Instruction-Following LLMs for Grammatical Error Correction: Analyzing Neutral-Anchored Instructional Sensitivity Across Editing Modes

Tolgahan Türker[1] **and Gülşen Eryiğit**[2]
Faculty of Computer and Informatics
Istanbul Technical University
Istanbul, Türkiye
turkert21@itu.edu.tr[1]
gulsen.cebiroglu@itu.edu.tr[2]

Abstract

Grammatical Error Correction (GEC) requires models to make edit decisions under competing objectives: correcting errors while either minimizing changes or maximizing fluency. However, we lack a principled characterization of how instruction-following Large Language Models (LLMs) shift their edit decisions across such *editing modes*, and whether standard evaluation setups faithfully reflect these shifts. We address this gap by defining three modes—*Neutral*, *Minimal-Edit*, and *Fluency-Edit*—and measuring *neutral-anchored* performance shifts to quantify instructional sensitivity. We benchmark seven LLMs, including proprietary and open-weight models, in a unified zero-shot prompting schema on CoNLL-2014, BEA-2019, and JFLEG datasets. The Minimal-Edit instruction mitigates over-editing and typically boosts precision; in some settings, strong models also improve recall, suggesting more selective and effective corrections. In contrast, the Fluency-Edit instruction often encourages broader paraphrastic rewriting that may improve perceived fluency while lowering GLEU, suggesting both a metric–objective mismatch and a shift away from targeted local correction. Notably, Claude-Sonnet-4.5 demonstrates superior zero-shot capabilities, outperforming previously reported scores and matching or even exceeding few-shot results across CoNLL-2014 ($F_{0.5}$: 67.05), BEA-2019 ($F_{0.5}$: 64.91), and JFLEG ($GLEU$: 66.09).

1 Introduction

GEC is a well-established natural language processing task that involves automatically correcting grammatical errors in text. From a computational perspective, GEC is commonly modeled as a text-to-text transformation problem, where an errorful input is transformed into a corrected output while preserving the author's intended meaning. Depending on the target application, GEC systems face distinct requirements: high-quality text generation use cases necessitate fluency-focused revisions on the text to achieve more natural-sounding expression, whereas pedagogical settings often prioritize conservative, minimal edits to gradually introduce learner feedback tailored to the learner's proficiency level to facilitate a more effective and pedagogically grounded learning experience (Wood et al., 1976; Lyster and Ranta, 1997).

These divergent objectives imply an inherent trade-off: strictly adhering to minimal edits often leaves unnatural or awkward phrasing untouched, whereas aggressively pursuing fluency often leads to broader rewrites that can inadvertently shift the author's intended meaning. For over a decade, shared tasks and benchmarks for GEC (e.g., CoNLL (Ng et al., 2013, 2014), BEA (Bryant et al., 2019), JFLEG (Napoles et al., 2017), MultiGEC (Masciolini et al., 2025)) have both driven progress and standardized evaluation practices, making the *minimal-edit* and *fluency-edit* distinction central to assessment of GEC systems.

Recent advances in LLMs—particularly their ability to follow complex instructions and produce high-quality outputs—have motivated researchers to investigate their effectiveness for a wide range of tasks, which have traditionally relied on task-specific models developed over many years. In the context of English GEC, a growing body of work (Wu et al., 2023; Fang et al., 2023; Coyne et al., 2023; Loem et al., 2023; Davis et al., 2024) has begun to explore this potential, analyzing the out-of-the-box performance of LLMs under zero- and few-shot settings across standard benchmarks. While these studies provide a valuable baseline, there remains an opportunity to systematically assess contemporary models' intrinsic capability by examining how their performance shifts across different editing modes, especially when these objectives are specified through structurally consistent instructional framing. To address this, we present an eval-

Proceedings of the 21st Workshop on Innovative Use of NLP for Building Educational Applications, pages 234–247

uation of six contemporary LLMs—ranging from high-capacity systems to efficient mini variants, and including both proprietary and open-weight families—together with a legacy model used to assess the quality of instructions, under a tripartite instructional framework in a zero-shot setting. We define three distinct editing modes: *Neutral*, *Minimal-Edit*, and *Fluency-Edit*, using structurally consistent prompts to observe how models modulate their output in response to different instructional goals. By quantifying the shifts from a *neutral-anchored* baseline, we characterize the instructional sensitivity of these models. To support future benchmarking efforts in the GEC community, our complete experimental framework, including implementation code, instructions, and model-generated outputs, is released for public use. [1]

Our primary contributions are: (I) To the best of our knowledge, we achieve the state-of-the-art performances for English prompt-based GEC, with Claude-Sonnet-4.5 achieving the highest reported zero-shot $F_{0.5}$ scores on CoNLL-2014 (67.05) and BEA-2019 (64.91), thereby redefining the zero-shot baseline for the task, (II) we provide empirical evidence for an evolution in instructional sensitivity, contrasting the behavioral rigidity of legacy models like GPT-3.5-Turbo with the higher instructional sensitivity of current frontier models, (III) we characterize the precision-recall (P-R) dynamics across editing modes, demonstrating that while minimal-edit instruction typically acts as a precision filter, it can effectively circumvent the traditional P-R trade-off in top-tier models by aligning generative priors with GEC-specific objectives, (IV) we provide an analysis for the metric-objective mismatch on the JFLEG benchmark, showing that as models exhibit higher instructional sensitivity, their native-like enhancements are penalized by the rigid word-matching constraints of GLEU.

2 Related Works

Initial prompt-based GEC studies primarily focused on the GPT-3.5 family. Wu et al. (2023) compared ChatGPT against specialized tools like Grammarly[2] and GECTOR (Omelianchuk et al., 2020), noting that its performance on automated metrics was hindered by a tendency to go beyond localized edits to modify surface expressions and sentence structures. This was further characterized by Fang et al. (2023), who demonstrated that ChatGPT's human-like fluency often comes at the expense of precision on traditional benchmarks due to its over-correction tendencies. Coyne et al. (2023) extended this analysis to GPT-4, demonstrating a clear performance gap between benchmarks: while the model set a new high score on the fluency-based JFLEG dataset, it significantly underperformed on BEA-2019. Their findings confirm that the GEC performance of LLMs is highly dependent on the targeted edit settings, as prompt-driven fluency often results in extensive revisions that are penalized by minimal-edit metrics.

While initial work established a performance baseline, subsequent research has shifted toward investigating the instructional sensitivity of LLMs and the extent to which their correction behaviors can be strategically modulated. Loem et al. (2023) demonstrated that GPT-3 can be effectively steered toward either *minimal-edit* or *fluency-edit* objectives through instructions. However, their analysis revealed that while instructions can guide the model, the provided examples for few-shot settings play a more critical role in controlling editing behavior and achieving the desired correction style. This sensitivity to prompting is further emphasized by Davis et al. (2024), who conducted an extensive comparison between seven open-source and three commercial LLMs across four benchmarks. They found that while commercial models remain superior in fluency-oriented tasks, open-source models can achieve competitive performance if the prompting strategy is carefully optimized. The requirement for structured, objective-specific instructions is further reinforced by Zeng et al. (2024), showing that a uniform prompting approach is insufficient for addressing the varying error patterns across different language proficiency levels, as over-correction tendencies become more pronounced in advanced learner texts.

A critical challenge in modern GEC is the inherent bias of traditional metrics, where LLMs are often penalized for valid, native-like corrections that deviate from the conservative, minimal edits found in standard references (Östling et al., 2024). To mitigate the metric penalties associated with non-mandatory edits, Staruch et al. (2025) explored methods to steer models toward the principle of least change, effectively constraining the output to adhere to minimal-edit requirements. Alternatively, Liang et al. (2025) introduced edit-wise

[1] https://github.com/tolgahanturker/gec-llm-eval

[2] https://www.grammarly.com

Neutral	Minimal-Edit	Fluency-Edit
ROLE: You are a helpful language error correction assistant. **TASK:** Correct the sentence provided inside `<input>...</input>` tags. **RULES:** **Process:** • Read the full sentence to understand its meaning. • Identify errors in the sentence. • Apply edits. **Constraints:** • Preserve the original meaning. **Output:** • If the sentence is already correct, return it unchanged. • Return ONLY the corrected sentence as plain text. • Do NOT include explanations, steps, or additional text.	**ROLE:** You are a helpful language teaching assistant for second language learners. **TASK:** Correct the sentence provided inside `<input>...</input>` tags with MINIMAL edits. **RULES:** **Process:** • Read the full sentence to understand its meaning. • Identify grammatical, spelling, and punctuation errors. • Apply only the edits necessary to fix those errors. **Constraints:** • Do NOT rewrite for style or fluency. • Preserve the original meaning, tone, and sentence structure as much as possible. **Output:** • If the sentence is already correct, return it unchanged. • Return ONLY the corrected sentence as plain text. • Do NOT include explanations, steps, or additional text.	**ROLE:** You are a native English speaker. **TASK:** Correct grammatical errors in the sentence provided inside `<input>...</input>` tags and make it sound natural to a native speaker of English. **RULES:** **Process:** • Read the full sentence to understand its meaning. • Identify errors. • Apply edits to correct errors and improve fluency of the sentence. **Constraints:** • Preserve the original meaning and intent. **Output:** • If the sentence is already correct, return it unchanged. • Return ONLY the corrected sentence as plain text. • Do NOT include explanations, steps, or additional text.

Table 1: Instructions defining the three editing modes: *Neutral*, *Minimal-Edit*, and *Fluency-Edit*.

preference optimization, an alignment technique that helps models distinguish between mandatory grammatical fixes and optional enhancements.

A different methodological angle pursues automatic optimization of the prompt. Chernodub et al. (2025) proposed APIO, an automatic prompt induction and optimization method that iteratively refines a list of instructions through beam search over an LLM-generated candidate pool, using validation-set performance as the optimization signal.

Building on these efforts to modulate model behavior, our work introduces a structured tripartite instructional framework to systematically characterize how contemporary LLMs navigate these competing objectives of minimal-edit and fluency-edit.

3 Instructions

To isolate the impact of different editing modes, we designed a unified prompting schema following widely accepted best practices for prompt engineering. Each prompt is structured as a consistent interface between the user and the model, consisting of five core components: role framing, task description, processing rules, constraints, and output formatting. This modular design ensures that model behavior is strictly governed by the intended correction regime while minimizing extraneous output.

We define three editing modes to probe model capabilities: (I) *Neutral*, which serves as an anchor to observe default correction behavior without specific constraints; (II) *Minimal-Edit*, which restricts changes to the smallest set of edits required for correctness while preserving sentence structure; and (III) *Fluency-Edit*, which targets native-like naturalness and flow while only requiring meaning preservation.

All prompts utilize a consistent interface where the source text is encapsulated in `<input>...</input>` tags. To eliminate the need for post-processing on model-generated outputs, we enforce a strict output contract: models are required to return only the corrected sentence, without any additional text. Table 1 shows the instructions we used.

Unlike the other modes, the *Fluency-Edit* instruction explicitly incorporates the term *grammatical errors* in the task to provide a tighter constraint on the model's editing behavior. This design choice aims to mitigate the common over-editing issue observed in prompt-based GEC studies by guid-

Track	Model	Organization	Access	Inference Platform / API
High-Capacity	GPT-4.1-2025-04-14	OpenAI	Proprietary	OpenAI API
	Claude-Sonnet-4.5	Anthropic	Proprietary	Anthropic API
	Llama-3.3-70B-Instruct	Meta	Open-weight	Azure AI Foundry
	Mistral-Large-3	Mistral AI	Open-weight	Azure AI Foundry
Efficient Mini	GPT-5-mini-2025-08-07	OpenAI	Proprietary	OpenAI API
	Llama-4-Scout-17B-16E-Instruct	Meta	Open-weight	Azure AI Foundry
Legacy	GPT-3.5-Turbo-0125	OpenAI	Proprietary	OpenAI API

Table 2: Summary of evaluated LLMs.

ing the model to prioritize grammatical correctness while striving for native-like flow. To avoid potential conservative bias from corrective roles, we used a *native speaker* persona to prioritize naturalness. As a robustness check, we also tested other personas to see if this choice contributed to the performance decline detailed in Section 5. However, preliminary evaluations—including tests with alternative persona (role) assignments—revealed that these instructional refinements did not alleviate the persistent performance degradation in this mode.

4 Experiments

As listed in Table 2, in our experiments, we evaluate seven models across three tracks to cover various scales and deployment tiers. The High-Capacity track represents the performance upper bound for our experiments, comprising large-capacity commercial models known for superior instruction-following alongside leading open-weight representatives. To reflect the industry trend toward compact intelligence, the Efficient Mini track includes models optimized for low-latency and cost-efficiency. Finally, the Legacy track includes GPT-3.5-Turbo as a historical reference point to measure the quality of the instructions used in the study and the progress made by newer generations in zero-shot GEC.

Experiments are conducted on three standard benchmarks: CoNLL-2014 (Ng et al., 2014), utilizing alternative gold references to better account for the diverse but valid outputs of LLMs in minimal-edit tasks; BEA-2019 (W&I + LOCNESS) (Bryant et al., 2019), providing a large-scale evaluation across the full CEFR spectrum and native student essays; and JFLEG (Napoles et al., 2017), which targets overall naturalness and native-like flow to assess fluency-oriented rewriting capacity.

Our experimental framework is intended to minimize formatting or inference-related inconsistencies, allowing the performance metrics to more accurately reflect the actual correction capabilities of the models. To do that, we utilize a series of alignment steps that first optimize the source text and then reformat the generated outputs to meet specific benchmark requirements.

Pre-processing. Since the source sentences in the benchmarks are typically provided in tokenized format, we need to detokenize the sentences before LLM processing to provide a more natural linguistic context. Following the practices established by Davis et al. (2024), we employ the *Moses detokenizer* (Koehn et al., 2007) and implement a specific rule to resolve split negative contractions (e.g., *do n't* → *don't*). We expect this processing to help align the input format with the models' pre-trained capabilities.

Evaluation Alignment. To facilitate precise scoring, model-generated responses must be re-tokenized to match the specific requirements of each benchmark's evaluation script. For the BEA-2019 test set, we conduct tokenization via *spaCy*[3] to facilitate the creation of `.m2` files, directly aligning with the official shared task proposal (Bryant et al., 2019). For CoNLL-2014, we apply the *NLTK tokenizer*[4] to generate text-based result files, following the conventions suggested in the shared task (Ng et al., 2014). These steps ensure that the model outputs are perfectly aligned with the gold references before being processed by the M^2 and ERRANT scorers.

Inference Details. All evaluations are conducted in a zero-shot setting to assess the out-of-the-box capabilities of the models. To minimize sampling variability, we set the temperature parameter to 0 for all models. Furthermore, to accommodate the varying Tokens Per Minute (TPM) and Requests Per Minute (RPM) constraints of the different API providers, a tailored `delay_per_request` value is implemented for each model. This en-

[3] https://spacy.io/
[4] https://www.nltk.org/

sured optimal throughput while strictly adhering to provider-specific rate-limit quotas. The corresponding TPM and RPM values are provided in the GitHub repository. The maximum output length is set to 2048 tokens to avoid overly long generations.

Experimental Design. To align the instructions with each benchmark's objectives, and to quantify how much model performance shifts from the Neutral prompt (which imposes no specific editing bias), we evaluate CoNLL-2014 and BEA-2019 under Neutral and Minimal-Edit instructions, and JFLEG under Neutral and Fluency-Edit instructions. This design probes whether models can suppress inherent over-editing tendencies under the minimal-edit mode or, conversely, leverage their generative capacity to produce more fluent, natural-sounding output. For this measurement to be interpretable, we restrict our evaluation to a zero-shot setting in order to isolate the effect of instructional framing: introducing in-context demonstrations would conflate instruction-driven behavioral shifts with example-driven imitation, undermining the *neutral-anchored* sensitivity measurements that motivate this work.

5 Results and Analysis

Table 3 summarizes our findings across model scales and editing settings. We categorize the evaluated LLMs into three tracks: High-Capacity, Efficient Mini, and Legacy. For each model, results are reported under three distinct instructions—*Neutral*, *Minimal-Edit*, and *Fluency-Edit*. Following standard practice and recommendations provided in the recent survey by Bryant et al. (2023), we report the $F_{0.5}$ score using the MaxMatch (M^2) scorer (Dahlmeier and Ng, 2012) for CoNLL-2014, the ERRANT scorer (Bryant et al., 2017) for BEA-2019, and we employ GLEU metric (Napoles et al., 2015) for JFLEG.

To benchmark our results, we include two comparative tiers: (I) Prior LLM Baselines, which present best findings from prior prompt-based LLM evaluations reported in the literature, and (II) single-system state-of-the-art (SOTA) task-specific models.

5.1 Comparative Evaluation

Minimal-Edit Mode. As shown in Table 3, the majority of the evaluated LLMs demonstrate a substantial performance leap over prior prompt-based studies under minimal-edit evaluation settings. In our experiments, **Claude-Sonnet-4.5** emerges as the top-performing model, achieving the best reported results among prompt-based LLM evaluation studies—presented in the middle part of the table—with an $F_{0.5}$ score of **67.05** on CoNLL-2014 and **64.91** on BEA-2019. Notably, these scores do not merely surpass previous prompt-based GEC studies; they redefine the expected zero-shot baseline for the task.

Results indicate a notable shift in the GEC zero-shot baseline across CoNLL-2014 and BEA-2019. On CoNLL-2014, all evaluated LLMs under the minimal-edit setting outperform prior prompt-based studies, regardless of shot count. Similarly, the majority of high-capacity models surpass previous BEA-2019 prompt-based baselines. While Mistral-Large-3 and Llama-4-Scout trail the 16-shot peak of 57.41, both exceed the zero-shot performance of 53.07 reported by Loem et al. (2023). This suggests that recent open-weight models have reached a level of inherent instruction-following capability where their zero-shot baselines can match or even exceed the few-shot results of earlier models.

A more recent prompt-based comparison point is APIO (Chernodub et al., 2025), which reports a BEA-2019 $F_{0.5}$ of 59.40 using GPT-4o with a validation-tuned, automatically optimized 10-instruction prompt. APIO is methodologically distinct from the few-shot baselines discussed above: rather than supplying in-context demonstrations, it iteratively refines the prompt itself on a validation set drawn from BEA-2019-Dev. Despite this active optimization, three of our four high-capacity models—Claude-Sonnet-4.5 (64.91), GPT-4.1 (64.70), and Llama-3.3-70B (61.61)—surpass APIO under the minimal-edit setting using fixed, manually authored prompts that receive no benchmark-specific tuning. Importantly, a direct comparison between automatic prompt optimization and fixed prompts is not possible from these results, as the underlying models differ. What these results do indicate is that careful zero-shot prompt design, when paired with contemporary high-capacity LLMs from a model class comparable to GPT-4o, surpasses APIO on BEA-2019.

However, when compared to previous single-system SOTA models, a distinct performance disparity emerges between the two benchmarks. While our top-performing LLM, Claude-Sonnet-4.5, achieves highly competitive results on CoNLL-2014 (67.05 $F_{0.5}$, trailing Liang et al. (2025)'s

Track	Model	CoNLL-2014			BEA-2019			JFLEG
		P	R	$F_{0.5}$	P	R	$F_{0.5}$	GLEU
High-Capacity	**Llama-3.3-70B**†							
	Neutral	59.45	60.18	59.60	50.14	70.13	53.17	63.68
	Minimal-edit	67.12	53.99	64.01	60.45	66.76	61.61	-
	Fluency-edit	-	-	-	-	-	-	62.04
	Mistral-Large-3†							
	Neutral	49.64	64.06	51.98	35.11	68.84	38.93	59.88
	Minimal-edit	60.15	60.25	60.17	50.46	70.57	53.51	-
	Fluency-edit	-	-	-	-	-	-	50.06
	GPT-4.1‡							
	Neutral	57.26	65.84	58.79	46.58	72.54	50.17	65.34
	Minimal-edit	70.37	52.73	65.96	63.75	68.77	64.70	-
	Fluency-edit	-	-	-	-	-	-	60.37
	Claude-Sonnet-4.5‡							
	Neutral	64.58	61.71	63.98	57.63	70.67	59.84	**66.09**
	Minimal-edit	69.14	59.82	**67.05**	63.48	71.33	**64.91**	-
	Fluency-edit	-	-	-	-	-	-	63.82
Efficient Mini	**Llama-4-Scout**†							
	Neutral	58.12	60.78	58.63	46.88	69.49	50.15	61.11
	Minimal-edit	63.06	58.17	62.02	53.78	69.11	56.28	-
	Fluency-edit	-	-	-	-	-	-	58.70
	GPT-5-Mini‡							
	Neutral	44.33	66.55	47.51	29.77	66.29	33.46	55.12
	Minimal-edit	62.52	61.34	62.28	54.97	70.52	57.51	-
	Fluency-edit	-	-	-	-	-	-	48.08
Legacy	**GPT-3.5-Turbo**‡							
	Neutral	65.05	53.96	62.48	54.70	65.97	56.63	64.67
	Minimal-edit	65.93	53.65	63.05	54.43	65.91	56.39	-
	Fluency-edit	-	-	-	-	-	-	63.99
Prior LLM Baselines								
Fang et al. (2023)	GPT-3.5-Turbo§	51.3^{3}	62.4^{3}	53.2^{3}	34.6^{1}	69.7^{1}	38.4^{1}	63.5^{3}
Coyne et al. (2023)	GPT-3.5 td-003*	-	-	-	-	-	49.66^{2}	63.40^{2}
Coyne et al. (2023)	GPT-4-0314	-	-	-	-	-	52.79^{2}	65.02^{2}
Loem et al. (2023)	GPT-3.5 td-003*	-	-	57.06^{16}	-	-	$\mathbf{57.41^{16}}$	$\mathbf{69.25^{64}}$
Davis et al. (2024)	Stable Beluga 2	-	-	57.2^{0}	-	-	-	61.3^{0}
Davis et al. (2024)	GPT-3.5-Turbo-0613	-	-	57.2^{1}	-	-	-	62.5^{4}
Omelianchuk et al. (2024)	GPT-4-0613	59.0^{0}	55.4^{0}	$\mathbf{58.2^{0}}$	-	-	-	-
Chernodub et al. (2025)	GPT-4o-2024-05-13	-	-	-	-	-	$\mathbf{59.40^{opt}}$	-
Single-System SOTA								
Stahlberg and Kumar (2021)	Seq2Seq Transformer	72.8	49.5	66.6	72.1	64.4	70.4	**64.7**
Liang et al. (2025)	Mistral-7b-EPO	76.71	52.56	**70.26**	78.16	68.07	75.91	-
Staruch et al. (2025)	Gemma 2 (27b) TS	77.38	47.88	68.89	82.28	67.03	**78.70**	62.42

Table 3: GEC performance across different model families under three editing modes. Superscripts denote the number of few-shot samples. †: Open-weight, ‡: Proprietary, §: Snapshot unspecified, td-003*: text-davinci-003, opt: Optimized prompt with induced instructions, not few-shot. Model names in the top part are abbreviated for clarity; full model identifiers can be found in Table 2.

70.26 by 3.21 points), a clear performance gap remains on BEA-2019. Even under the minimal-edit setting, the best zero-shot LLM performance (64.91) still trails behind the current empirical upper bound of 78.70 (Staruch et al., 2025), highlighting the continued necessity of task-specific solutions for high-complexity GEC datasets.

This performance gap—rather than indicating a fundamental lack of GEC capability in LLMs—highlights the effectiveness of domain-specific training and benchmark-centric optimization. Specialized models benefit from exposure to the specific correction styles and error distributions inherent in the benchmark training sets. This exposure allows them to internalize benchmark-specific repair patterns—patterns that are not accessible

to general-purpose LLMs in a zero-shot setting, where the only available signal is the instructional framing and task constraints provided in the prompt.

Fluency-edit Mode. The results on the JFLEG benchmark reveal a counter-intuitive relationship between fluency constraint and model performance. As shown in Table 3, **Claude-Sonnet-4.5** again emerges as the leader among the evaluated models, achieving a peak GLEU score of **66.09** under the neutral setting. Interestingly, across all model tracks, the fluency-edit instruction leads to a slight but consistent performance degradation compared to the neutral prompt. This observation suggests a potential overshooting effect: when explicitly tasked with maximizing fluency while maintaining adherence to grammatical correctness, LLMs can be driven toward excessive paraphrasing that diverges from the gold-standard references. A discussion of this phenomenon is provided in Appendix A.

Claude-Sonnet-4.5, GPT-4.1, and GPT-3.5-Turbo surpass prior zero-shot baselines. Although trailing the 64-shot peak (69.25 GLEU) of Loem et al. (2023), our zero-shot top result (66.09) exceeds their zero-shot baseline of 64.51. Notably, GPT-3.5-Turbo outperforms Efficient Mini models, suggesting smaller contemporary architectures still trail their larger predecessors on this benchmark.

Compared to previous SOTA results, the fluency performance of most of the LLMs is remarkably competitive. Claude-Sonnet-4.5 and GPT-4.1 both surpass the specialized model baseline of 64.7 reported by Stahlberg and Kumar (2021). Most of the high-capacity models and even the legacy GPT-3.5-Turbo outperform the recent task-specific model (Staruch et al., 2025) on this benchmark. This suggests that for fluency-centric GEC, the broad linguistic knowledge of general-purpose LLMs is now reaching a level where they can exceed the performance of models specifically tuned for error correction, as long as the prompt maintains a balance between fluency correction and original structure.

To further investigate the linguistic drivers behind the consistent performance decline observed in the Fluency-Edit setting, we conduct a qualitative case study on a representative subset of models. This analysis reveals that the lower scores often stem from a metric–objective mismatch rather than a failure in instruction adherence. While the Fluency-Edit prompt successfully triggers more naturalizing, global paraphrastic reformulations, these high-quality enhancements are frequently penalized for their structural divergence from the rigid n-gram boundaries of the gold references. We provide a granular discussion of these behavioral shifts and model-specific failure cases in Appendix A.

5.2 Prompt Quality through Legacy Benchmarking

To assess the effectiveness of instructions used in this study, we conduct a comparative analysis using GPT-3.5-Turbo as a reference point against prior prompt-based studies. We acknowledge that an exact one-to-one comparison is hindered by the temporal nature of proprietary model snapshots; while earlier studies utilized versions such as gpt-3.5-turbo-0613 or text-davinci-003, our experiments employ the 0125 snapshot, which was the most recent available version at the time of this study.

As shown in Table 3, our minimal-edit prompt yields a substantial performance gain on GPT-3.5-Turbo compared to previous scores reported for GPT-3.5 family. On CoNLL-2014, our zero-shot configuration achieves an $F_{0.5}$ of 63.05, which not only beats the zero-shot baselines but also exceeds every few-shot score recorded in prompt-based GEC studies to date. Remarkably, this performance even surpasses the results reported for subsequent and more advanced iterations; for instance, our zero-shot GPT-3.5-Turbo score significantly outperforms the 58.2 $F_{0.5}$ achieved by GPT-4-0613 in Omelianchuk et al. (2024). This trend is even more pronounced on BEA-2019, where it reaches 56.63 and 56.39 $F_{0.5}$ in neutral and minimal-edit settings respectively, nearly rivaling the 16-shot performance (57.41) of Loem et al. (2023) and outperforming Fang et al. (2023)'s 38.4 and even Coyne et al. (2023)'s 52.79 with GPT-4.

Regarding the evaluation on JFLEG, GPT-3.5-Turbo GLEU results for both neutral (64.67) and fluency-edit (63.99) settings outperforms all GPT-3.5 family zero-shot and low-shot baselines reported in prior work (Fang et al., 2023; Coyne et al., 2023; Davis et al., 2024). Although Loem et al. (2023) achieves a higher score with an extensive 64-shot setup (69.25), our zero-shot performance for GPT-3.5-Turbo surpasses their zero-shot baseline of 64.51.

Collectively, these results demonstrate that the primary driver of performance in prompt-based GEC is not merely the raw parameter count of the

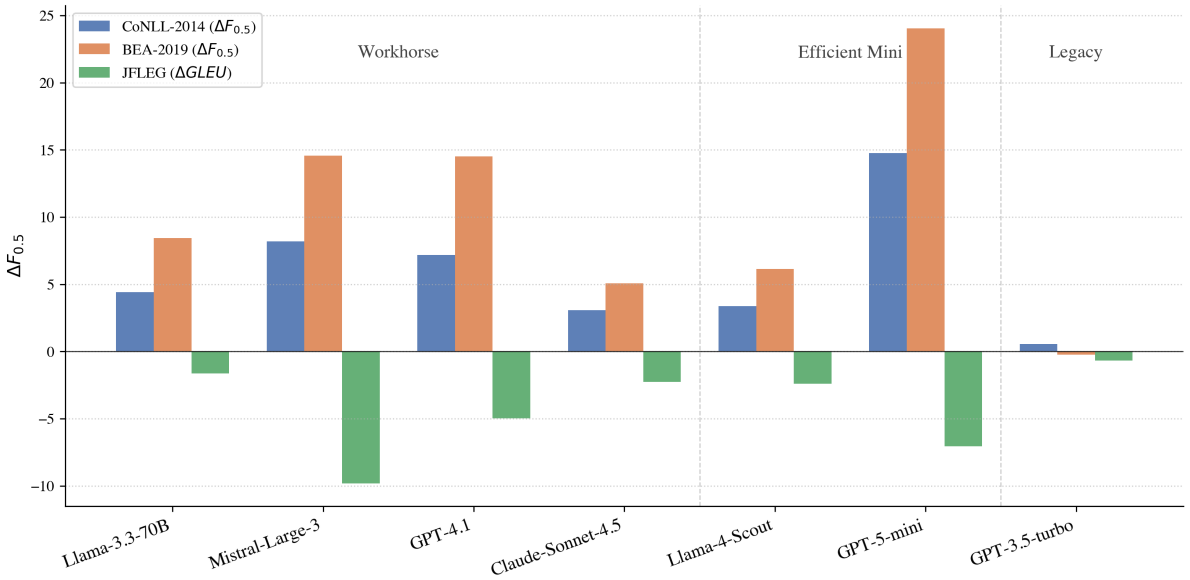

Figure 1: Delta changes in performance when switching from the Neutral baseline to Minimal-Edit or Fluency-Edit

underlying model, but the clarity of the task framing and instructional specificity of the prompts. We argue that the consistent superiority of our prompts—even over significantly larger models like GPT-4—stems from their ability to navigate the correction-preservation trade-off with higher sensitivity. In the minimal-edit configuration, our instructions impose rigorous decision boundaries that successfully inhibit the over-editing bias common in LLMs, allowing the model to prioritize precision over aggressive rewriting. Conversely, in the fluency setting, score divergence reflects a shift toward paraphrastic reformulation rather than instructional failure. This overshooting effect triggers a metric–objective mismatch, where idiomatic enhancements are penalized by rigid n-gram boundaries of the metric (Appendix A).

5.3 Instructional Sensitivity and P-R Dynamics

Editing mode transitions reveal distinct behavioral shifts. $F_{0.5}$ score deltas (Figure 1) highlight P-R dynamics, offering insights into a model's instructional sensitivity. Given that $F_{0.5}$ weights precision twice as much as recall, these performance shifts (deltas) reflect models' capacity to refine their output under restrictive constraints.

Minimal-Edit instruction consistently improves $F_{0.5}$ scores over the Neutral baseline, a gain primarily driven by sharp increases in Precision as can be seen in Table 3. By explicitly requiring models to preserve original structure and apply minimal changes, we effectively mitigate the common zero-shot pitfall of over-editing. A notable exception is the legacy GPT-3.5-Turbo. It exhibits behavioral rigidity, with minimal performance deltas indicating a limited response to varying instructional objectives and constraints. This implies that the model is reaching a performance ceiling under the neutral setting and is leaving little headroom for further optimization through explicit instructional objectives.

One of the important trends observed in Table 3 is the significant boost in Precision at the expense of Recall when switching to the Minimal-Edit mode. GPT-4.1 serves as a textbook example of this dynamic on CoNLL-2014, where its Precision jumps from 57.26 to 70.37 (+13.11), while its Recall undergoes an equivalent drop from 65.84 to 52.73 (−13.11). This suggests that for certain models, the Minimal-Edit instruction acts primarily as a conservative filter, successfully suppressing the inherent tendency to perform excessive rewrites. However, for Claude-Sonnet-4.5, GPT-5-mini, and Mistral-Large-3, the Minimal-Edit instruction effectively circumvents this traditional P-R trade-off on the BEA-2019 dataset. Unlike the typical inverse relationship, these models exhibit simultaneous gains in both metrics. This phenomenon indicates that the instruction does not merely impose a restrictive bottleneck; rather, it aligns the model's generative priors with given GEC-specific objectives. By providing a clearer operational boundary, the prompt enables these models to target actual

errors more accurately while inherently reducing the noise generated by non-corrective rewriting.

Claude-Sonnet-4.5 exhibits a unique performance profile, maintaining the highest $F_{0.5}$ scores with significantly narrower deltas than its peers (Figure 1). Unlike other models that require explicit minimal-edit constraints to mitigate over-editing, Claude's strong neutral performance suggests an intrinsic minimalist prior. This out-of-the-box maturity indicates the model is relatively better aligned for conservative GEC, bypassing the need for the explicit guidance required by other models.

5.4 Error Type Analysis

The aggregate $F_{0.5}$ shifts reported in Section 5.3 characterize *how much* model behavior changes between editing modes, but they leave the precision-recall mechanics underspecified at the linguistic level. To address this, we conduct an ERRANT-based error-type breakdown for our top-performing model, Claude-Sonnet-4.5, on the BEA-2019 test set under both neutral and minimal-edit instructions. Table 5 (Appendix B) reports precision, recall, and $F_{0.5}$ for the 15 most frequent error categories, which jointly cover approximately 77.27% of the gold-reference edits enumerated by ERRANT for this evaluation.

The breakdown reveals that the aggregate minimal-edit benefit decomposes into three distinct behavioral patterns: rewrite suppression in open-ended substitution categories such as R:VERB and R:OTHER; simultaneous precision and recall gains in rule-governed morphosyntactic categories such as R:MORPH and R:VERB:SVA; and minimal change in already-saturated categories such as R:SPELL. The full breakdown and accompanying analysis are presented in Appendix B.

6 Conclusion

In this study, we evaluated six contemporary LLMs to analyze their performance in GEC. We tested these models under three different editing modes using *Neutral*, *Minimal-Edit*, and *Fluency-Edit* instructions to analyze their responses.

Claude-Sonnet-4.5 set a new performance record for prompt-based GEC, with its zero-shot results outperforming even previously reported few-shot scores. Using a minimal-edit instruction helps improve precision and overall $F_{0.5}$ scores across various model scales. This approach reduces the tendency of models to edit sentences more than necessary. Furthermore, we found that switching to the Fluency-Edit instruction led to a consistent drop in GLEU scores. This is because models perform rewrites that are mostly correct but do not match the exact words in human references, leading to penalties from the n-gram-based metric. Our results show a clear difference in how models follow instructions. The legacy GPT-3.5-Turbo is markedly less sensitive to editing modes, exhibiting consistently smaller *neutral-anchored* shifts than contemporary LLMs. We argue that the design of the prompt is as critical as the size of the model for effective grammar correction. While general LLMs are becoming competitive with specialized systems, future evaluation methods must account for the fact that GLEU may not accurately capture the quality of fluency improvements.

7 Limitations

We acknowledge some constraints in our study that should be considered. First, while we attribute the performance drop in Fluency-Edit mode to a metric-objective mismatch, this claim remains to be validated through large-scale human evaluation to confirm if LLM-generated rewrites are preferred over gold standards. Second, our experimental design intentionally restricts evaluation to a zero-shot setting in order to isolate instructional sensitivity from in-context example effects. Although few-shot prompting is known to improve absolute performance on GEC, augmenting our best configuration with in-context demonstrations was beyond the scope of this study and is left as a complementary direction for the broader research community. Third, recent work shows that LLM behavior can be sensitive to prompt formatting variations (e.g., delimiter choice, instruction ordering, or layout conventions) (Sclar et al., 2024); our study, which adopts a single fixed structural template, does not address this dimension, and the format-level robustness of editing-mode shifts remains an open empirical question. Fourth, our error-type analysis characterizes instructional sensitivity within zero-shot settings but does not provide a direct head-to-head comparison with fine-tuned specialized systems at the category level. Such a comparison is currently constrained by the absence of category-level breakdowns in published fine-tuned baselines and remains unexplored. Finally, our English-only focus limits generalizability to morphologically rich or low-resource languages.

References

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

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

Christopher Bryant, Zheng Yuan, Muhammad Reza Qorib, Hannan Cao, Hwee Tou Ng, and Ted Briscoe. 2023. Grammatical error correction: A survey of the state of the art. *Computational Linguistics*, 49(3):643–701.

Artem Chernodub, Aman Saini, Yejin Huh, Vivek Kulkarni, and Vipul Raheja. 2025. APIO: Automatic prompt induction and optimization for grammatical error correction and text simplification. In *Proceedings of the 15th International Conference on Recent Advances in Natural Language Processing - Natural Language Processing in the Generative AI Era*, pages 234–239, Varna, Bulgaria. INCOMA Ltd., Shoumen, Bulgaria.

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. *Preprint*, arXiv:2303.14342.

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

Christopher Davis, Andrew Caines, 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*, pages 11952–11967, Bangkok, Thailand. Association for Computational Linguistics.

Tao Fang, Shu Yang, Kaixin Lan, Derek F. Wong, Jinpeng Hu, Lidia S. Chao, and Yue Zhang. 2023. Is ChatGPT a Highly Fluent Grammatical Error Correction System? A Comprehensive Evaluation. *arXiv preprint*.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.

Jiehao Liang, Haihui Yang, Shiping Gao, and Xiaojun Quan. 2025. Edit-wise preference optimization for grammatical error correction. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 3401–3414, Abu Dhabi, UAE. Association for Computational Linguistics.

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. 4. 07/2023 (BEA 2023).

Roy Lyster and Leila Ranta. 1997. Corrective feedback and learner uptake: Negotiation of form in communicative classrooms. *Studies in second language acquisition*, 19(1):37–66.

Arianna Masciolini, Andrew Caines, Orphée De Clercq, Joni Kruijsbergen, Murathan Kurfalı, Ricardo Muñoz Sánchez, Elena Volodina, and Robert Östling. 2025. The MultiGEC-2025 shared task on multilingual grammatical error correction at NLP4CALL. In *Proceedings of the 14th Workshop on Natural Language Processing for Computer Assisted Language Learning*, pages 1–33, Tallinn, Estonia. University of Tartu Library.

Courtney Napoles, Keisuke Sakaguchi, Matt Post, and Joel Tetreault. 2015. Ground truth for grammatical error correction metrics. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 588–593, Beijing, China. Association for Computational Linguistics.

Courtney Napoles, Keisuke Sakaguchi, and Joel Tetreault. 2017. JFLEG: A fluency corpus and benchmark for grammatical error correction. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 229–234, Valencia, Spain. Association for Computational Linguistics.

Hwee Tou Ng, Siew Mei Wu, Ted Briscoe, Christian Hadiwinoto, Raymond Hendy Susanto, and Christopher Bryant. 2014. The CoNLL-2014 shared task on grammatical error correction. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task*, pages 1–14, Baltimore, Maryland. Association for Computational Linguistics.

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.

Kostiantyn Omelianchuk, Andrii Liubonko, Oleksandr Skurzhanskyi, Artem Chernodub, Oleksandr Korniienko, and Igor Samokhin. 2024. Pillars of grammatical error correction: Comprehensive inspection of contemporary approaches in the era of large language models. In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, pages 17–33, Mexico City, Mexico. Association for Computational Linguistics.

Robert Östling, Katarina Gillholm, Murathan Kurfalı, Marie Mattson, and Mats Wirén. 2024. Evaluation of really good grammatical error correction. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 6582–6593, Torino, Italia. ELRA and ICCL.

Melanie Sclar, Yejin Choi, Yulia Tsvetkov, and Alane Suhr. 2024. Quantifying language models'sensitivity to spurious features in prompt design or: How i learned to start worrying about prompt formatting. In *International Conference on Learning Representations*, volume 2024, pages 25055–25083.

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, Online. Association for Computational Linguistics.

Ryszard Staruch, Filip Gralinski, and Daniel Dzienisiewicz. 2025. Adapting LLMs for minimal-edit grammatical error correction. In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 118–128, Vienna, Austria. Association for Computational Linguistics.

David Wood, Jerome S Bruner, and Gail Ross. 1976. The role of tutoring in problem solving. *Journal of child psychology and psychiatry*, 17(2):89–100.

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

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.

A Qualitative Analysis

While the quantitative results provide an empirical foundation, they do not fully capture the nuanced linguistic modifications of the selected LLMs. In particular, the Fluency-Edit instruction on the JFLEG benchmark yields a consistent performance decline as presented in Table 3. To clarify this discrepancy, we examine the relationship between model outputs and metric responses through a representative case study (Table 4).

For this analysis, we select three models that represent distinct behavioral profiles based on the quantitative results: Claude-Sonnet-4.5 as the overall performance leader, Mistral-Large-3 for its high instructional sensitivity (max Δ), and GPT-3.5-Turbo as a representative of behavioral rigidity (min Δ).

Table 4 provides a representative illustration of the broader quantitative trends discussed earlier, serving as a clear instance of the model-specific behavioral patterns identified in previous subsections. For each model, the table displays the generated outputs under both Neutral and Fluency-Edit instructions, alongside their respective sentence-level GLEU scores which are evaluated for each hypothesis against the set of four gold references.

Claude-Sonnet-4.5 is particularly informative because it is our strongest overall model, and its outputs reveal a subtle but important distinction between reference proximity and perceived fluency. Under the Neutral setting, it corrects the clear grammatical error *didn't expected* to *didn't expect* and also repairs *were flock to* to *were flocking to*, but otherwise remains highly conservative. Notably, its output remains very close to Ref3, but the omission of the infinitival marker *to* after *expect* leaves the *didn't expect* construction slightly incomplete and somewhat less natural from a native-speaker perspective. Despite this residual awkwardness, the output receives a relatively strong sentence-level GLEU score (0.69), which is consistent with the metric's sensitivity to n-gram overlap across the available references rather than to fine-grained judgments of naturalness. Under the Fluency-Edit

Original Sentence:
As a result, she could get a job that she badly wanted but she didn't expected because many people were flock to this company.

Gold References:
Ref0: As a result, she was able to get a job that she badly wanted but didn't expect because so many people were flocking to this company.
Ref1: As a result, she could get a job that she badly wanted but she didn't expect to get because many people flocked to this company.
Ref2: As a result, she could get a job that she badly wanted but didn't expect to get because many people were flocking to this company.
Ref3: As a result, she could get a job that she badly wanted but she didn't expect to because many people were flocking to this company.

Model	Mode	Model Output	GLEU
Claude-4.5	Neutral	As a result, she could get a job that she badly wanted but she didn't expect because many people were flocking to this company.	0.69
	Fluency	As a result, she got a job that she badly wanted but didn't expect because many people were flocking to this company.	0.62
Mistral-L3	Neutral	As a result, she could get a job that she badly wanted but didn't expect because many people were flocking to this company.	0.64
	Fluency	As a result, she got the job she badly wanted, though she hadn't expected it because so many people were flocking to the company.	0.29
GPT-3.5	Neutral	As a result, she could get a job that she badly wanted but she didn't expect because many people were flocking to this company.	0.69
	Fluency	As a result, she could get a job that she badly wanted but she didn't expect because many people were flocking to this company.	0.69

Table 4: Comparison of outputs of the models for the Sentence ID 217 in the JFLEG test set with sentence level GLEU scores.

setting, it makes a more ambitious change by replacing *could get* with *got*, which moves the sentence toward a more natural and direct formulation. However, it still leaves the construction *but didn't expect* unresolved, rather than producing the more complete and idiomatic *didn't expect to* or *didn't expect to get it*. In other words, the fluency-oriented instruction does trigger a more naturalizing edit, but it does not fully repair the most awkward local structure. This makes Claude's pair of outputs especially useful for illustrating that higher fluency intent does not automatically translate into a uniformly better correction, while higher GLEU under the neutral prompt may partly reflect reference closeness rather than superior overall well-formedness.

Mistral-Large-3 exhibits an even clearer contrast between conservative correction and fluency-driven editing. Under the Neutral setting, its output is nearly identical to Claude's neutral response: it corrects the obvious grammatical errors but preserves most of the original sentence structure, yielding a relatively strong GLEU score (0.64). This suggests that, even without explicit editing guidance, the model already performs a balanced correction that lies somewhere between minimal editing and moderate fluency improvement. Under the Fluency-Edit setting, however, it shifts much more decisively into reformulation: *could get a job* becomes *got the job*, *but* becomes *though*, *didn't expect* becomes *hadn't expected it*, *many people were flocking to this company* becomes *so many people were flocking to the company*. These revisions make the output more globally rewritten and, at first glance, more polished. At the same time, they substantially increase lexical and structural distance from the references, which likely explains the sharp drop in sentence-level GLEU (0.29). More broadly, this example is consistent with the general qualitative findings: among the evaluated models, Mistral-Large-3 exhibits the largest behavioral shift, making it the clearest case of high instructional sensitivity. Accordingly, the drop in GLEU reflects not only the behavior of the metric, but also a broader change in the model's response to the Fluency-Edit prompt. This is partly consistent with a metric–objective mismatch, insofar as the prompt encourages native-like reformulation while the metric still rewards proximity to a limited reference set. At the same time, the example indicates that the lower score cannot be explained by the metric alone. Some of Mistral's changes are plausibly motivated by fluency, but they also extend the revision beyond targeted local repair. Overall, the Fluency-Edit

prompt shifts the model from balanced correction toward broader paraphrastic rewriting, reducing reference alignment without delivering a clearly proportional gain in local correction quality.

GPT-3.5-Turbo shows a different pattern, remaining unchanged across prompts. Its Neutral and Fluency-Edit outputs are identical, indicating that the fluency instruction has little or no effect on its correction behavior in this example. This is also in line with the general quantitative results: among the evaluated models, GPT-3.5-Turbo shows the smallest change, consistent with its overall profile of low instructional sensitivity. At the sentence level, this low sensitivity is reflected in a restrained editing strategy: the model fixes the obvious grammatical errors but leaves the larger phrasing intact, including the somewhat incomplete *she didn't expect* segment. This yields a relatively high GLEU score (0.69), likely because the output remains reference-close and avoids broader paraphrastic divergence. One interpretation is that GPT-3.5, as a legacy model, is comparatively rigid in following higher-level stylistic instructions such as *make it more fluent* or *more native-like*, especially when these instructions compete with a simpler default strategy of surface-level error correction. This contrasts with Mistral's behavior: whereas Mistral shifts toward more aggressive rewriting under the fluency-edit prompt, GPT-3.5 remains largely unchanged in how it edits. In this sense, the example also provides a concrete instance of the broader behavioral rigidity observed in the quantitative results.

B Error Type Analysis

This appendix complements the Error Type Analysis introduced in Section 5.4 by presenting the full ERRANT-based breakdown and a regime-by-regime discussion of the observed shifts. Table 5 presents a per-category breakdown for the 15 most frequent ERRANT categories, reporting their gold-edit frequency, precision, recall, and $F_{0.5}$ under neutral and minimal-edit instructions, together with the $\Delta F_{0.5}$ between the two modes.

Of the 15 categories examined, 14 exhibit a positive $\Delta F_{0.5}$ when switching from neutral to minimal-edit, with a mean gain of $+2.95$ points. Crucially, this improvement is not driven by a uniform precision boost across the board: while the average precision increases by $+3.70$ points, the average recall stays nearly flat (-0.13). The category-level distribution, however, hides three distinct behavioral profiles that together explain the P-R dynamics observed in Section 5.3.

The largest $F_{0.5}$ gains concentrate in open-ended substitution categories—R:VERB, R:PREP, and R:OTHER—which often involve multiple plausible alternatives or contextually optional changes, and where the minimal-edit prompt produces large precision improvements alongside stable or declining recall, indicating that a substantial portion of the neutral baseline's edits in these categories are filtered out under the minimal-edit constraint. R:VERB ($\Delta F_{0.5}$=+8.68) is the clearest case: precision jumps by $+13.75$ points while recall drops by -7.87, indicating that the minimal-edit instruction sharply suppresses non-mandatory verb substitutions. R:OTHER ($\Delta F_{0.5}$=+5.97; ΔP=+7.35, ΔR=-1.52) and R:PREP ($\Delta F_{0.5}$=+5.25; ΔP=+6.53, ΔR=-1.03) follow a similar pattern. This pattern provides a category-level localization of the "over-editing" tendency identified in prior prompt-based GEC studies (Wu et al., 2023; Fang et al., 2023), and our results indicate that the minimal-edit mode substantially suppresses it in these categories.

A second, distinct group of categories shows simultaneous gains in both precision and recall. R:MORPH (ΔP=+9.14, ΔR=+1.77), R:VERB:FORM (ΔP=+2.98, ΔR=+2.69), R:VERB:SVA (ΔP=+2.40, ΔR=+1.86), and R:NOUN:NUM (ΔP=+3.33, ΔR=+1.39) all improve on both axes. These categories share a common property: they are governed by relatively deterministic grammatical rules—subject-verb agreement, morphological inflection, number marking—rather than admitting multiple plausible alternatives. For such categories, the minimal-edit instruction yields gains beyond what a purely restrictive filter would predict. A filter operating on neutral outputs could remove false positives and raise precision, but it could not introduce true positives that the neutral run failed to produce. The simultaneous improvement therefore suggests that the prompt influences the correction process itself rather than acting only as a post-hoc constraint, consistent with the observation in Section 5.3 that in top-tier models the minimal-edit prompt may not simply impose a restrictive bottleneck.

A third pattern emerges in error categories where neutral performance is already near ceiling, leaving little headroom for the minimal-edit instruc-

Error Type	Freq.	Neutral			Minimal-Edit			$\Delta F_{0.5}$
		P	R	$F_{0.5}$	P	R	$F_{0.5}$	
M:PUNCT	761	64.56	61.98	64.03	64.90	66.10	65.14	+1.11
R:OTHER	629	34.43	44.76	36.10	41.78	43.24	42.07	+5.97
R:ORTH	464	56.36	68.45	58.42	57.29	71.12	59.61	+1.19
R:SPELL	343	87.79	96.85	89.47	88.36	97.38	90.03	+0.56
R:PREP	340	65.86	76.62	67.76	72.39	75.59	73.01	+5.25
R:NOUN:NUM	313	74.26	87.11	76.52	77.59	88.50	79.55	+3.03
M:DET	306	78.75	75.08	77.98	79.80	77.45	79.32	+1.34
U:DET	257	71.09	83.68	73.30	74.63	77.82	75.24	+1.94
R:VERB:TENSE	223	64.00	68.97	64.94	67.56	68.16	67.68	+2.74
R:VERB	214	52.71	57.87	53.67	66.46	50.00	62.35	+8.68
R:VERB:FORM	200	76.11	84.31	77.62	79.09	87.00	80.56	+2.94
R:MORPH	193	64.02	85.79	67.44	73.16	87.56	75.65	+8.21
R:VERB:SVA	167	75.38	90.36	77.96	77.78	92.22	80.29	+2.33
R:PUNCT	161	36.19	46.34	37.85	32.92	49.69	35.30	−2.55
R:DET	134	67.11	71.33	67.91	70.40	65.67	69.40	+1.49

Table 5: ERRANT error-type breakdown for Claude-Sonnet-4.5 on BEA-2019 under Neutral and Minimal-Edit instructions. We report the 15 most frequent error categories, which together account for approximately 77.27% of all gold edits in the test set. Frequency is computed as $\mathrm{TP} + \mathrm{FN}$ from the Minimal-Edit run; the Neutral counterpart yields a near-identical ranking, with minor count differences arising from ERRANT's hypothesis-conditioned alignment. $\Delta F_{0.5}$ denotes the absolute change in $F_{0.5}$ from the Neutral to the Minimal-Edit instruction.

tion to act on. R:SPELL improves only marginally ($\Delta F_{0.5}$=+0.56) because it operates close to a saturation point with both precision (87.79%) and recall (96.85%) already very high under neutral mode. R:ORTH (Δ=+1.19) and M:PUNCT (Δ=+1.11) behave similarly. The instructional framing has limited influence on phenomena whose correction is largely deterministic and where model behavior is already well-calibrated.

R:PUNCT is the only category in the top-15 that shows a negative $\Delta F_{0.5}$ (-2.55), driven by a precision drop (-3.27) accompanied by a recall gain ($+3.35$). This pattern runs counter to the dominant trend and likely reflects a category-specific interaction. Under the minimal-edit mode, the model attempts more punctuation replacements, but its replacement choices are not sufficiently selective. Replacement punctuation is intrinsically context-dependent, with multiple plausible alternatives often available for the same source position. The observed precision drop may reflect the difficulty of aligning model choices with the specific replacements chosen by reference annotators, though we cannot rule out a genuine selectivity issue without examining individual cases.