CoCoGEC: Counterfactual Generation for Robust Grammatical Error Correction

Qianyu Wang, Xiaoman Wang, Yuanyuan Liang, Xinyuan Li, Yunshi Lan*
East China Normal University
{wangqianyu, xmwang, leonyuany, xyli}@stu.ecnu.edu.cn, yslan@dase.ecnu.edu.cn

Abstract

Grammatical error correction (GEC) systems are usually trained and evaluated on GEC benchmarks, but their performance often drops sharply once the surrounding context is slightly perturbed or extended. This indicates that the existing GEC models usually fail to understand the error patterns in the varying contexts. In this paper, we thoroughly investigate the counterfactuals for GEC tasks, where the subtle changes to the contexts could lead to the label flipping issue. We propose CoCoGEC, a counterfactual generation framework that creates copies of training instances with error-irrelevant contexts altered. Our framework systematically generates counterfactuals by (1) generating intra- and inter-sentence counterfactuals that maintain the error patterns as well as syntax of the original instances by altering the word-level and sentence-level contexts; (2) revising the generated counterfactuals by selecting the instances with flipped labels and high GEC Mutual Information (MI) coefficient. Extensive experiments show that our method substantially improves the stability of GEC models, outperforming a set of data augmentation baselines. Particularly, it could achieve absolute $F_{0.5}$ gains of $+9.9$, $+11.3$, and $+20.8$ points on the perturbed BEA-19*,CoNLL-14*, and TEM-8* data set.Our code is released at https://github.com/Quinnok/CoCoGEC.

1 Introduction

Grammatical Error Correction (GEC) aims to automatically detect and correct grammatical errors in text, supporting applications such as intelligent writing assistants and computer-assisted language learning. It has attracted increasing attention from both academia and industry in recent years (Katinskaia and Yangarber, 2023, 2024; Li et al., 2025; Kovalchuk et al., 2025). However, we observe a substantial gap between the well-trained GEC models and their applications to the real world.

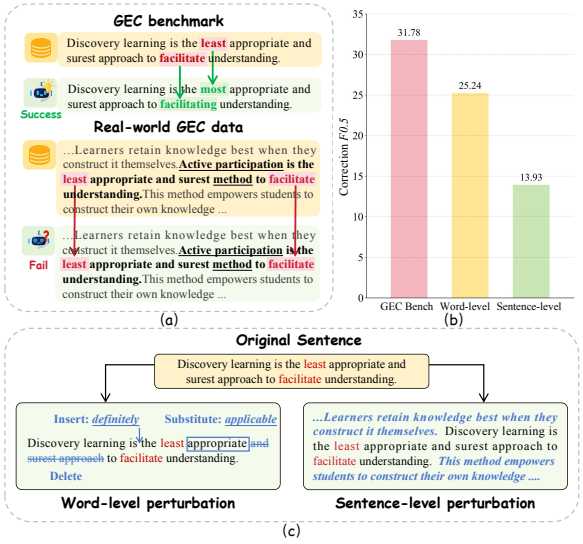

Figure 1: Motivation for CoCoGEC. (a) Context shift between standard benchmarks and real-world inputs. (b) Robustness drop on TEM-8 under perturbations with GPT-4. (c) The illustrative examples of two types of counterfactuals.

Figure 1 (a) illustrates an example from the BEA-19 task where "*least*" should be corrected to "*most*" and "*facilitate*" to "*facilitating*" in the context of "*Discovery learning*". While GPT-4 correctly revises this sentence on standard GEC benchmarks, GEC models often fail when encountering similar errors in diverse real-world contexts. For example, in a longer passage about "*Active participation*", the same erroneous phrase reappears. However, GPT-4 leaves it unchanged and thus exhibits under-correction errors.

To quantify this robustness gap, we conduct a preliminary experiment. We generate the "real-world" data via two augmentation methods: (1) word-level perturbation, involving random token alterations in test sentences from TEM-8 (Yang, 2017); and (2) sentence-level perturbation, constructed by randomly combining sentences from

*Corresponding author

Findings of the Association for Computational Linguistics: ACL 2026, pages 4004–4019
July 2-7, 2026

the test set[1]. As shown in Figure 1 (b), there is an observable performance drop between the original GEC data and "real-world" data, especially for sentence-level perturbation.

Prior studies (Zhang et al., 2023; Wang et al., 2024a) have revealed that current GEC models are vulnerable to seemingly harmless perturbations. But these studies mainly focus on noise-based attacks or broad augmentation, rather than interpreting the potential perturbations that fundamentally affect GEC models. For example, Wan et al. (2020) and Park et al. (2023) use noise injection, Lichtarge et al. (2019) and Stahlberg and Kumar (2021) generate pseudo corpora, and Wang et al. (2024a) and Li and Lan (2025) propose contextual augmentation. These approaches expand or re-distribute training data, but the perturbations are often random or coarsely controlled, limiting their ability to explain the performance gap.

In this paper, we address robustness to word- and sentence-level perturbations by asking: *how can we make a GEC model focus more on error patterns while ignoring varying context?* To this end, we propose a novel **CoCoGEC** method, inspired by counterfactual analysis. The intuition behind CoCoGEC is to create copies of training instances with their error-irrelevant contexts altered. We identify two types of decoupled counterfactuals for GEC data, which aim to alter the word-level and sentence-level contexts without influencing the original error patterns in a sentence, but could confuse the prediction of a GEC model. We display the "counterfactual" GEC in Figure 1 (c). With the counterfactuals, the GEC model would learn to put more emphasis on the error patterns when learning how to correct a sentence.

CoCoGEC uses large language models (LLMs) to generate span-controlled intra-sentence variants that substitute error-irrelevant spans while keeping the gold correction edits valid. It also constructs inter-sentence variants by attaching coherent, error-free prefixes and suffixes to emulate discourse-level context shifts. We enforce an edit-level fidelity constraint to filter invalid candidates, then rank the remaining counterfactuals with a GEC mutual-information score and keep the most challenging ones for augmentation. The experimental results consistently verify that CoCoGEC improves the robustness of GEC models.

[1]Implementation details of the preliminary experiment can be found in Appendix A.

The main contributions of this work are:

- To the best of our knowledge, this is the first study to explore counterfactuals for GEC tasks by characterizing potential perturbations with three criteria.
- We introduce CoCoGEC, a counterfactual generation pipeline tailored to contexts in GEC, which systematically constructs intra-sentence and inter-sentence variants without influencing the original error pattern, but could confuse the prediction of a GEC model.
- We propose a novel GEC mutual information coefficient that captures the dependence between the varying context and the model predictions for identifying high-quality counterfactuals.
- We demonstrate on the RobustGEC benchmark that COCOGEC consistently improves robustness under both intra- and inter-level perturbations, without sacrificing performance on standard test settings.

2 Related Work

Robust GEC. Modern GEC systems mainly fall into sequence-to-sequence generation (Vaswani et al., 2017; Junczys-Dowmunt et al., 2018), sequence-to-edit correction (Awasthi et al., 2019; Stahlberg and Kumar, 2020; Omelianchuk et al., 2020; Qorib and Ng, 2023) (including hybrid detection–correction variants (Li et al., 2023; Li and Wang, 2024)), and recent LLM-based pipelines with prompting or light supervision (Loem et al., 2023; Coyne et al., 2023; Katinskaia and Yangarber, 2024; Tang et al., 2024). Despite strong benchmark performance, existing models can be brittle under small contextual shifts, motivating robustness-oriented training and data construction. Robustness is typically pursued along two complementary directions. Model-centric methods improve stability by explicitly regularizing invariance—via adversarial objectives (Dang et al., 2021), distillation-style constraints (Xia et al., 2022), or consistency-based post-training on constructed variants and hard cases (e.g., RobustGEC/TemplateGEC/CLEME2.0/CSA) (Zhang et al., 2023; Li et al., 2023; Ye et al., 2024; Tang et al., 2023). Data-centric methods instead broaden supervision by synthesizing training pairs through noise injection (Solyman et al., 2023; Sun et al.,

2023), back-translation (Fang et al., 2023), and contextual or edit-based augmentation (Wang et al., 2024a; Ye et al., 2023), sometimes coupled with robustness-oriented annotation or curricula (Li and Lan, 2025; Zhang et al., 2025). However, much of this augmentation primarily targets error diversity or reweighting, and indiscriminate synthetic data may even degrade GEC performance (Park et al., 2023). In contrast, our approach is data-centric: we generate context-decoupled counterfactuals with an edit-subset constraint ($E' \subseteq E$) to target context robustness, and they can be used with various GEC backbones.

Counterfactual Analysis Beyond GEC. Counterfactual data augmentation (CDA) improves robustness by generating controlled perturbations that preserve or systematically modify labels, encouraging models to rely on invariant features and generalize out of distribution (Wang et al., 2024b; Jiang et al., 2024). Recent progress largely comes from strengthening controllability and label fidelity in generation: diffusion-based frameworks provide a powerful mechanism for robust synthesis and transfer (Xin et al., 2024; Chen et al., 2024; Bae et al., 2025; Wang and Wan, 2022), while optimization-driven formulations enforce invariance through reinforcement learning, information bottlenecks, and contrastive objectives (Chen et al., 2021; Sreedhar et al., 2025; Chang et al., 2024; Choi et al., 2022). In parallel, counterfactuals have shifted from rule-based edits to more controllable generative pipelines, including distillation and LLM-driven synthesis (Chen et al., 2023b; Youssef et al., 2024; Howard et al., 2022; Treviso et al., 2023; Zhou et al., 2023), as well as explanation-oriented designs that improve interpretability and faithfulness (Yang et al., 2024; An et al., 2025).

Collectively, these studies offer broad tools for building context-invariant NLP models, but they focus on classification tasks, leaving counterfactual generation for structured prediction and GEC comparatively underexplored. Our work brings CDA to GEC by designing contextual counterfactuals tailored to error correction, rather than label-flipping counterfactuals commonly used in discriminative settings.

3 Method

3.1 Definition of Counterfactuals for GEC

Existing studies (Wang et al., 2024c; Verma et al., 2024) have made a formal definition of counterfactuals in general machine learning tasks. A counterfactual example c usually disturbs a model to predict an instance x as an alternative class y' instead of its original class y by making *minimal yet necessary* changes to x as follows:

$$\begin{aligned} & arg\min_{c} \quad \text{dist}(x, c) \\ & \text{s.t.} \quad f(c) \neq f(x) \end{aligned}$$

where f is a task-specific model $f : \mathtt{X} \in \mathbb{R}^d \rightarrow \mathtt{Y}$ to bridge the mapping from x to y and $dist(\cdot, \cdot)$ is a distance function that measures the cost of changes required to alter the prediction.

The above definition outlines the fundamental principles of the counterfactual generation problem. Considering the distinct problem formulation of the GEC task, we identify the counterfactuals for GEC as conducting a subtle change to the source text, but resulting in a subset of the original edits. Motivated by the example in Figure 1, we mainly focus on the counterfactuals for intra-sentence and inter-sentence. We consider a counterfactual to be shown in the form $c = p \oplus s' \oplus q$, where s' minimally modifies the s, and p and q are a prefix and a suffix to the source text, respectively. As a result, we have:

$$\begin{aligned} & \underset{c}{\text{argmin}} \quad \text{dist}(s, c) \\ & = \text{syntax\_dist}(s, s') + \text{semantic\_dist}(s', p \oplus q) \\ & \text{s.t.} \quad \mathcal{E}' \subseteq \mathcal{E} \end{aligned} \tag{1}$$

Formally, (s, t) denotes the annotated source and target text for grammatical error correction. Ideally, $f(\cdot)$ is a GEC model which takes the source text as the input and perfectly produces the corrected text, such that $f(s) = t$ and $f(c) = t'$. The edit mapping of the original text $s \rightarrow t$ and counterfactual text $c \rightarrow t'$ are denoted as $\mathcal{E}$ and $\mathcal{E}'$, respectively.

Regarding Equation (1), we interpret the countertfactuals for GEC as follows:

- **Minimal syntactic revision to source text**. For the revision in intra-sentence, we make minor verbal adjustments, which may change the semantics of the sentence rather than its syntax. We denote it as minimal $syntax\_dist(s, s')$.
- **Semantic coherence to the revision**. For the revision in inter-sentence, we append a prefix and a suffix to the revised source text, but keep semantic coherence to the revised source text, which prevents the illogical flow of the counterfactuals. We define minimal $semantic\_dist(s', p \oplus q)$.

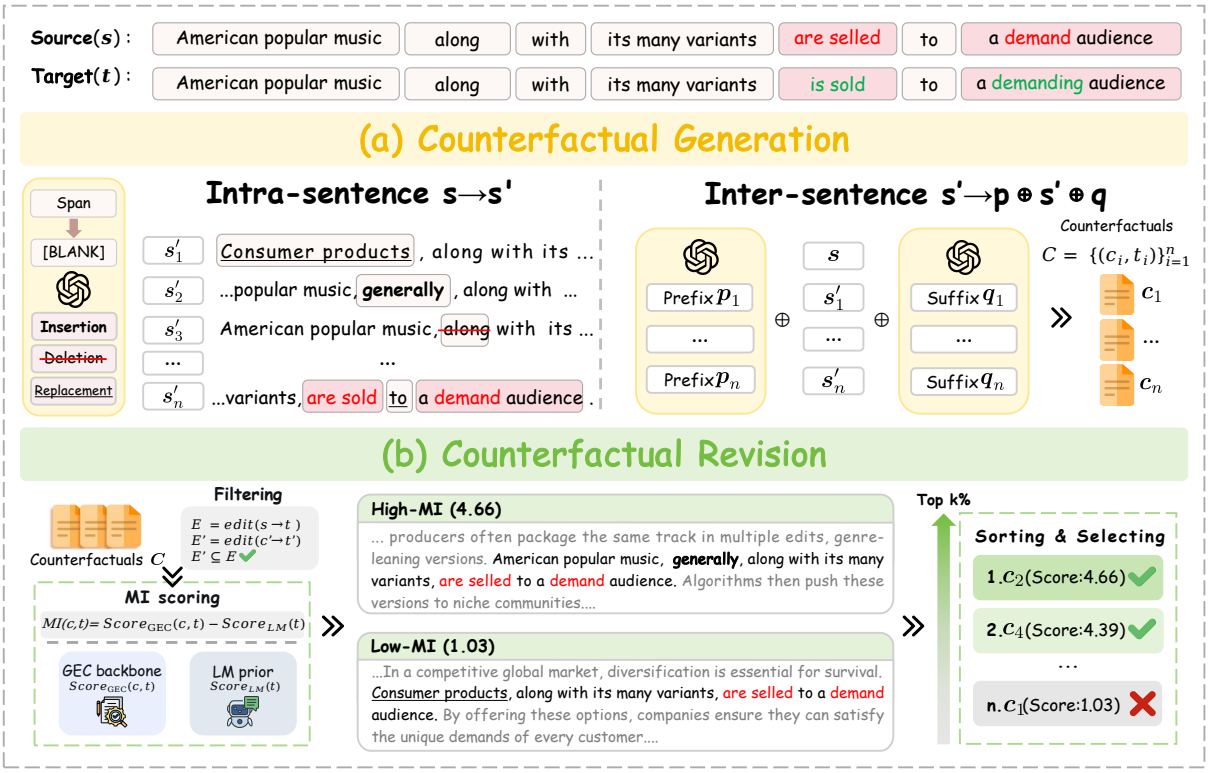

Figure 2: Overview of CoCoGEC: we generate span-controlled intra-sentence variants and attach coherent prefix/suffix to form long-context counterfactuals, then filter by edit-set consistency and rank by GEC mutual information to keep the most confusing yet valid augmentations.

- **Flipped edit labels**. We consider that a good counterfactual for GEC could flip the prediction of a GEC model, where the original edits cannot be recognized well. To avoid intertwined grammatical errors, new edits cannot be introduced. Hence, we deem $\mathcal{E}'$ as a subset of $\mathcal{E}$ as the outcomes of the counterfactuals.

3.2 Counterfactual Generation with LLMs

Next, we generate counterfactuals following the above definition. We first target at constructing the syntactically similar s' to s, which could flip the original error annotations. However, it is not trivial to control the perturbation of the source sentence. Unlike the label-flipping objective in discriminative settings, our goal in GEC is to *preserve* the original grammatical errors and *not include* new errors. Thus, we would like the error-irrelevant context to be perturbed while keeping the syntax unchanged.

We propose a pipeline of counterfactuals generation for GEC with the integration of intra-sentence and inter-sentence perturbation, which corresponds to s' and $p \oplus q$ defined in Equation (1), accordingly. Specifically, we first distill counterfactuals for s' generation with LLMs in a controllable manner. Then, we generate p and q for the purpose of the inter-sentence perturbation. At last, we extract the counterfactuals satisfying our objective of flipping the original edits while not introducing new edits. This pipeline results in an integrated counterfactual in the format of $p \oplus s' \oplus q$.

Intra-sentence counterfactual generation of s'. To generate s', we target at minimal syntactic revision to source text s. Prior study (Chen et al., 2023b) introduces $\mathcal{DISCO}$ method to prompt LLMs via in-context learning to generate counterfactuals for natural language inference, which controls the spans for variant generation. We follow the principle of $\mathcal{DISCO}$ and distill intra-sentence counterfactuals with LLMs as follows:

- We first segment each source sentence s into a sequence of candidate spans $\mathcal{W} = \{w|w \in s\}$ using a Flair-based chunker (Akbik et al., 2019). To prevent unexpected edits to the erroneous regions, we discard all spans overlapping with $\mathcal{E}$, keeping only spans outside the gold edits.

- For each retained span $w \in \mathcal{W}$, we sample a masked variant by replacing w with a special token `[BLANK]`, and feed into an LLM with an

instruction asking the model to fill the blank with an alternative phrase, which achieves insertion, deletion, or replacement for w. We generate multiple variants for each s.

As shown in Figure 2, the illustrative example "*American popular music, along with its many variants, are selled to a demand audience.* " can have multiple intra-sentence counterfactuals with the unchanged syntax. To preserve alignment between the perturbed source and its correction, we apply LLM filling only to non-error spans. Whenever a span w in the source sentence s is replaced by an LLM-generated fragment $\tilde{w}$, we simultaneously replace the aligned span in the gold target t with the same $\tilde{w}$. As a result, we collect a set of counterfactual sentence pairs, denoted as (s', t').

Inter-sentence counterfactual generation of $p \oplus q$. To probe long-range context and expose GEC models to paragraph-level correction, we generate prefix and suffix for s'. To ensure the semantic coherence of the revision, for each pair (s', t'), we automatically attach error-free context on both sides by sampling a short grammatical prefix p and suffix q from an LLM. As the example shown in Figure 2, the intra-sentence counterfactual is mentioned in the middle of long contexts. Eventually, we obtain $c = p \oplus s' \oplus q$ as defined in Equation (1). We attach the same prefix and suffix to t', and the sentence pair (c, t') forms a pre-defined counterfactual. It is worth noting that since p and q are error-free contexts, the edit mapping $c \rightarrow t'$ will not introduce new errors beyond the span s. We denote the set of generated counterfactuals as $\mathcal{C} = \{(c_i, t'_i)\}_{i=1}^{n}$

3.3 Counterfactual Revision with GEC Mutual Information

Edit-subset counterfactual filtering. To ensure that label flipping occurs in c within the span s, we conduct filtering. Specifically, we employ ERRANT (Bryant et al., 2017b) to conduct the edit mapping of $s \rightarrow t$ and $c \rightarrow t'$ to produce $\mathcal{E}$ and $\mathcal{E}'$, respectively. If a sentence pair satisfies $\mathcal{E}' \subseteq \mathcal{E}$, we keep it; otherwise, we abandon it.This results in a set of counterfactuals that follows our pre-defined principle.

Mutual-information scoring and selection. The above steps ensure a counterfactual to be *valid* but not *optimal*. An optimal counterfactual in GEC tasks should be a hard negative that is highly related to the correct form but particularly challenging for a GEC model to distinguish. In other words, the flipped edits should be difficult to detect via a GEC model. In Figure 2(b), we show two counterfactual candidates derived from the source sentence about "*American popular music*". The retained candidate makes only a minor contextual insertion (e.g., "*generally*") while keeping the original subject, forming a challenging near-miss that remains compatible with the gold target. By contrast, another candidate changes the subject to "*Consumer products*", which becomes semantically misaligned with the original correction target and is therefore ranked low and filtered out. To turn the large pool of generated counterfactuals into high-quality examples. We propose a novel Mutual-Information–based scoring function to measure the GEC mutual information of a counterfactual. Previous studies on counterfactual data augmentation (Chen et al., 2018; Plyler and Chi, 2025) adapted Mutual Information (MI) to counterfactual generation, holding the following assumption:

- Seeking a counterfactual is referred to as the maximum mutual information criterion:

$$I(c; y) = \mathbb{E}_{\mathtt{C},\mathtt{Y}}[\log \frac{P_{\mathtt{C},\mathtt{Y}}(c, y)}{P_{\mathtt{C}}(c) P_{\mathtt{Y}}(y)}]$$

  where c denotes the counterfactual and y denotes the original prediction. $I(c; y)$ measures the dependence between the counterfactual and the original prediction.

When it comes to the GEC task, we tend to measure the dependence between the erroneous source text c and the original target text t as follows:

$$\begin{aligned} \text{argmax}_{c \in \mathcal{C}} I(c; t) &= \mathbb{E}_{\mathtt{C},\mathtt{T}}[\log \frac{P_{\mathtt{C},\mathtt{T}}(c, t)}{P_{\mathtt{C}}(c) P_{\mathtt{T}}(t)}] \\ &= \mathbb{E}_{\mathtt{C},\mathtt{T}}[\log P_{\mathtt{T}|\mathtt{C}}(t|c) - \log P_{\mathtt{T}}(t)] \end{aligned}$$

As we can see from the formula, a good counterfactual c in the GEC task should have a high probability $P_{\mathtt{T}|\mathtt{C}}(t|c)$ of transferring a counterfactual c to the original target text t, where the perturbed errors in c do not alter the original prediction effectively. This is also influenced by the fluency of the prediction t, a smaller $\log P_{\mathtt{T}}(t)$ indicates a less fluent t, which makes the GEC model more confused.

In GEC tasks, we approximate these two terms using neural network-based GEC models. We employ a GEC model with a Seq2seq framework and

compute the joint probability of the sequential tokens in the target text:

$$\log P_{\mathrm{T|C}}(t|c) = \frac{1}{|t|}\sum_{i=1}^{|t|}\log \mathrm{GEC}_{\mathrm{Seq2seq}}\big(w_i \mid c, w_{<i}\big),$$

where w_i denotes the generation of i-th token in t.

For the GEC model with the Seq2Edit framework, we approximate with the joint probability of the sequential operations leading to the target text:

$$\log P_{\mathrm{T|C}}(t|c) = \frac{1}{|c|}\sum_{i=1}^{|c|}\log \mathrm{GEC}_{\mathrm{Seq2edit}}\big(e_i \mid c, e_{<i}\big)$$

where e_i denotes the operation to i-th token in c.

Regarding $\log P_{\mathrm{T}}(t)$, we employ GPT-2-medium (Radford et al., 2019) to compute the joint probability of the tokens in the target text.

$$\log P_{\mathrm{T}}(t) = \frac{1}{|t|}\sum_{i=1}^{|t|}\log \mathrm{LM}\big(w_i \mid w_{<i}\big)$$

where w_i denotes the generation of i-th token in t.

We compute an MI score for each counterfactual in $\mathcal{C}$, sort them in descending order, and select the top k percent to form the final set. MI scoring reflects our desiderata: a high-quality counterfactual is a close near-miss to the correct form rather than a random, malformed utterance.

4 Experimental Setup

4.1 Dataset

We conduct all experiments on RobustGEC (Zhang et al., 2023), which augments BEA-19, CoNLL-14, and TEM-8 with robustness-oriented perturbations to error-irrelevant context. Each original sentence pair has several human-generated variants, and we split the data by case into training, development, and test sets in a 7:1:2 ratio, keeping all variants of a case in the same split. For long-context evaluation, we use GPT-4[2] to add a shared prefix p and suffix q to each source–target pair, yielding long-context test sets denoted BEA-19*, CoNLL-14*, and TEM-8*. Unless otherwise noted, models are trained and tuned only on the original sentence pairs.

4.2 Evaluation Metrics

We report standard edit-based metrics (precision, recall, and $F_{0.5}$) computed with ERRANT (Bryant et al., 2017a). For RobustGEC, we additionally report CRS and P-CRS (Zhang et al., 2023) to quantify correction consistency across context-perturbed variants (case-level and pairwise, respectively). For attacked sets, following CSA (Tang et al., 2023), we further report SR and TR, measuring recovery at the sentence and token levels; higher values indicate better robustness.

4.3 Comparative Methods

We compare CoCoGEC with representative robustness and augmentation baselines. As backbones, we use GECToR-large (Omelianchuk et al., 2020) for Seq2Edit, T5-large (Raffel et al., 2020) for Seq2Seq, and Qwen3-8B for LLM-based GEC. CPR (Zhang et al., 2023) improves contextual consistency via KL-divergence regularization. $\mathcal{DISCO}$ (Chen et al., 2023a) distills counterfactual data with LLMs. TypeDA (Li and Lan, 2025) augments training data with type-aware LLM annotation through masked modeling and error filling. We also include zero-shot GEC baselines with GPT-4o (Achiam et al., 2023) and LLaMA3-8B as representative general-purpose LLMs.

4.4 Implementation Details

We generate counterfactuals by chunking candidate spans with Flair[3] and enforcing edit constraints using gold edit sets extracted by ERRANT[4], followed by basic hygiene filtering (length control, empty/degenerate removal, and de-duplication). We estimate the likelihood term $\log P(t \mid c)$ with an ensemble of GEC scorers, and implement fine-tuning with LLaMA-Factory and LoRA (Zheng et al., 2024; Hu et al., 2022). Templates and remaining details are in Appendix B.3.

5 Results and Analyses

5.1 Main Results on RobustGEC

Table 1 reports the performance of CoCoGEC and data-augmentation baselines on the three RobustGEC benchmarks. We summarize three main observations:

- **CoCoGEC improves robustness across different backbones.** Across Seq2Edit (GECToR-large), Seq2Seq (T5-large), and LLM-based (Qwen3-8B) backbones, CoCoGEC consistently improves $F_{0.5}$ on all three benchmarks. For Qwen3-8B in particular, CoCoGEC yields

[2] https://openai.com/research/gpt-4

[3] https://github.com/flairNLP/flair

[4] https://github.com/chrisjbryant/errant

Method	Data Size	BEA-19*			CoNLL-14*			TEM-8*		
		Prec.	Rec.	$F_{0.5}$	Prec.	Rec.	$F_{0.5}$	Prec.	Rec.	$F_{0.5}$
GECToR-large	-	26.86	24.54	26.36	22.28	21.31	22.08	39.51	34.27	38.35
+ CPR method	36K	28.32	27.55	28.16	18.09	20.51	18.54	49.81	22.88	40.32
+ $\mathcal{DISCO}$	42K	31.49	21.05	28.65	26.88	17.79	24.38	47.27	26.46	40.83
+ TypeDA	30K	26.87	25.64	26.63	23.87	22.30	23.54	39.51	36.89	38.96
+ CoCoGEC	25K	**51.55**	11.27	**30.07**	**34.52**	12.22	**25.29**	**52.90**	31.76	**46.77**
T5-large	-	27.27	38.22	28.92	23.64	25.80	24.04	23.18	38.39	25.18
+ CPR method	36K	27.29	38.70	28.99	25.91	21.63	24.93	24.47	38.76	26.41
+ $\mathcal{DISCO}$	42K	27.70	39.28	29.43	24.06	26.76	24.56	24.85	39.33	26.83
+ TypeDA	30K	27.96	41.42	29.90	24.16	27.72	24.80	25.64	41.19	27.74
+ CoCoGEC	25K	**32.93**	31.52	**32.64**	**27.61**	22.43	**26.40**	**33.16**	35.02	**33.51**
Qwen3-8B	-	26.68	37.34	28.29	22.22	23.72	22.50	31.11	46.44	33.32
+ CPR method	36K	27.48	40.64	29.38	25.77	18.43	23.88	33.25	44.94	35.07
+ $\mathcal{DISCO}$	42K	26.20	38.01	27.93	21.56	27.40	22.52	36.64	46.98	38.33
+ TypeDA	30K	25.94	48.40	28.59	21.05	31.41	22.54	27.02	51.69	29.88
+ CoCoGEC	25K	**36.63**	46.12	**38.22**	**36.78**	25.64	**33.83**	**55.70**	48.50	**54.09**
Qwen3-4B	-	22.14	22.11	22.13	13.08	9.29	12.10	24.07	24.34	24.13
Qwen3-14B	-	26.86	39.57	28.70	23.27	27.40	23.99	33.48	54.49	36.28
Qwen3-235B	-	37.01	43.82	38.20	29.88	34.62	30.72	46.88	64.79	49.63
GPT-4o	-	34.43	48.89	36.60	33.72	34.10	33.80	45.60	65.97	48.60
LLaMA3-8B	-	23.09	44.52	25.55	19.3	29.01	20.68	16.24	48.30	18.72

Table 1: Main results on RobustGEC test sets. We denote the perturbed data subsets in RobustGEC as BEA-19*, CoNLL-14*, and TEM-8*. CoCoGEC consistently improves $F_{0.5}$ over CPR, $\mathcal{DISCO}$ generation, and TypeDA across all backbone–dataset pairs, with the largest absolute gains on the long-context TEM-8* data.

Method	Source			Word-level Perturbation			Sentence-level Perturbation			Combined Perturbation		
	Prec.	Rec.	$F_{0.5}$	Prec.	Rec.	$F_{0.5}(\Delta\downarrow)$	Prec.	Rec.	$F_{0.5}(\Delta\downarrow)$	Prec.	Rec.	$F_{0.5}(\Delta\downarrow)$
GECToR	62.4	38.7	55.6	58.6	38.3	53.0 (2.6)	38.4	33.6	37.3 (18.3)	39.5	34.4	38.3 (17.3)
+ CPR	60.4	39.8	54.7	58.1	40.3	53.4 (1.3)	39.8	34.2	38.5 (16.2)	40.8	29.9	38.0 (16.7)
+ TypeDA	62.9	40.9	56.8	60.1	36.8	53.3 (3.4)	40.1	33.6	38.6 (18.2)	47.3	26.5	40.9 (15.9)
+ $\mathcal{DISCO}$	69.1	37.3	59.0	65.7	37.3	57.0 (2.0)	48.2	30.6	43.3 (15.9)	39.5	36.9	39.0 (20.1)
+ CoCoGEC	66.3	45.2	**60.6**	64.6	46.5	**59.9** (**0.7**)	63.7	32.5	**53.4** (**7.2**)	52.90	31.76	**46.7** (**14.0**)

Table 2: Performance breakdown on TEM-8 across disturbance settings. $\Delta\downarrow$ denotes the drop from the Source setting, computed as $F_{0.5}^{\text{Source}} - F_{0.5}^{\text{Perturbed}}$; lower Δ indicates better robustness.

absolute $F_{0.5}$ gains of $+9.9$, $+11.3$, and $+20.8$ points on BEA-19*, CoNLL-14*, and TEM-8*, respectively, indicating more context-invariant correction behavior, especially in long-context scenarios.

- **CoCoGEC outperforms existing augmentation methods.** Compared with noise injection (CPR), distillation-based augmentation ($\mathcal{DISCO}$), and type-aware augmentation (TypeDA), CoCoGEC attains the highest $F_{0.5}$ in every backbone–dataset pair in Table 1. On the context-perturbed TEM-8* benchmark in particular, it surpasses all baselines by a clear margin, suggesting that controllable context counterfactuals provide stronger training signals than random or loosely controlled perturbations, even when using fewer counterfactual instances.

- **CoCoGEC narrows the gap between compact models and large LLM baselines.** Although large LLMs (e.g., GPT-4o, LLaMA3-8B, and Qwen3-235B) are strong zero-shot baselines, they can still be sensitive to contextual perturbations. When fine-tuned with CoCoGEC, Qwen3-8B matches or slightly surpasses these zero-shot LLM baselines on RobustGEC perturbed subsets, suggesting that counterfactual, data-centric optimization can be a parameter-efficient alternative to simply scaling model size.

5.2 Breakdown by Perturbation Type

Since Section 5.1 shows consistent gains across GECToR, T5, and Qwen3, we use GECToR-large as a representative Seq2Edit backbone for a detailed robustness analysis on TEM-8. Table 2 further decomposes performance into the unperturbed *Source* setting, word-level, sentence-level,

Model	ATK1			ATK2			ATK3			ATK4			ATK5		
	$F_{0.5}$	TR↑	SR↑	$F_{0.5}$	TR↑	SR↑	$F_{0.5}$	TR↑	SR↑	$F_{0.5}$	TR↑	SR↑	$F_{0.5}$	TR↑	SR↑
Qwen3-8B	7.98	27.39	27.39	15.44	38.59	27.05	18.73	30.77	16.26	20.33	23.54	8.31	21.29	18.47	3.90
Qwen3-8B +CoCoGEC	**9.48**	**37.40**	**37.40**	**15.84**	**38.81**	**28.21**	**19.41**	**32.14**	**17.11**	**21.22**	**24.13**	**8.70**	**22.16**	**19.19**	**4.17**

Table 3: Robustness to perturbation number from 1 to 5. CoCoGEC maintains a consistent advantage over the baseline across most attack settings, indicating improved robustness under stronger perturbations.

Method	Prec.	Rec.	$F_{0.5}$	CRS↑	P-CRS↑
GECToR	39.51	34.27	38.35	6.41	64.30
Counterfactual Generation					
+ GPT-based c	40.31	34.22	39.00	6.42	65.36
+ s'	47.27	26.46	40.83	10.45	71.33
+ $p \oplus q$	45.79	32.42	42.33	10.02	66.50
Counterfactual Revision					
+ $k = 100\%$	44.05	33.08	41.31	11.83	67.93
+ $k = 30\%$	52.90	31.76	**46.77**	**16.98**	**72.52**

Table 4: Ablation of counterfactual components on TEM-8 with GECToR, showing that span-/document-level edits and MI-based selection ($k = 30\%$) give the best $F_{0.5}$ and robustness.

and combined perturbations. Vanilla GECToR suffers the largest drop in $F_{0.5}$ under sentence-level and combined perturbations, while word-level perturbation alone is less harmful, suggesting that long-range contextual shifts are the main source of brittleness. Robustness-oriented augmentations reduce this drop, and CoCoGEC achieves the smallest $F_{0.5}$ degradation in all settings while maintaining strong source performance, indicating that training with context-decoupled counterfactuals stabilizes GEC in long-context scenarios.

5.3 Ablation Studies

We ablate CoCoGEC on TEM-8 with GECToR to disentangle the effects of counterfactual generation and revision. As shown in Table 4, adding only global context rewriting (`+ GPT-based` c) yields small but consistent improvements, while intra-sentence variants s' and inter-sentence expansion $p \oplus q$ each bring larger gains in $F_{0.5}$ and robustness metrics. The revision stage further improves performance: selecting a subset of counterfactuals with MI-based ranking ($k = 30\%$) outperforms keeping all generated instances ($k = 100\%$) in both $F_{0.5}$ and CRS/P-CRS[5]. Overall, this shows that selective filtering is crucial, as keeping all instances can dilute the gains. These results indicate that robustness gains come from controllable span edits combined with selective filtering, rather than simply enlarging the training set.

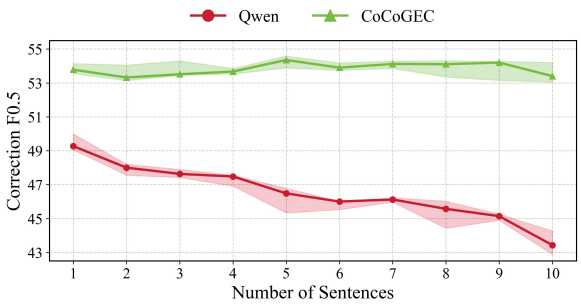

Figure 3: Robustness to perturbation length on Qwen3-8B on TEM-8*. As context grows, the vanilla model degrades, whereas the CoCoGEC-trained model keeps higher and more stable $F_{0.5}$, indicating better long-range robustness.

5.4 Robustness to Perturbation Length and Number

To evaluate robustness under varying perturbation length and number, we use the LLM-based Qwen3-8B as the backbone. We test on TEM-8 with sentence-level perturbations and increasing context length. Specifically, for each error-containing sentence, we concatenate k preceding and k following sentences ($k \in \{1, \ldots, 10\}$) as additional context, and we further apply word-level attacks at m different positions ($m \in \{1, \ldots, 5\}$) to vary the perturbation number. We compare the vanilla Qwen3-8B model with Qwen3-8B trained with CoCoGEC. Figure 3 shows that the vanilla model degrades as the context window grows, whereas CoCoGEC maintains higher and more stable $F_{0.5}$, indicating reduced sensitivity to long-range context. Table 3 reports results under word-level perturbations with increasing attack number: CoCoGEC consistently achieves higher $F_{0.5}$, TR, and SR across attack budgets, showing stronger tolerance to error-position shifts and denser error layouts.

6 Conclusion

We examined the robustness gap in grammatical error correction and proposed CoCoGEC, a contextual counterfactual generation framework that

[5]Definitions of CRS and P-CRS are given in Appendix B.6.

preserves intended corrections while varying the surrounding discourse. Results on robustness-oriented and standard GEC benchmarks indicate that CoCoGEC improves correction accuracy and markedly enhances resilience to word-level and sentence-level contextual variations, pointing to contextual counterfactual generation as an effective data-centric approach to robust GEC.

Limitations

CoCoGEC currently relies on external large language models (LLMs) to generate span-controlled counterfactuals. This introduces a dependency on the particular LLM and prompting setup used for augmentation, and future work could explore lighter-weight or fully self-contained generators to further reduce this reliance.

Acknowledgement

The authors would like to thank the anonymous reviewers for their insightful comments. This work is supported by the Chenguang Program of Shanghai Education Development Foundation and Shanghai Municipal Education Commission under Grant 24CGA26.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Alan Akbik, Tanja Bergmann, Duncan Blythe, Kashif Rasul, Stefan Schweter, and Roland Vollgraf. 2019. FLAIR: An easy-to-use framework for state-of-the-art NLP. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 54–59.

Dezhi An, Fang Wang, Jun Lu, and Shengcai Zhang. 2025. Self-explaining counterfactual data augmentation for nlp. In *2025 IEEE 6th International Seminar on Artificial Intelligence, Networking and Information Technology (AINIT)*, pages 1–6.

Abhijeet Awasthi, Sunita Sarawagi, Rasna Goyal, Sabyasachi Ghosh, and Vihari Piratla. 2019. Parallel iterative edit models for local sequence transduction. In *Proceedings of EMNLP-IJCNLP*.

Suyoung Bae, YunSeok Choi, Hyojun Kim, and Jee-Hyong Lee. 2025. Salad: Improving robustness and generalization through contrastive learning with structure-aware and llm-driven augmented data. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 12724–12738.

Christopher Bryant, Mariano Felice, and Ted Briscoe. 2017a. 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.

Christopher Bryant, Mariano Felice, and Ted Briscoe. 2017b. Automatic annotation and evaluation of grammatical error correction. In *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 127–136.

Mingshan Chang, Min Yang, Qingshan Jiang, and Ruifeng Xu. 2024. Counterfactual-enhanced information bottleneck for aspect-based sentiment analysis. *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 17736–17744.

Hao Chen, Rui Xia, and Jianfei Yu. 2021. Reinforced counterfactual data augmentation for dual sentiment classification. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 269–278.

Jianbo Chen, Le Song, Martin J. Wainwright, and Michael I. Jordan. 2018. Learning to explain: An information-theoretic perspective on model interpretation. In *Proceedings of the 35th International Conference on Machine Learning*.

Xiang Chen, Tianyu Gao, and Antoine Bosselut. 2023a. Disco: Distilling counterfactuals with large language models. In *Proceedings of EMNLP*.

Zeming Chen, Qiyue Gao, Antoine Bosselut, Ashish Sabharwal, and Kyle Richardson. 2023b. DISCO: Distilling counterfactuals with large language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5514–5528.

Zhuowei Chen, Lianxi Wang, Yuben Wu, Xinfeng Liao, Yujia Tian, and Junyang Zhong. 2024. An effective deployment of diffusion lm for data augmentation in low-resource sentiment classification. *arXiv preprint arXiv:2409.03203*.

Seungtaek Choi, Myeongho Jeong, Hojae Han, and Seung-won Hwang. 2022. C2l: Causally contrastive learning for robust text classification. In *Proceedings of the AAAI conference on artificial intelligence*, pages 10526–10534.

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*.

Kai Dang, Jiaying Xie, and Jie Liu. 2021. Leveraging adversarial training to facilitate grammatical error correction. In *Artificial Neural Networks and Machine Learning – ICANN 2021 (LNCS 12891–12895), Part I*, pages 67–78.

Tao Fang, Xuebo Liu, Derek F. Wong, Runzhe Zhan, Liang Ding, Lidia S. Chao, Dacheng Tao, and Min Zhang. 2023. Transgec: Improving grammatical error correction with translationese. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 3614–3633.

Phillip Howard, Gadi Singer, Vasudev Lal, Yejin Choi, and Swabha Swayamdipta. 2022. Neurocounterfactuals: Beyond minimal-edit counterfactuals for richer data augmentation. In *Findings of EMNLP*, pages 5056–5072.

Edward J. Hu, Yelong Shen, Phil Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Junqi Jiang, Francesco Leofante, Antonio Rago, and Francesca Toni. 2024. Robust counterfactual explanations in machine learning: A survey.

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 NAACL-HLT*.

Anisia Katinskaia and Roman Yangarber. 2023. Grammatical error correction for sentence-level assessment in language learning. In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*, pages 488–502.

Anisia Katinskaia and Roman Yangarber. 2024. GPT-3.5 for grammatical error correction. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 7831–7843.

Roman Kovalchuk, Mariana Romanyshyn, and Petro Ivaniuk. 2025. Introducing omnigec: A silver multilingual dataset for grammatical error correction. *arXiv preprint arXiv:2509.14504*.

Wei Li, Wen Luo, Guangyue Peng, and Houfeng Wang. 2025. Explanation based in-context demonstrations retrieval for multilingual grammatical error correction. In *Proceedings of the 2025 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 4881–4897.

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.

Xinyuan Li and Yunshi Lan. 2025. Large language models are good annotators for type-aware data augmentation in grammatical error correction. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 199–213.

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.

Jared Lichtarge, Chris Alberti, Shankar Kumar, Noam Shazeer, Niki Parmar, and Simon Tong. 2019. Corpora generation for grammatical error correction. 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 3291–3301.

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.

Kostiantyn Omelianchuk, Vitaliy Atrasevych, Artem Chernodub, and Oleksandr Skurzhanskyi. 2020. Gector – grammatical error correction: Tag, not rewrite. In *Proceedings of BEA Workshop at ACL*, pages 163–170.

Chanjun Park, Seonmin Koo, Seolhwa Lee, Jaehyung Seo, Sugyeong Eo, Hyeonseok Moon, and Heuiseok Lim. 2023. Synthetic alone: Exploring the dark side of synthetic data for grammatical error correction. *CoRR*.

Mitchell Plyler and Min Chi. 2025. Iterative counterfactual data augmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Muhammad Reza Qorib and Hwee Tou Ng. 2023. System combination via quality estimation for grammatical error correction. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12746–12759.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, pages 1–67.

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.

Aiman Solyman, Marco Zappatore, Zhenyu Wang, and 1 others. 2023. Optimizing the impact of data augmentation for low-resource grammatical error correction. *Journal of King Saud University – Computer and Information Sciences*.

KC Sreedhar, T Kavya, JVS Prasad, and V Varshini. 2025. A novel metric-based counterfactual data augmentation with self-imitation reinforcement learning (sil). *International Journal of Advanced Computer Science & Applications*.

Felix Stahlberg and Shankar Kumar. 2020. Sequence transduction using span-level edit operations. In *Proceedings of EMNLP*.

Felix Stahlberg and Shankar Kumar. 2021. Synthetic data generation for grammatical error correction with tagged corruption models. In *Proceedings of the 16th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 37–47.

Jingbo Sun, Weiming Peng, Zhiping Xu, Shaodong Wang, and Jihua Song. 2023. Incorporating syntactic cognitive in multi-granularity data augmentation for chinese grammatical error correction. In *International Conference on Neural Information Processing*.

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.

Zecheng Tang, Kaifeng Qi, Juntao Li, and Min Zhang. 2023. Beyond hard samples: Robust and effective grammatical error correction with cycle self-augmenting. *CoRR*.

Marcos Treviso, Alexis Ross, Nuno M Guerreiro, and André FT Martins. 2023. Crest: A joint framework for rationalization and counterfactual text generation. *arXiv preprint arXiv:2305.17075*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

Sahil Verma, Varich Boonsanong, Minh Hoang, Keegan Hines, John Dickerson, and Chirag Shah. 2024. Counterfactual explanations and algorithmic recourses for machine learning: A review. *ACM Computing Surveys*.

Zhaohong Wan, Xiaojun Wan, and Wenguang Wang. 2020. Improving grammatical error correction with data augmentation by editing latent representation. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2202–2212.

Ke Wang and Xiaojun Wan. 2022. Counterfactual representation augmentation for cross-domain sentiment analysis. *IEEE Transactions on Affective Computing*, pages 1979–1990.

Yixuan Wang, Baoxin Wang, Yijun Liu, Qingfu Zhu, Dayong Wu, and Wanxiang Che. 2024a. Improving grammatical error correction via contextual data augmentation. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 10898–10910.

Yongjie Wang, Xiaoqi Qiu, Yu Yue, Xu Guo, Zhiwei Zeng, Yuhong Feng, and Zhiqi Shen. 2024b. A survey on natural language counterfactual generation.

Yongjie Wang, Xiaoqi Qiu, Yu Yue, Xu Guo, Zhiwei Zeng, Yuhong Feng, and Zhiqi Shen. 2024c. A survey on natural language counterfactual generation. *arXiv preprint arXiv:2407.03993*.

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

Dancheng Xin, Jiawei Yuan, and Yang Li. 2024. Diffusion based counterfactual augmentation for dual sentiment classification. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 4901–4911.

Heerin Yang, Sseung-won Hwang, and Jungmin So. 2024. Relation-based counterfactual data augmentation and contrastive learning for robustifying natural language inference models. *arXiv preprint arXiv:2410.20710*.

Yang Yang. 2017. Test for english majors-band 8 (tem8) in china. *Journal of Language Teaching and Research*, pages 1229–1233.

J. Ye, Z. Xu, Y. Li, X. Cheng, L. Song, Q. Zhou, Hai-Tao Zheng, Y. Shen, and X. Su. 2024. CLEME2.0: Towards more interpretable evaluation by disentangling edits for grammatical error correction. *arXiv preprint arXiv:2407.00934*.

Jingheng Ye, Yinghui Li, Yangning Li, and Hai-Tao Zheng. 2023. Mixedit: Revisiting data augmentation and beyond for grammatical error correction. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 10161–10175.

Paul Youssef, Christin Seifert, Jörg Schlötterer, and 1 others. 2024. Llms for generating and evaluating counterfactuals: A comprehensive study. In *Findings of the association for computational linguistics: EMNLP 2024*, pages 14809–14824.

Ding Zhang, Yangning Li, Lichen Bai, Hao Zhang, Yinghui Li, Haiye Lin, Hai-Tao Zheng, Xin Su, and Zifei Shan. 2025. Loss-aware curriculum learning for chinese grammatical error correction. In *ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.

Yue Zhang, Leyang Cui, Enbo Zhao, Wei Bi, and Shuming Shi. 2023. RobustGEC: Robust grammatical error correction against subtle context perturbation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 16780–16793.

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, Zheyan Luo, and Zhixin Ma. 2024. Llamafactory: Unified efficient fine-tuning of 100+ language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pages 156–166.

Xiaoling Zhou, Ou Wu, and Michael K Ng. 2023. Implicit counterfactual data augmentation for robust learning. *arXiv preprint arXiv:2304.13431*.

A Prompt Templates for Counterfactual Generation

We instantiate our span-edit template over the training corpus to create a set of prompt–sentence pairs, denoted `gec_examples_with_span`, for eliciting span-level edits from a large language model (LLM). All prompts described in this section operate in a *sentence* mode: given a full sentence, the LLM is asked either to fill a masked span or produce additional surrounding context. This section summarizes the templates used in our counterfactual generation pipeline.

A.1 Intra-sentential Editing

Global instruction. All span-edit prompts follow the "Span-edit prompt (sentence mode)" template below and differ only in the concrete {sentence} and span positions.

Unified span-edit template. We use a single template for replacement, deletion, and insertion. The selected span (or insertion position) is masked with `[BLANK]`, and the LLM is required to output only the content that should fill `[BLANK]`. An empty prediction corresponds to a deletion. For insertion, we treat the insertion position as a zero-length span between two tokens and apply the same template.

Span-edit prompt (sentence mode)

```
Task Instruction: Fill in the [BLANK] with
a word or phrase, or leave it empty.
- Follow standard English grammar.
- No grammatical errors are allowed.
- Do not copy from the original sentence.
- The final sentence must be logically
correct and sound natural to native
speakers.
- Output only the content that should
replace [BLANK]. Do not output the full
sentence.
Sentence: {sentence}
[BLANK] should be:
```

A.2 Inter-sentential Expansion for Context Augmentation

We perform inter-sentential expansion by generating a fluent context before and after an input sentence. Given a training sample with source sentence s and its correction t, we use an instruction-following LLM to generate a prefix p and a suffix q such that p naturally leads into s and q naturally follows s, while keeping s unchanged. The expanded source and target are constructed as $p \oplus s \oplus q$ and $p \oplus t \oplus q$, respectively. To improve diversity, we randomly sample the number of prefix/suffix sentences (e.g., 3–5) for each sample. If the LLM output cannot be parsed into the required two-block format or violates constraints (e.g., repeating s), we fall back to the original sample without expansion.

Inter-sentential expansion prompt (prefix/-suffix)

```
You are given a sentence S in English. Your
task is to write additional context BEFORE
and AFTER S.
RULES:
1) In the <<<PREFIX>>> block, write about
{n_pre} fluent English sentences that
smoothly lead into S.
2) In the <<<SUFFIX>>> block, write about
{n_suf} fluent English sentences that
naturally follow S.
3) Do NOT change S at all. Keep it EXACTLY
AS-IS.
4) Do NOT include S itself in the blocks.
5) Write only in English.
6) Output EXACTLY TWO blocks:
<<<PREFIX>>>
...text before S...
<<<SUFFIX>>>
...text after S...
S: {sentence}
```

A.3 LLM-based Correction Prompt

For instruction-following LLMs used as correction models (or LLM-based scorers), we adopt the following sentence-level GEC prompt:

Grammar error correction prompt (GEC)

```
You are an experienced English teacher who
specializes in grammatical error correction
(GEC). You are given exactly one sentence as
input. Correct it with the fewest possible
edits (minimal edit distance).
Requirements:
1) Correct grammar, spelling, and word
choice only. 2) Keep the original structure
and meaning; do not paraphrase or reorder.
3) If the sentence is already correct,
return it unchanged. 4) Output format:
return exactly one sentence on a single
line, with no explanations or extra text.
Original: {sentence}
Corrected:
```

B Additional Experimental Details

B.1 Dataset Statistics

Table 5 reports corpus statistics for each split, including sentence counts, source-token volumes, and error-type breakdowns.

Properties	train	dev	test_BEA-19	test_CoNLL-14	test_TEM-8
#Sent	22848	2538	3018	1578	1590
#Tokens(src)	1.14M	125.1K	242.1K	120.1K	142.4K
Avg src length (tok)	49.92	49.28	80.23	76.08	89.56
Avg #sentences in src	2.87	2.85	4.90	4.51	4.78
Errorful %	78.4%	78.4%	71.8%	87.5%	100.0%
#Edits per example	1.999	1.963	2.095	2.369	2.029
#Missing errors per example	0.373	0.357	0.217	0.278	0.282
#Redundant errors per example	0.205	0.192	0.304	0.479	0.223
#Substitution errors per example	1.407	1.394	1.560	1.606	1.524
#Word-Order errors per example	0.014	0.019	0.014	0.007	0.000

Table 5: Dataset statistics across splits. Errorful % is the percentage of examples with at least one ERRANT edit between (src, tgt). Sentence counts are reported using spaCy segmentation and a rule-based heuristic (punctuation/newline boundaries).

B.2 Gold Validity Check

After enforcing the edit-subset constraint ($E' \subseteq E$), we apply a lightweight gold-validity check to ensure that the revised target t' remains valid for the perturbed source c'. We discard candidates if (i) t' fails automatic grammaticality checking, (ii) a frozen external GEC verifier further edits t', or (iii) t' shows obvious semantic drift.

Quality-control pipeline. We further apply a filtering pipeline to ensure that generated contexts are valid distractors without introducing new errors or leaking the correction.

Grammaticality check (ERRANT). Using the same ERRANT-based edit extraction pipeline as in the main experiments, we verify that generated prefixes and suffixes introduce no additional grammatical errors. Most generated contexts pass this check, yielding a Context Robustness Score (CRS) of 99.6%.

Fluency and coherence filtering. We use a perplexity filter to remove non-fluent generations and a semantic-similarity constraint to filter severe semantic drift, ensuring natural and coherent augmented contexts.

Leakage prevention. We remove candidates that may directly reveal the target correction, reducing reliance on accidental lexical cues.

Manual verification. Manual inspection of sampled cases shows that the retained counterfactuals generally preserve the original error pattern while varying only the surrounding context, supporting the validity of our pipeline.

B.3 Hyperparameters

For the LLM-based scorer, we fine-tune Qwen3-8B with LoRA (Hu et al., 2022) using LLaMA-Factory (Zheng et al., 2024); the LoRA configuration is summarized in Table 6.

Parameter	Value
Target modules	all
LoRA rank	16
Learning rate	5×10^{-5}
Training epochs	2
Per-device batch size	2
Gradient accumulation steps	8
Warmup ratio	0.1
Max sequence length	1024
Scheduler	cosine
Precision	bf16

Table 6: LoRA configuration for fine-tuning Qwen3-8B.

We also train and evaluate GECToR-large and T5-large backbones. For GECToR-large, we follow the official implementation and training hyperparameters used in RobustGEC (Zhang et al., 2023). For T5-large, we follow the text-to-text training recipe of Rothe et al. (2021) and use the public `gec-t5` implementation.[6] For $\mathcal{DISCO}$ and TypeDA, we implement the methods as described in the original papers and run them with the authors' recommended hyperparameters and filtering rules, without additional tuning beyond adapting them to our GEC data. Unless otherwise specified, all baseline settings are kept identical to the referenced implementations.

Model	BEA-19 dev			CoNLL-14 test			BEA-19 test		
	P	R	$F_{0.5}$	P	R	$F_{0.5}$	P	R	$F_{0.5}$
GECToR	68.8	38.8	59.6	75.4	40.9	64.5	79.0	56.2	73.1
+ CoCoGEC	72.2	42.1	**63.2**	72.6	54.6	**68.2**	80.0	57.2	**74.1**

Table 7: Evaluating GECToR-large with and without CoCoGEC on its original BEA-19 and CoNLL-14 benchmarks.

B.4 Standard GEC Performance on Original Benchmarks

Table 7 reports standard GEC results on the original BEA-19 dev/test and CoNLL-14 test sets. Compared with the vanilla GECToR backbone, the CoCoGEC-augmented model attains similar or slightly higher $F_{0.5}$ scores on all three benchmarks (up to +3.7 on CoNLL-14), indicating that improving robustness on RobustGEC does not degrade conventional GEC performance.

[6] https://github.com/gotutiyan/gec-t5

B.5 Supplementary Cross-Lingual Results on VisCGEC

Following the reviewer suggestion on cross-lingual and cross-domain validation, we additionally evaluate COCOGEC on the Chinese VisCGEC benchmark. Results are shown in Table 8. We follow the official VisCGEC evaluation protocol and compare the baseline and COCOGEC-augmented models under the same training and decoding settings, without additional model-specific tuning.

Method	Precision	Recall	$F_{0.5}$
Baseline	32.30	20.23	28.86
COCOGEC (ours)	**43.53**	16.12	**32.49**

Table 8: Supplementary results on the Chinese VisCGEC benchmark.

As shown in Table 8, COCOGEC also improves $F_{0.5}$ on VisCGEC, suggesting that the proposed context-decoupled counterfactual augmentation is not limited to the English RobustGEC setting and can generalize to a different language and benchmark.

B.6 Context Robustness Metrics: CRS and P-CRS

Following RobustGEC (Zhang et al., 2023), we report two context-robustness metrics: Context Robustness Score (CRS) and Pair-wise Context Robustness Score (P-CRS). Each GEC case contains one original sentence and a set of context-perturbed variants. CRS measures strict stability: it counts a case as correct only if the model outputs exactly identical corrections for *all* variants within the same case,

$$\mathrm{CRS} = \frac{\#\mathrm{Case}_C}{\#\mathrm{Case}_T}. \tag{2}$$

P-CRS is more lenient and evaluates stability at the original⇔perturb pair level,

$$\text{P-CRS} = \frac{\#\text{P-sample}_C}{\#\text{P-sample}_T}. \tag{3}$$

For example, if a case has one original and five perturbed variants where four variants share the same correction as the original, then CRS is 0 while P-CRS is $4/5$.

	TEM-8	
Method	CRS↑	P-CRS↑
Seq2Edit (GECToR)	6.41	64.30
+ CPR method	7.93	72.22
+ TypeDA	3.39	63.09
+ $\mathcal{DISCO}$	8.25	69.77
+ CoCoGEC	**12.45**	**72.52**
Seq2Seq (T5)	0.75	53.20
+ CPR method	2.26	59.22
+ TypeDA	0.75	53.13
+ $\mathcal{DISCO}$	0.75	51.54
+ CoCoGEC	**4.15**	**63.54**
LLM(Qwen3-8B)	16.60	67.62
+ CPR method	7.93	72.22
+ TypeDA	3.01	50.26
+ $\mathcal{DISCO}$	6.41	55.24
+ CoCoGEC	**16.98**	**72.52**
GPT-4o	8.67	**69.57**
LLaMA3-8B	1.13	56.83

Table 9: CRS and P-CRS on TEM-8. CoCoGEC improves robustness over vanilla GECToR and T5, while the effects on LLM-based baselines are mixed.

B.7 CRS and P-CRS on TEM-8

Table 9 summarizes CRS and P-CRS on TEM-8*. For Seq2Edit (GECToR) and Seq2Seq (T5), CoCoGEC improves robustness over the corresponding vanilla models for both GECToR and T5. For GECToR, $\mathcal{DISCO}$ yields the highest CRS/P-CRS among the training variants, while CoCoGEC still provides a clear gain over the vanilla baseline. For Qwen3, CoCoGEC gives a modest improvement in both CRS and P-CRS, whereas other recipes exhibit trade-offs (e.g., CPR increases P-CRS but lowers CRS), suggesting that stability improvements for LLM-based correction may be sensitive to the training recipe.

B.8 Case Study

We present representative long-context cases to qualitatively compare model behaviors. For each case, we report the original input, the expanded input, the gold correction, and model predictions from different systems (e.g., GECToR, T5, Qwen3, ChatGPT, and LLaMA).

Case Study: Contextual counterfactuals for "an sad person"

Line	Utterance
Original s	*The key to* ***comfort*** *a person is to try and avoid a debate over* ***if*** *your loved one is sick and instead* ***of look*** *for common ground.*
Original t	*The key to* ***comforting*** *a person is to try and avoid a debate over* ***whether*** *your loved one is sick and instead* ***look*** *for common ground.*
Counterfactual s'	*The key to* ***comfort*** *a sad person is to try and avoid a debate over* ***if*** *your loved one is sick and instead* ***of look*** *for common ground.*
Counterfactual t'	*The key to* ***comforting*** *a sad person is to try and avoid a debate over* ***whether*** *your loved one is sick and instead* ***look*** *for common ground.*
Difference s **vs.** s'	*The counterfactual* s' *inserts the descriptor "sad" before "person", while the local error pattern around* ***comfort / if / "instead of look"*** *remains unchanged.*
GECToR-large on s	*The key to* ***comforting*** *a person is to try and avoid a debate over* ***if*** *your loved one is sick and instead* ***of looking*** *for common ground.*
GECToR-large on s'	*The key to* ***comforting*** *a sad person is to try and avoid a debate over* ***if*** *your loved one is sick and instead* ***of looking*** *for common ground.*
T5-large on s	*The key to* ***comforting*** *a person is to try and avoid a debate over* ***if*** *your loved one is sick and instead* ***of looking*** *for common ground.*
T5-large on s'	*The key to* ***comforting*** *a sad person is to try and avoid a debate over* ***if*** *your loved one is sick and instead* ***of looking*** *for common ground.*
Qwen3-8B on s	*The key to* ***comforting*** *a person is to try and avoid a debate over* ***whether*** *your loved one is sick and instead* ***of looking*** *for common ground.*
Qwen3-8B on s'	*The key to* ***comforting*** *a sad person is to try and avoid a debate over* ***whether*** *your loved one is sick and instead* ***of looking*** *for common ground.*
GPT-4o on s	*The key to* ***comforting*** *a person is to try* ***to*** *avoid a debate over* ***whether*** *your loved one is sick and instead* ***look*** *for common ground.*
GPT-4o on s'	*The key to* ***comforting*** *a sad person is to try* ***to*** *avoid a debate over* ***whether*** *your loved one is sick and instead* ***of looking*** *for common ground.*
LLaMA3-8B on s	*The key to* ***comforting*** *a person is to try* ***to*** *avoid* ***debating whether*** *your loved one is sick and instead* ***look*** *for common ground.*
LLaMA3-8B on s'	*The key to* ***comforting*** *a sad person is to try* ***to*** *avoid* ***debating if*** *your loved one is sick and instead* ***look*** *for common ground.*
Qwen3-8B+COCOGEC on s	*The key to* ***comforting*** *a sad person is to try and avoid a debate over* ***whether*** *your loved one is sick and instead* ***look*** *for common ground.*
Qwen3-8B+COCOGEC on s'	*The key to* ***comforting*** *a person is to try and avoid a debate over* ***whether*** *your loved one is sick and instead* ***look*** *for common ground.*

Table 10: Line-by-line case study of a contextual counterfactual pair (s, s') for "an sad person". Red bold spans mark residual errors relative to the gold targets, while dark-green bold spans highlight canonical corrections.