous sentence> <corrected sentence>

Table 7: The prompts for the proposed method. {text} and {label} means the input text and correct sentence (label) for labeled GEC data. {source} represents the test input text.