# How Far Can Prompting Go for Minimal-Edit Ukrainian Grammatical Error Correction?

**Kateryna Karpo**[υ,σ] **Artem Chernodub**[ζ]
[υ]Ukrainian Catholic University [σ]YouScan [ζ]Zendesk

## Abstract

Fine-tuned Large Language Models (LLMs) dominate in Ukrainian grammatical error correction (GEC), while API-accessed LLMs remain nearly untested on minimal-edit benchmarks. We evaluate 11 commercial LLMs from four providers and one open-source Ukrainian model on the UNLP 2023 Shared Task benchmark, GEC-only track, comparing zero-shot, few-shot, minimal-edits, and LLM-assisted prompt optimization strategies. Our best configuration (Gemini 3.1-Pro) reaches $F_{0.5}$ = 69.22, closing over 90% of the gap to fine-tuned SOTA ($F_{0.5}$ = 73.14). For zero-shot prompts, only Claude models benefit from Ukrainian instructions. However, the best overall results for all models use Ukrainian minimal-edits prompts, whose language-specific rules require Ukrainian to express precisely. LLM-assisted prompt optimization on top of minimal-edits + few-shot achieves the highest score. Detailed minimal-edits instructions yield the largest gains for punctuation and case errors but cause the model to abandon several low-frequency categories. Delving into error analysis, we identify five recurring overcorrection patterns tied to Ukrainian-specific linguistic phenomena. Code, prompts, and outputs are publicly available.[1][2][3]

## 1 Introduction and Related Work

Grammatical error correction (GEC) systems operate under two paradigms (Bryant et al., 2023). Minimal-edit correction targets only clear grammatical, spelling, and punctuation errors, preserving the author's wording. Fluency-oriented correction additionally permits lexical substitutions, syntactic restructuring, and stylistic improvements. The minimal-edit setting is especially relevant for educational tools, where feedback should pinpoint errors rather than rewrite learner text, and for writing assistants that must preserve authorial voice.

For English, a high-resource language with decades of GEC research, this distinction is well established. Minimal-edit evaluation is the standard in shared tasks such as CoNLL-2014 (Ng et al., 2014) and BEA-2019 (Bryant et al., 2019), while JFLEG (Napoles et al., 2017) targets fluency. Staruch et al. (2025) recently achieved state-of-the-art single-model minimal-edit results on BEA-2019 by adapting a decoder-only LLM. The MultiGEC-2025 shared task (Masciolini et al., 2025) extended the two-track paradigm to twelve European languages, confirming it as a cross-lingual standard.

For Ukrainian, GEC infrastructure has only recently begun to emerge. The UNLP 2023 Shared Task (Syvokon and Romanyshyn, 2023) introduced the first benchmark with two parallel tracks: GEC-only (minimal-edit) and GEC+Fluency, both evaluated with span-based $F_{0.5}$. Since then, research has shifted toward fluency (Saini et al., 2024), with Luhtaru et al. (2024) pushing GEC+Fluency SOTA to $F_{0.5}$ = 74.09 with a fine-tuned Llama 2 model, surpassing the original winner ($F_{0.5}$ = 68.17; Bondarenko et al., 2023). The GEC-only track, however, has seen no new results. Ukrainian was also included in MultiGEC-2025, where the winning team's fine-tuned Gemma 2 scored GLEU = 79.55 on minimal edits vs. GLEU = 68.03 for a one-shot Llama 3.1 baseline. Most recently, Kovalchuk et al. (2025) introduced silver-standard GEC corpora for multiple languages including Ukrainian and fine-tuned multilingual models on them; however, their work centers on training data creation and finetuning rather than prompting strategies for API-accessed LLMs.

To the best of our knowledge, most of Ukrainian GEC systems available to date rely on fine-tuned models that require dedicated GPU infras-

[1]Correspondence: a.chernodub@gmail.com
[2]https://github.com/katerynkarpo/gec_unlp_2026
[3]This work was conducted as part of Kateryna Karpo's M.Sc. thesis at the Ukrainian Catholic University, Faculty of Applied Sciences.



*Proceedings of the Fifth Ukrainian Natural Language Processing Conference (UNLP 2026)*, pages 136–154

tructure. Commercial API-accessed LLMs offer a lightweight alternative, yet remain nearly untested for Ukrainian minimal-edit GEC. The only published result is from Katinskaia and Yangarber (2024), who evaluated GPT-3.5 (specifically, `gpt-3.5-turbo-0613`) in a zero-shot setting on the UNLP 2023 Shared Task test set, GEC-only track (henceforth UNLP-2023-test (GEC only)), and obtained $F_{0.5} = 27.4$, far below the fine-tuned SOTA of 73.14.

This paper is the first to test newer API-accessed models on this benchmark and to explore whether better prompting strategies can close the gap with fine-tuned systems.

## 2 Experimental Setup

**Data.** We use the GEC-only track of the UNLP 2023 Shared Task (Syvokon and Romanyshyn, 2023), which is built on the UA-GEC corpus. We adopt UA-GEC's own train and valid splits as our training and validation sets (31,038 and 1,422 sentences) and report all final numbers on the shared task's test set (1,274 sentences), whose gold annotations are held out from participants and never inspected during prompt development. We use the training and validation sets for prompt development (few-shot exemplar selection and prompt engineering) and report only on the test set.

**Models.** We evaluate commercial, API-accessed LLMs from four providers and one open-source Ukrainian model, Lapa v0.1.2 (Paniv et al., 2025).[4] We report exact snapshot identifiers for reproducibility. From OpenAI, we use GPT-4.1 (`gpt-4.1-2025-04-14`), GPT-4.1-mini (`gpt-4.1-mini-2025-04-14`), GPT-5.1 (`gpt-5.1-2025-11-13`), GPT-5.2 (`gpt-5.2-2025-12-11`), and GPT-5.4 (`gpt-5.4-2026-03-05`). From Moonshot, we use Kimi-K2 (`kimi-k2-0905-preview`; `0905` denotes a dated preview build). The Google and Anthropic APIs do not expose dated snapshot identifiers; we use Gemini 3-Flash (`gemini-3-flash-preview`), Gemini 3-Pro (`gemini-3-pro-preview`), Gemini 3.1-Pro (`gemini-3.1-pro-preview`), Claude Sonnet 4.6 (`claude-sonnet-4.6`), and Claude Opus 4.6 (`claude-opus-4.6`).[5]

[4]Provider documentation: OpenAI `https://platform.openai.com`; Anthropic `https://docs.anthropic.com`; Google `https://ai.google.dev`; Moonshot `https://platform.moonshot.ai`.

[5]Inference-time parameters differ: GPT-4.1 and Kimi use temperature/top-$p$; Claude and Gemini use either temperature or a reasoning effort budget; GPT-5.x uses an effort level (`low`/`medium`/`high`). We set temperature 0 where available, default effort for Claude and Gemini, and `medium` for GPT-5.x.

### 2.1 Research Questions

We address the following four research questions:

**RQ1: What is the zero-shot minimal-edit GEC performance of current LLMs on Ukrainian relative to fine-tuned SOTA, and how sensitive is it to prompt language?** We systematically compare 2025–2026 commercial LLMs on UNLP-2023-test (GEC only) against the fine-tuned SOTA of $F_{0.5} = 73.14$, and test both English and Ukrainian prompt variants to assess whether instruction language affects correction quality for a morphologically rich, low-resource language. To the best of our knowledge, the only published LLM baseline for Ukrainian GEC is the GPT-3.5 zero-shot result (English) from Katinskaia and Yangarber (2024), which we include for reference.

**RQ2: Can prompting strategies reduce overcorrection compared to zero-shot baselines?** We evaluate how each of the four prompting strategies affects the precision–recall trade-off: (1) zero-shot, (2) few-shot, (3) minimal-edits + zero-shot, and (4) minimal-edits + few-shot.

**RQ3: Can LLM-assisted prompt optimization improve over manually crafted prompts?** We apply an LLM-assisted prompt optimization pipeline built on Claude Code skills, an agentic system that iteratively generates, evaluates, and refines GEC prompts using the full evaluation loop as feedback.

**RQ4: Where do minimal-edits instructions help and where do they fail?** We compare per-error-type performance between a standard zero-shot prompt and our best optimized prompt using ERRANT category breakdowns, identifying which error types benefit most from detailed minimal-edits instructions and which remain resistant to prompt-based improvement.

**Prompting strategies.** We compare four manually engineered prompting configurations that vary in prompt detail (general vs. minimal-edits) and use of examples (zero-shot vs. few-shot).

1. **Zero-shot** (A.1): a general system prompt that instructs the model to correct grammatical and spelling errors and return the original sentence if no errors are found. No examples are provided.

2. **Few-shot** (A.2): the zero-shot prompt augmented with source–target correction pairs from the training set, covering spelling, punctuation, and morphological errors as well as already-correct sentences.

3. **Minimal-edits + zero-shot** (A.3): a detailed system prompt enumerating which error types to correct, Ukrainian-specific conventions (e.g., dash vs. hyphen in dialogue, у/в 'u/v' alternation, vocative case in forms of address, etc.), and categories of changes to avoid. No examples are provided.

4. **Minimal-edits + few-shot** (A.4): combines the detailed minimal-edits system prompt with few-shot correction examples from the training set, providing both rule-based guidance and concrete demonstrations.

Strategies (1)–(4) are tested with both English (EN) and Ukrainian (UA) prompt text to address RQ1. For subsequent experiments (RQ2–RQ4), we use EN for zero-shot and few-shot prompts (where it performs best for most models; see RQ1) and UA for minimal-edits variants. This was an intentional design choice: the minimal-edits rules reference specific Ukrainian word forms, morphological categories, and language-specific conventions (e.g., vocative case paradigms, euphonic preposition alternation) that cannot be adequately expressed in English. Because the prompt language and strategy are tied together in this comparison, we treat them as a single design decision. To test whether an LLM can improve over these handcrafted prompts, we also apply LLM-assisted prompt optimization, inspired by automatic prompt optimization methods (see Ramnath et al., 2025, for a survey), on top of the best minimal-edits + few-shot prompt (Appendix A.5.2; RQ3).

**Evaluation.** We use the official UNLP-2023 evaluation pipeline (GEC only), which computes span-level Precision (P), Recall (R), and $F_{0.5}$ using a Ukrainian adaptation of ERRANT. Per-error-type scores are extracted from the ERRANT alignment for the per-error-type analysis (RQ4).

## 3 Prompt Design

**Zero-shot.** We use a single-sentence system prompt (Appendix A.1), adapted from Loem et al. (2023):

> `Reply with a corrected version of the sentence with all grammatical and spelling errors fixed. If there are no errors, reply with a copy of the original sentence. Input sentence: {sentence}. Corrected sentence:`

The Ukrainian version is its direct translation:

> Надай виправлену версію речення з виправленими всіма граматичними та орфографічними помилками. Якщо помилок немає, надай копію оригінального речення. Вхідне речення: {sentence}. Виправлене речення:

**Few-shot.** The few-shot prompt extends the zero-shot instruction with source–target correction pairs from the training set (Appendix A.2), e.g.:

> `[Same header as zero-shot prompt]`
> `Input:` Так само потерпає Україна і сьогодні від того що насправді талановитим людям заважають працювати...
> `Output:` Так само потерпає Україна і сьогодні від того, що насправді талановитим людям заважають працювати...
> *(Input: 'Ukraine suffers the same today from the fact that truly talented people are prevented from working...'*
> *Output: 'Ukraine suffers the same today from the fact, that truly talented people are prevented from working...')*

Exemplars are drawn from the UA-GEC training split because it is the only publicly available Ukrainian GEC corpus at the required scale and annotation quality. Since this split is public, it may have been seen by commercial LLMs during pretraining; drawing exemplars from an independent Ukrainian GEC corpus would be a cleaner control, but no comparable dataset currently exists. We therefore treat our numbers as establishing prompting baselines on this benchmark and revisit this risk in the Limitations section.

**Minimal-edits.** The zero-shot and few-shot prompts give only a generic correction instruction ("fix all grammatical and spelling errors"), which provides no guidance on correction scope. In practice, this leads LLMs to overcorrect: rephrasing sentences, substituting synonyms, or "improving" stylistically acceptable constructions. Since the ERRANT-based $F_{0.5}$ metric penalizes unnecessary edits, such overcorrection directly hurts precision.

The minimal-edits prompt addresses this with a two-part structure (Appendix A.3). The first part explicitly declares the minimal-edit constraint: "correct only clear-cut errors while preserving the original wording". The second part provides a

specific taxonomy of 16 Ukrainian GEC error categories (spelling, punctuation, case, gender, number, aspect, tense, etc.), followed by language-specific conventions (e.g., у/в 'u/v' alternation before consonants/vowels, em-dash in dialogue) and strict rules on what not to change (no synonym substitution, no quote style normalization, no changes when in doubt):

> ...
> Виправляй ЛИШЕ такі типи помилок:
> *('Fix ONLY the following types of errors:')*
> 1. Орфографія: явні орфографічні помилки
> *('1. Spelling: obvious spelling errors')*
> 2. Пунктуація: пропущені або зайві коми, крапки...
> *('2. Punctuation: missing or extra commas, periods...')*
> 3. G/Case: некоректне вживання відмінкової форми
> *('3. G/Case: incorrect use of case form')*
> ...

The minimal-edits + few-shot variant (Appendix A.4) combines this detailed system prompt with few-shot examples.

**LLM-assisted prompt optimization.** Inspired by automatic prompt optimization methods (Ramnath et al., 2025), we develop a semi-automatic approach in which an LLM proposes prompt edits but a human reviews and accepts them. Our method borrows ideas from several automatic prompt optimization papers: like ProTeGi (Pryzant et al., 2023), we use LLM-generated "textual gradients" derived from error analysis to guide prompt edits; following PromptAgent (Wang et al., 2024), we cluster prediction–reference mismatches into recurring linguistic patterns (e.g., "unnecessary dash normalization", "missed comma before subordinate conjunction") to produce domain-expert-style prompt sections; and as in OPRO (Yang et al., 2024), we maintain an optimization history of previous candidates and their scores to inform each iteration.

**LLM-assisted prompt optimization design.** We implemented this pipeline as a Claude Code skill powered by Claude Opus 4.6, which acts as both error analyst and prompt engineer. Starting from the best manual prompt (usually minimal-edits + few-shot), the agent iteratively: (1) evaluates the candidate on the validation set, recording span-level TP/FP/FN; (2) clusters mismatches into linguistic patterns ranked by frequency; (3) modifies the prompt via rule insertion (an explicit prohibition in the "do not change" section) or example insertion (a targeted input–output pair, including "no-change" examples); (4) accepts the change only if $F_{0.5}$ improves, otherwise reverts. The cycle repeats until gains plateau.

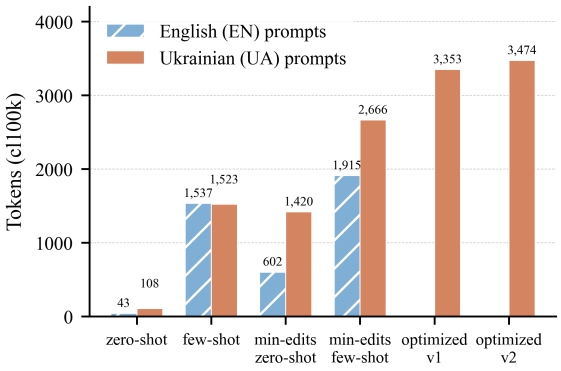



Figure 1: Prompt length (in tokens, cl100k_base tokenizer[6]) across prompting strategies for English (EN) and Ukrainian (UA) prompts. Optimized variants are available in UA only. The minimal-edits + few-shot + optimized-v2 prompt (A.5.2) is ~32× longer than the zero-shot UA baseline (A.1.2; 3,474 vs. 108 tokens).

**Optimization setup.** Due to cost and time constraints, we could not run the optimization loop separately for every model. Instead, we selected the best-performing manual prompt, minimal-edits + few-shot (A.4), and optimized it in two rounds: first on GPT-4.1-mini, producing minimal-edits + few-shot + optimized-v1 (A.5.1), and then starting from that result on Gemini 3-Flash, producing minimal-edits + few-shot + optimized-v2 (A.5.2). We then transferred these prompts to the remaining models without further tuning: GPT and Claude models are evaluated with optimized-v1, Gemini models with optimized-v2 (Table 3). We acknowledge that per-model optimization would give a more complete picture; we report these preliminary results as a useful reference point.

**Prompt length.** Figure 1 shows how prompt length grows across strategies, from 43 tokens for zero-shot (EN, A.1.1) to 3,474 tokens for minimal-edits + few-shot + optimized-v2 (UA, A.5.2).

## 4 Experimental Results

**RQ1: Zero-shot performance and prompt language (Table 1).** We start with zero-shot prompts, the simplest and most widely used setup for LLM-based GEC, and test whether prompting in Ukrainian rather than English improves results.

[6] https://github.com/openai/tiktoken

| Model | Lang. | Prec. | Rec. | $F_{0.5}$ | UA? |
|---|---|---|---|---|---|
| *Baseline: fine-tuned* | | | | | |
| mBART50-large† | – | 78.52 | 50.60 | 70.71 | |
| mT5-large‡ | – | **76.81** | **61.39** | **73.14** | |
| *Baseline: LLM zero-shot* | | | | | |
| GPT-3.5* | EN | 25.80 | 36.20 | 27.40 | |
| *API-accessed (zero-shot)* | | | | | |
| GPT-4.1 | EN | 37.17 | 55.54 | 39.80 | |
| GPT-4.1 | UA | 30.24 | 58.41 | 33.47 | |
| GPT-4.1-mini | EN | 39.06 | 51.92 | 41.09 | |
| GPT-4.1-mini | UA | 36.73 | 53.03 | 39.13 | |
| GPT-5.1 (medium) | EN | 36.06 | 60.15 | 39.20 | |
| GPT-5.1 (medium) | UA | 32.35 | 62.61 | 35.82 | |
| GPT-5.2 (medium) | EN | 34.04 | 66.31 | 37.71 | |
| GPT-5.2 (medium) | UA | 29.89 | 66.90 | 33.60 | |
| GPT-5.4 (medium) | EN | 36.87 | 62.96 | 40.20 | |
| GPT-5.4 (medium) | UA | 32.73 | 65.35 | 36.36 | |
| Claude Sonnet 4.6 | EN | 41.79 | 45.83 | 42.54 | |
| Claude Sonnet 4.6 | UA | 42.70 | 47.76 | 43.63 | + |
| Claude Opus 4.6 | EN | 47.60 | 46.20 | 47.30 | |
| Claude Opus 4.6 | UA | **49.20** | 51.60 | **49.70** | + |
| Gemini 3-Flash | EN | 39.09 | 64.85 | 42.46 | |
| Gemini 3-Flash | UA | 35.91 | **67.38** | 39.61 | |
| Gemini 3-Pro | EN | 37.89 | 60.54 | 40.96 | |
| Gemini 3-Pro | UA | 36.30 | 63.58 | 39.71 | |
| Gemini 3.1-Pro | EN | 41.50 | 58.03 | 44.01 | |
| Gemini 3.1-Pro | UA | 39.16 | 62.01 | 42.28 | |
| Kimi-K2 | EN | 34.24 | 52.74 | 36.82 | |
| Kimi-K2 | UA | 29.03 | 60.11 | 32.38 | |
| *Open-source (zero-shot)* | | | | | |
| Lapa v0.1.2§ | EN | 24.24 | 23.52 | 24.09 | |
| Lapa v0.1.2§ | UA | 28.57 | 32.38 | 29.26 | + |

Table 1: RQ1: Zero-shot GEC performance on UNLP-2023-test (GEC only). Lang.: prompt language, Ukrainian (UA) or English (EN). Bold marks the best result per column among API-accessed models. UA?: + indicates that the UA prompt yields a higher $F_{0.5}$ than the EN prompt for the same model. †Syvokon and Romanyshyn (2023); ‡Gomez et al. (2023); *Katinskaia and Yangarber (2024), EN; §Paniv et al. (2025), Ukrainian open-source LLM.

**Zero-shot performance.** All current models dramatically outperform the GPT-3.5 baseline of $F_{0.5}$ = 27.4 reported by Katinskaia and Yangarber (2024). The best zero-shot system, Claude Opus 4.6 with a UA prompt, reaches $F_{0.5}$ = 49.70, nearly doubling the GPT-3.5 score. Notably, GPT-5.x reasoning models do not outperform the older GPT-4.1-mini ($F_{0.5}$ = 41.09), despite their stronger general benchmarks. We attribute this to over-correction: reasoning-optimized models tend to over-interpret the correction task, producing more extensive rewrites that ERRANT penalizes as false positives.

Even the best zero-shot result remains 23.4 $F_{0.5}$ points below the fine-tuned SOTA of 73.14. Comparing the best zero-shot system (Claude Opus 4.6 UA: P=49.20, R=51.60) against the fine-tuned reference (P=76.81, R=61.39), the gap is primarily driven by precision (27.6-point difference) rather than recall (9.8-point difference). Without task-specific fine-tuning, zero-shot models lack the calibration to suppress spurious corrections.

Models differ in their precision–recall profiles. Gemini variants are high-recall correctors (R=60–67%) with low precision (P=36–42%), flagging many candidates, many of which are spurious. Claude models show a more balanced profile with precision and recall within 5 points of each other, which is favorable under $F_{0.5}$'s precision weighting. GPT-5.x models lean toward high recall and low precision, similar to Gemini.

**Prompt language.** Surprisingly, Claude is the only family where UA prompts improve performance: Claude Opus 4.6 gains 2.4 points (47.30 → 49.70) and Claude Sonnet 4.6 gains 1.1 points (42.54 → 43.63), with both precision and recall improving simultaneously. For all other models, UA prompts degrade $F_{0.5}$ by 1–6 points, consistently trading precision for recall and amplifying overcorrection. For reference, we also include Lapa v0.1.2[7], an open-source Ukrainian LLM, which scores $F_{0.5}$ = 29.26 with a UA prompt, above GPT-3.5 but well below the API-accessed models.

**RQ2: Prompting strategies (Table 2).** We select the six best-performing models from RQ1 (one to two per provider, excluding lower-scoring variants) and test whether few-shot examples and minimal-edits constraints can reduce the overcorrection observed in RQ1.

**Best results and gap to SOTA.** Combining few-shot examples with minimal-edits instructions, minimal-edits + few-shot yields the best or near-best $F_{0.5}$ for every model. Claude Opus 4.6 ($F_{0.5}$ = 63.73) and Gemini 3.1-Pro (63.68) effectively tie despite different zero-shot starting points. GPT-5.4 shows the largest absolute gain (+19.5

[7]Lapa (Paniv et al., 2025) is an open-source Ukrainian LLM fine-tuned with GEC-style prompts different from ours, and the only non-API-accessed model in our evaluation. We include it as a reference point for open-weight Ukrainian-centric models, though a full comparison with open-source alternatives is beyond the scope of this work.

| Model | Prompting Strategy | Lang. | Prec. | Rec. | $F_{0.5}$ |
|---|---|---|---|---|---|
| *Baseline: fine-tuned SOTA* | | | | | |
| mT5-large‡ | – | – | 76.81 | 61.39 | 73.14 |
| GPT-4.1-mini | zero-shot (A.1.1) | EN | 39.06 | 51.92 | 41.09 |
| | few-shot (A.2.1) | EN | 44.75 | 59.73 | 47.11 |
| | minimal-edits + zero-shot (A.3.1) | UA | 46.66 | 51.52 | 47.56 |
| | minimal-edits + few-shot (A.4.1) | UA | 47.16 | 51.48 | 47.97 |
| GPT-5.4 | zero-shot (A.1.1) | EN | 36.87 | 62.96 | 40.20 |
| | few-shot (A.2.1) | EN | 46.24 | 66.67 | 49.26 |
| | minimal-edits + zero-shot (A.3.1) | UA | 57.05 | 63.18 | 58.18 |
| | minimal-edits + few-shot (A.4.1) | UA | 60.02 | 58.40 | 59.69 |
| Claude Sonnet 4.6 | zero-shot (A.1.1) | EN | 41.79 | 45.83 | 42.54 |
| | few-shot (A.2.1) | EN | 52.52 | 56.26 | 53.23 |
| | minimal-edits + zero-shot (A.3.1) | UA | 62.44 | 48.55 | 59.06 |
| | minimal-edits + few-shot (A.4.1) | UA | 61.96 | 49.10 | 58.88 |
| Claude Opus 4.6 | zero-shot (A.1.1) | EN | 47.60 | 46.20 | 47.30 |
| | few-shot (A.2.1) | EN | 56.83 | 55.93 | 56.65 |
| | minimal-edits + zero-shot (A.3.1) | UA | 67.63 | 47.08 | 62.20 |
| | minimal-edits + few-shot (A.4.1) | UA | **68.54** | 49.75 | 63.73 |
| Gemini 3-Flash | zero-shot (A.1.1) | EN | 39.09 | 64.85 | 42.46 |
| | few-shot (A.2.1) | EN | 48.48 | **72.01** | 51.87 |
| | minimal-edits + zero-shot (A.3.1) | UA | 53.29 | 66.40 | 55.48 |
| | minimal-edits + few-shot (A.4.1) | UA | 60.28 | 66.18 | 61.38 |
| Gemini 3.1-Pro | zero-shot (A.1.1) | EN | 41.50 | 58.03 | 44.01 |
| | few-shot (A.2.1) | EN | 54.92 | 66.25 | 56.86 |
| | minimal-edits + zero-shot (A.3.1) | UA | 60.49 | 65.24 | 61.38 |
| | minimal-edits + few-shot (A.4.1) | UA | 63.76 | 63.35 | **63.68** |

Table 2: RQ2: Effect of prompting strategies on UNLP-2023-test (GEC only). Zero-shot and few-shot use EN prompts; minimal-edits variants use UA prompts (see Section 3 for rationale). Lang.: prompt language, Ukrainian (UA) or English (EN). For each model, we report the best configuration per strategy. Bold marks the best result per column among API-accessed models. ‡Gomez et al. (2023).

points), recovering from the weakest zero-shot result to a competitive 59.69.

However, even the best prompted result falls 9.4 points below the fine-tuned SOTA of 73.14, with the gap concentrated in precision (P=76.81 vs. 68.54). The minimal-edits instruction suppresses the most egregious false positives, but a long tail of borderline corrections remains that likely requires task-specific fine-tuning.

**Few-shot gains.** Adding few-shot examples to the zero-shot prompt produces moderate but reliable gains of +6–13 $F_{0.5}$ points, driven by improvements in both precision and recall. The gains are largest for Gemini 3.1-Pro (+12.85) and smallest for GPT-4.1-mini (+6.02).

**Minimal-edits instructions matter most.** The minimal-edits constraint has a larger effect than few-shot examples. Switching from a generic EN prompt to a UA minimal-edits instruction, even without few-shot examples, already matches or exceeds few-shot-only performance for five out of six models. The most striking case is GPT-5.4: minimal-edits + zero-shot alone yields $F_{0.5}$ = 58.18, a full 9 points above its few-shot score of 49.26. The mechanism is a sharp precision increase (+8–21 points across models) with modest recall change, meaning the constraint reduces unnecessary edits without hurting the model's ability to catch real errors.

**Overall trends.** Across all six models, $F_{0.5}$ improves consistently along the progression: zero-shot < few-shot < minimal-edits + zero-shot ≤ minimal-edits + few-shot, with one notable exception: Claude Sonnet 4.6 peaks at minimal-edits + zero-shot (59.06) and slightly drops with the addition of few-shot examples (58.88). As discussed in Section 3, the minimal-edits prompts are written in Ukrainian by design, so we cannot fully separate the effect of prompt language from the effect of the prompting strategy itself.

**RQ3: LLM-assisted prompt optimization (Table 3).**

| Model | Prompting Strategy | Lang. | Prec. | Rec. | $F_{0.5}$ |
|---|---|---|---|---|---|
| *Baseline: fine-tuned SOTA* | | | | | |
| mT5-large‡ | – | – | 76.81 | 61.39 | 73.14 |
| GPT-4.1-mini | minimal-edits + few-shot (A.4.1) | UA | 47.16 | 51.48 | 47.97 |
| | minimal-edits + few-shot + optimized-v1 (A.5.1) | UA | 55.75 | 51.49 | 54.84 |
| GPT-5.4 | minimal-edits + few-shot (A.4.1) | UA | 60.02 | 58.40 | 59.69 |
| | minimal-edits + few-shot + optimized-v1 (A.5.1) | UA | 63.94 | 51.75 | 61.07 |
| Claude Sonnet 4.6 | minimal-edits + zero-shot[8] (A.3.1) | UA | 62.44 | 48.55 | 59.06 |
| | minimal-edits + few-shot + optimized-v1 (A.5.1) | UA | 66.73 | 36.58 | 57.29 |
| Claude Opus 4.6 | minimal-edits + few-shot (A.4.1) | UA | 68.54 | 49.75 | 63.73 |
| | minimal-edits + few-shot + optimized-v1 (A.5.1) | UA | 66.54 | 38.27 | 57.98 |
| Gemini 3-Flash | minimal-edits + few-shot (A.4.1) | UA | 60.28 | **66.18** | 61.38 |
| | minimal-edits + few-shot + optimized-v2 (A.5.2) | UA | 69.98 | 62.87 | 68.43 |
| Gemini 3.1-Pro | minimal-edits + few-shot (A.4.1) | UA | 63.76 | 63.35 | 63.68 |
| | minimal-edits + few-shot + optimized-v2 (A.5.2) | UA | **70.77** | 63.63 | **69.22** |

Table 3: RQ3: Effect of LLM-assisted prompt optimization, evaluated on UNLP-2023-test (GEC only). Optimized-v1 was tuned on GPT-4.1-mini, optimized-v2 on Gemini 3-Flash; both were then transferred to other models. Lang.: prompt language, Ukrainian (UA) or English (EN). For each model, the first row shows the best manual prompt result (from Table 2); the second row shows the best optimized result. Bold marks the best result per column among API-accessed models. ‡Gomez et al. (2023).

**Best results and gap to SOTA.** The best optimized result is Gemini 3.1-Pro with minimal-edits + few-shot + optimized-v2 (A.5.2; $F_{0.5}$ = 69.22), followed closely by Gemini 3-Flash with the same prompt (68.43). This narrows the gap to fine-tuned SOTA from 9.5 to 3.9 points. The gain on the target model is precision-driven: precision rises from 63.76 to 70.77 (+7.0) while recall remains nearly unchanged. The remaining 3.9-point gap is concentrated in precision (70.77 vs. 76.81), suggesting that closing it likely requires more focused instructions.

**Improvement within Gemini.** Since the minimal-edits + few-shot + optimized-v2 prompt (A.5.2) was tuned directly on Gemini 3-Flash, its strong gain on that model ($F_{0.5}$: 61.38 → 68.43, +7.05) is expected. More notably, the same prompt transfers successfully to Gemini 3.1-Pro, which achieves the overall best result ($F_{0.5}$ = 69.22), indicating that optimization on a smaller model within the family can benefit larger variants.

**Improvement on GPT.** GPT models show mixed results. GPT-4.1-mini gains +6.87 points (47.97 → 54.84), a substantial improvement but still the weakest absolute result. GPT-5.4 gains only +1.38 (59.69 → 61.07), suggesting that the stronger model already captures most correction patterns encoded in the optimized prompt.

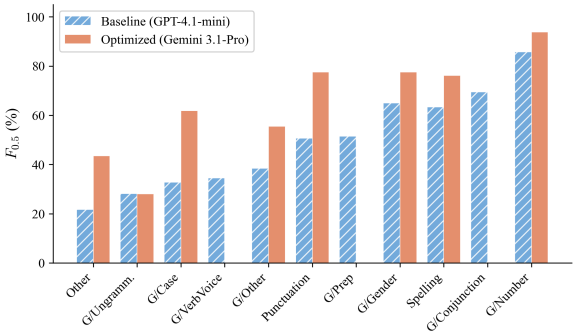



Figure 2: RQ4: Per-error-type $F_{0.5}$ on UNLP-2023-test (GEC only). Baseline: GPT-4.1-mini (zero-shot (A.1.1), EN); Optimized: Gemini 3.1-Pro (minimal-edits + few-shot + optimized-v2 (A.5.2), UA). Error types sorted as in Table 4 (by baseline overcorrection ratio, descending).

**No improvement on Claude.** The optimized prompt degrades both Claude models: Opus drops by −5.75 points (63.73 → 57.98) and Sonnet by −1.77 points (59.06 → 57.29), driven largely by a recall collapse (Opus: 49.75 → 38.27; Sonnet: 48.55 → 36.58). Claude appears to interpret the optimized rules too conservatively, suppressing genuine corrections alongside false positives. This finding shows that optimization on one model family does not necessarily guarantee improvement on another.

**RQ4: Where do minimal-edits instructions help and where do they fail? (Figure 2, Table 4).** We compare per-error-type $F_{0.5}$ between a standard

| Error Type | GPT-4.1-mini | Gemini 3.1-Pro |
|---|---|---|
| Other | 3.67 | 0.50 |
| G/Ungramm. | 2.50 | 2.00 |
| G/Case | 1.88 | 0.39 |
| G/VerbVoice | 1.50 | – |
| G/Other | 1.00 | 0.00 |
| Punctuation | 0.95 | 0.26 |
| G/Prep | 0.86 | – |
| G/Gender | 0.60 | 0.22 |
| Spelling | 0.59 | 0.25 |
| G/Conjunction | 0.40 | ∞ |
| G/Number | 0.17 | 0.00 |

Table 4: RQ4: Overcorrection ratio (FP/TP) per error type; lower is better. The two columns compare the weakest (GPT-4.1-mini, zero-shot (A.1.1), EN) and strongest (Gemini 3.1-Pro, minimal-edits + few-shot + optimized-v2 (A.5.2), UA) configurations from Tables 2–3; note that both model and prompt differ. Rows sorted by GPT-4.1-mini ratio (descending); overcorrection (more false positives than true positives) occurs above 1.0. "–" indicates no predictions for that type; ∞ indicates only false positives.

zero-shot prompt (GPT-4.1-mini) and our best optimized prompt (Gemini 3.1-Pro, minimal-edits + few-shot + optimized-v2; A.5.2) to identify which error categories benefit most from detailed minimal-edits instructions. Note that this comparison reflects the combined effect of model choice, prompt strategy, and prompt language; we select these two configurations as the weakest and strongest endpoints of our evaluation pipeline.

**Where minimal-edit instructions help.** The largest $F_{0.5}$ gains appear in categories amenable to explicit rules. Punctuation improves from 50.67 to 77.56 (+26.9), G/Case from 32.82 to 61.83 (+29.0), and G/Gender from 64.94 to 77.59 (+12.7). The overcorrection ratios in Table 4 confirm the mechanism: FP/TP drops from 0.95 to 0.26 for Punctuation, from 1.88 to 0.39 for G/Case, and from 0.60 to 0.22 for G/Gender. Spelling and G/Number are reliable under both configurations.

**Where minimal-edit instructions fail.** Three categories drop to $F_{0.5} = 0$ under the optimized prompt: G/Prep, G/VerbVoice, and G/Conjunction. The baseline achieves non-trivial $F_{0.5}$ scores on these types (51.47, 34.48, and 69.44), but the detailed minimal-edits rules cause the model to avoid these corrections entirely. G/UngrammaticalStructure remains persistently overcorrected in both settings ($F_{0.5}$: 28.17 → 28.04), indicating a structural difficulty that instructions cannot resolve.

**Ukrainian-specific overcorrection patterns.** Error analysis reveals five recurring patterns specific to Ukrainian, driven by the interaction between English-calibrated correction heuristics and Ukrainian linguistic norms. Below, we report false positive counts out of 1,274 test sentences.

**En-dash over-normalization** (– → —; ~49 FP, 3.8% of sentences). Em-dashes are obligatory in direct speech but not elsewhere; the model generalizes the rule indiscriminately.

**Dialogue reformatting** (~40 FP, 3.1% of sentences). The model converts acceptable quote-style dialogue («текст», — сказав ' "text," — said') to dash-style, applying a real Ukrainian norm where none was required. A single prohibition rule was sufficient to suppress this pattern.

**Synonym and register substitution** (~30 FP, 2.4% of sentences). Acceptable words are replaced with literary alternatives (знаходиться 'is located' → перебуває 'is situated'), violating the minimal-edit constraint.

**Euphonic preposition alternation** (в/у 'v/u', з/із/зі 'z/iz/zi'; ~17 FP, 1.3% of sentences). Ukrainian preposition choice is phonetically conditioned; the model both over- and under-corrects within the same category.

**Collapse of morphological variants.** Ukrainian admits multiple grammatically correct surface forms (навчались/навчалися 'studied', їх/їхній 'their'); the model collapses these to a single preferred form. This space of acceptable alternations is open-ended and cannot be exhaustively covered by prompt examples.

These patterns share a common cause: the model's correction heuristics are calibrated to English, where most of these alternations do not exist. Overall, our strongest prompt (Gemini 3.1-Pro, minimal-edits + few-shot + optimized-v2 (A.5.2), UA) significantly reduces overcorrection for high-frequency, rule-based categories (Table 4), but at the cost of the model becoming too conservative on low-frequency grammatical categories.

## 5 Conclusion

We presented the first systematic evaluation of prompting strategies for minimal-edit Ukrainian GEC using API-accessed LLMs. While fine-tuned models currently dominate GEC benchmarks, we show that prompting alone can be competitive. On the UNLP-2023-test (GEC only), our best configuration (Gemini 3.1-Pro with LLM-assisted opti-

mization) reaches $F_{0.5} = 69.22$, closing over 90% of the gap between the previous API-accessed result of Katinskaia and Yangarber (2024) (GPT-3.5, $F_{0.5} = 27.4$) and the fine-tuned SOTA of Gomez et al. (2023) (mT5-large, $F_{0.5} = 73.14$).

Our findings yield four takeaways. First, for zero-shot and few-shot prompts, English is sufficient for most models; only Claude benefits from Ukrainian prompts (RQ1). Our best overall results, however, use Ukrainian minimal-edits prompts, as the language-specific rules they encode require Ukrainian to express precisely. Second, the minimal-edits strategy provides the largest gains, outperforming both zero-shot and few-shot baselines across all models (RQ2). Third, LLM-assisted prompt optimization yields further improvements on the model family it was optimized for, but does not transfer reliably across families (RQ3). Fourth, minimal-edits instructions yield the largest per-category gains for punctuation and case errors, but cause the model to abandon several low-frequency grammatical categories entirely, revealing a precision-recall tradeoff inherent to detailed prompting (RQ4).

## Limitations

Our study has several limitations:

1. We evaluate on a single benchmark (UNLP-2023-test (GEC only)); results may not generalize to other Ukrainian GEC datasets or domains.

2. Although we include a single open-source model (Lapa v0.1.2) as a reference point, we do not systematically compare against open-weight models (e.g., Llama 3, Mistral, Lapa, MamayLM) that could be prompted or fine-tuned without API costs, leaving this as future work.

3. API-accessed models are opaque and subject to unannounced updates, making exact reproducibility difficult.

4. Although the UNLP-2023-test (GEC only) gold annotations are held out, the upstream UA-GEC train and valid splits are publicly available. Since UA-GEC train is the source of our few-shot exemplars, it is plausible that commercial LLMs encountered similar sentences and annotation patterns during pretraining, which could inflate recall on a corpus sharing the same annotation conventions. A cleaner control would draw exemplars from an independent Ukrainian GEC corpus, but no comparable dataset currently exists, so we treat our results as establishing initial prompting baselines; prior API-accessed results for Ukrainian GEC at this scale are essentially absent.

5. We run each configuration only once; since LLM outputs are not fully deterministic, reproduced scores may differ/cos slightly.

6. Our LLM-assisted prompt optimization pipeline optimizes $F_{0.5}$ on the validation set, which may overfit to its error distribution.

7. All our experiments are evaluated with span-based $F_{0.5}$ computed by ERRANT, the official metric of the UNLP 2023 Shared Task, GEC-only track (Syvokon and Romanyshyn, 2023); this differs from GLEU used in MultiGEC-2025 (Masciolini et al., 2025), so scores are not directly comparable across benchmarks.

8. Optimized prompts are substantially longer (roughly 32× the zero-shot baseline; see Figure 1), which may increase token cost and latency. Prompt caching, now widely supported by providers, amortizes much of this overhead, making the net cost hard to estimate.

## Ethical Considerations

In accordance with the conference policy on AI-based writing assistance, we disclose that ChatGPT, Claude, Gemini, and Grammarly were used for drafting, editing, and proofreading. All AI-generated text was reviewed by the authors, who take full responsibility for the final content.

## Acknowledgments

We are deeply grateful to YouScan for fostering an inspiring environment that encourages both research and professional development. We also express our appreciation to the Faculty of Applied Sciences at the Ukrainian Catholic University for supporting this work as part of an M.Sc. thesis program. We gratefully acknowledge Mariana Romanyshyn and Oleksiy Syvokon for their assistance with the UNLP 2023 Shared Task. Finally, we extend our sincere gratitude to the anonymous reviewers for their insightful feedback and dedicated efforts in refining this manuscript.

## References


Maksym Bondarenko, Artem Yushko, and Andrii Shportko. 2023. Comparative study of models trained on synthetic data for Ukrainian grammatical error correction. In *Proceedings of the Second Ukrainian Natural Language Processing Workshop (UNLP)*, pages 103–113, Dubrovnik, Croatia. Association for Computational Linguistics.

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, 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.

Frank Palma Gomez, Alla Rozovskaya, and Dan Roth. 2023. A low-resource approach to the grammatical error correction of Ukrainian. In *Proceedings of the Second Ukrainian Natural Language Processing Workshop (UNLP)*, pages 114–119, Dubrovnik, Croatia. Association for Computational Linguistics.

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, Torino, Italia. ELRA and ICCL.

Roman Kovalchuk, Mariana Romanyshyn, and Petro Ivaniuk. 2025. Introducing OmniGEC: A silver multilingual dataset for grammatical error correction. In *Proceedings of the Fourth Ukrainian Natural Language Processing Workshop (UNLP)*, pages 162–178, Vienna, Austria. 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.

Agnes Luhtaru, Taido Purason, Martin Vainikko, and Maali Helin Del. 2024. To err is human, but llamas can learn it too. In *Proceedings of the Third Ukrainian Natural Language Processing Workshop (UNLP)*, Torino, Italia. Association for Computational Linguistics.

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 (NLP4CALL 2025)*, pages 1–33, Tallinn, Estonia. University of Tartu Library.

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.

Yurii Paniv, Bohdan Didenko, Mykola Haltiuk, Vladyslav Humennyy, Andrian Kravchenko, Roman Kyslyi, Viktoriia Makovska, Artem Orlovskyi, Bohdan Ruban, Maksym-Yurii Rudko, Anastasiia Senyk, Nazarii Drushchak, Dmytro Chaplynskyi, and Mariana Romanyshyn. 2025. Lapa LLM v0.1.2 — the most efficient Ukrainian open-source language model.

Reid Pryzant, Dan Iter, Jerry Li, Yin Lee, Chenguang Zhu, and Michael Zeng. 2023. Automatic prompt optimization with "gradient descent" and beam search. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7957–7968, Singapore. Association for Computational Linguistics.

Kiran Ramnath, Kang Zhou, Sheng Guan, Soumya Smruti Mishra, Xuan Qi, Zhengyuan Shen, Shuai Wang, Sangmin Woo, Sullam Jeoung, Yawei Wang, Haozhu Wang, Han Ding, Yuzhe Lu, Zhichao Xu, Yun Zhou, Balasubramaniam Srinivasan, Qiaojing Yan, Yueyan Chen, Haibo Ding, and 2 others. 2025. A systematic survey of automatic prompt optimization techniques. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 33078–33110, Suzhou, China. Association for Computational Linguistics.

Aman Saini, Artem Chernodub, Vipul Raheja, and Vivek Kulkarni. 2024. Spivavtor: An instruction tuned Ukrainian text editing model. In *Proceedings of the Third Ukrainian Natural Language Processing Workshop (UNLP)*, pages 95–108, Torino, Italia. ELRA and ICCL.

Ryszard Staruch, Filip Graliński, 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.

Oleksiy Syvokon and Mariana Romanyshyn. 2023. The UNLP 2023 shared task on grammatical error correction for Ukrainian. In *Proceedings of the Second Ukrainian Natural Language Processing Workshop (UNLP)*, pages 120–131, Dubrovnik, Croatia. Association for Computational Linguistics.

Xinyuan Wang, Chenxi Li, Zhen Wang, Fan Bai, Haotian Luo, Jiayou Zhang, Nebojsa Jojic, Eric P. Xing, and Zhiting Hu. 2024. PromptAgent: Strategic planning with language models enables expert-level prompt optimization. In *The Twelfth International Conference on Learning Representations*, Vienna, Austria. OpenReview.net.

Chengrun Yang, Xuezhi Wang, Yifeng Lu, Hanxiao Liu, Quoc V. Le, Denny Zhou, and Xinyun Chen. 2024. Large language models as optimizers. *arXiv preprint arXiv:2309.03409*.

# A Prompts

Below we list all prompts in both English (EN) and Ukrainian (UA) versions. The placeholder `{sentence}` is replaced with the input sentence at inference time.

## A.1 Zero-shot prompts

The baseline prompt provides only a task description with no examples.

### A.1.1 Zero-shot prompt (EN)

```
Reply with a corrected version of the sentence with all grammatical and spelling errors fixed.
If there are no errors, reply with a copy of the original sentence.
Input sentence: <input_text>
Corrected sentence:
```

### A.1.2 Zero-shot prompt (UA)

Надай виправлену версію речення з виправленими всіма граматичними та орфографічними помилками.
Якщо помилок немає, надай копію оригінального речення.
Вхідне речення: <input_text>
Виправлене речення:

*(English: 'Provide a corrected version of the sentence with all grammatical and spelling errors fixed. If there are no errors, provide a copy of the original sentence. Input sentence: <input_text>. Corrected sentence:')*

## A.2 Few-shot prompts

The few-shot prompt prepends source–target pairs selected from the training set to cover spelling, punctuation, and morphological error types.

### A.2.1 Few-shot prompt (EN)

```
[Same header as Prompt 1]

Examples:

Input: Так само потерпає Україна і сьогодні від того що насправді талановитим людям заважають працювати усілякі посередності "у руля".
Output: Так само потерпає Україна і сьогодні від того, що насправді талановитим людям заважають працювати усілякі посередності "у руля".

Input: Це пов'язано з тим, що такі колективні рухи молекул води сильно збільшують характерні часи процесів які відбуваються в системі.
Output: Це пов'язано з тим, що такі колективні рухи молекул води сильно збільшують характерні часи процесів, які відбуваються в системі.

Input: Я ніколи не навчався у медичному коледжі, кажу, - Я лиш підійшов як звичайна людина подивитись чи можу бути чимось корисний.
Output: Я ніколи не навчався у медичному коледжі, кажу, - Я лиш підійшов як звичайна людина подивитись, чи можу бути чимось корисний.

Input: Це у місті швидка приїжджає, забирає хворого і везе у лікарню; якщо ж до лікарні кілька годин льоту то хворого можна і не довезти.
Output: Це у місті швидка приїжджає, забирає хворого і везе у лікарню; якщо ж до лікарні кілька годин льоту, то хворого можна і не довезти.

Input: Найбільше він любив тримати в руках старанно орнаментовані стародавні шолом і меч, котрі своїм золотом, здається гріли старого.
Output: Найбільше він любив тримати в руках старанно орнаментовані стародавні шолом і меч, котрі своїм золотом, здається, гріли старого.

Input: - Я часто казав тобі, що ти дурненька, - сказав він.
Output: — Я часто казав тобі, що ти дурненька, — сказав він.

Input: Така традиція також походить з Візантії, прикладом є зображення Андроніка II Палеолога.
Output: Така традиція також походить із Візантії, прикладом є зображення Андроніка II Палеолога.

Input: Як і більшість ділових людей, він не знав напамять жодного вірша і не памятав жодної казки, а тому щоразу мусив імпровізувати.
Output: Як і більшість ділових людей, він не знав напам'ять жодного вірша і не памятав жодної казки, а тому щоразу мусив імпровізувати.

Input: Настя, привіт! я хотіла уточнити про завдання Олі.
Output: Насте, привіт! Я хотіла уточнити про завдання Олі.

Input: Я принесу твої улюблені солодощі та обніму тебе міцно-міцно.
Output: Я принесу твої улюблені солодощі та обніму тебе міцно-міцно.

Input: Смакота ще та, скажу я вам))
Output: Смакота ще та, скажу я вам))

Input: У той час глибокий сенс народної мудрості нам був ще недоступним через брак досвіду.
Output: У той час глибокий сенс народної мудрості нам був ще недоступним через брак досвіду.

Input: Каналізація в будинках еволюціонує повільно, але все ж таки змінюється.
Output: Каналізація в будинках еволюціонує повільно, але все ж таки змінюється.

Input sentence: {input_text}
Corrected sentence:
```

### A.2.2 Few-shot prompt (UA)

[Той самий заголовок, що і в Prompt 1]
*(English: '[Same header as Prompt 1]')*

Приклади: *(‘Examples:’)*

Вхід: Так само потерпає Україна і сьогодні від того що насправді талановитим людям заважають працювати усілякі посередності "у руля".
Вихід: Так само потерпає Україна і сьогодні від того, що насправді талановитим людям заважають працювати усілякі посередності "у руля".

Вхід: Я ніколи не навчався у медичному коледжі, кажу, - Я лиш підійшов як звичайна людина подивитись чи можу бути чимось корисний.
Вихід: Я ніколи не навчався у медичному коледжі, кажу, - Я лиш підійшов як звичайна людина подивитись, чи можу бути чимось корисний.

Вхід: Це у місті швидка приїжджає, забирає хворого і везе у лікарню; якщо ж до лікарні кілька годин льоту то хворого можна і не довезти.
Вихід: Це у місті швидка приїжджає, забирає хворого і везе у лікарню; якщо ж до лікарні кілька годин льоту, то хворого можна і не довезти.

Вхід: Найбільше він любив тримати в руках старанно орнаментовані стародавні шолом і меч, котрі своїм золотом, здається гріли старого.
Вихід: Найбільше він любив тримати в руках старанно орнаментовані стародавні шолом і меч, котрі своїм золотом, здається, гріли старого.

Вхід: - Я часто казав тобі, що ти дурненька, - сказав він.
Вихід: — Я часто казав тобі, що ти дурненька, — сказав він.

Вхід: Така традиція також походить з Візантії, прикладом є зображення Андроніка II Палеолога.
Вихід: Така традиція також походить із Візантії, прикладом є зображення Андроніка II Палеолога.

Вхід: Як і більшість ділових людей, він не знав напамять жодного вірша і не памятав жодної казки, а тому щоразу мусив імпровізувати.
Вихід: Як і більшість ділових людей, він не знав напам’ять жодного вірша і не памятав жодної казки, а тому щоразу мусив імпровізувати.

Вхід: Настя, привіт! я хотіла уточнити про завдання Олі.
Вихід: Насте, привіт! Я хотіла уточнити про завдання Олі.

Вхід: Я принесу твої улюблені солодощі та обніму тебе міцно-міцно.
Вихід: Я принесу твої улюблені солодощі та обніму тебе міцно-міцно.

Вхід: Смакота ще та, скажу я вам))
Вихід: Смакота ще та, скажу я вам))

Вхід: У той час глибокий сенс народної мудрості нам був ще недоступним через брак досвіду.
Вихід: У той час глибокий сенс народної мудрості нам був ще недоступним через брак досвіду.

Вхід: Каналізація в будинках еволюціонує повільно, але все ж таки змінюється.
Вихід: Каналізація в будинках еволюціонує повільно, але все ж таки змінюється.

Вхідне речення: {input_text}
Виправлене речення:

*(English:* Вхід/Вихід = *‘Input’/‘Output’;* Вхідне речення = *‘Input sentence’;* Виправлене речення = *‘Corrected sentence’. The few-shot examples are the same Ukrainian GEC sentence pairs as in the EN variant (A.2.1), with Ukrainian keywords.)*

## A.3 Minimal-edits zero-shot prompts

This prompt replaces the generic system prompt with a detailed minimal-edit instruction containing Ukrainian-specific grammar rules. No few-shot examples are included.

### A.3.1 Minimal-edits zero-shot prompt (UA)

Ти — система виправлення українських граматичних помилок. Внось МІНІМАЛЬНІ зміни, щоб виправити ЛИШЕ явні граматичні, орфографічні та пунктуаційні помилки. НЕ переписуй, не перефразовуй і не замінюй слова синонімами. Точно зберігай оригінальне формулювання.

Виправляй ЛИШЕ такі типи помилок:

1. Орфографія: явні орфографічні помилки (друкарські помилки, неправильні літери).
2. Пунктуація: пропущені або зайві коми, крапки, знаки питання; використання тире (—) замість дефіса (-) у діалогах та вставних конструкціях.
3. G/Case: некоректне вживання відмінкової форми (зокрема кличний відмінок при звертаннях).
4. G/Gender: некоректне вживання форми роду.
5. G/Number: некоректне вживання форми числа.
6. G/Aspect: некоректне вживання форми виду дієслова.
7. G/Tense: некоректне вживання часової

форми дієслова.
8. G/VerbVoice: некоректне вживання форми стану дієслова.
9. G/PartVoice: некоректне вживання форми стану дієприкметника.
10. G/VerbAForm: некоректне вживання аналітичної форми дієслова.
11. G/Prep: некоректне вживання прийменника.
12. G/Participle: некоректне вживання дієприслівника.
13. G/UngrammaticalStructure: порушення граматичних норм у синтаксичних конструкціях.
14. G/Comparison: некоректна форма ступенів порівняння.
15. G/Conjunction: некоректне вживання сполучників.
16. G/Other: інші граматичні помилки.

ВАЖЛИВІ ПРАВИЛА УКРАЇНСЬКОЇ МОВИ:
- Прийменник «у» вживається перед приголосними (у школі, у місті, у готелі), «в» — перед голосними та на початку речення.
- Прийменник «об» вживається перед голосними (об одинадцятій), «о» — перед приголосними.
- У діалогах вживається тире (—), а не дефіс (-): «Текст», — сказав він. — Текст далі.
- Вставні слова (може, мабуть, звичайно, здається) виділяються комами з обох боків.
- Кличний відмінок при звертаннях: Настя → Насте, Олег → Олеже, мама → мамо.

СУВОРІ ПРАВИЛА:
- Виправляй ЛИШЕ явні помилки, перелічені вище
- Для кожної помилки вноси НАЙМЕНШУ можливу зміну
- НІКОЛИ не замінюй слова синонімами і не перефразовуй (залишай «буду йти», НЕ змінюй на «піду»)
- НІКОЛИ не змінюй слово на інше, якщо воно не написане з помилкою
- Зберігай оригінальний стиль лапок ("або «»), НЕ перетворюй один тип лапок на інший
- Зберігай оригінальне використання великих/малих літер, якщо це не явна помилка
- Якщо граматична форма є прийнятною, залишай її, навіть якщо можлива й інша форма
- Якщо є сумнів, НЕ змінюй
- Якщо помилок немає, повертай оригінальний текст БЕЗ ЗМІН
- Поверни ЛИШЕ виправлений текст

*English translation of the above prompt:*

*You are a Ukrainian grammatical error correction system. Make MINIMAL changes to fix ONLY obvious grammatical, spelling, and punctuation errors. Do NOT rewrite, rephrase, or substitute synonyms. Preserve the original wording exactly.*

*Fix ONLY the following types of errors:*
*1. Spelling: obvious spelling errors (typos, wrong letters).*
*2. Punctuation: missing or extra commas, periods, question marks; use of em-dash (—) instead of hyphen (-) in dialogues and parenthetical constructions.*
*3. G/Case: incorrect case form (especially vocative case in forms of address).*
*4. G/Gender: incorrect gender form.*
*5. G/Number: incorrect number form.*
*6. G/Aspect: incorrect verb aspect form.*
*7. G/Tense: incorrect verb tense form.*
*8. G/VerbVoice: incorrect verb voice form.*
*9. G/PartVoice: incorrect participle voice form.*
*10. G/VerbAForm: incorrect analytical verb form.*
*11. G/Prep: incorrect preposition usage.*
*12. G/Participle: incorrect adverbial participle usage.*
*13. G/UngrammaticalStructure: grammatical norm violations in syntactic constructions.*
*14. G/Comparison: incorrect comparative/superlative form.*
*15. G/Conjunction: incorrect conjunction usage.*
*16. G/Other: other grammatical errors.*

*IMPORTANT RULES OF UKRAINIAN:*
*- Preposition “u” is used before consonants (u shkoli, u misti), “v” before vowels and at sentence start.*
*- Preposition “ob” is used before vowels (ob odynadtsiatii), “o” before consonants.*
*- Em-dash (—) is used in dialogues, not hyphen (-).*
*- Parenthetical words (maybe, probably, of course, it seems) are set off by commas on both sides.*
*- Vocative case in forms of address: Nastia → Naste, Oleh → Olezhe, mama → mamo.*

*STRICT RULES:*
*- Fix ONLY obvious errors listed above*
*- For each error, make the SMALLEST possible change*
*- NEVER substitute synonyms or rephrase*
*- NEVER change a word to another unless it is misspelled*
*- Preserve original quote style, do NOT convert one type to another*
*- Preserve original capitalization unless it is a clear error*
*- If a grammatical form is acceptable, leave it even if another form is possible*
*- If in doubt, do NOT change*
*- If there are no errors, return the original text WITHOUT CHANGES*
*- Return ONLY the corrected text*

### A.4 Minimal-edits few-shot prompts

This prompt combines the minimal-edits system prompt (Appendix A.3) with few-shot examples from the training set.

#### A.4.1 Minimal-edits few-shot prompt (UA)

Ти — система виправлення українських граматичних помилок. Внось МІНІМАЛЬНІ зміни, щоб виправити ЛИШЕ явні граматичні, орфографічні та пунктуаційні помилки. НЕ переписуй, не перефразовуй і не замінюй слова синонімами. Точно зберігай оригінальне

формулювання.

Виправляй ЛИШЕ такі типи помилок:

1. Орфографія: явні орфографічні помилки (друкарські помилки, неправильні літери).
2. Пунктуація: пропущені або зайві коми, крапки, знаки питання; використання тире (—) замість дефіса (-) у діалогах та вставних конструкціях.
3. G/Case: некоректне вживання відмінкової форми (зокрема кличний відмінок при звертаннях).
4. G/Gender: некоректне вживання форми роду.
5. G/Number: некоректне вживання форми числа.
6. G/Aspect: некоректне вживання форми виду дієслова.
7. G/Tense: некоректне вживання часової форми дієслова.
8. G/VerbVoice: некоректне вживання форми стану дієслова.
9. G/PartVoice: некоректне вживання форми стану дієприкметника.
10. G/VerbAForm: некоректне вживання аналітичної форми дієслова.
11. G/Prep: некоректне вживання прийменника.
12. G/Participle: некоректне вживання дієприслівника.
13. G/UngrammaticalStructure: порушення граматичних норм у синтаксичних конструкціях.
14. G/Comparison: некоректна форма ступенів порівняння.
15. G/Conjunction: некоректне вживання сполучників.
16. G/Other: інші граматичні помилки.

ВАЖЛИВІ ПРАВИЛА УКРАЇНСЬКОЇ МОВИ:
- Прийменник «у» вживається перед приголосними (у школі, у місті, у готелі), «в» — перед голосними та на початку речення.
- Прийменник «об» вживається перед голосними (об одинадцятій), «о» — перед приголосними.
- У діалогах вживається тире (—), а не дефіс (-): «Текст», — сказав він. — Текст далі.
- Вставні слова (може, мабуть, звичайно, здається) виділяються комами з обох боків.
- Кличний відмінок при звертаннях: Настя → Насте, Олег → Олеже, мама → мамо.

СУВОРІ ПРАВИЛА:
- Виправляй ЛИШЕ явні помилки, перелічені вище
- Для кожної помилки внось НАЙМЕНШУ можливу зміну
- НІКОЛИ не замінюй слова синонімами і не перефразовуй (залишай «буду йти», НЕ змінюй на «піду»)
- НІКОЛИ не змінюй слово на інше, якщо воно не аписане з помилкою
- Зберігай оригінальний стиль лапок ("або «»), НЕ перетворюй один тип лапок на інший
- Зберігай оригінальне використання великих/малих літер, якщо це не явна помилка
- Якщо граматична форма є прийнятною, залишай її, навіть якщо можлива й інша форма
- Якщо є сумнів, НЕ змінюй
- Якщо помилок немає, повертай оригінальний текст БЕЗ ЗМІН

Приклади:

Вхід: Так само потерпає Україна і сьогодні від того що насправді талановитим людям заважають працювати усілякі посередності "у руля".
Вихід: Так само потерпає Україна і сьогодні від того, що насправді талановитим людям заважають працювати усілякі посередності "у руля".

Вхід: Я ніколи не навчався у медичному коледжі, кажу, - Я лиш підійшов як звичайна людина подивитись чи можу бути чимось корисний.
Вихід: Я ніколи не навчався у медичному коледжі, кажу, - Я лиш підійшов як звичайна людина подивитись, чи можу бути чимось корисний.

Вхід: Це у місті швидка приїжджає, забирає хворого і везе у лікарню; якщо ж до лікарні кілька годин льоту то хворого можна і не довезти.
Вихід: Це у місті швидка приїжджає, забирає хворого і везе у лікарню; якщо ж до лікарні кілька годин льоту, то хворого можна і не довезти.

Вхід: Найбільше він любив тримати в руках старанно орнаментовані стародавні шолом і меч, котрі своїм золотом, здається гріли старого.
Вихід: Найбільше він любив тримати в руках старанно орнаментовані стародавні шолом і меч, котрі своїм золотом, здається, гріли старого.

Вхід: - Я часто казав тобі, що ти дурненька, - сказав він.
Вихід: — Я часто казав тобі, що ти дурненька, — сказав він.

Вхід: Така традиція також походить з Візантії, прикладом є зображення Андроніка II Палеолога.
Вихід: Така традиція також походить із Візантії, прикладом є зображення Андроніка II Палеолога.

Вхід: Як і більшість ділових людей, він не знав напамять жодного вірша і не памятав жодної казки, а тому щоразу мусив імпровізувати.
Вихід: Як і більшість ділових людей, він не знав напам'ять жодного вірша і не памятав жодної казки, а тому щоразу мусив імпровізувати.

Вхід: Настя, привіт! я хотіла уточнити про завдання Олі.
Вихід: Насте, привіт! Я хотіла уточнити про завдання Олі.

Вхід: Я принесу твої улюблені солодощі та обніму тебе міцно-міцно.
Вихід: Я принесу твої улюблені солодощі та обніму тебе міцно-міцно.

Вхід: Смакота ще та, скажу я вам))
Вихід: Смакота ще та, скажу я вам))

Вхід: У той час глибокий сенс народної мудрості нам був ще недоступним через брак досвіду.
Вихід: У той час глибокий сенс народної мудрості нам був ще недоступним через брак досвіду.

Вхід: Каналізація в будинках еволюціонує повільно, але все ж таки змінюється.
Вихід: Каналізація в будинках еволюціонує повільно, але все ж таки змінюється.

Вхідне речення: {input_text}
Виправлене речення:

*English: The system prompt is identical to the minimal-edits zero-shot prompt (A.3.1); see the English translation there. The few-shot examples are the same Ukrainian GEC sentence pairs as in the few-shot EN variant (A.2.1). Keywords:* Приклади = *'Examples';* Вхід/Вихід = *'Input'/'Output'.*

## A.5 Optimized prompts

The following prompts were produced by LLM-assisted prompt optimization (Section 3). Each was derived from its parent prompt via iterative refinement on the validation set (see Section 3 for the optimization procedure).

### A.5.1 Minimal-edits + few-shot + optimized-v1 (UA): optimized on GPT-4.1-mini

Ти — система виправлення українських граматичних помилок. Внось МІНІМАЛЬНІ зміни, щоб виправити ЛИШЕ безсумнівні граматичні, орфографічні та пунктуаційні помилки. НЕ переписуй, не перефразовуй і не замінюй слова синонімами.

Виправляй:
- Орфографічні помилки (друкарські помилки, пропущені/зайві літери, неправильне написання разом/окремо: незважати → не зважати, буд-якому → будь-якому).
- Пунктуацію: пропущені коми перед підрядними сполучниками (що, який, бо, чи, коли, щоб, де, поки), при звертаннях, при вставних словах (мабуть, може, звичайно). У прямій мові дефіс (-) замінюй на тире (—): "текст сказав → "текст — сказав.
- Відмінкові помилки (зокрема кличний відмінок у звертаннях: Привіт Настя → Привіт, Насте).
- Узгодження роду, числа, відмінка в словосполученнях.
- Прийменники: о/об (об одинадцятій, о третій); з → зі/із перед збігом приголосних (з зображенням → зі зображенням, з Візантії → із Візантії).

НЕ змінюй:
- НІКОЛИ не замінюй слова синонімами і не змінюй форми слів на альтернативні (виказував, кучею, достойною, вивести — залишай як є).
- НЕ переформатовуй діалоги: якщо діалог оформлений лапками ("/«»), зберігай лапки, НЕ замінюй їх на тире.
- НЕ змінюй граматичні форми, які є допустимими варіантами: відмінкові форми прикметників (недоступним/недоступний), варіанти дієслів (навчались/навчалися), форми займенників (їх/їхній) — якщо форма граматично допустима, залишай її.
- Стиль та тон тексту: неформальний текст (чати, смс) залишай як є — не додавай крапки в кінці, не прибирай смайлики )).
- Порядок слів у реченні.
- Великі/малі літери, крім початку речення після крапки.
- Лапки: зберігай оригінальний стиль.
- Розділові знаки кінця речення: НЕ змінюй . на ? або навпаки.
- НЕ додавай тире (—) там, де його не було в оригіналі, окрім прямої мови.
- Дефіс у складених словах та повторах (міцно-міцно, дере-дере-дере, все-таки, все ж таки — залишай як є).
- Якщо сумніваєшся — НЕ змінюй.

[Ті самі приклади, що і в Prompt 2] *('[Same examples as in Prompt 2]')*

Поверни ЛИШЕ виправлений текст. *('Return ONLY the corrected text.')*

*English translation of the instruction part:*

*You are a Ukrainian grammatical error correction system. Make MINIMAL changes to fix ONLY unambiguous grammatical, spelling, and punctuation errors. Do NOT rewrite, rephrase, or substitute synonyms.*

*Fix:*

*- Spelling errors (typos, missing/extra letters, incorrect joined/separate writing: nezvazhaty → ne zvazhaty, bud-yakomu → bud'-yakomu).*
*- Punctuation: missing commas before subordinate conjunctions (shcho, yakyi, bo, chy, koly, shchob, de, poky), in forms of address, with parenthetical words (mabut', mozhe, zvychaino). In direct speech, replace hyphen (-) with em-dash (—).*
*- Case errors (especially vocative in address: Pryvit Nastia → Pryvit, Naste).*
*- Gender, number, case agreement in phrases.*
*- Prepositions: o/ob; z → zi/iz before consonant clusters.*

*Do NOT change:*
*- NEVER substitute synonyms or change word forms to alternatives — leave as is.*
*- Do NOT reformat dialogues: if dialogue uses quotes, keep quotes, do NOT replace with dashes.*
*- Do NOT change grammatically acceptable variant forms.*
*- Text style and tone: leave informal text (chats, SMS) as is.*
*- Word order, capitalization (except after period), quote style, sentence-final punctuation.*
*- Do NOT add em-dashes where there were none, except in direct speech.*
*- Hyphens in compound words and repetitions — leave as is.*
*- If in doubt — do NOT change.*

### A.5.2 Minimal-edits + few-shot + optimized-v2 (UA): optimized on Gemini 3-Flash

The prompt below was derived from minimal-edits + few-shot + optimized-v1 (A.5.1) by further LLM-assisted optimization on Gemini 3-Flash over 5 iterations on the validation set. It includes additional Ukrainian-specific rules discovered during optimization.

Ти — система виправлення українських граматичних помилок. Внось МІНІМАЛЬНІ зміни, щоб виправити ЛИШЕ безсумнівні граматичні, орфографічні та пунктуаційні помилки. НЕ переписуй, не перефразовуй і не замінюй слова синонімами.

Виправляй:
- Орфографічні помилки (друкарські помилки, пропущені/зайві літери, неправильне написання разом/окремо: незважати → не зважати, буд-якому → будь-якому).
- Пунктуацію: пропущені коми перед підрядними сполучниками (що, який, бо, чи, коли, щоб, де, поки), при звертаннях, при вставних словах (мабуть, може, звичайно). У прямій мові дефіс (-) замінюй на тире (—): "текст сказав → "текст — сказав.
- Відмінкові помилки (зокрема кличний відмінок у звертаннях: Привіт Настя → Привіт, Насте).
- Узгодження роду, числа, відмінка в словосполученнях.
- Прийменники: о/об (об одинадцятій, о третій); з → зі/із перед збігом приголосних (з зображенням → зі зображенням, з Візантії → із Візантії).

НЕ змінюй:
- НІКОЛИ не замінюй слова синонімами і не змінюй форми слів на альтернативні (виказував, кучею, достойною, вивести — залишай як є).
- НЕ переформатовуй діалоги: якщо діалог оформлений лапками ("/«»), зберігай лапки, НЕ замінюй їх на тире.
- НЕ змінюй граматичні форми, які є допустимими варіантами: відмінкові форми прикметників (недоступним/недоступний), варіанти дієслів (навчались/навчалися), форми займенників (їх/їхній) — якщо форма граматично допустима, залишай її.
- Стиль та тон тексту: неформальний текст (чати, смс) залишай як є — не додавай крапки в кінці, не прибирай смайлики )).
- Порядок слів у реченні.
- Великі/малі літери, крім початку речення після крапки.
- Лапки: зберігай оригінальний стиль.
- Розділові знаки кінця речення: НЕ змінюй . на ? або навпаки.
- НЕ додавай тире (—) там, де його не було в оригіналі, окрім прямої мови.
- Дефіс у складених словах та повторах (міцно-міцно, дере-дере-дере, все-таки, все ж таки — залишай як є).
- Якщо сумніваєшся — НЕ змінюй.

Приклади:

Вхід: Так само потерпає Україна і сьогодні від того що насправді талановитим людям заважають працювати усілякі посередності "у руля".
Вихід: Так само потерпає Україна і сьогодні від того, що насправді талановитим людям заважають працювати усілякі посередності "у руля".

Вхід: Це пов'язано з тим, що такі колективні рухи молекул води сильно збільшують характерні часи процесів які відбуваються в системі.
Вихід: Це пов'язано з тим, що такі колективні рухи молекул води сильно збільшують характерні часи процесів, які відбуваються в системі.

Вхід: Я ніколи не навчався у медичному коледжі, кажу, - Я лиш підійшов як звичайна людина подивитись чи можу бути чимось корисний.
Вихід: Я ніколи не навчався у медичному коледжі, кажу, - Я лиш підійшов як звичайна людина подивитись, чи можу бути чимось корисний.

Вхід: Це у місті швидка приїжджає, забирає хворого і везе у лікарню; якщо ж

до лікарні кілька годин льоту то хворого можна і не довезти.
Вихід: Це у місті швидка приїжджає, забирає хворого і везе у лікарню; якщо ж до лікарні кілька годин льоту, то хворого можна і не довезти.

Вхід: Найбільше він любив тримати в руках старанно орнаментовані стародавні шолом і меч, котрі своїм золотом, здається гріли старого.
Вихід: Найбільше він любив тримати в руках старанно орнаментовані стародавні шолом і меч, котрі своїм золотом, здається, гріли старого.

Вхід: - Я часто казав тобі, що ти дурненька, - сказав він.
Вихід: — Я часто казав тобі, що ти дурненька, — сказав він.

Вхід: Така традиція також походить з Візантії, прикладом є зображення Андроніка II Палеолога.
Вихід: Така традиція також походить із Візантії, прикладом є зображення Андроніка II Палеолога.

Вхід: Як і більшість ділових людей, він не знав напамять жодного вірша і не памятав жодної казки, а тому щоразу мусив імпровізувати.
Вихід: Як і більшість ділових людей, він не знав напам'ять жодного вірша і не памятав жодної казки, а тому щоразу мусив імпровізувати.

Вхід: Настя, привіт! я хотіла уточнити про завдання Олі.
Вихід: Насте, привіт! Я хотіла уточнити про завдання Олі.

Вхід: Я принесу твої улюблені солодощі та обніму тебе міцно-міцно.
Вихід: Я принесу твої улюблені солодощі та обніму тебе міцно-міцно.

Вхід: Смакота ще та, скажу я вам))
Вихід: Смакота ще та, скажу я вам))

Вхід: У той час глибокий сенс народної мудрості нам був ще недоступним через брак досвіду.
Вихід: У той час глибокий сенс народної мудрості нам був ще недоступним через брак досвіду.

Вхід: Каналізація в будинках еволюціонує повільно, але все ж таки змінюється.
Вихід: Каналізація в будинках еволюціонує повільно, але все ж таки змінюється.

Вхід: "Досить добре вийшов з Весту, чи не так? спитав міліціонер.
Вихід: "Досить добре вийшов з Весту, чи не так?"— спитав міліціонер.

Вхід: Не знаю як у інших, а у мене в житті траплялось не так багато див.
Вихід: Не знаю, як у інших, а у мене в житті траплялось не так багато див.

Вхід: Хочеться закінчити у дусі книг з самодопомоги.
Вихід: Хочеться закінчити у дусі книг із самодопомоги.

Вхід: Наступного дня тим же автобаном повернулися назад, звідти – ще півтори години літаком.
Вихід: Наступного дня тим же автобаном повернулися назад, звідти — ще півтори години літаком.

Вхід: От читаєш такі новини і жалкуєш, що населенню України Господь дав все, крім совісті і мозгів.
Вихід: От читаєш такі новини і жалкуєш, що населенню України Господь дав усе, крім совісті і мозгів.

Поверни ЛИШЕ виправлений текст. *('Return ONLY the corrected text.')*

*English: The instruction structure is the same as optimized-v1 (A.5.1); see the translation there. This version includes additional few-shot examples (lines 6–16 above) and was further refined on Gemini 3-Flash. Keywords:* Виправляй = *'Fix';* НЕ змінюй = *'Do NOT change';* Приклади = *'Examples';* Вхід/Вихід = *'Input'/'Output'.*

## B Inference Pipeline and Structured Output

Each input sentence is processed independently through a single LLM call, with no cross-sentence batching. Before any model invocation, the pipeline applies a lightweight passthrough rule: lines matching the document-marker pattern `# <digits>` (e.g. `# 0001`) are emitted verbatim and never sent to the LLM. These markers delimit document boundaries in the UA-GEC corpus and carry no correctable content, so routing them around the model both saves tokens and prevents spurious edits.

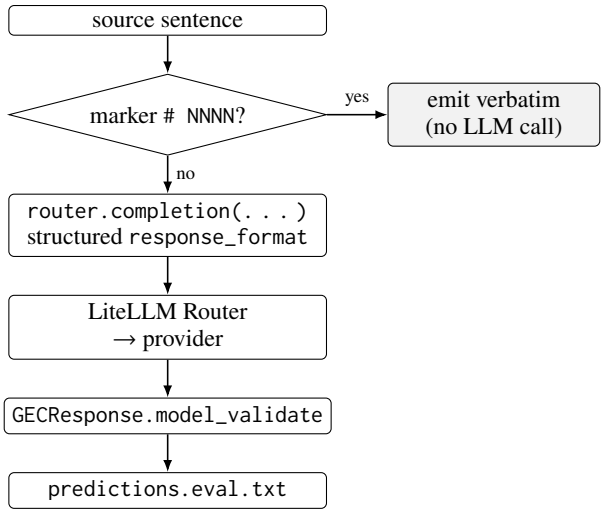
For all remaining sentences, the agent issues one chat completion through a LiteLLM router that abstracts over the underlying provider (OpenAI, Anthropic, Google, Moonshot). The system message is the configured prompt template; the user message is the raw source sentence. To eliminate free-form post-processing of model output, we constrain the response with a JSON schema derived from a Pydantic model and attached to the request as a strict `response_format`. Field descriptions declared on the Pydantic model propagate into the schema and act as in-band instructions to the model.

**Schema definition.** The response contract is declared once as a Pydantic class:

```
class GECResponse(BaseModel):
    corrected_sentence: str = Field(
        ..., description="Corrected version of
            the input sentence")
```

**Generated JSON schema.** At call time, the class is converted to JSON Schema, all object nodes are closed with `additionalProperties: false`, and the result is wrapped into the provider-agnostic `response_format` envelope:

```
{
  "type": "json_schema",
  "json_schema": {
    "name": "GECResponse",
    "strict": true,
    "schema": {
      "type": "object",
      "additionalProperties": false,
      "required": ["corrected_sentence"],
      "properties": {
        "corrected_sentence": {
          "type": "string",
          "description": "Corrected version of
              the input sentence"
        }
      }
    }
  }
}
```




| Parameter | Value (from run YAML) |
|---|---|
| model | gpt-4.1-mini |
| temperature | 0.0 |
| top_p | 0.1 |
| reasoning_effort | None |
| timeout | 90 s |
| response_format | json_schema(GECResponse) |

Figure 3: Per-sentence inference flow. Document markers bypass the LLM; all other sentences go through one structured-output call. Decoding parameters are read verbatim from the run YAML, ensuring deterministic replay.

The decoding parameters in Figure 3 are read from the run YAML and the exact configuration file is copied into the output directory alongside `results.json`, so a run can be replayed bit-for-bit given the same provider model snapshot. The returned payload is validated with `model_validate`, so any schema violation is caught deterministically rather than being masked by string heuristics. If a provider rejects `json_schema`, the client transparently retries with `response_format = {"type":"json_object"}`; for the single-field case, a final recovery path extracts the corrected sentence from malformed JSON to keep evaluation aligned. This design ensures that every non-marker sentence yields exactly one validated correction, making the sentence-to-prediction mapping bijective and the run reproducible given a fixed configuration.