FinnGEC: Benchmarking Grammatical Error Correction for Finnish

Anh-Duc Vu,[†‡] **Mikhail Zolotilin,**[‡] **Jue Hou,**[†‡]
Anisia Katinskaia,[◇] **Yiheng Wu,**[‡] **Roman Yangarber**[‡]
[†]Department of Computer Science, University of Helsinki, Finland
[‡]Department of Digital Humanities, University of Helsinki, Finland
[◇]Swiss Data Science Center, EPFL & ETH Zürich, Switzerland
firstname.lastname@helsinki.fi anisia.katinskaia@epfl.ch

Abstract

Grammatical error correction (GEC) is a natural language processing task critical for improving language quality, supporting communication efficacy, and for language learning and teaching. To date, most research in GEC has focused on major, resource-rich languages such as English, while lower-resource languages remain underexplored. In this paper, we focus on GEC for Finnish. We build a dataset based on data from real-world language learners. We explore various approaches to GEC, including fine-tuning transformer models and zero-shot LLM prompting. We also adapt ERRANT, a popular GEC evaluation tool, for the Finnish language, to evaluate the performance of the models. Our results indicate that the performance of GEC for Finnish is promising, but requires further research. To the best of our knowledge, this is the first in-depth exploration of GEC for Finnish; we provide benchmarks, datasets, and code for GEC for Finnish—by releasing our training and test data and the code for Finnish ERRANT—to support further research on this important task.

1 Introduction

Grammatical error correction (GEC) is important in many areas of natural language processing (NLP), including tools for improving language quality, supporting effective communication, and language learning and teaching. Writing is a fundamental skill in second-language (L2) learning. A common approach to support L2 learners and teachers is to employ GEC systems to provide feedback to learners on samples of their writing. The objective of GEC is to identify grammatical errors, spelling errors, punctuation errors, stylistic inaccuracies, and many other types of errors, and propose corrections for them (Bryant et al., 2023). Research on GEC is decades-old, and many approaches, models, and datasets have been proposed (Bryant et al., 2019; Rothe et al., 2021; Cao et al., 2025; Deng et al., 2025; Park et al., 2025). However, most studies to date focus on major languages, such as English, because of their rich resources, while GEC for lower-resource languages is still underexplored.

In this paper, we focus on the GEC task for Finnish, which suffers from the scarcity of GEC data: high-quality annotated data for training and testing GEC models is difficult to collect for Finnish. This complicates the training and evaluation of GEC models in Finnish. To tackle this problem, we build GEC datasets based on language-learning data from real learners. We experiment with various approaches, including fine-tuning transformer models and prompt-based methods. We adapt ERRANT (Bryant et al., 2017)—a GEC performance metric widely used, e.g., for English—to the Finnish language, and evaluate the performance of various models with our datasets. To the best of our knowledge, this is the first work that provides benchmarks for the GEC task for Finnish.

The main contributions of this paper are as follows:

- We build a Finnish GEC dataset based on real language-learning data. The dataset is based on actual errors made by real-world learners of Finnish as L2. We release this dataset to facilitate future research on GEC.
- We experiment with a number of approaches to GEC for Finnish. This provides benchmarks for future research on this task.
- We adapt ERRANT to the Finnish language and use it to evaluate model performance on our datasets. We also release the Finnish ERRANT to the research community.

The paper is organized as follows. We review related work in Section 2. We describe the sources of our data and the pre-processing steps in Section 3. We discuss our GEC methodology in Section 4. We present our experimental setup and analyze the

Proceedings of the 21st Workshop on Innovative Use of NLP for Building Educational Applications, pages 479–491

results in Section 5. We conclude and outline directions for future work in Section 6.

2 Related Work

GEC is typically approached as a monolingual sequence-to-sequence task, where an incorrect sentence is rewritten into its corrected version. This formulation enables the application of recurrent neural architectures, including machine translation (MT) models (Ji et al., 2017; Junczys-Dowmunt et al., 2018; Zhao et al., 2019; Rothe et al., 2021). Since the emergence of transformer models, multilingual encoder-decoder models pre-trained at scale, such as mT5 (Xue et al., 2021), have become the dominant backbone for both monolingual and multilingual GEC (Rothe et al., 2021). Given the data-intensive nature of these methods, the models are commonly pre-trained on *synthetic* datasets generated through various data-augmentation strategies, i.e., random word perturbations, Wikipedia edits, back-translation, and error-pattern imitation (Kiyono et al., 2019; Grundkiewicz et al., 2019; Takahashi et al., 2020). More recently, Wang et al. (2024) proposed contextual data augmentation to ensure more consistent error distributions in the synthetic data.

An alternative line of work formulates GEC as a sequence *tagging* task, where sentences are annotated with token transformation labels and subsequently edited based on predicted transformations (Malmi et al., 2019; Stahlberg and Kumar, 2020; Omelianchuk et al., 2020; Tarnavskyi et al., 2022). Earlier work also explored adversarial training for GEC: Raheja and Alikaniotis (2020) used a Generative Adversarial Network (GAN)-based setup, where an MT-based generator corrects errors, while a discriminator evaluates correction quality. Parnow et al. (2021) combined a sequence labeler with an error detector in a similar fashion. Li and Wang (2024) showed that explicitly decoupling error detection and correction improves performance in general language models.

Large Language Models for GEC: The emergence of LLMs has significantly impacted GEC research. GPT-3 has been shown to effectively perform GEC, generating fluent corrections that often go beyond reference corrections (Loem et al., 2023). Yet LLMs are also prone to over-editing, which can vary across languages and cause semantic shift (Katinskaia and Yangarber, 2024). Alignment-enhanced methods have been proposed to mitigate over-corrections in both sequence-to-sequence and decoder-only LLMs, for example by training models on both good and bad corrections to instruct the model to prefer the good ones (Yang and Quan, 2024; Liang et al., 2025).

Recent work has also explored LLM-based generation of synthetic data for GEC. Luhtaru et al. (2024) fine-tuned Llama 2 models for artificial error generation, producing synthetic errors that closely resemble human-made errors and achieving state-of-the-art results on German, Ukrainian and Estonian. Potter and Yuan (2024) extended this approach to code-switched text (text mixing multiple languages) generation, addressing the challenge of correcting errors in multilingual English as a Second Language (ESL) learner writing. Koo et al. (2024) proposed Knowledge-Augmented GEC (KAGEC), incorporating external knowledge retrieval to enhance LLMs' context-aware generation capabilities, particularly for non-English languages, including Korean.

Low-Resource and Multilingual GEC: For low-resource scenarios, Korotkova et al. (2019) investigated zero-shot GEC using Transformer-based MT models. Cross-lingual transfer has emerged as a promising direction, leveraging multilingual pre-trained representations to transfer correction capabilities across languages (Yamashita et al., 2020; Sun et al., 2022). Palma Gomez and Rozovskaya (2025) recently proposed selective data augmentation with round-trip MT (translating text into an intermediate language and back), and demonstrated substantial improvements for Russian and Ukrainian GEC. The MultiGEC-2025 shared task (Masciolini and Volodina, 2025)[1] covered 12 languages, including Estonian, the closest relative of Finnish in the Uralic language family.

Finnish GEC remains a low-resource problem with limited dedicated infrastructure. The closest related language with established GEC support is Estonian, which has recently benefited from LLM-based error generation (Luhtaru et al., 2024). For Finnish, Creutz (2024) compared GPT-3.5, GPT-4, and Claude on authentic beginner-level learner texts, finding that GPT-4 achieves the best correction performance among these models.

Evaluation and Quality Control: Automatic evaluation of GEC systems relies on gold-standard reference-based metrics computed over *edits*—

[1] https://spraakbanken.gu.se/en/compsla/multigec-2025

words or phrases that have been changed, added, or removed when comparing the model's output and the reference. The M2 scorer (Dahlmeier and Ng, 2012) is an early standard. It first aligns the source sentence (to be corrected) with both the model's output and the reference to identify which tokens were changed, inserted, or deleted, and then checks how many of the model's edits match the reference edits. ERRANT (Bryant et al., 2017) extended this approach by automatically labeling each aligned edit with an *error type*, enabling fine-grained evaluation at the level of error types, and it has become a widely accepted framework for GEC evaluation. An alternative approach avoids the alignment step by computing an N-gram F-score directly between the hypothesis and the reference without pairwise alignment, making it computationally more efficient (Koyama et al., 2024).

Kobayashi et al. (2024) demonstrated that GPT-4 achieves state-of-the-art performance as a GEC evaluator, surpassing traditional automatic metrics in correlation with human judgments. A post-processing approach has also been explored, where a classifier judges whether each proposed correction is grammatically acceptable and filters out invalid ones (Cao et al., 2024). Goto et al. (2025b) proposed a ranking-based aggregation procedure that better aligns with human judgments across metric types. To support reproducible comparisons, Goto et al. (2025a) released `gec-metrics`, a unified library implementing a wide range of GEC evaluation metrics through a consistent interface.

3 Data

3.1 Corpus of Learner Errors

We begin by building a corpus of errors from data collected from learners of Finnish as L2, who use Revita, a language learning platform (Hou et al., 2025). The platform offers learners the possibility to practice with authentic real-world texts. The platform offers multiple-choice and "cloze" (fill-in-the-blank) exercises, where the base form—*lemma*—of a word is given to the learner, and serves as a hint in the context of 2–3 sentences. We refer to the complete exercise context as a "snippet." The learner's task is to produce the correct form of the word that best suits the context.

We build the *learner-error* corpus by collecting *actual observed* errors from learner data. This allows us to collect a dataset with two principal advantages. First, our dataset covers multiple contexts and error types, including subject-verb agreement, tense, preposition, noun phrase agreement, government relations, etc. Errors may appear in any position in the sentence. This is in contrast to other existing GEC datasets, such as BLiMP (Warstadt et al., 2020), which has the following limitations:

- they target only certain error types, such as Subject-Verb Agreement (SVA);
- each instance can contain only one erroneous word, and the word is tagged with only one error type;
- this also means that some of the errors appear in a fixed role in a sentence: e.g., in SVA, only the predicate can contain an error.

This means that the set of errors observable in BLiMP is severely restricted by the rules used in their collection process.

Secondly, as different learners practice with the same lemma, they commit different mistakes, including spelling, grammatical, and syntactic errors. Further, a particular mistake made by a learner may contain multiple grammatical errors; e.g., it is quite common that a learner chooses an incorrect tense, person, and number all at the same time. We believe this is a more realistic scenario, and can better test the true performance of a GEC model.

We should note that our work on GEC is driven by its applications for language teaching and language learning (Katinskaia and Yangarber, 2021, 2023; Katinskaia et al., 2019). In L2 teaching, having robust GEC is crucial for several reasons. One is to support the generation of meaningful feedback for text (e.g., essays) produced by learners. When learners perform exercises, GEC is used to detect *alternative-correct* answers, also called multiple-admissible answers. Multiple admissibility occurs when more than one form of a word fits syntactically and semantically in a given context. In such situations, it is crucial that a model can **acknowledge** the correctness of alternative answers.

The learner-error corpus contains approximately 3.7K snippets. Each snippet contains one or more errors. Overall, there are approximately 23K distinct errors from 890 distinct texts in the dataset.

The exercise types from the learning platform are cloze and multiple-choice tasks. Therefore, the learner-error corpus contains primarily morphological substitution errors. However, since some exercises involve multi-word answer slots (e.g., compound verb forms), the dataset also contains a small number of insertion and deletion errors

# of errors	Proportion in (%) learner data	Wiki data
0	52.6	50.0
1	19.2	18.1
2	13.8	18.4
3	6.1	9.7
4	3.0	2.0
5	1.8	1.2
≥6	3.5	0.7

Table 1: Training data: proportion of snippets with different numbers of errors.

arising from word count differences between the learner response and the reference.

3.2 From Learner Errors to Instances

We next describe the process of building the datasets of *instances* for training and testing models for GEC, from the learner-error corpus. Each instance is a "parallel" pair: a snippet (sequence of words and punctuation) containing some errors, paired with a *reference*—an error-free version of the same snippet. While a snippet with errors may be corrected in *multiple ways* in real life, we provide only a single error-free reference. The reference corresponds exactly to the correct snippet found in the original text with which the learners were practicing.

We split the snippets into three sets: approximately 1K snippets reserved for testing, and the remaining 2.7K snippets split into a training set (85%) and a validation set (15%)—at the snippet level: this means that *all instances* derived from a given snippet (see below) always fall into the same split, and are never distributed across more than one split. These three splits are randomly sampled from the error corpus. To ensure that there is no leakage across the splits, we filtered the validation and test sets to remove any snippets with identical references in the training set.

Each snippet contains one or more error *slots*—positions where some learner has made a mistake. Each error slot may have multiple distinct incorrect variants—committed by different learners at different times. We generate GEC instances by combining different error slots within each snippet. For example, a snippet with two error slots, each having 4 distinct learner-error variants, yields 25 instances in total: 16 two-error instances formed by pairing each variant from the first slot with each variant from the second slot ($4 \times 4 = 16$), 8 single-error instances formed by combining each variant from one slot with the correct form of the other ($4 + 4 = 8$), and one "clean" instance where both slots are filled with the correct form (which constitutes a *negative* instance for GEC).

For the test set, we apply this combinatorial expansion fully, to maximize the coverage of error types. When applied to the 1K test snippets, this yields approximately 70K instances. We downsample the dataset to 9,981 instances using stratified sampling. The idea is to preserve the distribution of error density—the ratio of error slots to sentence length—while also limiting the number of instances assembled from the same source snippet. This prevents snippets with more error slots from dominating over snippets with fewer error slots, ensuring a more balanced representation across source texts. Lastly, we supplement the test set with zero-error instances sampled directly from texts in the language learning platform, to evaluate accurately the over-correction rate (OC). This results in a total of 17,854 instances in the test set.

For the training and validation sets, we apply a different expansion strategy to maximize the contextual diversity for each error pattern. We first scan a snippet by its sentence boundaries and segment the snippet into overlapping windows of different sentence sizes. A window may range from one standalone sentence to multiple contiguous sentences. For example, a 3-sentence snippet would yield six possible windows: three single-sentence windows, two two-sentence windows, and one full three-sentence window. We always include the full snippet window; we randomly select another 3 from the remaining windows, so each snippet contributes at most 4 training pairs. This allows us to generate training instances that vary in size and avoid high similarity due to overlapping contexts. Too much overlap in the training data would risk that the models would try to memorize the contexts.

For each window, we identify all error slots that fall within the window and apply a set of substitutions for each slot, but not necessarily for all slots. This means only a subset of the slots may be filled with their incorrect variants, while the others remain untouched. To maintain diversity of error combinations, the same error slot, even if it appears in multiple windows, will be substituted only once with an incorrect variant across all windows of the same snippet. We expect that this approach will serve to build a more robust model.

We also include an error-free instance from each snippet. The goal is to prevent the model from

Source	Training	Validation	Test
L2 learner data	10,668	1,372	17,854†
Wikipedia data	1,850,000	150,000	—

Table 2: Dataset statistics after expansion and augmentation. †The test set includes 9,981 error-containing instances and additional zero-error instances drawn from the platform, used to evaluate the over-correction rate.

over-correcting: proposing corrections regardless of whether an error is present. Additionally, we supplement the training and validation sets with clean zero-error instances drawn directly from the platform—following the same approach used for the test set—resulting in approximately 52.6% zero-error instances in the training data, to reduce over-correction. In total, the training set contains 10,668 pairs and the validation set contains 1,372 pairs. The distribution of errors is shown in Table 1.

We note that error-correction pairs in our data are not one-to-one: the same erroneous form may map to different correct forms depending on the context, and vice versa. The model must therefore learn context-sensitive correction rather than memorizing fixed error pairs.

3.3 Augmentation with External Corpus

Although our dataset covers a diverse range of actual error types, composing instances by error combinations would introduce a bias into the training process: the same reference snippet may appear in multiple training pairs, causing the model to overfit to specific snippet contexts rather than learning generalizable correction patterns. To mitigate this, we augment the training data with an external corpus.

We augment the training data with sentences from Finnish Wikipedia (1.2M documents), split into snippets of 1–3 consecutive sentences (8–80 words). We retain only snippets that contain words or phrases for which real learner errors exist in the learner-error corpus. For each snippet, several such words or phrases are selected at random and replaced with incorrect variants drawn from that corpus, producing erroneous training pairs. We also include a large number of clean, error-free pairs to balance the dataset and discourage the model from over-correcting. This yields 1,850,000 training and 150,000 validation pairs (Table 2).

The actual error count distribution in training is shown in Table 1.

The distribution of errors in the validation set is nearly the same. This allows us to have much more training data with diverse contexts and references.

4 Methodology

4.1 LLM Prompting

Recent advances in LLMs have shown very strong performance on various NLP tasks. Researchers have been exploring the potential of LLMs for GEC in particular (Staruch et al., 2025; Davis et al., 2024; Katinskaia and Yangarber, 2024). We consider the prompt-based approach to be a strong baseline for the GEC task. In this work, we focus on the zero-shot setting and prompt the LLMs with only one instance at a time. The prompt is presented in Appendix B. We focus the prompt on errors in morphology and agreement, as these constitute the dominant error types in our dataset, as described in Section 3. We plan to explore the few-shot setting in future work.

In this work, we use OpenAI GPT-5.4 and Claude Haiku 4.5 as prompt-based baselines, and evaluate their performance with the full balanced test set—17,854 instances.

4.2 Fine-tuning Encoder-Decoder and Decoder-Only Models

We experiment with two encoder-decoder transformer models. **FinT5** is a 260M-parameter monolingual T5 model[2] pretrained exclusively on Finnish text, giving it strong morphological knowledge of Finnish. **mBART** (Tang et al., 2020) is a 611M-parameter multilingual model pretrained on 50 languages, with Finnish accounting for only a small fraction of its pretraining data. We also experiment with **Poro2** (Zosa et al., 2025), a decoder-only model pretrained on Finnish and English with 8B parameters. More specifically, we experiment with Poro2-Instruct, created through supervised fine-tuning (SFT) and Direct Preference Optimization (DPO) of the Poro 2 8B Base model.

We adopt a two-stage training process for all models to mitigate overfitting caused by repeated error combinations in our L2 learner dataset:

1. **Stage 1—Fine-tune with augmented data:** We fine-tune the models on synthetic Wikipedia data with artificially injected learner errors. This aims to teach the model to perform GEC across diverse contexts.

2. **Stage 2—Fine-tune with L2 learner data:** We continue fine-tuning on the L2 learner

[2] Finnish-NLP/t5-small-nl24-finnish

Error type	Meaning	Examples (with description)
VERB:FORM	Verb Form	sanoa → sanon (infinitive vs. finite verb)
VERB:TENSE	Tense	menee → meni (present vs. past)
VERB:VOICE	Voice	sanon → sanotaan (active vs. passive)
VERB:MOOD	Mood	tulee → tulisi (indicative vs. conditional)
VERB:PART	Participle	juokseva → juossut (present vs. past)
PERSON	Person	sanon → sanot (1 person vs. 2 person)
DEGREE	Degree	iso → isompi (positive vs. comparative)
CASE	Case	kaupunki → kaupungin (nominative vs. genitive)
NUMBER	Number	koira → koirat (singular vs. plural)
NOUN_POSS	Possessive	koira → koirani (“dog” vs. “my dog”)
CLITIC	Clitic	-kin, -kaan, -han, etc. (incorrect clitics)
AGR	Agreement	(Phrase-level agreement errors)
WO	Word Order	(Word order errors)
WORD_CHOICE	Word Choice	(same POS, different lemma)
CONJ	Conjunction	ja → mutta
PART	Particle	(Particle errors)
VOWEL_HARM	Vowel Harmony	talossa → talossä
COMPOUND	Compound	autotalli → auto talli (compound joined vs. split)
SPELL	Spelling	(Typos)
PUNCT	Punctuation	! → ?
INSERT	Insertion	(Missing words)
DELETE	Deletion	(Removed words)
OTHER	Other	—

Table 3: Error types for Finnish GEC.

dataset to adapt the model to the distribution of real Finnish learner errors in practice.

For FinT5 and mBART, both stages perform full parameter fine-tuning without layer freezing, while Poro2 is trained with LoRA adapters (Hu et al., 2022) on all linear layers. In Stage 1, each model is trained for up to 2 epochs on the augmented Wikipedia data, with the best checkpoint selected according to validation loss. In Stage 2, training continues from the Stage 1 final version and uses only the learner data. Stage 2 is configured to run up to 5 epochs, and to retain the best checkpoint according to validation loss. More detailed information about the hyperparameters is available in Appendix A.

4.3 Evaluation with Finnish ERRANT

ERRor ANnotation Toolkit (ERRANT) (Bryant et al., 2017) is widely used for evaluating GEC models. It requires reference corrections, and measures the performance in terms of an edit-based F-score. Besides computing scores for the overall performance, it also evaluates the different error types. The original ERRANT is designed for English and is developed with 25 English error types. In our work, we extend ERRANT according to the Finnish grammar and common error types. We present our error types in Table 3. We apply the Stanza parser (Qi et al., 2020) to parse the sentences and extract information about the errors.

We adapt the Finnish ERRANT with multi-label annotation. Each edit can be tagged with multiple error types simultaneously, reflecting the compounding nature of Finnish morphological errors. It is common for a Finnish learner to make a mistake with multiple errors at the same time. For example, a learner can submit the answer *sanon* ("I say," indicative mood, present tense 1st person singular), while the correct answer is *sanoi-vat* ("they said," past tense 3rd person plural). In this case, we observe three errors within one edit: tense, person, and number. This adaptation treats the evaluation as a multi-label classification problem and avoids the information loss inherent in the single-label tagging scheme as in the native ERRANT.

Before the evaluation, all texts are normalized: Unicode NFC normalization, full-width character normalization, quote character normalization, and whitespace normalization. This avoids penalizing models for surface formatting differences unrelated to grammatical correctness.

We evaluate all models using precision (P), re-

Model	Architecture	Precision	Recall	$F_{0.5}$	OC %
FinT5 (260M)	encoder-decoder	33.89	17.44	28.52	23.4
mBART (611M)	encoder-decoder	50.46	9.71	27.44	3.6
Poro2 (8B)	decoder-only	36.52	24.30	33.18	13.0

Table 4: Stage-1 performance after fine-tuning with augmented Wikipedia data. OC—over-correction rate

Model	Architecture	Precision	Recall	$F_{0.5}$	OC %
OpenAI GPT-5.4	—	47.27	60.10	**49.37**	57.0
Claude Haiku 4.5	—	39.23	42.31	39.81	47.9
FinT5 (260M)	encoder-decoder	48.17	40.25	**46.35**	23.4
mBART (611M)	encoder-decoder	51.13	22.84	40.98	9.9
Poro2 (8B)	decoder-only	40.54	43.36	41.07	22.6

Table 5: Stage-2 Performance of prompt-based (top), and fine-tuned (bottom) models after fine-tuning on L2 learner data. OC—over-correction rate.

call (R), and $F_{0.5}$:

$$F_\beta = \frac{(1+\beta^2) \cdot P \cdot R}{\beta^2 \cdot P + R} \quad (1)$$

with $\beta = 0.5$ to weight precision twice as heavily as recall, following standard GEC practice (Bryant et al., 2017). We also report the over-correction rate (OC), defined as the proportion of error-free input sentences for which the model attempts to modify something.

5 Experiments and Results

5.1 Fine-Tuning Performance

Table 4 presents the performance after Stage 1 fine-tuning on augmented Wikipedia data. Poro2 achieves the best $F_{0.5}$ of 33.18% at this stage. Among the encoder-decoder models, FinT5 achieves $F_{0.5}$ of 28.52% with an over-correction rate of 23.4%. mBART achieves the highest precision (50.46%) and the lowest over-correction rate (3.6%), suggesting it is more conservative in making corrections.

Table 5 presents the performance after Stage 2 fine-tuning on L2 learner data, alongside the prompt-based baselines. All fine-tuned models improve substantially over Stage 1. FinT5 achieves $F_{0.5}$ of 46.35% and mBART achieves 40.98%. Poro2 reaches 41.07%, slightly outperforming mBART.

Among the prompt-based approaches, GPT-5.4 achieves the best overall $F_{0.5}$ of 49.37%, outperforming all fine-tuned models, while Claude Haiku 4.5 reaches $F_{0.5}$ of 39.81%.

5.2 Error Analysis

Table 6 shows the full breakdown by $F_{0.5}$ for error types for all models. Vowel harmony errors (VOWEL:HARMONY) are the easiest to correct across all models, with all models scoring above 72%. Poro2 achieves the highest $F_{0.5}$ of 86.96%, followed by FinT5 at 83.80%. Claude Haiku has the lowest performance on the vowel harmony errors with 72.14% $F_{0.5}$.

Verb form errors (VERB:FORM) are also relatively easy to correct, with FinT5 achieving $F_{0.5}$ of 75.22% and Poro2 at 69.86%. mBART and Claude Haiku also perform competitively at 60.57% and 61.78%, suggesting that verb form correction is a tractable task for most models as it can be learned from local morphological patterns.

Case errors (CASE), on the other hand, require a deeper understanding of Finnish syntax.[3] This is reflected by clear performance gaps between models. GPT-5.4 achieves the highest $F_{0.5}$ of 70.15% on case errors, followed by FinT5 at 63.09%. Claude Haiku and Poro2 both reach around 57%, while mBART reaches only 45.42%. This highlights the advantage of Finnish-specific pretraining. Both FinT5 (260M, Finnish-specific) and Poro2 (8B, Finnish and English) have clearly better performance than mBART (611M, 45.42%), the only multilingual model in the comparison, despite mBART being larger than FinT5.

Insertion, deletion, and punctuation errors (INSERT, DELETE, PUNCT) are the most challenging error types across all models, with most scoring very low on $F_{0.5}$. Unlike morphological er-

[3]We should note that Finnish has an extremely complex nominal case system, with at least 14 cases, possibly more, depending on the chosen theory.

Error Type	GPT-5.4	Claude Haiku	FinT5	mBART	Poro2
VOWEL:HARMONY	81.78	72.14	83.80	77.07	**86.96**
VERB:FORM	67.50	61.78	**75.22**	60.57	69.86
SPELL	**69.56**	60.04	48.71	55.65	54.63
CASE	**70.15**	57.22	63.09	45.42	57.14
DEGREE	**72.89**	70.75	41.79	15.12	44.03
VERB:VOICE	58.99	55.69	**60.86**	34.15	53.29
VERB:PART	**62.88**	52.84	50.70	36.67	44.64
NUMBER	**61.68**	50.50	57.30	44.82	52.78
CLITIC	**55.56**	45.59	48.08	14.85	24.76
WORD:CHOICE	**54.52**	42.69	33.34	27.64	41.75
VERB:TENSE	52.60	43.45	**53.17**	28.71	33.40
PERSON	**59.56**	48.32	38.55	35.71	44.08
OTHER	29.66	28.27	**37.69**	18.81	28.44
VERB:MOOD	27.97	27.90	22.94	**28.14**	18.30
COMPOUND	18.48	**20.71**	12.89	12.23	15.58
DELETE	**7.90**	6.11	2.68	5.98	3.75
INSERT	5.80	8.40	**16.86**	9.96	13.39
PUNCT	1.12	1.38	**2.83**	0.00	1.72

Table 6: $F_{0.5}$ by error type across all models.

rors, which can be corrected by modifying the word form, insertion and deletion errors require the model to determine whether a word is extraneous or missing in the context. This is also expected, as the cloze and multiple-choice exercise format produces primarily morphological substitutions, leaving the models with limited training signal for these error categories.

5.3 Manual FP Inspection

One known limitation of reference-based evaluation (Choshen et al., 2018) is that a correction is penalized if it differs from the reference, even if it is valid. As we are experimenting with a single-reference dataset, this limitation is even more pronounced. To mitigate this issue, we conducted a manual inspection of false positives (FPs) flagged by Finnish ERRANT. Due to resource constraints, we focus on four models: FinT5, mBART, Poro2, and GPT-5.4. We randomly sampled 100 instances from the intersection of FP instances produced by four models. Each FP instance may contain different edits from different models, but they all include false positives flagged by Finnish ERRANT. This reduces the workload but still allows us to inspect our model's behavior comprehensively. A native Finnish expert then judged each instance at the sentence level, assigning either **Correct** (the model made a valid correction that ERRANT penalized) or **Wrong** (the model genuinely made an incorrect correction).

Table 7 shows the results of the manual FP inspection. Our human evaluation shows that 51% (mBART), 61% (Poro2), 65% (FinT5), and 85% (GPT-5.4) of these flagged edits are in fact valid corrections, confirming that ERRANT underestimates true model precision.

The ranking of valid corrections reflects the overall $F_{0.5}$ ranking, suggesting that better-performing models also generate high-quality corrections even when they do not match the references. Furthermore, the large gap between GPT-5.4 (85%) and the best fine-tuned model FinT5 (65%) suggests that reference-based metrics systematically underestimate LLM performance more than fine-tuned model performance. Fine-tuned models are trained to reproduce the reference and tend to stay closer to it, while LLMs generate more diverse but potentially valid corrections.

6 Conclusion

In this paper, we present the first benchmarks for grammatical error correction for Finnish. We construct a dataset from L2 learner data. The dataset covers a diverse range of morphological and syntactic error types. We adapt ERRANT to Finnish to enable a systematic evaluation and benchmark a range of approaches spanning from prompt-based methods to fine-tuning transformer-based models.

Our results show that language-specific pre-

Model	*Valid*	*Invalid*
mBART (611M)	51	49
Poro2 (8B)	61	39
FinT5 (260M)	65	35
OpenAI GPT-5.4	**85**	**15**

Table 7: Manual inspection of 100 false positives flagged by ERRANT. *Valid*: the GEC model made a valid correction. *Invalid*: the GEC model made an incorrect correction.

training is a more impactful factor than model scale. FinT5 (260M parameters) outperforms the larger multilingual mBART model and outperforms Claude Haiku 4.5, a compact commercial LLM, particularly on morphologically complex error types such as case. Poro2, a decoder-only model fine-tuned under the same two-stage setup, achieves competitive results with mBART, demonstrating that decoder-only architectures can also be adapted effectively for Finnish GEC.

Finnish GEC remains an open challenge. For future research, we plan to use few-shot prompting techniques to reduce the over-correction problem and to further improve LLM-based GEC performance.

In terms of data preparation, expanding the training data to cover more error types, such as insertion, deletion, and punctuation, is an important direction. Our results show that the models currently receive insufficient training for these error categories. Scaling up training data through systematic synthetic error generation, either via LLM-based error injection or knowledge distillation from frontier models, is a potential direction for improving fine-tuned model performance.

In addition, we also plan to experiment with larger models. FinT5 already achieves competitive performance with compact commercial LLMs such as Claude Haiku 4.5 despite its small size, and experimenting with a larger Finnish-specific model may close the gap to frontier LLMs further.

Acknowledgements

This work was supported in part by the Research Council of Finland, Project "Know-AI" (Grant 1359285), and by Strategic Research Council Project "DALAI-Fin" (Grant 1373225).

Data Availability

The full training dataset, derived from learner data collected via Revita, is available upon request by contacting the authors, subject to licensing constraints from the original data.

Limitations

Our dataset is sourced from a single language learning platform, which may limit generalization to other Finnish L2 learner populations. The raw L2 learner dataset is also relatively small, with 10,668 training, 1,372 validation, and 9,981 error-containing test pairs. Each instance provides only a single reference correction, though multiple valid corrections may exist for the same error. The dataset consists primarily of morphological substitution errors, as the cloze and multiple-choice exercise format produces mainly morphological substitutions; insertion, deletion, and punctuation errors are underrepresented, limiting the training signal available for these error categories.

Evaluation relies solely on Finnish ERRANT, an automatic metric adapted from English. For prompt-based approaches, only zero-shot prompting was evaluated, leaving few-shot settings for future work. Sequence-labeling approaches such as GECToR (Omelianchuk et al., 2020) are out of scope for this benchmark paper and left for future work.

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.

Bin Cao, Kai Jiang, Fayu Pan, Chenlei Bao, and Jing Fan. 2024. Improving grammatical error correction by correction acceptability discrimination. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 8818–8827, Torino, Italia. ELRA and ICCL.

Yayu Cao, Tianxiang Wang, Lvxiaowei Xu, Zhenyao Wang, and Ming Cai. 2025. CxGGEC: Construction-guided grammatical error correction. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6143–6156, Vienna, Austria. Association for Computational Linguistics.

Leshem Choshen, Lior Bar, and Omri Abend. 2018. Inherent biases in reference-based evaluation for grammatical error correction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 632–641, Melbourne, Australia.

Mathias Creutz. 2024. Correcting challenging Finnish learner texts with claude, GPT-3.5 and GPT-4 large language models. In *Proceedings of the Ninth Workshop on Noisy and User-generated Text (W-NUT 2024)*, pages 1–10, San Ġiljan, Malta. Association for Computational Linguistics.

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.

Jiayi Deng, Chen Chen, Chunyan Hou, and Xiaojie Yuan. 2025. InstructGEC: Enhancing unsupervised grammatical error correction with instruction tuning. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 110–122, Abu Dhabi, UAE. Association for Computational Linguistics.

Takumi Goto, Yusuke Sakai, and Taro Watanabe. 2025a. gec-metrics: A unified library for grammatical error correction evaluation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pages 524–534, Vienna, Austria. Association for Computational Linguistics.

Takumi Goto, Yusuke Sakai, and Taro Watanabe. 2025b. Rethinking evaluation metrics for grammatical error correction: Why use a different evaluation process than human? In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1165–1172, Vienna, Austria. Association for Computational Linguistics.

Roman Grundkiewicz, Marcin Junczys-Dowmunt, and Kenneth Heafield. 2019. Neural grammatical error correction systems with unsupervised pre-training on synthetic data. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 252–263, Florence, Italy. Association for Computational Linguistics.

Jue Hou, Anh-Duc Vu, Anisia Katinskaia, and Roman Yangarber. 2025. AI-assisted second-language teaching and learning in the Zone of Proximal Development. *Studies in language assessment*, 14(2):92–122.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *ICLR*.

Jianshu Ji, Qinlong Wang, Kristina Toutanova, Yongen Gong, Steven Truong, and Jianfeng Gao. 2017. A nested attention neural hybrid model for grammatical error correction. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 753–762, Vancouver, Canada. Association for Computational Linguistics.

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

Anisia Katinskaia, Sardana Ivanova, and Roman Yangarber. 2019. Multiple admissibility: Judging grammaticality using unlabeled data in language learning. In *Proceedings of the 7th Workshop on Balto-Slavic Natural Language Processing*, pages 12–22, Florence, Italy.

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

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

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.

Shun Kiyono, Jun Suzuki, Masato Mita, Tomoya Mizumoto, and Kentaro Inui. 2019. An empirical study of incorporating pseudo data into grammatical error correction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1236–1242, Hong Kong, China. Association for Computational Linguistics.

Masamune Kobayashi, Masato Mita, and Mamoru Komachi. 2024. Large language models are state-of-the-art evaluator for grammatical error correction. In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, pages 68–77, Mexico City, Mexico. Association for Computational Linguistics.

Seonmin Koo, Jinsung Kim, Chanjun Park, and Heuiseok Lim. 2024. Search if you don't know! knowledge-augmented Korean grammatical error correction with large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 96–125, Miami, Florida, USA. Association for Computational Linguistics.

Elizaveta Korotkova, Agnes Luhtaru, Krista Liin, Maksym Del, Daiga Deksne, and Mark Fishel. 2019. Grammatical error correction and style transfer via zero-shot monolingual translation. *Preprint*, arXiv:1903.11283.

Shota Koyama, Ryo Nagata, Hiroya Takamura, and Naoaki Okazaki. 2024. n-gram F-score for evaluating grammatical error correction. In *Proceedings of the 17th International Natural Language Generation Conference*, pages 303–313, Tokyo, Japan. Association for Computational Linguistics.

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

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.

Agnes Luhtaru, Taido Purason, Martin Vainikko, Maksym Del, and Mark Fishel. 2024. To err is human, but llamas can learn it too. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 12466–12481, Miami, Florida, USA. Association for Computational Linguistics.

Eric Malmi, Sebastian Krause, Sascha Rothe, Daniil Mirylenka, and Aliaksei Severyn. 2019. Encode, tag, realize: High-precision text editing. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5054–5065, Hong Kong, China. Association for Computational Linguistics.

Arianna Masciolini and Elena Volodina. 2025. MultiGEC-2025: Multilingual grammatical error correction shared task. `https://spraakbanken.gu.se/en/compsla/multigec-2025`. NLP4CALL Workshop, NoDaLiDa 2025.

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.

Frank Palma Gomez and Alla Rozovskaya. 2025. Low-resource grammatical error correction: Selective data augmentation with round-trip machine translation. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 25749–25770, Vienna, Austria. Association for Computational Linguistics.

Taehee Park, Heejin Do, and Gary Lee. 2025. Leveraging what's overfixed: Post-correction via LLM grammatical error overcorrection. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 28195–28207, Suzhou, China. Association for Computational Linguistics.

Kevin Parnow, Zuchao Li, and Hai Zhao. 2021. Grammatical error correction as GAN-like sequence labeling. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3284–3290, Online. Association for Computational Linguistics.

Tom Potter and Zheng Yuan. 2024. LLM-based code-switched text generation for grammatical error correction. In *Proceedings of the 2024 Conference on*

Empirical Methods in Natural Language Processing, pages 16957–16965, Miami, Florida, USA. Association for Computational Linguistics.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 101–108, Online. Association for Computational Linguistics.

Vipul Raheja and Dimitris Alikaniotis. 2020. Adversarial Grammatical Error Correction. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3075–3087, Online. Association for Computational Linguistics.

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

Felix Stahlberg and Shankar Kumar. 2020. Seq2Edits: Sequence transduction using span-level edit operations. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5147–5159, 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.

Xin Sun, Tao Ge, Shuming Ma, Jingjing Li, Furu Wei, and Houfeng Wang. 2022. A unified strategy for multilingual grammatical error correction with pre-trained cross-lingual language model. *arXiv preprint arXiv:2201.10707*.

Yujin Takahashi, Satoru Katsumata, and Mamoru Komachi. 2020. Grammatical error correction using pseudo learner corpus considering learner's error tendency. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 27–32, Online. Association for Computational Linguistics.

Yuqing Tang, Chau Tran, Xian Li, Peng-Jen Chen, Naman Goyal, Vishrav Chaudhary, Jiatao Gu, and Angela Fan. 2020. Multilingual translation with extensible multilingual pretraining and finetuning. *Preprint*, arXiv:2008.00401.

Maksym Tarnavskyi, Artem Chernodub, and Kostiantyn Omelianchuk. 2022. Ensembling and knowledge distilling of large sequence taggers for grammatical error correction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3842–3852, Dublin, Ireland. Association for Computational Linguistics.

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

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman. 2020. BLiMP: The benchmark of linguistic minimal pairs for English. *Transactions of the Association for Computational Linguistics*, 8:377–392.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.

Ikumi Yamashita, Satoru Katsumata, Masahiro Kaneko, Aizhan Imankulova, and Mamoru Komachi. 2020. Cross-lingual transfer learning for grammatical error correction. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 4704–4715, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Haihui Yang and Xiaojun Quan. 2024. Alirector: Alignment-enhanced Chinese grammatical error corrector. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 2531–2546, Bangkok, Thailand. Association for Computational Linguistics.

Wei Zhao, Liang Wang, Kewei Shen, Ruoyu Jia, and Jingming Liu. 2019. Improving grammatical error correction via pre-training a copy-augmented architecture with unlabeled data. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 156–165, Minneapolis, Minnesota. Association for Computational Linguistics.

Elaine Zosa, Jouni Louma, Kai Hakala, Antti Virtanen, Mika Koistinen, Risto Luukkonen, Akseli Reunamo, Sampo Pyysalo, and Jonathan Burdge. 2025. Poro 2: Continued pretraining for language acquisition. LumiOpen.

A Training Details

Tables 8–10 report the hyperparameters used for each model across both training stages.

Hyperparameter	Stage 1	Stage 2
Optimizer	Adafactor	Adafactor
Learning rate	5×10^{-5}	1×10^{-5}
Weight decay	0.01	0.01
Batch size	8	8
Grad. accum.	4 (eff. 32)	4 (eff. 32)
Max length	512	512
Warmup steps	500	50
Max epochs	2	5

Table 8: FinT5 hyperparameters.

Hyperparameter	Stage 1	Stage 2
Optimizer	AdamW	AdamW
Learning rate	3×10^{-5}	1×10^{-5}
Weight decay	0.01	0.01
Batch size	8	8
Grad. accum.	4 (eff. 32)	4 (eff. 32)
Max length	512	512
Warmup steps	500	50
Max epochs	2	5

Table 9: mBART hyperparameters.

B Zero-shot prompt

The following text is our prompt for LLM:

```
You are a Finnish grammatical error correction (GEC) system. Your task is to correct grammatical errors in Finnish sentences while making minimal changes.

Rules:
- Fix ONLY grammatical errors (morphology, case endings, verb conjugation, agreement).
- Do NOT paraphrase or change the meaning.
- Do NOT change vocabulary unless the word itself is wrong.
- Make the MINIMAL number of changes necessary.
- Output ONLY the corrected sentence — no explanation, no commentary.
- If the sentence has no errors, output it unchanged.
```

Hyperparameter	Stage 1	Stage 2
Quantization	4-bit	
LoRA rank	64	
LoRA alpha	128	
LoRA target	All Linear	
LoRA Dropout	0.05	
Num. Base Params	8B	
Num. Trainable Params	167M (2.01%)	
Optimizer	AdamW	
Learning rate	2e-4	1e-5
Max epochs	2	

Table 10: Poro2 hyperparameters. *All linear* refers to gate_proj, o_proj, v_proj, q_proj, up_proj, down_proj and k_proj.