To Err Is Human, but Llamas Can Learn It Too

Anonymous ACL submission

Abstract

This study explores enhancing grammatical error correction (GEC) through artificial error generation (AEG) using language models (LMs). Specifically, we fine-tune Llama 2-based LMs for error generation and find that this approach yields synthetic errors akin to human errors. Next, we train GEC Llama models with the help of these artificial errors and outperform previous state-of-the-art error correction models, with gains ranging between 0.8 and 6 $F_{0.5}$ points across all tested languages (German, Ukrainian, and Estonian). Moreover, we demonstrate that generating errors by fine-tuning smaller sequence-to-sequence models and prompting large commercial LMs (GPT-3.5 and GPT-4) also results in synthetic errors beneficially affecting error generation models. We openly release trained models for error generation and correction and all the synthesized error datasets for the covered languages.

1 Introduction

The grammatical error correction (GEC) task aims to correct spelling and grammatical errors in text, making it valuable for a wide range of people. The best-performing GEC approaches currently use deep learning models (Junczys-Dowmunt et al., 2018; Omelianchuk et al., 2020; Rothe et al., 2021, and several others), which are known to be data-hungry. Simultaneously, the amount of openly available error correction data is severely limited, even for high-resource languages like German, Arabic, and Czech (Bryant et al., 2023). This lack of data complicates the development of effective GEC systems for these and other even less-resourced languages.

The scarcity of correction data is commonly addressed through the creation of synthetic data, where errors are automatically added into correct sentences – also called artificial error generation (AEG). In low-resource settings, the overwhelmingly most employed approach for AEG is applying random probabilistic perturbation (deletion, insertion, replacement) of words and/or characters (Grundkiewicz et al., 2019; Rothe et al., 2021; Náplava and Straka, 2019, and others). Alternatives include usage of intricate hand-crafted rules and confusion sets (Rozovskaya and Roth, 2010; Xu et al., 2019; Kara et al., 2023; Bondarenko et al., 2023) and automatically learning to generate errors (Xie et al., 2018; Kiyono et al., 2019; Stahlberg and Kumar, 2021) – also referred to as *back-translation* (BT)[*]. However, to the best of our knowledge, none of the related work on AEG uses pre-trained foundation models or applies this methodology in a low-resource setting.

This gap is precisely the focus of the present work: we are using pre-trained language models for synthetic error generation and demonstrate the simplicity and effectiveness of the approach in low-resource scenarios. We approach the task by fine-tuning open language models (LMs) based on Llama 2 (Touvron et al., 2023) for error generation and correction, resulting in quality AEG data and state-of-the-art GEC models even when very limited human error data is available. Our analysis shows that the resulting errors can be categorized similarly to human errors. We also compare fine-tuning approach to prompting commercial LMs (GPT-3.5 and GPT-4: OpenAI, 2023) to perform AEG, as well as include other open models commonly employed for GEC and tune them for AEG: mT5 (Rothe et al., 2021; Palma Gomez et al., 2023) and NLLB (Luhtaru et al., 2024).

Our final goal and evaluation setting is improving grammatical error correction for low-resource languages. In particular, we focus on German, Ukrainian, and Estonian GEC. Our experimental results show that Llama-based language models with fewer learned parameters can beat state-of-the-art results achieved with a bigger model. When

[*]by analogy with the machine translation technique (Sennrich et al., 2016)

pre-trained on our LM-generated synthetic errors, the resulting GEC models achieve the best current results on the included benchmarks in all three evaluated cases, including previous state-of-the-art and 4-shot GPT-4.

We publicly release AEG and GEC models from our work and the generated data. The datasets include one million sentences for German, Ukrainian, and Estonian, each processed with three different models, as well as an additional set of 100k sentences with GPT models.

In summary, our contributions are as follows:

- We show that pre-trained language models can be fine-tuned to generate high-quality synthetic errors even with limited data.
- We compare the influence of different models applied to AEG (LLama/GPT/mT5/NLLB) on subsequent GEC models.
- We achieve new state-of-the-art GEC results across all tested languages with Llama 2-based models outperforming related work as well as GPT-4.
- We openly release GEC and AEG models as well as AEG datasets to facilitate future research[†].

The paper is structured as follows. We outline related work in Section 2, methodology experimental settings in Section 3, and results in Section 4. Additional questions on the same topic are discussed in Section 6 and the paper is concluded in Section 5.

2 Related Work

The use of synthetic data is a common concept in GEC. The first effective neural method proposed by Junczys-Dowmunt et al. (2018) approaches GEC as low-resource Machine Translation (MT), making it a relatively resource-heavy method encouraging synthetic data generation. Over the years, there have been different approaches to deliberately introducing errors into monolingual text, like rule-based and probabilistic methods, methods based on confusion sets and error patterns, models trained for error generation and using round-trip translation (Bryant et al., 2023).

[†]Models: huggingface.co/anonymous-acl/models, datasets: huggingface.co/datasets/anonymous-acl/aeg_data

One widely adopted approach to generating synthetic data involves the probabilistic addition of errors to monolingual corpora. This technique encompasses inserting, deleting, substituting, or moving characters or words without considering the context, as described by Grundkiewicz et al. (2019), Zhao et al. (2019), and Rothe et al. (2021). Additionally, Grundkiewicz et al. (2019) introduced a "reverse speller" approach that suggests word replacements from confusion sets based on the speller's corrections. This method has been applied to several languages such as German, Czech, Russian, Ukrainian, Icelandic and Estonian (Náplava and Straka, 2019; Trinh and Rozovskaya, 2021; Náplava et al., 2022; Palma Gomez et al., 2023; Ingólfsdóttir et al., 2023; Luhtaru et al., 2024). As we show later, errors generated with the context-free probabilistic method differ from human errors and thus cover a much smaller number of error types, shown by significantly lower GEC recall.

Learned methods of error generation typically require more resources. Before the widespread adoption of transformers and MT, various studies explored alternative approaches for training models for error generation. For instance, Felice and Yuan (2014) and Rei et al. (2017) utilized statistical machine translation to generate errors, while Xie et al. (2018) and Yuan et al. (2019) experimented with convolutional neural networks (CNNs) for this purpose. Additionally, Kasewa et al. (2018) investigated using RNN-based sequence-to-sequence models with attention mechanisms.

Moving towards more modern MT architectures, Htut and Tetreault (2019) tested various model frameworks, including transformers, and Kiyono et al. (2019) specifically employed transformer models. Both of the latter studies trained models from scratch, utilizing datasets ranging from approximately 500,000 to over a million error correction examples to train the artificial error generation system. In contrast, our work generates up to 1 million sentences with synthetic error while using between 9k and 33k human error sentences to fine-tune the base models.

During the last few years, there has been no one error-generation method that has proved its superiority. It depends on language and available resources. For English Stahlberg and Kumar (2021) train Seq2Edit models (Stahlberg and Kumar, 2020) from scratch for learning to create diverse sets of errors. As mentioned in the beginning, synthetic probabilistic errors have found wide use for different languages. For instance, Ingólfsdóttir et al. (2023) combine probabilistic character/word permutations with a rule-based approach for Icelandic and Kara et al. (2023) curate special rules for generating Turkish data.

Next, we present the key methodological details of our work.

3 Methodology and Experiments

The primary target of our work is to apply generative language models to artificial error generation (AEG) via fine-tuning. Additionally, we experiment with prompting large language models to perform the same task and include two seq2seq models that are fine-tuned to do the same.

The efficiency of proposed AEG solutions is evaluated using them to improve grammatical error correction (GEC). Thus, we also fine-tune generative LMs to perform the GEC task and compare the results to prompting-based GEC results and related work.

Our approach's general pipeline is straightforward. First, we fine-tune a language model (LM) to generate errors using human error data, where correct sentences are input and erroneous sentences are output. Next, we use this AEG LM to create synthetically erroneous sentences from correct ones. Then, we fine-tune another LM on this synthetic dataset to correct grammatical errors, reversing the direction of the sentence pairs from the first step. We continue by fine-tuning the GEC LM on a smaller dataset with human errors. Finally, we apply the models to erroneous sentences in benchmark test sets and evaluate the results.

Next, we describe the technical details of our implementation and the experimental setup.

3.1 Data

We use two distinct types of data in our work. Firstly, we rely on datasets containing examples of grammatical error corrections to train our error generation systems and correction models. Secondly, we incorporate monolingual data to create synthetic datasets by introducing errors. See an overview of used data in Table 1.

Corpus	Language	Train	Test
UT-L2 GEC	ET	8,935	-
EstGEC-L2	ET	-	2,029
UA-GEC	UK	31,038	1,271
FM	DE	19,237	2,337
ENC 2021	ET	1M/100k	-
CC-100	UK/DE	1M/100k	-

Table 1: Data used for training and testing.

We use the language learners' corpus from the University of Tartu (UT-L2 GEC) (Rummo and Praakli, 2017) for gold data in Estonian. In Ukrainian, we use the UA-GEC corpus (Syvokon et al., 2023) used in the UNLP 2023 Shared Task on Grammatical Error Correction for Ukrainian (Syvokon and Romanyshyn, 2023), using the GEC+Fluency data for training. For German, we rely on the widely used Falko-Merlin (FM) corpus (Boyd, 2018).

For monolingual Estonian data, we employ the Estonian National Corpus 2021 (Koppel and Kallas, 2022). We randomly sample equal sets from the latest Wikipedia, Web, and Fiction subsets and shuffle these together. For Ukrainian and German, we use the CC-100 dataset (Conneau et al., 2020; Wenzek et al., 2020). Depending on the experiments, we sample the required number of sentences from the larger corpora (i.e., one million or 100 thousand sentences or a set equal to gold corpora sizes).

3.2 Models and Training

Llama-2-based models. We fine-tune models that have been enhanced with bilingual capabilities using continued pre-training from Llama-2-7B (Touvron et al., 2023). For Estonian, we use Llammas-base[‡], and for German, LeoLM[§]. For Ukrainian, we apply continued pre-training to replicate the conditions of Estonian LM by training with 5B tokens from CulturaX (Nguyen et al., 2023) with 25% of the documents being in English and the rest in Ukrainian. For GEC and AEG fine-tuning, we formatted the training data with a prompt (see Table 12 and 13) loosely based on Alpaca (Taori et al., 2023). During fine-tuning, the loss is calculated on the tokens of the correct sentence. Fine-tuning details (including hyperparameters) are discussed in Appendix A.1.

Other models we use are NLLB (Team et al., 2022) and mT5 (Xue et al., 2021). Specifically, we use the NLLB-200-1.3B-Distilled and mt5-large (1.2B parameter) models for our experiments and train NLLB models using Fairseq (Ott et al., 2019)

[‡]huggingface.co/tartuNLP/Llammas-base
[§]huggingface.co/LeoLM/leo-hessianai-7b

and mT5 with HuggingFace Transformers (Wolf et al., 2020). When training in two stages, first with synthetic data and later with human errors, we keep the state of the learning rate scheduler, following the fine-tuning approach rather than retraining as defined by Grundkiewicz et al. (2019). See Appendices A.2 and A.3 for further details.

3.3 Generation

Fine-tuned models. We use sampling instead of beam search to generate the synthetic errors and sample from the top 50 predictions with a temperature of 1.0. During error correction, beam search with a beam size of 4 is used without sampling as regularly.

Prompt engineering. We perform iterative prompt engineering, analyzing intermediate qualitative results and updating the prompt. For instance, we initially started with a simple 2-shot prompt (temperature = 0.1) asking GPT-3.5 to add grammatical and spelling mistakes into the input text but noticed that some error types were missing. We then improved the prompt by specifying the missing error types, adding two more examples, and upping the temperature. Our final prompt uses four examples and a model temperature of 1.0. See Appendix D for the prompts. We randomly pick the examples from each language's train set for few-shot prompting. When comparing the prompting between GPT-4-Turbo and GPT-3.5-Turbo, we use an identical random set of examples to ensure comparability.

Finally, we converged on using GPT-3.5-turbo for more massive error generation (100,000 sentence pairs per language). The motivation for that is partially financial (as GPT-4/GPT-4-turbo are several times more expensive) as well as performance-driven (see Figure 1 and description for details).

We apply simple post-processing to the resulting set because, in some cases, parts from the prompt are duplicated in the output. If the model didn't generate a response due to safety model activation or the response was too short or too long compared to the input sentence, we replaced the output with the source text (equivalent to adding no errors).

The precise model versions we prompt are `gpt-4-1106-preview` for GPT-4-Turbo (using the OpenAI API) and `gpt-3.5-turbo` (GPT-3.5-Turbo) and `gpt-4` (GPT-4) (using Azure OpenAI API, version 0613 for both).

Probabilistic errors. We generate rule-based synthetic errors as done in prior work (Grundkiewicz et al., 2019; Náplava and Straka, 2019; Palma Gomez et al., 2023; Luhtaru et al., 2024) using the same method and also employing the Aspell speller[¶] for replacing subwords.

3.4 Automatic Evaluation of Models

We evaluate the performance of our GEC models using test sets and evaluation metrics consistent with those employed in previous works (see datasets in Table 1).

For Estonian, we evaluate our models using the Estonian learner language corpus (EstGEC-L2)[‖], alongside a modified version of the MaxMatch scorer[**], following Luhtaru et al. (2024). The Estonian scorer also outputs recall per error category, accounting for both other errors within the word order error scope and not accounting for these. We report the ones that do consider other errors separately. For Ukrainian, our evaluation methodology aligns with that of the UNLP 2023 Shared Task (Syvokon and Romanyshyn, 2023), utilizing the CodaLab platform for submissions to a closed test set that uses the ERRANT scorer for evaluation (Bryant et al., 2017). We follow the GEC+Fluency track setting since it encompasses a wider range of challenging errors. For German, we use the test set from the Falko-Merlin (FM) corpus (Boyd, 2018) that several works have reported their scores on and the original MaxMatch scorer (Dahlmeier and Ng, 2012).

3.5 Human Evaluation of Generated Data

In addition to evaluating the quality of our data in terms of its usefulness for training better models, we perform a detailed evaluation of generated data in Estonian. We apply the same annotation scheme Allkivi-Metsoja et al. (2022) used for annotating test and development sets to artificially generated sentences. This comparison allows us to assess the error distribution between training and generated data and to see whether the errors can be categorized into the same classes.

We select 100 random sentences from sets generated by Llama-based models, GPT-3.5-Turbo and GPT-4-Turbo[††], for annotation and also annotate

[¶] `aspell.net`

[‖] `github.com/tlu-dt-nlp/EstGEC-L2-Corpus/`

[**] `github.com/TartuNLP/estgec/tree/main/M2_scorer_est`

[††] We also considered annotating probabilistic denoising errors, but these contained very few edits that could be categorized based on the annotation scheme.

Method	Estonian			Ukrainian			German		
	P	R	$F_{0.5}$	P	R	$F_{0.5}$	P	R	$F_{0.5}$
GPT-4-turbo (4-shot)	70.86	57.35	67.67	39.62	42.13	40.1	64.15	69.34	65.12
GPT-4 (4-shot)	70.04	**59.03**	67.52	36.25	37.77	36.54	65.22	**69.75**	66.08
Old SOTA (rel. work)	71.27	55.38	67.40	79.13	43.87	68.17	-	-	75.96
Llama + gold	71.52	55.23	67.54	79.98	51.76	72.12	76.86	65.60	74.31
Llama + 1M prob + gold	72.59	54.72	68.14	80.37	53.19	72.92	78.22	67.65	75.85
Llama + 1M BT + gold	**73.85**	57.83	**69.97**	**82.03**	**53.41**	**74.09**	**79.08**	68.66	**76.75**

Table 2: Comparison of Llama 2-based models (denoted as Llama) after extended pre-training and GEC fine-tuning: Models without synthetic data (LLM + gold) versus models with synthetic data generated with a probabilistic reverse-speller method (LLM + 1M prob + gold) and back-translation style learned synthetic data (LLM + 1M BT + gold). State-of-the-art benchmarks include Luhtaru et al. (2024) for Estonian (NLLB-200-1.3B-Distilled with mixed synthetic and translation data training), Bondarenko et al. (2023) for Ukrainian (mBART-based model with synthetic data), and Rothe et al. (2021) for German (mT5 xxl with multilingual synthetic data and GEC fine-tuning).

Lang/Model	Llama	NLLB	mT5
ET (AEG only)	65.30	65.34	59.40
ET (AEG + gold)	69.97	69.73	68.57
UK (AEG only)	28.39	27.04	16.79
UK (AEG + gold)	74.09	72.30	72.51
DE (AEG only)	71.29	69.13	54.96
DE (AEG + gold)	76.75	76.28	74.77

Table 3: $F_{0.5}$-scores for Llama-based models fine-tuned with 1M sentences generated with different AEG models and then further fine-tuned with gold GEC data. The errors are generated with 7B Llama-2-based models, 1.3B NLLB model and 1.2B mT5 model.

100 sentences from the training set. We add labels for problematic errors generated by the model, such as hallucinations and truncation of words important for understanding the meaning of sentence (HALL), synonym swaps (SYN), optional edits (O), corrections of mistakes in original sentences (INACC), and transformations that make the original word unrecognizable (UNREC).

4 Results

In this section, we evaluate the performance of Llama-based models for GEC and AEG tasks. We then compare the AEG effectiveness between NLLB and mT5 models against Llama-based models to see if smaller, more efficient models can generate quality data. Separately, we assess AEG through prompting with GPT-3.5-turbo versus Llama models with trained error generation. Finally, we examine the quality of generated errors against human data and probabilistic reverse-speller errors and compare the error type distributions for Estonian.

4.1 Artificial Error Generation and Correction with Llama

We compare LLama-based large language model (LLM) fine-tuning error corrections across three configurations: (1) the baseline approach of training exclusively on human error GEC data, (2) the established related work approach of training on probabilistic reverse-speller AEG data and then continuing training with human error GEC data, and (3) our approach of training on back-translation style AEG data produced by fine-tuned Llama-based models first, followed by fine-tuning on human data.

The resulting scores are compared in Table 2, along with previous state-of-the-art (SOTA) scores and results of GEC via 4-shot prompting of GPT-4/GPT-4-turbo. Results show that llama-based models, further enhanced through continued pre-training, exhibit strong correction capabilities across languages in our study. Even without synthetic data, these models outperform current state-of-the-art (SOTA) methods in Estonian and Ukrainian error correction, and are not too far behind in German, trailing the best score by less than two points. However, it's important to note the discrepancy in model sizes for a fair comparison; our 7B Llama model significantly exceeds the NLLB-200-1.3B-Distilled model (Team et al., 2022) used for Estonian (Luhtaru et al., 2024) and the mBART model (Tang et al., 2021) for Ukrainian (Bondarenko et al., 2023) in size. At the same time,

Lang/Model	Prompting GPT-3.5-turbo (100k)			Fine-tuning Llama (100k)		
	P	R	$F_{0.5}$	P	R	$F_{0.5}$
ET (AEG only)	71.72	44.20	63.78	67.57	50.89	63.41
ET (AEG + gold)	71.11	56.56	67.63	71.51	56.51	67.91
UK (AEG only)	28.61	22.16	27.04	40.00	19.87	33.26
UK (AEG + gold)	80.82	51.33	72.49	80.89	50.31	72.12
DE (AEG only)	70.55	49.61	65.05	70.07	59.11	67.56
DE (AEG + gold)	78.06	67.06	75.58	78.80	67.52	76.25

Table 4: Scores of Llama-based models fine-tuned with 100k sentences generated by Llama-based model fine-tuned for error generation and GPT-3.5-model prompted to add errors.

it is smaller than the 13B mT5-xxl model used for German (Rothe et al., 2021).

Incorporating synthetic data as a preliminary step to fine-tuning significantly enhances performance across all languages and synthetic data types. Notably, our back-translation style synthetic data consistently delivers superior precision and recall compared to the probabilistic reverse-speller method. This approach results in a 2-2.4 point increase in the $F_{0.5}$ score relative to solely using gold data for fine-tuning. Conversely, the gains from using probabilistic reverse-speller data are more modest, ranging from 0.6 to 1.5 points, highlighting the enhanced utility of our learned AEG errors.

Our systems consistently outperform GPT-4 models in terms of precision across all languages studied. However, GPT-4 models exhibit higher recall rates for Estonian and German. This discrepancy indicates that while our systems are more accurate in identifying correct instances, GPT-4 models better retrieve a broader range of relevant errors in these languages. On the other hand, the performance of GPT-4 models on the Ukrainian test set is notably lower compared to other methods and languages.

4.2 Artificial Error Generation with Smaller Models

Since error generation with 7B Llama-based models can be costly and time-consuming and many other architectures have proved useful for correction, we also explore smaller models for AEG: the 1.3B NLLB model and 1.2B mT5-large. The goal here is to see if these can also produce useful errors.

Table 3 shows the results of the analysis. Both models can learn valuable information that improves performance beyond what is achieved with fine-tuning on gold data alone. Notably, errors generated by the NLLB model are particularly effective, delivering results close to those achieved by LLM-generated errors in Estonian and German, almost matching the performance of LLama-based models. However, for Ukrainian, NLLB-generated errors fall behind probabilistic reverse-speller errors. This is likely because the dataset contains many special punctuation characters that get normalized during preprocessing (see more in Appendix C).

The mT5 models, in contrast, appear less adept at error generation. The errors produced by mT5 lag behind those from probabilistic reverse speller for Ukrainian and German and offer only a minimal improvement for Estonian.

We can also see that the scores before gold fine-tuning highlight that Ukrainian scores are notably low across all methods. However, these scores recover well after fine-tuning, suggesting the synthetic data may not align well with the text domain or error types specific to the Ukrainian language. Estonian and German models show higher scores for models trained with just AEG data and improve less drastically with fine-tuning.

4.3 Artificial Error Generation with Prompting

To assess the capability of generating errors without additional LM training, we utilize advanced commercial models, specifically exploring the efficiency of error generation through prompting GPT-3.5-turbo with datasets comprising 100,000 sentences. We later also explore the effectiveness of GPT-4-Turbo in a more limited setting (see Section 4.4).

The generation cost depends on the sum of input

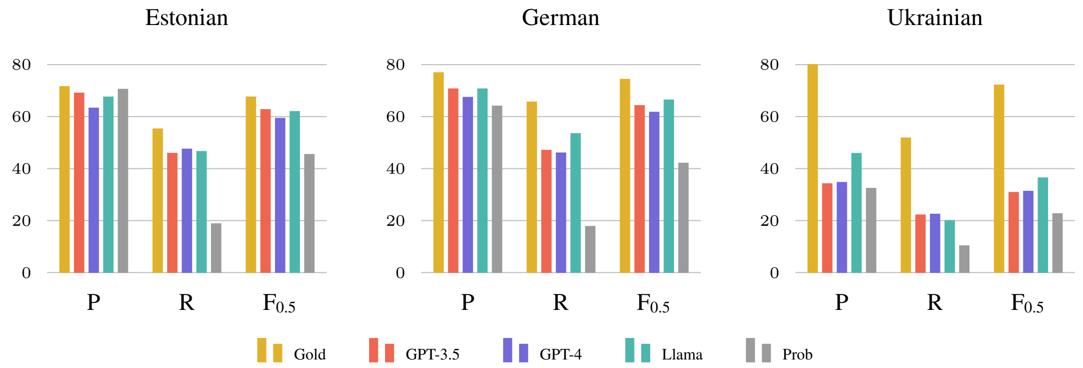

Figure 1: Quality of generated errors compared to gold and probabilistic, as shown by GEC results of tuning Llama-based models on same-sized synthetic or human (gold) error sets. GPT-3.5-turbo and GPT-4-turbo errors are generated via prompting, Llama stands for Llama 2-based model fine-tuned on the AEG task.

and completion tokens. Ukrainian, our most expensive language, had the highest number of tokens per 100,000 sentences: 98 million input and 12 million completion tokens. The cost for input tokens with GPT-3.5-Turbo in USD is \$147, and for completion tokens, it is \$25 – in total, \$172 for generating 100,000 Ukrainian sentences. In comparison, the costs with GPT-4-Turbo would have been \$983 and \$370, respectively[‡‡].

Table 4 shows the results of continued pre-training Llama-based models on the same amount of sentences (100,000) with synthetic errors from prompting or fine-tuning. In terms of error correction quality after gold fine-tuning, employing GPT-3.5-turbo for prompting and fine-tuning Llama-2-based models are both viable strategies for artificial error generation, as they lead to very close $F_{0.5}$ scores in all three languages (with a slight difference in favor of fine-tuning errors for German: 75.58 vs 76.25).

Analyzing the performance before gold fine-tuning reveals distinct differences between the two methods. For Estonian and German, recall rates are significantly higher with fine-tuning than prompting, though precision is slightly compromised. Conversely, Ukrainian exhibits the reverse pattern. However, it's important to note that any disparities observed before gold fine-tuning are greatly diminished after training on actual error correction examples. The most considerable remaining difference is under 0.7 points for German, with smaller discrepancies for Estonian and Ukrainian.

When comparing LLama model scores for 100k to the ones with only gold tuning (see Table 2), we can see that although scores increase more modestly, only 100k examples of synthetic data increase the scores more for German (almost 2 $F_{0.5}$-score points), a bit for Estonian (around 0.4 points) and stay the same for Ukrainian with higher precision and lower recall. The scores for models trained with 100k sentences are mostly lower than those trained with 1M reverse-speller errors, which indicates that the data quantity jump from 100,000 to 1M plays a significant role.

‡‡ https://openai.com/pricing

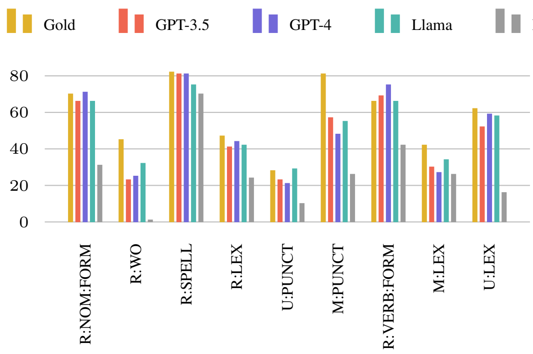

Figure 2: Recall scores for most frequent categories in Estonian EstGEC-L2 test set. The first letter corresponds to the operation type (R - replaced, M - missing, U - unnecessary).

4.4 Quality Compared to Human Data

Finally, we run a direct comparison between human errors and artificial ones. To do so we train models using the same number of sentences as the respective human error set sizes: 19k sentence pairs for German, 33k for Ukrainian, and 9k sentence pairs for Estonian. We include comparing these models to ones based on one million probabilistic sentences.

Our findings indicate that the precision of all synthetic data closely matches that of high-quality (gold) data in both Estonian and German, as illustrated in Figure 1. A notable distinction, however, is observed in recall rates. For Estonian and German, the recall for errors generated by LLMs is more comparable to human-generated (gold) data than errors produced through probabilistic methods.

Ukrainian scores with synthetic data are substantially worse than gold data, regardless of the AEG method. Still, recall for LLM-generated errors is significantly higher than for simple probabilistic errors. This might be due to a larger mismatch in the text domain or error frequency. Ukrainian UA-GEC data predominantly contains punctuation errors (43%) and has a two times smaller error rate than German (8.2 vs 16.8) (Syvokon et al., 2023).

Comparing GPT-3.5-Turbo with GPT-4-Turbo, we find similar performance overall. However, for Estonian, GPT-4-Turbo exhibits higher recall but lower precision. For German, GPT-4-Turbo shows reductions in both precision and recall. Performance is nearly identical for Ukrainian between the two models. Overall, the $F_{0.5}$ scores of GPT-4-Turbo are slightly lower for Estonian and German and marginally higher for Ukrainian compared to GPT-3.5-Turbo.

When analyzing the recall for various error categories in Estonian, it is evident that our models trained with AEG data particularly face challenges in inserting missing punctuation marks and correcting errors related to word order, as depicted in Figure 2. Errors generated probabilistically excel in identifying spelling mistakes and can correct certain errors in noun and verb forms. However, they generally perform poorly in addressing issues beyond spelling errors.

4.5 Evaluation of Generated Errors: Case Study with Estonian

We labeled 100 LM-generated sentences from different sets to determine if the errors made by models are similar to those in the training corpus.

Based on the annotations, we can categorize a large proportion of the changes according to the annotation scheme, but there is still a considerable amount of problematic edits (25-45%) (see Figure 3 and Table 7 in Appendix B). The human evaluation also indicates that the models differ in their error rates. GPT models generate fewer problematic errors overall, but the error category distribution seems more similar to human data with Llama-based models. This is likely due to a fine-tuning approach instead of prompting.

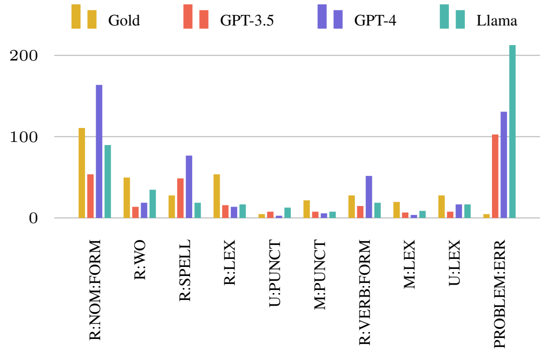

Figure 3: Error type count in Estonian based on annotating 100 randomly selected sentences (R - replaced, M - missing, U - unnecessary)

As mentioned in the last section, compared to human data, all models trained with generated data, correct far fewer word order and missing punctuation errors, and lexical changes are not well corrected either. These results can be partially explained by examining the different error types in generated data, where the same types are not as well represented as in human data. Most problematic edits involve generating lexical errors, which often were synonymous or changed the original meaning of the sentence, which could explain the poor performance in correcting lexical errors. On the other hand, verb or nominal form and spelling errors were better or almost as well corrected as by a model trained with gold data, and the data contained more errors in these categories. This shows that correction recall is closely tied to the error types present in the training data, and the data generated with our approach generates realistic error types that help correction in these categories.

5 Conclusion

In conclusion, our research demonstrates the significant potential of Llama-based LMs in addressing the challenges of GEC for low-resource languages. We have successfully developed state-of-the-art systems for Estonian, Ukrainian, and German by leveraging these models as both correctors and synthetic data generators. We also explore other methods for AEG and show that prompting stronger commercial LLMs is another way of generating high-quality data, and fine-tuning smaller models also has potential when the resources are more limited.

6 Limitations

Our work focuses on three languages, recognizing that numerous other languages with grammar error correction (GEC) datasets exist outside our study's scope. We selected languages based on recent relevant research activities: Ukrainian due to its recent Shared Task; Estonian, a newly emerging language in GEC research; and German for comparison with a robust 13B model. To comprehensively validate our method, further exploration across additional languages is necessary.

Our objective was not to devise the optimal system exhaustively. Therefore, several avenues remain unexplored, such as varying generation methods, testing different temperatures, and adjusting parameters. Moreover, we capped the generation of synthetic sentences at one million, below the volume utilized in many (though not all) synthetic data studies. Questions about the ideal amount of data needed its dependency on the quality of synthetic and gold examples, remain unanswered.

Furthermore, our study lacks human evaluation of GEC systems, a component for more reliably assessing the real-world efficacy of GEC systems.

References

Kais Allkivi-Metsoja, Karina Kert Kaisa Norak, Silvia Maine, and Pille Eslon. 2022. Error classification and annotation of learner language for developing estonian grammar correction.

Maksym Bondarenko, Artem Yushko, Andrii Shportko, and Andrii Fedorych. 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.

Adriane Boyd. 2018. Using Wikipedia edits in low resource grammatical error correction. In *Proceedings of the 2018 EMNLP Workshop W-NUT: The 4th Workshop on Noisy User-generated Text*, pages 79–84, Brussels, Belgium.

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.

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

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online.

Steven Coyne, Keisuke Sakaguchi, Diana Galvan-Sosa, Michael Zock, and Kentaro Inui. 2023. Analyzing the performance of gpt-3.5 and gpt-4 in grammatical error correction.

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.

Mariano Felice and Zheng Yuan. 2014. Generating artificial errors for grammatical error correction. In *Proceedings of the Student Research Workshop at the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 116–126, Gothenburg, Sweden.

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.

Phu Mon Htut and Joel Tetreault. 2019. The unbearable weight of generating artificial errors for grammatical error correction. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 478–483, Florence, Italy.

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

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.

Atakan Kara, Farrin Marouf Sofian, Andrew Bond, and Gözde Şahin. 2023. GECTurk: Grammatical error correction and detection dataset for Turkish. In *Findings of the Association for Computational Linguistics: IJCNLP-AACL 2023 (Findings)*, pages 278–290, Nusa Dua, Bali. Association for Computational Linguistics.

Sudhanshu Kasewa, Pontus Stenetorp, and Sebastian Riedel. 2018. Wronging a right: Generating better errors to improve grammatical error detection. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4977–4983, Brussels, Belgium.

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.

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

Kristina Koppel and Jelena Kallas. 2022. Eesti keele ühendkorpuste sari 2013–2021: mahukaim eestikeelsete digitekstide kogu. *Eesti Rakenduslingvistika Ühingu aastaraamat Estonian Papers in Applied Linguistics*, 18:207–228.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium.

Agnes Luhtaru, Elizaveta Korotkova, and Mark Fishel. 2024. No Error Left Behind: Multilingual Grammatical Error Correction with Pre-trained Translation Models. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2024)*.

Jakub Náplava and Milan Straka. 2019. Grammatical error correction in low-resource scenarios. In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, pages 346–356, Hong Kong, China.

Jakub Náplava, Milan Straka, Jana Straková, and Alexandr Rosen. 2022. Czech grammar error correction with a large and diverse corpus. *Transactions of the Association for Computational Linguistics*, 10:452–467.

Thuat Nguyen, Chien Van Nguyen, Viet Dac Lai, Hieu Man, Nghia Trung Ngo, Franck Dernoncourt, Ryan A. Rossi, and Thien Huu Nguyen. 2023. Culturax: A cleaned, enormous, and multilingual dataset for large language models in 167 languages.

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.

OpenAI. 2023. Gpt-4 technical report.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota.

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 114–120, Dubrovnik, Croatia.

Marek Rei, Mariano Felice, Zheng Yuan, and Ted Briscoe. 2017. Artificial error generation with machine translation and syntactic patterns. In *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 287–292, Copenhagen, Denmark.

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.

Alla Rozovskaya and Dan Roth. 2010. Generating confusion sets for context-sensitive error correction. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 961–970, Cambridge, MA.

Ingrid Rummo and Kristiina Praakli. 2017. TÜ eesti keele (võõrkeelena) osakonna õppijakeele tekstikorpus [the language learner's corpus of the department of estonian language of the university of tartu]. In *EAAL 2017: 16th annual conference Language as an ecosystem, 20-21 April 2017, Tallinn, Estonia: abstracts, 2017, p. 12-13.*

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany.

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.

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

Oleksiy Syvokon, Olena Nahorna, Pavlo Kuchmiichuk, and Nastasiia Osidach. 2023. UA-GEC: Grammatical error correction and fluency corpus for the Ukrainian language. In *Proceedings of the Second Ukrainian Natural Language Processing Workshop (UNLP)*, pages 96–102, Dubrovnik, Croatia.

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 132–137, Dubrovnik, Croatia.

Yuqing Tang, Chau Tran, Xian Li, Peng-Jen Chen, Naman Goyal, Vishrav Chaudhary, Jiatao Gu, and Angela Fan. 2021. Multilingual translation from denoising pre-training. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3450–3466, Online.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford alpaca: An instruction-following llama model. `https://github.com/tatsu-lab/stanford_alpaca`.

NLLB Team, Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loic Barrault, Gabriel Mejia Gonzalez, Prangthip Hansanti, John Hoffman, Semarley Jarrett, Kaushik Ram Sadagopan, Dirk Rowe, Shannon Spruit, Chau Tran, Pierre Andrews, Necip Fazil Ayan, Shruti Bhosale, Sergey Edunov, Angela Fan, Cynthia Gao, Vedanuj Goswami, Francisco Guzmán, Philipp Koehn, Alexandre Mourachko, Christophe Ropers, Safiyyah Saleem, Holger Schwenk, and Jeff Wang. 2022. No language left behind: Scaling human-centered machine translation.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Viet Anh Trinh and Alla Rozovskaya. 2021. New dataset and strong baselines for the grammatical error correction of Russian. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4103–4111, Online.

Guillaume Wenzek, Marie-Anne Lachaux, Alexis Conneau, Vishrav Chaudhary, Francisco Guzmán, Armand Joulin, and Edouard Grave. 2020. CCNet: Extracting high quality monolingual datasets from web crawl data. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4003–4012, Marseille, France.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Huggingface's transformers: State-of-the-art natural language processing.

Ziang Xie, Guillaume Genthial, Stanley Xie, Andrew Ng, and Dan Jurafsky. 2018. Noising and denoising natural language: Diverse backtranslation for grammar correction. 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 619–628, New Orleans, Louisiana.

Shuyao Xu, Jiehao Zhang, Jin Chen, and Long Qin. 2019. Erroneous data generation for grammatical error correction. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 149–158, Florence, Italy.

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.

Zheng Yuan, Felix Stahlberg, Marek Rei, Bill Byrne, and Helen Yannakoudakis. 2019. Neural and FST-based approaches to grammatical error correction. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 228–239, Florence, Italy.

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.

A Training details

A.1 Llama-based models

The models are trained on 4 AMD MI250x GPUs (each acting as 2 GPUs).

For fine-tuning, we used a learning rate of 5e-6 linearly decayed to 5e-7 (10%). The learning rate was selected from {4e-5, 2e-5, 1e-5, 5e-6, 2.5e-6} based on highest Estonian GEC development set $F_{0.5}$ score. The models were trained for three epochs, although we chose the first epoch since it almost always achieved the highest $F_{0.5}$ score. Table 5 provides an overview of the hyperparameters.

For GEC and AEG fine-tuning, sentences are in non-tokenized format or detokenized (for Estonian and German). The crawled data used for AEG is normalized with Moses (Koehn et al., 2007) for Estonian and German.

For continued pre-training, we follow the parameters used by Llammas-base (see Table 6). The training data is packed to fill the whole sequence length.

Parameter	Value
LR	5e-6
LR_{final}	5e-7
LR-schedule	linear
Epochs	3
Max sequence length	1024
Batch size (total)	128
Gradient clipping	1.0
Weight decay	0.1
Optimizer	AdamW
Precision	bf16
DeepSpeed	Zero Stage 2

Table 5: Llama-based GEC model fine-tuning parameters.

Parameter	Value
LR	2e-5
LR_{final}	2e-6
LR-schedule	linear
Updates	19080
Max sequence length	1024
Batch size (total)	256
Gradient clipping	1.0
Weight decay	0.1
Optimizer	AdamW
Precision	bf16
DeepSpeed	Zero Stage 2

Table 6: Llama continued pre-training parameters.

A.2 NLLB-based models

We follow the training process specified by Luhtaru et al. (2024), including hyperparameters. The training is conducted on an AMD MI250x GPU. We are training the AEG models for 20 epochs and picking the 15th after arbitrary manual evaluation and testing sets on checkpoints 5, 10, 15, and 20. The data for NLLB models is first normalized with Moses script, and we use the SentencePiece model (Kudo and Richardson, 2018) for untokenized text.

A.3 mT5-based models

To learn to generate errors, we train on reversed human GEC data for three epochs with batch size 32, max sequence length of 128, half-precision training, and a learning rate of 0.0001 without warmup and scheduling. For generation, we use top 50 probabilistic sampling.

B Problematic edits

We further explore the human annotation results discussed in section 4.5. Table 7 displays the percentage of problematic error types out of all errors generated by the model.

C NLLB correction

The GEC performance of the NLLB model without any synthetic data is in Table 8. The zero-shot results for Estonian and German are significantly higher than for Ukrainian. We notice that the Ukrainian dataset contains characters not present in NLLB vocabulary, like special quotation marks, which the normalization script unifies but appear

https://github.com/pluiez/NLLB-inference/blob/main/preprocess/normalize-punctuation.perl

Type	Llama	GPT-3.5	GPT-4
O	10.83	4.71	9.07
HALL	22.72	11.11	3.75
SYN	6.16	6.4	7.5
INACC	2.12	5.39	1.38
UNREC	3.82	6.73	3.94
Total %	45.65	34.34	25.64

Table 7: Percentages of problematic edits.

as errors while testing. In addition, the Ukrainian test set contains far fewer edits, which, especially in a zero-shot scenario, means worse scores because NLLB paraphrases more rigorously (Luhtaru et al., 2024).

D Prompts

We present the prompts used to generate 1) 100,000 sets with GPT-3.5-Turbo and 2) preliminary sets with GPT-4-Turbo in Tables 9, 10, 11 for Estonian, German, and Ukrainian respectively.

Lang	Zero-shot			Gold fine-tuning		
	P	R	$F_{0.5}$	P	R	$F_{0.5}$
Estonian	43.89	45.31	44.17	61.14	49.48	58.39
Ukrainian	8.24	31.57	9.67	35.62	34.1	35.31
German	43.66	41.52	43.22	73.71	67.75	72.44

Table 8: Zero-shot and gold fine-tuning scores of NLLB-200-1.3B-Distilled models on Ukrainian UA-GEC gec+fluency test set.

Muuda sisendteksti, genereerides sinna vigu, mida võib teha eesti keele õppija. Väljundtekstina tagasta sisendtekst, kuhu oled genereerinud vead. Sisendteksti genereeri õigekirja-, grammatika-, sõnavaliku-, sõnajärje-, kirjavahemärgi- ning stiilivigu. Kui sisendtekstis on vigu, siis ära neid paranda, vaid genereeri vigu juurde. Ülesande kohta on neli näidet:

Sisendtekst: `{correct}`
Väljundtekst: `{incorrect}`

Sisendtekst: `{correct}`
Väljundtekst: `{incorrect}`

Sisendtekst: `{correct}`
Väljundtekst: `{incorrect}`

Sisendtekst: `{correct}`
Väljundtekst: `{incorrect}`

Sisendtekst: `{input}`
Väljundtekst:

Table 9: GPT prompt - Estonian

Erzeugen Sie im Eingabetext Fehler, wie sie jemand, der Deutsch lernt, machen könnte. Geben Sie als Ausgabetext den Eingabetext zurück, in den Sie Fehler eingefügt haben. Erzeugen Sie Rechtschreib-, Grammatik-, Wortwahl-, Wortreihenfolge-, Zeichensetzungs- und Stilfehler im Eingabetext. Sollten im Eingabetext bereits Fehler vorhanden sein, korrigieren Sie diese nicht, sondern erzeugen Sie zusätzliche Fehler. Es gibt vier Beispiele für die Aufgabe:

Eingabetext: `{correct}`
Ausgabetext: `{incorrect}`

Eingabetext: `{correct}`
Ausgabetext: `{incorrect}`

Eingabetext: `{correct}`
Ausgabetext: `{incorrect}`

Eingabetext: `{correct}`
Ausgabetext: `{incorrect}`

Eingabetext: `{input}`
Ausgabetext:

Table 10: GPT prompt - German

Змініть вхідний текст шляхом генерації в ньому помилок, які міг би зробити учень, що вивчає українську мову. На виході повертайте вхідний текст, у який ви внесли помилки. У вхідному тексті генеруйте помилки правопису, граматики, вибору слів, порядку слів, розділових знаків та стилю. Якщо у вхідному тексті є помилки, то не виправляйте їх, а генеруйте додаткові помилки. Далі наведені чотири приклади до цієї задачі

Вхідний текст: `{correct}`
Вихідний текст: `{incorrect}`

Вхідний текст: `{correct}`
Вихідний текст: `{incorrect}`

Вхідний текст: `{correct}`
Вихідний текст: `{incorrect}`

Вхідний текст: `{correct}`
Вихідний текст: `{incorrect}`

Вхідний текст: `{input}`
Вихідний текст:

Table 11: GPT prompt - Ukrainian

Instruction:
Reply with a corrected version of the input sentence in `{language}` with all grammatical and spelling errors fixed. If there are no errors, reply with a copy of the original sentence.

Input:
`{input}`

Response:
`{correction}`

Table 12: Llama-based model GEC instruction format loosely based on Alpaca (Taori et al., 2023). The instruction is based on Coyne et al. (2023).

Instruction: Reply with a grammatically incorrect version of the `{language}` input sentence. ### Input: `{input}` ### Response: `{correction}`

Table 13: Llama-based model AEG instruction format loosely based on Alpaca (Taori et al., 2023).