Encode Errors: Representational Retrieval of In-Context Demonstrations for Multilingual Grammatical Error Correction

Anonymous ACL submission

Abstract

Grammatical Error Correction (GEC) involves detecting and correcting incorrect usage of grammar. While large language models (LLMs) with in-context learning (ICL) capabilities have shown significant progress on various natural language processing (NLP) tasks, their few-shot performance on GEC remains suboptimal. This is mainly due to the challenge of retrieving suitable in-context demonstrations that capture error patterns instead of semantic similarity. In this paper, we demonstrate that LLMs can inherently capture information related to grammatical errors through their internal states. We extract from these states the Grammatical Error Representation (GER), an informative and semantically neutral encoding of grammatical errors. Our novel GER-based retrieval method significantly boosts performance in ICL settings on multilingual GEC datasets, improving the precision of correction. For high-resource languages, our results on 8B-sized open-source models match those of closed-source models such as Deepseek2.5 and GPT-4o-mini. For low-resource languages, our $F_{0.5}$ scores surpass the baseline by a factor of 1.25. This method provides a more precise and resource-efficient solution for multilingual GEC, offering a promising direction for interpretable GEC research. The code will be available upon acceptance.

1 Introduction

Grammatical Error Correction (GEC) is an important research field in natural language processing (NLP), as it requires language models to understand the syntax, semantics, and pragmatics underlying the subtle structures of natural sentences (Bryant et al., 2023). Initially considered a specific case of machine translation (Yuan and Briscoe, 2016; Junczys-Dowmunt et al., 2018), GEC has evolved with two dominant approaches. Text-to-text methods (Katsumata and Komachi, 2020; Sun et al., 2021; Ingólfsdóttir et al., 2023) construct pairs of erroneous input and corrected output sentences and train encoder-decoder models, while text-to-edit approaches (Stahlberg and Kumar, 2020; Omelianchuk et al., 2020) rely on the encoder's capabilities to identify errors and make corrections.

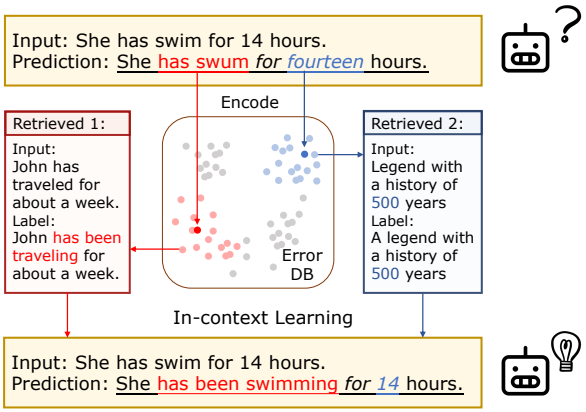

Figure 1: A minimal working example demonstrating the workflow of representational retrieval. Given an erroneous input with predictions containing both under-correction (marked in red) and over-correction (marked in blue), we first transform the error information detected by the model into Grammatical Error Representation (GER). Then, we retrieve GER-adjacent demonstrations from the database, which exhibit similar error patterns to the input. These demonstrations guide the model to make more precise corrections and alleviate over-corrections.

As Large Language Models (LLMs) come to prominence, they have achieved considerable results in GEC (Maeng et al., 2023; Zeng et al., 2024). However, LLMs that are not specifically adapted for GEC tasks face two main challenges: misalignment and over-correction (Loem et al., 2023). These models often produce corrections misaligned with human-annotated labels, and they may over-correct error-free parts, rewriting them into more fluent forms. This behavior violates the

minimum edit distance principle (Nagata and Sakaguchi, 2016).

Since few-shot inference is widely used to bridge alignment gaps in downstream tasks through in-context learning (ICL), LLM-based GEC systems have leveraged correction examples from databases to improve performance and interpretability (Davis et al., 2024; Song et al., 2024). However, vanilla retrieval methods based on sentence embedding or k-nearest neighbors (kNN) struggle to meet the unique needs of grammatical error selection (Vasselli and Watanabe, 2023). Grammatical errors are typically localized structural issues that are independent of word meanings, but model embeddings combine syntax and semantics into a single vector, making it difficult to retrieve samples with similar error patterns.

In this paper, we argue that despite the alignment problem in GEC tasks, language-proficient models can smoothly distinguish wrong from right and identify error patterns. This suggests that we should focus less on the generation capabilities of LLMs, but more on their internal knowledge about grammatical errors. We probe for two key questions: *How does a language model encode grammatical errors internally?* and *can we extract grammatical error representations that are disentangled from semantics?*

In this paper, we introduce a novel method to extract the Grammatical Error Representations (GER), a precise and interpretable representation of grammatical errors with less semantic noise, for guiding the retrieval of in-context demonstrations. Specifically, we compute error vectors (EV) by applying PCA to the difference between the hidden states of erroneous and correct tokens. We then project the hidden states of errors onto the EV to obtain the GER. As shown in Figure 1, our GER preserves the proximity of fine-grained errors: during retrieval, each detected error aligns with similar error patterns. Additionally, over-corrected tokens are queried for similar over-correction cases in the database, improving the precision of the correction process. During inference, the number of retrieved examples dynamically adjusts based on the detected errors in the sentence, allowing for more efficient use of computational resources.

We conduct extensive experiments to demonstrate our consistent outperformance on GEC datasets across five languages. Without additional training or generation, we obtain high-quality and interpretable demonstrations for ICL. Our results surpass state-of-the-art (SOTA) GEC retrieval methods, increasing $F_{0.5}$ by up to 9 points for high-resource languages like English, and by a factor of 1.25 for low-resource languages like Estonian. On open-source 8B-sized models, our approach yields results comparable to contemporary closed-source LLMs like Deepseek2.5 (Liu et al., 2024a) and GPT-4o-mini (Achiam et al., 2023).

Our contributions are summarized as follows:

- We introduce a novel method to disentangle grammatical errors from semantic information and into grammatical error representations (GER), a high-quality encoding for grammatical errors.
- We develop an effective retriever to query examples with similar error patterns based on GER, enabling powerful ICL with LLMs across multilingual datasets.
- To the best of our knowledge, we are the first to explore the relationship between grammatical errors and LLM representations, offering new insights for utilizing LLMs' representations to guide GEC tasks.

2 Related Works

2.1 Grammatical Error Correction

Grammatical Error Correction (GEC) systems have wide applications in proofreading, education, and second language acquisition (Kaneko et al., 2022; Caines et al., 2023; Liang et al., 2023). Research has primarily focused on two Transformer-based approaches: sequence-to-sequence generation (Yuan and Briscoe, 2016; Junczys-Dowmunt et al., 2018; Li et al., 2022) and sequence-to-edit tagging (Awasthi et al., 2019; Omelianchuk et al., 2020). Given the local and sparse nature of grammatical errors, researchers often generate synthetic data (Stahlberg and Kumar, 2024), incorporate additional information (Zhang et al., 2022; Fei et al., 2023), or add extra processing steps during inference (Lai et al., 2022; Zhou et al., 2023; Zhang et al., 2023; Li and Wang, 2024) to boost performance. Recent work also explores LLMs for GEC, either through direct correction generation (Loem et al., 2023) or instruction tuning (Fan et al., 2023). Despite challenges like over-correction and misalignment in LLMs (Vasselli and Watanabe, 2023), human evaluations often rate their corrections highly (Zeng et al., 2024).

2.2 Interpretable Representations in LLMs

Although LLMs are often seen as black boxes due to their vast number of parameters, recent research has shown that they develop emergent structures within their representations (Elhage et al., 2021; Zou et al., 2023). In the simplest case, a single dimension within the model is sufficient to characterize a specific behavior (Arditi et al., 2024; Sheng et al., 2024); more complex circuits may involve dozens of neurons distributed across different layers interacting to form meaningful components (Wang et al., 2023). These interpretable components can be understood and controlled through techniques like adding, deleting, replacing, or tuning (Liu et al., 2024b; Wu et al., 2024). Our work is the first to explore and utilize LLMs' representations related to grammatical errors.

2.3 In-Context Learning in GEC

LLMs have demonstrated the ability to align their generated results to the knowledge domain and style of several in-context examples (Brown et al., 2020; Saakyan and Muresan, 2024). The few-shot inference paradigm avoids the additional parameters and computational costs of fine-tuning with downstream tasks.

The selection of examples in the prompt largely affects the performance of ICL. Researchers have increased retrieval results by filtering the data, (He et al., 2021; Peng et al., 2023) or optimizing query encodings and retrieval algorithms (Li and Qiu, 2023; Wang et al., 2024). The most helpful examples usually share similar encodings to the query, along with sufficient diversity to increase information entropy. However, for GEC tasks, the selection goal is hard to achieve. Due to the entanglement of syntax and semantics, the error encodings tend to retrieve examples with similar meanings instead of analogous error types (Vasselli and Watanabe, 2023; Song et al., 2024). Recent works tackle this entanglement by having models write error explanations, which are then used to retrieve errors based on the explanation embeddings (Li et al., 2025). Despite the improved retrieval performance, these methods still suffer from coarse sentence-level granularity and the semantic noise introduced by generated explanations. Moreover, no work has yet addressed the issue of over-correction.

3 Methods

In this section, we describe a novel method for extracting vectors that characterize grammatical error information and using them to create semantically neutral grammatical error representations (GER). GER from the training dataset is stored in a database, where each error is associated with its original and corrected texts. During inference, the model retrieves similar correction examples based on GER to guide corrections, with the flexibility to dynamically adjust the number of examples depending on the complexity of the input sentence. The final GEC prediction is generated by combining the retrieved examples with a correction template.

3.1 Extraction of Error Vectors

Given a GEC dataset $\mathcal{S} = \{(x^{(k)}, y^{(k)})\}_{k=1}^{N}$, each sample consists of a potentially erroneous text x and its parallel corrected text y. x is prompted with an initial correction prompt, which can be zero-shot or filled with random initial demonstrations [1]. During the generation of the initial prediction $\hat{y}$, we extract the hidden state at the i-th position from the t-th layer of the model, denoted as $\mathbf{h}_i^{(t)}$, obtaining the set $\mathcal{H}^{(t)}$. The choice of the specific layer t is discussed in 5.2. For simplicity, the subsequent formulas omit the layer index.

$$\hat{y} = \text{LLM}\big(\text{prompt}_{\text{init}}(x)\big) \tag{1}$$

$$\mathcal{H}^{(t)} = \left\{ \mathbf{h}_i^{(t)} \mid \forall i \in \{1, \ldots, |\hat{y}|\} \right\} \tag{2}$$

By comparing x and $\hat{y}$, we identify all edits made by the LLM and collect the set of edited positions $\mathcal{E}$ and unedited positions $\mathcal{U}$. The corresponding hidden states, $\mathcal{H}_{\mathcal{E}}$ and $\mathcal{H}_{\mathcal{U}}$, contain the information necessary for the model to decide whether to correct. The difference between these sets captures the directions that guide the model from copying the original text to making corrections - precisely the information related to grammatical errors. We multiply this difference by a random sign variable $\alpha_{e,u} \in \{-1, 1\}$, which randomly changes the sign to enhance the weight of the error-related directions in the principal components.

$$\begin{aligned} \mathcal{E} &= \{i \mid \text{Align}(x, \hat{y})[i] = \text{Edited}\}_{i=1}^{|\hat{y}|} \\ \mathcal{U} &= \{i \mid \text{Align}(x, \hat{y})[i] = \text{Unedited}\}_{i=1}^{|\hat{y}|} \end{aligned} \tag{3}$$

[1] The selection of examples in the initial prompt is discussed in Section 5.3.

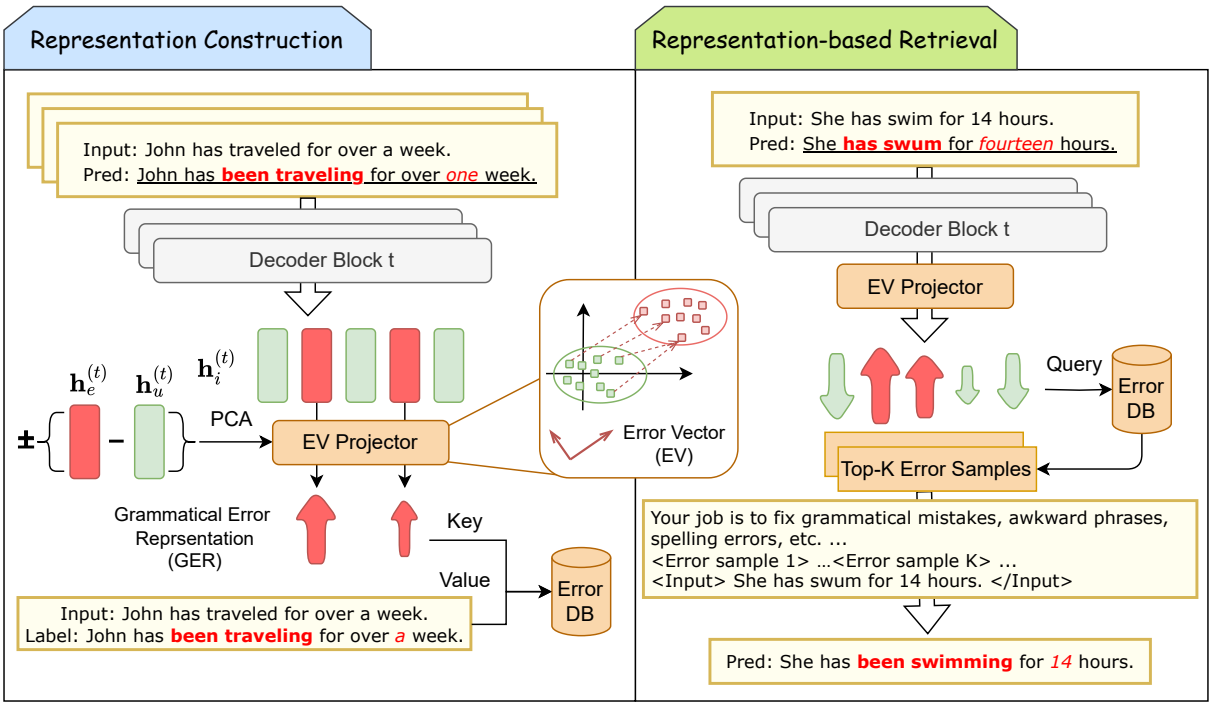

Figure 2: The pipeline for proposed representational retrieval for few-shot GEC. Left: The hidden states that best reflect the error information are extracted and transformed through PCA to obtain error vectors (EV). The projections onto EV, denoted as grammatical error representations (GER), are stored as keys in the database. Right: During inference, GER of the test input serves as the query to retrieve similar error patterns to aid correction.

$$\mathcal{H}_{\mathcal{E}} = \{\mathbf{h}_i \mid \forall i \in \mathcal{E}\}$$
$$\mathcal{H}_{\mathcal{U}} = \{\mathbf{h}_i \mid \forall i \in \mathcal{U}\} \quad (4)$$

$$\Delta\mathbf{H} = \{\alpha_{e,u}(\mathbf{h}_e - \mathbf{h}_u) \mid \forall e \in \mathcal{E}, \forall u \in \mathcal{U}\} \quad (5)$$

We apply Principal Component Analysis (PCA) to the difference $\Delta\mathbf{H}$, yielding a set of principal components $\mathbf{R}$. As shown in Section 5.1, $\mathbf{R}$ encapsulates information related to grammatical errors, with the first principal component $\mathbf{r}_1$ representing the simplicity of the error, indicating how easy it can be corrected. The first two principal components are sufficient for encoding simple error types disentangled from the text's meaning. We designate $\mathbf{R}$ as the **error vectors (EV)** of the model.

$$\Delta\mathbf{H} = \mathbf{U}\mathbf{\Sigma}\mathbf{R}^{\top} \quad (6)$$

3.2 Construction of GER Database

For each correction $e \in \mathcal{E}$, we average the difference between $\mathbf{h}_e$ and all corresponding $\mathbf{h}_u \in \mathcal{H}_{\mathcal{U}}$ in the same sentence, canceling out noise from token meanings and positional embeddings. We then apply PCA, projecting onto m principal components[2] to obtain the **grammatical error representation (GER)** $\mathbf{p}_e^{(m)}$. We omit dimension labeling where it is not necessary. GER serves as the key, with the corresponding pair (x, y) as the label, to construct the GER database $\mathcal{D}$.

$$\Delta\bar{\mathbf{h}}_e = \frac{1}{|\mathcal{U}|}\sum_{u\in\mathcal{U}}(\mathbf{h}_e - \mathbf{h}_u) \quad (7)$$

$$\mathbf{p}_e^{(m)} = \left[\mathbf{r}_1, \mathbf{r}_2, ..., \mathbf{r}_m\right]^{\top}\Delta\bar{\mathbf{h}}_e, \forall e \in \mathcal{E} \quad (8)$$

$$\mathcal{D} = \{(\mathbf{p}_e \rightarrow (x, y)) \mid \forall (x, y) \in \mathcal{S}, \forall e \in \mathcal{E}\} \quad (9)$$

3.3 Retrieval of In-Context Demonstrations

During inference, the test input $\widetilde{x} \in \widetilde{\mathcal{S}}$ undergoes the pipeline from Equation (1)-Equation (5) to obtain GER for every edit, which is then used as the query $\mathbf{q}_e$ to retrieve the K_e nearest neighbors from $\mathcal{D}$.

$$\mathcal{N}(\mathbf{q}_e) = \left\{(\mathbf{p}_e \rightarrow (x, y))^{(j)}\right\}_{j=1}^{K_e} \subseteq \mathcal{D} \quad (10)$$

Thanks to the fine-grained error encoding, we dynamically allocate the number of retrieved demonstrations K_s based on the complexity of each sentence's errors. Sentences deemed error-free by the

[2]The choice of dimensions for GER is discussed in Section 5.1.3.

model are not assigned examples, saving computational resources for sentences with more errors. We further reveal in Section 5.1 that the magnitude of the first dimension of GER $|\mathbf{p}_e^{(1)}|$ correlates with the simplicity of the error. Therefore, we prioritize retrieval for errors that have small $|\mathbf{p}_e^{(1)}|$, further optimizing resource allocation.

The retrieved examples are concatenated and combined with a few-shot correction template to prompt the final GEC prediction. The inference pipeline is illustrated in Figure 2. and the prompts used are listed in Appendix A.3.

4 Experiments

4.1 Datasets, Models, and Metrics

We evaluate the proposed method on five GEC datasets across four languages to testify to GER's ability to encode and retrieve errors. Following the multilingual setup in Li et al. (2025), we process the training dataset and use LlamaIndex (Liu, 2022) to construct the database and retriever.

For high-resource English (EN), we use the W&I+LOCNESS (Bryant et al., 2019) as the training dataset, and the CoNLL-14 (Ng et al., 2013) and BEA-19 (Bryant et al., 2019) datasets for testing. For medium-resource German (DE), we use the Falko-Merlin (Boyd, 2018) dataset for both training and testing. To showcase the generalizability of our method, we also include low-resource Romanian (RO) and Estonian (ET). For Romanian, we choose the RONACC (Cotet et al., 2020) training and test datasets; for Estonian, we use the Tartu L2 learner corpus (Rummo and Praakli, 2017) as the database and the L1 as the test data.[3]

Since GER requires the model's internal states, all experiments are conducted using recent open-source multilingual LLMs, including Meta's Llama-3.1-8B-Instruct (Dubey et al., 2024) and Qwen2.5-7B-Instruct (Yang et al., 2024) by Tongyi. We use the ERRANT toolkit (Bryant et al., 2017) to align edits between initial and final predictions and evaluate using precision, recall, and $F_{0.5}$ with M2Scorer (Dahlmeier and Ng, 2012).

Our method is compared with the following baselines:

- Random: Random selection of in-context demonstrations from the database;
- Semantic: kNN retrieval based on input text embeddings (Khandelwal et al., 2021);
- BM25: A term-based ranking function widely used in information retrieval (Robertson et al., 2009);
- Explanation: Retrieval based on the similarity of LLM-generated explanations for erroneous sentences (Li et al., 2025).

All experiments are conducted in an 8-shot setting. For all baseline methods, we retrieve 4 erroneous and 4 correct examples, following Li et al. (2025). Since our method dynamically determines the number of examples needed for each sentence, we retrieve 4 examples for each error and ensure that the average demonstration number is 8.

4.2 Main Results

During preliminary experiments, we found that the number of examples in the initial $\text{prompt}_{\text{init-n}}$ significantly impacts results. Thus, we present results in two configurations: "GER_0" refers to generating the initial predictions using 0-shot initial prompt $\text{prompt}_{\text{init-0}}$, and "$\text{GER}_8$" add 8 randomly chosen examples into the initial prompt $\text{prompt}_{\text{init-8}}$.

As Table 1 demonstrates, our GER-based retrieval methods consistently outperform other baseline methods in both prompt settings. In the GER_8 setting, our method exceeds the **explanation-based** SOTA by 4.36 and 4.56 points on the English CoNLL-14 and German Falko-Merlin datasets, respectively. Moreover, the BEA-19 dataset achieves a 9.15 higher $F_{0.5}$ than the *semantic* SOTA, nearly a 20% improvement. GER_0 still results in an improvement of around 3-5.6 points above SOTA, testifying to the effectiveness of our GER extraction and retrieval process.

On low-resource languages, GER retrieval yields even better results. For Romanian, the $F_{0.5}$ score improves by 4.33 points, while Estonian shows a 3.06 points improvement (nearly 25%). In GER_0, results are about 1 point lower but still surpass the SOTA. We hypothesize that low-resource languages benefit more from examples to help the model grasp syntax and generate corrections, as discussed in Section 5.3.

On the Qwen2.5 model, the results follow a similar trend to Llama3.1, confirming the generalizability of our approach across models. However, the advantage is slightly lower for low-resource lan-

[3]The detailed statistics of GEC datasets are placed in Appendix A.1.

Model	Method	English						German			Romanian			Estonian		
		CoNLL-14			BEA-19			Falko-Merlin			RONACC			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	44.20	63.43	47.05	59.62	54.53	58.53	45.38	43.87	45.07	10.34	21.10	11.52
	Semantic	55.21	51.56	54.44	45.51	62.84	48.17	60.03	54.15	58.75	49.16	46.73	48.65	11.04	22.61*	12.30
	BM25	54.58	51.58	53.95	44.18	62.95	46.98	59.65	**58.53**	58.80	50.50	48.70*	50.13	-	-	-
	Explanation	55.00	53.04	54.60	45.24	63.26	47.97	60.35	54.79	59.15	47.75	43.66	46.87	11.45	**24.52**	12.81*
	GER_0	58.60*	**55.33**	57.92*	47.86*	65.67*	53.75*	**66.39**	55.88	62.46*	54.00*	**49.51**	53.04*	14.25*	18.94	15.00*
	GER_8	**60.11**	54.75*	**58.96**	**55.63**	**67.28**	**57.63**	65.54	57.34*	**63.71**	**56.26**	48.28	**54.46**	**15.07**	20.12	**15.87**
Qwen2.5 (7B)	Random	54.43	53.50	54.24	44.84	63.62	47.65	55.25	48.06	53.65	32.85	28.25	31.81	6.21	17.67	7.14
	Semantic	55.27	52.65	54.73	45.48	63.40	48.21	57.81	48.57	55.76	37.97	34.59	37.24	6.41	**18.36**	7.37
	BM25	54.11	52.25	53.73	44.67	63.89*	47.53	57.21	50.18*	55.65	39.28	35.52*	38.46	-	-	-
	Explanation	55.67	51.60	54.81	47.22	62.31	49.62	57.33	47.63	55.08	33.64	30.23	32.90	6.54	18.21*	7.50
	GER_0	55.78	**56.94**	56.00*	49.12*	63.24	51.41*	61.09*	48.15	57.97*	**41.47**	30.60	38.72*	7.29*	11.75	7.89*
	GER_8	**57.53**	55.62	**57.13**	**52.37**	**67.37**	**54.81**	**60.31**	**51.90**	**58.42**	41.34*	**35.56**	**40.04**	**8.27**	**14.10**	**9.01**

Table 1: Results on multilingual GEC datasets by different retrieval methods. "Random" refers to retrieval baseline by random selection; "Semantic", "BM25", and "Explanation" retrieve demonstrations based on text embedding, BM25 matching, and LLM-generated explanations, respectively. "GER" refers to our representation-based retrieval methods, "GER_0" and "GER_8" refers to different initial prompts. The best results are marked in bold and the second-best results are marked with an asterisk (*).

Backbone	Method	Lang	EN	DE	ET
			$F_{0.5}$		
Fine-tuned GEC Single Model					
gT5 xxl	Rothe et al. (2021)	Mono	65.7	**76.0**	-
NLLB	Luhtaru et al. (2024)	Multi	65.2	73.9	**63.2**
BART	Zhou et al. (2023)	Mono	**69.6**	-	-
Inference of LLMs					
GPT-3.5-Turbo	Davis et al. (2024)	-	57.2	-	-
GPT-3.5-Turbo	Tang et al. (2024)	-	58.8	-	-
Deepseek2.5	Li et al. (2025)	-	**59.4**	63.4	**22.7**
GPT-4o-mini	Li et al. (2025)	-	58.7	**65.6**	19.9*
Llama3.1 (8B)	Ours	-	59.0*	63.7*	15.9

Table 2: The comparison of state-of-the-art (SOTA) models on multilingual GEC datasets. "EN", "DE", and "ET" stand for the CoNLL-14, Falko-Merlin, and Tartu-L1 datasets, respectively. Fine-tuned language models are labeled with their training data in the "Lang" column, where the "Mono" models are tuned separately for each language, and the "Multi" with multilingual mixed data. The best results are marked in bold and the second-best results are marked with an asterisk (*).

guages, likely due to Qwen2.5's smaller pre-trained corpus for these languages.

4.3 Comparison with SOTA

Current datasets reveal a persistent performance disparity in GEC tasks: while fine-tuned specialist models achieve state-of-the-art (SOTA) results across multilingual benchmarks (see Table 2), in-context learning (ICL) with LLMs exhibits significant accuracy gaps. Our representational retrieval method manages to achieve results comparable to some closed-source models on high-resource English and German, including Deepseek2.5 (Liu et al., 2024a) and GPT-4o-mini (Achiam et al., 2023). These promising results demonstrate the potential of utilizing interpretable components within the model to better align with human concepts and annotations of grammatical errors.

5 Analysis

5.1 Encoding Capacity of GER

The different principle components calculated by PCA, referred to as error vectors (EVs), capture various levels of error-related information in natural sentences. Our preliminary exploration of the first few EVs shows that the first EV represents the model's recognition and difficulty ranking of grammatical errors, while the second EV captures simple information about error types such as tense issues. In the following analysis section, unless stated otherwise, we use the GER_8 setup with Llama3.1-8B.

5.1.1 The First EV: Error Detector

We illustrate the first component of GER (first GER) obtained from the English training dataset in Figure 3. The figure presents a clear boundary between erroneous and correct tokens along the direction of the first EV, proving that the first GER can serve as an effective error detector.

Moreover, the magnitude of the first GER reflects the simplicity of errors in a relatively quantitative way. We classify the predicted tokens based on the confusion matrix, and plot the distribution of True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN) in Figure 3. Those cases with larger first GER are more likely to be precise corrections, while those with smaller values tend to be over-corrected. Hence, we construct the dynamic demonstration selection method, sparing the demonstration quota on errors with large values of the first GER, to save more computational resources for errors that are difficult and need to be corrected by referring to the exam-

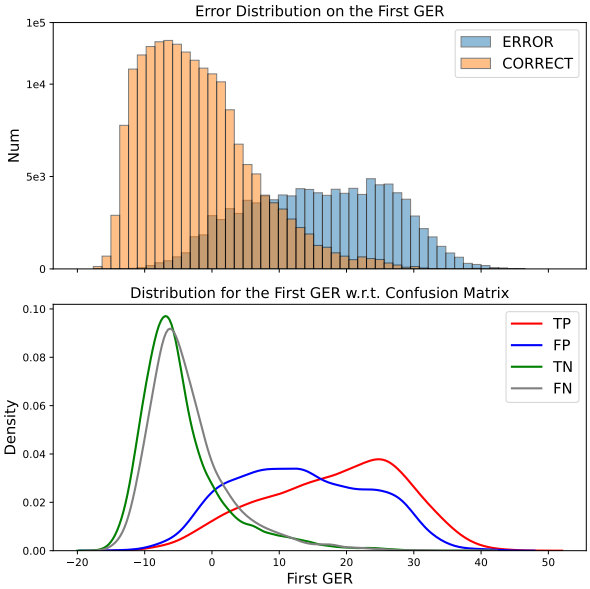

Figure 3: Distribution of the first GER component with respect to error/correct (up) and confusion matrix (down).

Method	EN			DE			ET		
	P	R	$F_{0.5}$	P	R	$F_{0.5}$	P	R	$F_{0.5}$
Dynamic	60.1	**54.8**	**59.0**	**65.5**	**57.3**	**63.7**	**15.1**	**20.1**	**15.9**
Random	59.8	52.6	58.2	64.1	55.5	62.2	13.9	20.0	14.8
Reverse	**60.7**	50.3	58.3	65.2	54.6	62.8	14.4	17.8	15.0

Table 3: Ablation of different demonstration selection methods of GER.

ples. We ablate the selection method by randomly selecting retrieved examples (Random) and prioritizing retrieval for errors that have large first GER (Reverse) in Table 3, validating the usefulness of dynamic selection.

5.1.2 The Second EV: Simple Error Classifier

On the first EV, we can distinguish between the wrong and the correct, but one dimension fails to provide detailed information. Introducing the second EV enables recognition of basic grammatical patterns. To validate this progression, we create a specialized test set[4] containing:

- Sport-domain sentences with present perfect progressive (ppp) tense errors;
- Art-domain sentences with simple past (sp) tense errors.

Cross-domain probes are designed as:

- Art-domain samples with ppp errors;
- Sport-domain samples with sp errors.

[4]Specific samples of the test set are placed in Appendix B.

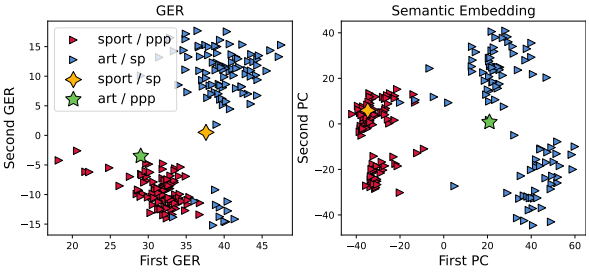

Figure 4: Distribution of different encoding methods on a manually created test set. "sport"/"art" refers to sentences in the sport/art domain, and "ppp"/"sp" refers to present perfect progressive / simple past tense errors. Cross-domain probes are marked as stars.

Dim.	EN			DE			ET		
	P	R	$F_{0.5}$	P	R	$F_{0.5}$	P	R	$F_{0.5}$
128	59.5	54.5	58.4	65.2	57.3	63.4	14.4	19.4	15.2
256	59.7	53.6	58.4	65.2	57.2	63.4	**15.1**	20.1	**15.9**
512	59.8	54.3	58.6	**65.5**	57.3	**63.7**	14.7	20.1	15.5
1024	**60.1**	**54.8**	**59.0**	65.4	**57.4**	63.6	14.9	20.4	15.8
2048	60.0	54.4	58.7	65.1	56.9	63.3	14.3	**20.7**	15.2

Table 4: Results across different dimensional configurations of GER.

Figure 4 shows that while semantic embeddings retrieve semantic-similar but error-mismatched examples, our 2-dimensional GER successfully clusters analogous errors across domains, demonstrating the proximity and semantic neutrality of GER.

5.1.3 Dimensionality Trade-offs in GER

Increasing the dimensionality of GER (m in $\mathbf{p}_e^{(m)}$) enhances its ability to encode fine-grained error patterns, but simultaneously amplifies the semantic noise it contains, causing GER to extract examples with semantic similarities over those sharing similar error types. Experimental results across different dimensional configurations are presented in Table 4: the more resources the model has about a particular language, the more dimensions it needs to encode errors in that language. At reduced dimensions, GER fails to distinguish complex errors; on the other hand, when the dimensions are too large, GER can identify some nuanced error cases but introduce more error-irrelevant samples, resulting in higher recall and lower precision.

5.2 Layer Selection

We select the layer used to extract GER based on the performance of grammatical error detection. The error detection performance with respect to each layer of the model is juxtaposed with the explained variance ratio of the first principal component in PCA (first EVR) in Figure 5. From the

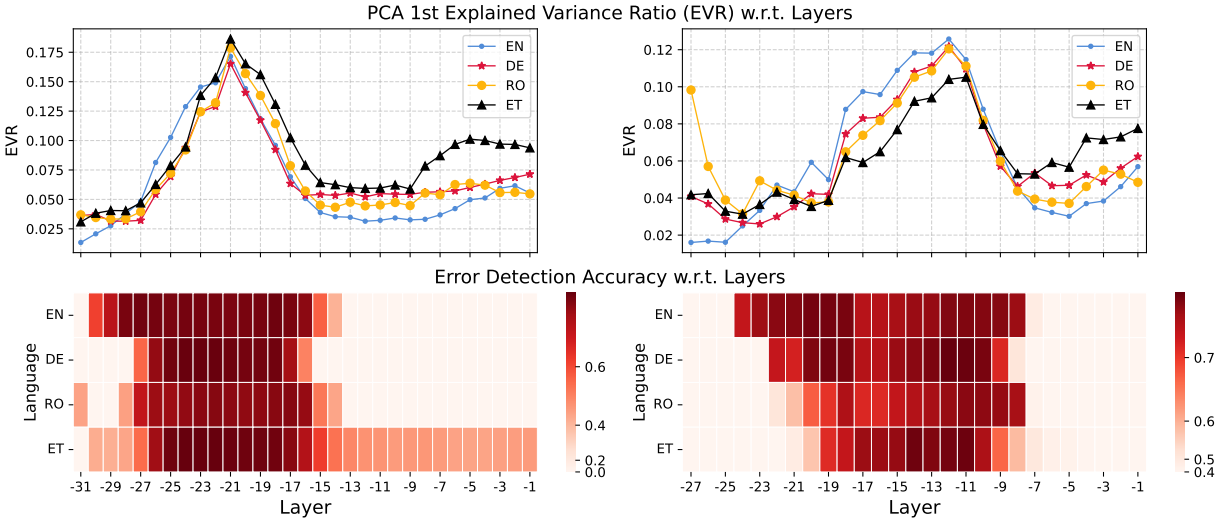

Figure 5: Upper: The explained variance ratio of the first principal component in PCA (first EVR) for layers. Lower: Accuracy of grammatical error detection task in each layer. We observe similar patterns for the trend of first EVR and error detection accuracy in Llama3.1 (left) and Qwen2.5 (right).

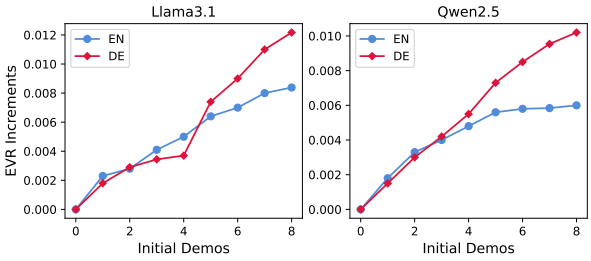

Figure 6: EVR increments of n-shot initial demonstrations relative to 0-shot.

upper figures, a spike of the first EVR is clearly depicted, coinciding with the most accurate layer in the lower images. The specific choice of layer differs with each model but maintains high consistency within the model across all languages, and all in the medium of the model (the 21st layer for 32-layer Llama3.1, and the 12th layer for 28-layer Qwen2.5). This suggests to us that there are specific components within the layer that are responsible for understanding and processing grammatical error information. We leave further research to future work.

5.3 Demonstration Selection for Initial Prompt

As observed in Section 4.2, even randomly selected examples in the initial prompt significantly improve results, although they affect the initial prediction and not the final output. We attribute this improvement to two factors: first, the few-shot initial prompt helps activate the model's correction capability and aligns the generate outputs with the example format. This alignment is particularly noticeable in low-resource languages such as Estonian, where zero-shot predictions usually include English tokens, introducing noise that hinders the PCA process for extracting EV. Second, from within the model, the initial prompt aligns EV inside the model toward the actual error space. Figure 6 reveals that the first explained variance ratio (EVR) increases as more initial examples are added, indicating that the model is refining its error space with each new demonstration. This suggests that the examples selected by GER may help the model better characterize the error space, which can be used iteratively in another round of generation to optimize EV. We leave this iterative approach for future work.

6 Conclusion

In this paper, we delve into the internals of LLMs and develop a novel method for extracting precise and interpretable grammatical error representations (GER) with less semantic noise. The effectiveness of GER in encoding fine-grained error patterns enables retrieval of high-quality error demonstrations, improving the few-shot performance of LLMs on GEC across diverse language settings.

Our preliminary exploration and successful utilization of LLMs' internal states highlight the potential of utilizing the model's inherent knowledge to strengthen GEC performance, alignment, and interpretability, all without the need for additional components or training resources.

Limitations

Our work explores and leverages the knowledge related to error correction within large models. However, the few-shot GEC capabilities of LLMs are far from fully realized. The later dimensions of our proposed error vectors contain detailed, fine-grained knowledge about error classification and correction, but they are difficult to separate, visualize, and utilize effectively. In addition, we did not address the scenario where long sentences with multiple errors outpace the utility of the 8-shot examples. In such cases, slicing the long sentence into smaller segments may yield better performance.

While we have encoded errors and used them for example retrieval in this work, the error information could be applied more broadly in the model's prediction pipeline, such as in controlling the decoding process. Future work could investigate simpler ways of representing error information, or develop methods to comprehensively combine and summarize this information for more effective manipulation of model-generated grammatical error corrections.

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*, abs/2303.08774.

Andy Arditi, Oscar Obeso, Aaquib Syed, Daniel Paleka, Nina Rimsky, Wes Gurnee, and Neel Nanda. 2024. Refusal in language models is mediated by a single direction. *CoRR*, abs/2406.11717.

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, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

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*, pages 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*, abs/2307.08393.

Teodor-Mihai Cotet, Stefan Ruseti, and Mihai Dascalu. 2020. Neural grammatical error correction for romanian. In *2020 IEEE 32nd International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 625–631. IEEE.

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, Øistein E. 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, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, pages 11952–11967. 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*, abs/2407.21783.

Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Nova DasSarma, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. 2021. A mathematical framework for transformer circuits. *Transformer Circuits Thread*. Https://transformer-circuits.pub/2021/framework/index.html.

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 *Natural Language Processing and Chinese Computing - 12th National CCF Conference, NLPCC 2023, Foshan, China, October 12-15, 2023, Proceedings, Part III*, volume 14304 of *Lecture Notes in Computer Science*, pages 69–80. Springer.

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.

Junxian He, Graham Neubig, and Taylor Berg-Kirkpatrick. 2021. Efficient nearest neighbor language models. In *Proc. of EMNLP*.

Svanhvít Lilja Ingólfsdóttir, Petur Orri Ragnarsson, Haukur Páll Jónsson, Haukur Barri Símonarson, Vilhjalmur Thorsteinsson, and Vésteinn Snæbjarnarson. 2023. Byte-level grammatical error correction using synthetic and curated corpora. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 7299–7316. Association for Computational Linguistics.

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, 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), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 7176–7187. 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. 2021. Nearest neighbor machine translation. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

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.

Jiquan Li, Junliang Guo, Yongxin Zhu, Xin Sheng, Deqiang Jiang, Bo Ren, and Linli Xu. 2022. Sequence-to-action: Grammatical error correction with action guided sequence generation. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*, pages 10974–10982. AAAI Press.

Wei Li, Wen Luo, Guangyue Peng, and Houfeng Wang. 2025. Explanation based in-context demonstrations retrieval for multilingual grammatical error correction. *arXiv preprint arXiv:2502.08507*.

Wei Li and Houfeng Wang. 2024. Detection-correction structure via general language model for grammatical error correction. *ArXiv preprint*, abs/2405.17804.

Xiaonan Li and Xipeng Qiu. 2023. Finding support examples for in-context learning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 6219–6235, Singapore. Association for Computational Linguistics.

Kaihui Liang, Sam Davidson, Xun Yuan, Shehan Panditharatne, Chun-Yen Chen, Ryan Shea, Derek Pham, Yinghua Tan, Erik Voss, and Luke K. 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@ACL 2023, Toronto, Canada, 13 July 2023*, pages 83–99. Association for Computational Linguistics.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. 2024a. Deepseek-v3 technical report. *ArXiv preprint*, abs/2412.19437.

Huanshuo Liu, Hao Zhang, Zhijiang Guo, Kuicai Dong, Xiangyang Li, Yi Quan Lee, Cong Zhang, and Yong Liu. 2024b. Ctrla: Adaptive retrieval-augmented generation via probe-guided control. *CoRR*, abs/2405.18727.

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.

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.

Ryo Nagata and Keisuke Sakaguchi. 2016. Phrase structure annotation and parsing for learner english. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics.

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, Sofia, Bulgaria. 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.

Guangyue Peng, Tao Ge, Si-Qing Chen, Furu Wei, and Houfeng Wang. 2023. Semiparametric language models are scalable continual learners. *ArXiv preprint*, abs/2303.01421.

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.

Ingrid Rummo and Kristiina Praakli. 2017. Tu eesti keele (voorkeelena) osakonna oppijakeele tekstikorpus [the language learners corpus of the department of estonian language of the university of tartu]. *Proc EAAL*.

Arkadiy Saakyan and Smaranda Muresan. 2024. ICLEF: In-context learning with expert feedback for explainable style transfer. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16141–16163, Bangkok, Thailand. Association for Computational Linguistics.

Shuqian Sheng, Yi Xu, Tianhang Zhang, Zanwei Shen, Luoyi Fu, Jiaxin Ding, Lei Zhou, Xiaoying Gan, Xinbing Wang, and Chenghu Zhou. 2024. Repeval: Effective text evaluation with LLM representation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, pages 7019–7033. Association for Computational Linguistics.

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.

Felix Stahlberg and Shankar Kumar. 2020. Seq2edits: Sequence transduction using span-level edit operations. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 5147–5159. Association for Computational Linguistics.

Felix Stahlberg and Shankar Kumar. 2024. Synthetic data generation for low-resource grammatical error correction with tagged corruption models. In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, pages 11–16, Mexico City, Mexico. Association for Computational Linguistics.

Xin Sun, Tao Ge, Furu Wei, and Houfeng Wang. 2021. Instantaneous grammatical error correction with shallow aggressive decoding. In *Proceedings of the 59th Annual Meeting of the Association for Computational*

Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021, pages 5937–5947. Association for Computational Linguistics.

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.

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.

Kevin Ro Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. 2023. Interpretability in the wild: a circuit for indirect object identification in GPT-2 small. In *The Eleventh International Conference on Learning Representations*.

Liang Wang, Nan Yang, and Furu Wei. 2024. Learning to retrieve in-context examples for large language models. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1752–1767, St. Julian's, Malta. Association for Computational Linguistics.

Muling Wu, Wenhao Liu, Xiaohua Wang, Tianlong Li, Changze Lv, Zixuan Ling, Jianhao Zhu, Cenyuan Zhang, Xiaoqing Zheng, and Xuanjing Huang. 2024. Advancing parameter efficiency in fine-tuning via representation editing. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 13445–13464. Association for Computational Linguistics.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. 2024. Qwen2. 5 technical report. *ArXiv preprint*, abs/2412.15115.

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.

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, Bo Zhang, Zhenghua Li, Zuyi Bao, Chen Li, and Min Zhang. 2022. 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.

Houquan Zhou, Yumeng Liu, Zhenghua Li, Min Zhang, Bo Zhang, Chen Li, Ji Zhang, and Fei Huang. 2023. 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.

Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, Shashwat Goel, Nathaniel Li, Michael J. Byun, Zifan Wang, Alex Mallen, Steven Basart, Sanmi Koyejo, Dawn Song, Matt Fredrikson, J. Zico Kolter, and Dan Hendrycks. 2023. Representation engineering: A top-down approach to AI transparency. *CoRR*, abs/2310.01405.

A Experimental Settings

A.1 Dataset Statistics

Our dataset usage is shown in Table 5. The training data samples used to construct the database are initially filtered by length with a minimum of 10 to ensure quality.

A.2 Model Settings

We utilize open-source LLMs such as Llama-3.1-8B-Instruct and Qwen2.5-7B-Instruct to implement representation extraction and demonstration retrieval.

To ensure reproducibility, we applied deterministic decoding (with temperature set to 0 and top_p set to 1.0) during inference. For the "Random" baseline, samples were selected using three different random seeds, and the results were averaged.

A.3 Prompt Settings

Throughout the entire experiment pipeline, we use the same prompt for GEC task as prior works (Tang

	Training Dataset (As Database)			Test Dataset	
Language	**Name**	**#Erroneous**	**#Correct**	**Name**	**#Total**
English	W&I+LOCNESS	20185	6839	CoNLL-14	1312
				BEA-19	4477
German	Falko-Merlin	11801	1916	Falko-Merlin	2337
Romanian	RONACC	6974	108	RONACC	1519
Estonian	Tartu-L2-Corpus	7156	4	Tartu-L1-Corpus	1453

Table 5: The statistics of GEC dataset used in experiments. For the training datasets, #Erroneous represents the number of erroneous samples, and #Correct refers to the number of correct samples. For the test datasets, Total indicates the total number of samples.

You are a 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 6: 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.

et al., 2024; Davis et al., 2024; Li et al., 2025), to form a fair comparison. The correction prompt is shown in Table 6.

B Cross-domain demonstration set

In Section 5.1.2, we used the web version of Deepseek-v3 to build 100 sport-domain sentences with present perfect progressive (ppp) tense errors, and 100 art-domain sentences with simple past (sp) tense errors. We then created cross-domain probes such as art-domain samples with ppp errors and sport-domain samples with sp errors to show the proximity and semantic neutrality of our GER. The created cases are demonstrated in Table 7.

Domain	Error Type	Case
Sport	**ppp**	Input: I have jogged along the riverbank for 45 minutes. Label: I have been jogging along the riverbank for 45 minutes.
	sp	Input: Yesterday, she try to hold her breath underwater. Label: Yesterday, she tried to hold her breath underwater.
Art	**ppp**	Input: Marcel Duchamp submits a urinal to an art show in 1917. Label: Marcel Duchamp submitted a urinal to an art show in 1917.
	sp	Input: For the entire week, Georgia O'Keeffe has painted her first giant flower close-up. Label: For the entire week, Georgia O'Keeffe has been painting her first giant flower close-up.

Table 7: The showing cases of manually constructed test set used in Section 5.1.2.