LLMs in alliance with Edit-based Models: Advancing In-Context Learning for Grammatical Error Correction by Specific Example Selection

Alexey Sorokin
MSU Institute for Artificial Intelligence
Yandex

Regina Nasyrova
MSU Institute for Artificial Intelligence

Abstract

We show that fewshot Grammatical Error Correction might be improved by using an encoder-based sequence labeling model, such as GECTOR, to select similar examples. We demonstrate this on three Russian GEC corpora and English BEA corpus. The effect is the most significant for the new LORuGEC corpus and reaches up to 5-10% F0.5-score depending on the model. The corpus is released in our paper and contains 348 train and 612 test examples. The corpus is designed for diagnostic purposes and is also equipped with writing rules' annotations.

These annotations allow to further improve few-shot error correction by contrastive tuning of GECTOR-like encoder on rule classification task. This holds for a broad class of large language models. The best results are obtained with 5-shot YandexGPT-5 Pro model, achieving F0.5-score of 83%.

1 Introduction

The task of Grammatical Error Correction (GEC) may be defined in two ways, depending on whether the main objective is to make the sentence *grammatical*, i.e. applying minimal edits until it is grammatically correct, or *fluent*, namely transforming a sentence, likely substantially, so that it sounds natural yet saves the initial meaning (Coyne et al., 2023). In the era of Large Language Models (LLMs), researchers studied their ability in both settings (see more in the Section 2.1) and concluded that LLMs outperform mainstream GEC models in the latter objective (Coyne et al., 2023), demonstrating more freedom and creativity in sentence modifications. However, this asset becomes a burden in the former setting, where LLMs' 'generous' edits are treated as overcorrections.

A reasonable thing to do to make LLMs predict more reliable corrections as well as leverage their fluency and language knowledge is to apply them in few-shot settings which proved to be valuable in many other NLP tasks, e.g. Machine Translation and Question Answering (Brown et al., 2020). As supported by Fang et al. (2023); Loem et al. (2023), in-context examples indeed enhance the quality and consistency of LLMs' corrections. However, the research of in-context learning in GEC pays little attention to example selection, the rare exception being Tang et al. (2024), using a syntactic structure similarity metric to select in-context examples.

We argue that sentences containing the errors of the same kind as the target ones may be much more beneficial as in-context examples rather than randomly selected ones. To prove this hypothesis, we present a novel approach to Grammatical Error Correction which makes use of a task-specific sequence labeling model (Omelianchuk et al., 2020) and retrieval-based few-shot learning. The sequence labeling model was trained to predict token-level edits, required to transform the source text into the grammatically correct one. We employ it to encode tokens in a sentence and choose the embeddings of the most likely edits as the representation of a sentence. After that, we use the retriever to select the closest sentence representations to the target one. As a result, the sentences corresponding to the selection are used as in-context demonstrations.

However, we assume that the notion of "errors of the same kind" may require an extension, involving the similarity of not only the edit but also the general pattern behind it. Since the same edits may occur in diverse contexts (e.g. comma insertions may be required before certain conjunctions or between subordinate clauses), the sentence with the same edit may not be informative enough. The model would not comprehend the utility of the given demonstration because it is unclear what it should pay attention to when sentences are completely distinct, apart from the edit.

That is why, we collect a new Linguistically Oriented Rule-annotated GEC dataset for Russian –

Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications, pages 517–534

LORuGEC, which consists of sentences representing the rules of Russian grammar that are considered to be complicated both for L1 learners and large language models. These errors are also underrepresented in the existing Russian GEC corpora, so we expect that the effect of in-context demonstrations would be the most prominent for this corpus.

We conduct experiments on Russian GEC datasets in zero-shot and few-shot (1-shot and 5-shot) settings. For the few-shot setting we study random example selection and retrieval-based selection with the GECTOR-like pretrained encoder. We additionally tune the retriever to select sentences related to the same rule. We choose several LLMs for testing and also present the results of their finetuned versions where possible.

Our main contribution is as follows:

- Novel GEC dataset for Russian, where sentences are also annotated for rules which are violated in them. The methodology of its collection makes it a challenging benchmark for LLMs, as it includes previously underrepresented cases.
- We are the first to apply the GECTOR-like(Omelianchuk et al., 2020) model for few-shot examples retrieval in grammatical error correction. The proposed approach yields considerably higher scores on LORuGEC dataset than random selection of examples for all models, supporting the impact of demonstrations' quality and design on the performance of LLMs.
- Contrastive tuning of the retriever on related data additionally improves the quality of corrections on LORuGEC.
- The proposed method may compete with LLMs' finetuning, especially if the training data is not large in size.

We make our data[1] and code[2] freely available.

2 Related work

2.1 Using LLMs for Grammatical Error Correction

Large Language Models gained prominence over the recent years as helpful tools for most Natural Language Processing tasks (Brown et al., 2020; DeepSeek-AI et al., 2025). Their abilities were also tested on the Grammatical Error Correction task. Wu et al. (2023); Fang et al. (2023) show that ChatGPT[3] performs worse, than commercial and conventional GEC models for English, being less prone to under-correction and mis-correction, but generating more fluent corrections, hence over-correcting, which is penalized severely by conventional metrics designed to evaluate minimal edits. Moreover, ChatGPT shows promising results for Multilingual GEC (Fang et al., 2023).

A more detailed analysis with fine-grained prompt and hyperparameter search was done in Coyne et al. (2023). They found that low temperature and suitable prompts increase the reliability of corrections produced by GPT-3.5(Ouyang et al., 2022) and GPT-4(OpenAI, 2023). Loem et al. (2023) proceed to research prompt-based methods for GEC, discovering that GPT-3(Brown et al., 2020) is much less prompt-sensitive and inconsistent, when supported with in-context examples.

Fang et al. (2023); Loem et al. (2023) propose that the investigation on the effect of example quality and design may be beneficial. An instance of it is introduced in Tang et al. (2024), where sentences with the same syntactically incorrect structure are adopted as in-context examples, significantly outperforming randomly selected ones. Advancing the choice of in-context examples, Robatian et al. (2025) propose Retrieval-Augmented Generation within In-Context Learning approach to improve Generative Error Correction in speech recognition systems. Other works also consider LLMs' instruction tuning and ensembling for GEC (Kaneko and Okazaki, 2023; Omelianchuk et al., 2024).

2.2 In-context learning for LLMs

Our work is an example of the so-called retrieval-based few-shot learning, where demonstration samples are selected according to some similarity measure between vectors. A review of retrieval-based in-context learning is presented in Xu et al. (2024). The early examples of this approach include Rubin et al. (2022) where retrieval-based selection of demonstrations was shown to improve performance for three sequence-to-sequence learning tasks. The authors also demonstrated that one may reach further gains by training the retriever to select examples that maximize the correct output probability. Margatina et al. (2023) verified the positive role

[1] https://github.com/ReginaNasyrova/LORuGEC
[2] https://github.com/AlexeySorokin/LORuGEC
[3] https://openai.com/index/chatgpt/

of similarity between test and in-context examples on a diverse range of models and tasks including classification and multiple choice datasets. Nori et al. (2023) demonstrated that using KNN-based few-shot example selection allows to adapt general models to medical domain without special tuning.

2.3 GEC corpora for Russian

There are three available Russian GEC datasets: RULEC-GEC(Rozovskaya and Roth, 2019), RU-Lang8(Trinh and Rozovskaya, 2021) and GERA(Sorokin and Nasyrova, 2025). The first one represents a subset of the Russian Learner Corpus of Academic Writing (RULEC)(Alsufieva et al., 2012), containing essays of the US students who were either learning Russian as a foreign language or heritage speakers. The authors comprised a list of 23 error type labels that cover (morpho)syntactic, lexical and spelling errors.

The RU-Lang8 Dataset constitutes a subset of the Lang-8 Corpus(Mizumoto et al., 2012) learner corpus, based on the language learning website[4]. Most texts in RU-Lang8 are much shorter, being small paragraphs or learners' questions. Unlike RULEC-GEC, RU-Lang8 has a more coarse-grained annotation, with error type labels representing operations of token replacement, deletion, insertion and change in word order.

As opposed to both datasets, GERA is based on Russian school texts and was annotated in line with a much more fine-grained label inventory, i.e. grammatical error types cover a broader list of parts of speech and grammatical categories, and there are different types of lexical and spelling errors depending on the erroneous construction.

2.4 Linguistically motivated data for GEC

Usually GEC corpora are based on real-world learner data, not a predefined error taxonomy. A partial example of error-driven approach was Volodina et al. (2021), where the four principal error types from existing data were selected to be included in the dataset. Similarly to LORuGEC, most examples in their corpus contain exactly one error.

More frequently, error taxonomies are used for collecting linguistic acceptability data. The most well-known example of such corpora are COLA(Warstadt et al., 2019) and BLIMP(Warstadt et al., 2020) for English. One may even convert a BLIMP-like dataset of minimal pairs to GEC format, by using the ungrammatical element of the pair as the source and the grammatical one – as the target, this approach was adopted in Volodina et al. (2021) for Swedish and Jentoft and Samuel (2023) for Norwegian. Concerning Russian language, BLIMP-like datasets of minimal pairs were introduced in the recent works of Graschenkov et al. (2024) and Taktasheva et al. (2024).

[4] https://lang-8.com/

3 LORuGEC: Corpus description

3.1 Motivation and data collection

Most existing GEC corpora consist of L2 learners' data. Even corpora based on native learners' data mostly reflect the real-world error distribution, underrepresenting complicated grammatical rules. Concerning the Russian language, existing corpora, such as RULEC-GEC(Rozovskaya and Roth, 2019), RU-Lang8(Trinh and Rozovskaya, 2021) and GERA(Sorokin and Nasyrova, 2025), contain very few examples of complex, "schoolbook" rules, making these corpora suboptimal for use in educational applications. Our primary goal is to fill this gap and collect a corpus of complex cases that represent the rules which are considered difficult for Russian L1 learners. The second goal of our project is to study, which rules present the highest complexity for modern LLMs in the task of Grammatical Error Correction.

Given our research goals, we organize the data collection and annotation process as follows:

1. Firstly, one of the paper authors (a bachelor in Linguistics) collected an initial set of about 10 rules that are known as difficult for Russian high school students. These rules covered various fields of writing, mostly punctuation, grammar and spelling. The list of rules was checked by another author of the paper and verified using several Russian grammar books.

2. For each of the selected rules, the annotators, which were students with linguistic backgrounds and Russian native speakers, were asked to collect up to 15 examples belonging to these rules. Since the collected examples were intended to be used for LLM benchmarking, several precautions were taken, which were expressed in the instruction (see more in Appendix A), as follows:

    - Preferably, choose sentences from different sources.

- Avoid using quotations from fiction.
- Refrain from selecting commonplace examples.

3. The collected examples were corrupted to simulate the common mistakes corresponding to particular rules. For example, if the rule governs the use of comma between the conjuncts, the comma was either deleted in the contexts where it was required or inserted when it must not be used. If there are multiple ways to introduce errors, the examples should cover them all. For instance, clauses with participles in Russian should be surrounded by commas, so possible corruptions included deletion of both commas, only the preceding comma or only the following one.

4. The collected examples were passed through the YandexGPT3 Pro[5] model. The goal of this stage was to identify complex sentences and make the dataset more challenging by including analogous examples.

5. After successfully completing the data collection for the initial set of 10 rules, the annotators were allowed to select the subsequent rules themselves. They were instructed to consult grammar reference books and cover all fields of written language, such as punctuation, spelling, grammar (in the narrow sense) and lexis. The process was supervised by the principal annotator (one of the authors) who checked the selection of rules and example cases, as well as their annotation. Since the source sentences were created by targeted manual corruption, the correct sentence was known in advance, thus reducing the correction ambiguity. The principal annotator additionally analyzed 100 random samples and found no disagreement with the annotators.

3.2 Data sources

While selecting the rules, annotators and authors used various resources, such as grammar reference books, teacher manuals and educational websites based on them, we refer to B for the full list of data sources. The textbooks that were used comply with Russian educational standards, some of them are specially approved by the Russian Academy of Sciences, for example, (Valgina et al., 2009).

[5] https://yandex.cloud/ru/docs/foundation-models/concepts/yandexgpt/models

3.3 Rules Description and Statistics

We gathered 48 rules from 4 grammar sections. The majority of them represent punctuation and spelling. We present the comprehensive list of rules in Appendix C.

We collected 960 pairs of sentences (an average of 20 sentences per rule), which were split into validation and test subsets so that for each rule at least 9 sentences or approximately two thirds of collected sentences would be allocated to the test partition. Consequently, the size of the test subset is twice as large as the size of the validation one (see Table 1). Additionally, unlike the latter, only the test subset includes initially correct sentences (for hypercorrection considerations). See more on the data format in Appendix D.

3.4 Comparison with other GEC corpora for Russian

Comparing to existing Russian GEC corpora, such as RULEC-GEC(Rozovskaya and Roth, 2019), RU-Lang8(Trinh and Rozovskaya, 2021) and GERA(Sorokin and Nasyrova, 2025), our data differs in several aspects:

- To the best of our knowledge, that is the only Russian GEC corpus where all the errors are matched with corresponding grammar rules instead of error type.

- Our corpus is deliberately created for evaluation and diagnostic purposes. Therefore, it has no training subset and is much smaller than other corpora (see Table 2). We do not want LLMs to acquire new capabilities on the validation set of our corpus, but rather to reveal the knowledge they already have.

  On the other hand, almost all sentences in our corpus contain errors and are supposed to be challenging in contrast to other GEC data.

- Since corpus examples were created via corruption, for the vast majority of mistakes there is only one possible correction, increasing the trustworthiness of evaluation scores.

- As shown in Table 3, LORuGEC has the highest fraction of pattern-based errors covered by a rule-based generator. These errors include punctuation errors, word form changes, deletion, insertion or replacement of closed word categories (prepositions, conjunctions and pronouns), spelling errors, etc. Despite this, the

Sample	Sentences	Correct source sentences	Sentences for complex rules (%)	Tokens
Validation	348	0	250 (71.84)	5,579
Test	612	31	419 (68.46)	10,131

Table 1: Statistics on the validation and test samples of LORuGEC.

Sample	Sentences	Tokens
RULEC-GEC	12,480	206,258
RU-Lang8	4,412	54,741
GERA	6,681	119,068
LORuGEC	960	15,710

Table 2: Quantitative comparison of GEC datasets for Russian.

corpus	P	R	F0.5	uncov., %
RULEC-GEC	50.4	32.6	45.5	42.0
RU-Lang8	60.8	37.9	54.2	48.8
GERA	74.3	47.0	66.6	33.7
LORuGEC	45.1	17.7	34.4	21.9

Table 3: Comparison of GEC model performance and difficult fraction (uncov., %) for different Russian GEC corpora. The model is Qwen2.5-7B finetuned on the concatenation of Russian GEC data.

GEC model finetuned on the concatenation of 3 Russian GEC corpora (see Section 5 for details) has much lower scores on LORuGEC than on other corpora. This implies that the main problem on LORuGEC is not to generate the suggestion but to discriminate between correct and incorrect variants.

4 Similar example retrieval

4.1 Approach description

We suppose that large language models may lack knowledge about specific Russian grammar rules. This information might be injected during inference via in-context example selection. A natural solution might be to select examples that belong to the same rule, i.e. resembling not only the required correction, but also the grammatical reasoning behind it. However, this restricts the method to a predefined bounded set of rules that prevents the model from real-world usage.

Our approach is to use an embedder to select training examples similar to the given test sentence. We want this embedder to reflect grammatical similarity. That is not the case for standard sentence embedders that assign similar vector representations to semantically similar sentences. To be used for similar examples retrieval, the embedder should be pretrained on a grammar-related task.

We decide to select the famous GECTOR model offered by (Omelianchuk et al., 2020). Their approach does not treat GEC as a Machine Translation task but reduces it to sequence labeling, taking into account the fact that most tokens in a sentence remain unchanged after the correction. GECTOR classifier, which is built upon a pretrained encoder[6], predicts the no-operation label KEEP for such tokens. In other cases, labels represent

- **elementary edit operations**, such as DELETE, REPLACEWITH_<TOKEN> (e.g., replace the current word with the word *on*) or INSERT_<TOKEN>, where <TOKEN> may refer to not only words, but also punctuation marks.
- **grammatical transformations** which mostly have to do with inflection (e.g., GRAM$SING, meaning 'put the current word in the singular form instead of plural').

Although the latter labels, the so-called G-labels, do not exactly correspond to rules of writing, mistakes from the same rule class often obtain the same label. Since the hidden states of encoder models reflect the similarity in their label space, this similarity is also related to rule similarity.

4.2 Implementation details

Although the retrieval based on embedding similarity is very common and is extensively used, e.g., in Retrieval-Augmented Generation (RAG), the adaptation of GECTOR to retrieval has several details. Firstly, as GECTOR operates on token level , it does not assign meaningful representation to the [CLS] token usually used for retrieval. We represent the sentence with the hidden states from the final encoder layer and select up to 3 hidden states

[6] https://huggingface.co/ai-forever/ruRoberta-large in our case.

corresponding to the most probable error positions. The probability of an error is predicted by the GECTOR model itself, using $1 - p(\text{KEEP})$, where KEEP is the no-edit label of the GECTOR model.

Since the original GECTOR model uses obsolete Python libraries and the sets of G-labels differ significantly between English and Russian, we reimplement the model by ourselves using HuggingFace Transformers[7] library. The details of its training are available in Appendix F.

4.3 Retriever finetuning

We suppose that pretraining on external data empowers the model with the basic information about grammatical error patterns, but the model might not have enough knowledge about rare or dataset-specific rules. Therefore, we propose to finetune the retriever on the task of rule classification using contrastive learning. The tuning is performed on the validation part of our dataset. The training objective is a standard triplet loss

$$L(h, h^+, h^-) = \max(\frac{\rho(h, h^+) - \rho(h, h^-) + \alpha}{t}, 0),$$

where ρ is the distance function (e.g., cosine), α is the margin and t is the temperature. We always use as h^+ the closest example with the same class label and as h^- – the closest example with another class label. In terms of contrastive learning literature, we use hard positives and hard negatives without in-batch negatives.

We retrieve the closest positive and negative examples once in epoch. After completing the epoch we recalculate the triples using the updated embedder. Further details are given in Appendix F.

5 Model evaluation

In this section we evaluate several LLMs on our corpus[8]. We select two open-source models: the open-source *Qwen-2.5 7B Instruct*[9](Yang et al., 2024) and *yandex/YandexGPT-5-Lite-8B-instruct*[10] as well as closed-source *YandexGPT-5 Pro*[11]. The latter two models are selected because they were largely trained on Russian data and the first one is chosen due to its excellent multilingual abilities. We evaluate additional models, such as LLama3-8B-Instruct(Meta, 2024) and GPT4o-2024-05-13(OpenAI, 2023), in Appendix G.1.

We compare several settings:

1. zero-shot prompt-based application of LLM. The prompt is provided in Appendix E.1.
2. few-shot prompt-based application of LLMs with different selection of in-context examples: random, the general purpose e5-base-multilingual[12](Wang et al., 2024) embedder, pretrained GECTOR and GECTOR with contrastive finetuning).
3. finetuning open-source LLMs on external Russian GEC data: RULEC-GEC, RU-Lang8 and GERA(Sorokin and Nasyrova, 2025).
4. further training of the finetuned LLMs on the validation part of our corpus.
5. LORA-based training of open-source LLMs only on the validation part of our corpus.

The hyperparameters of the finetuning are available in Table 9.

As is commonly done, we score the tokenized model outputs with M2scorer(Dahlmeier et al., 2013) and report precision, recall and F0.5 score, using F0.5 as the main metric. The results are given in Table 4. We make the following conclusions:

1. Finetuning on external GEC data is detrimental for LORuGEC. Since LORuGEC types of errors are rare in general GEC corpora, the finetuned model decides not to correct them, hence, its recall dramatically reduces.
2. With a single exception, the GECTOR retriever performs better than the random one, proving our first hypothesis: **during pretraining on general GEC data, the encoder learns the representations for error types**. Moreover, these representations are helpful even for rare classes of errors that the LLM was not able to learn. In contrast, the general-purpose e5-base-multilingual embedder produces much smaller improvements.

[7] https://huggingface.co/docs/transformers/index

[8] We restrict our attention to LLMs by two reasons: first, one of our goals is to study few-shot learning approach. Second, in contrast to English, LLMs outperform other approaches, such as encoder-decoder or GECTOR-like, on available Russian data.

[9] https://huggingface.co/Qwen/Qwen2.5-7B-Instruct

[10] https://huggingface.co/yandex/YandexGPT-5-Lite-8B-instruct

[11] https://yandex.cloud/ru/docs/foundation-models/concepts/yandexgpt/models

[12] https://huggingface.co/intfloat/multilingual-e5-base

Setup	Qwen2.5-7B			YandexGPT5-Lite			YandexGPT5-Pro		
	P	R	F0.5	P	R	F0.5	P	R	F0.5
zero-shot	43.3	34.0	41.0	66.4	51.0	62.6	76.5	66.7	74.3
1-shot, random	44.4	28.6	40.0	67.8	48.6	62.8	78.3	71.0	76.7
5-shot, random	47.2	30.2	42.4	68.5	56.3	65.6	**83.9**	**79.2**	**83.0**
1-shot, e5-base	44.6	29.5	40.5	69.4	49.4	64.2	81.6	69.7	78.9
5-shot, e5-base	47.0	31.8	42.9	68.8	56.8	66.0	81.8	72.2	79.7
1-shot, GECTOR	50.2	35.8	46.5	69.9	53.9	66.0	81.9	72.8	79.9
5-shot, GECTOR	54.3	41.7	51.2	70.0	62.4	68.3	82.7	76.7	81.4
1-shot, GECTOR+FT	*52.7*	*39.8*	*49.5*	*71.2*	*56.7*	*67.7*	*83.0*	*76.3*	*81.6*
5-shot, GECTOR+FT	**59.3**	**46.2**	**56.1**	*73.1*	*65.5*	*71.4*	83.5	78.1	82.3
ext. finetuning	45.1	17.7	34.4	67.0	35.4	56.9		NA	
ext.+LORuGEC finetuning	*50.1*	37.9	47.1	**77.4**	**73.6**	**76.6**		NA	
LORuGEC LORA finetuning	48.6	*42.6*	*47.3*	74.1	72.6	73.8		NA	

Table 4: Comparison of different LLMs on the LORuGEC test set in zero-shot, few-shot and finetuning modes. Ext. finetuning refers to training on the concatenation of other Russian GEC corpora. The best metric inside the same approach (e.g., 1-shot) is presented in italics and the best overall metric – in bold.

3. Contrastive finetuning of the embedder is also helpful: the 1-shot GECTOR+FT retrieval almost matches the performance of 5-shot GECTOR retrieval. This proves our second hypothesis: **In-domain contrastive tuning of the retriever improves the quality of few-shot error correction**. This also proves the usefulness of rule annotation that distinguishes our corpus from general GEC data.

4. The models of the YandexGPT-5 family handle "schoolbook" errors from LORuGEC much better than Qwen-2.5 does. The details of their training are not available, however, it is likely that they saw more high-quality Russian data than the multilingual Qwen model.

5.1 Detailed results and examples

In Table 5 we also report the results per category for different error types. For both compared models punctuation errors are the easiest and the lexical ones – the hardest. A plausible explanation of this fact is that punctuation rules are the most strict, mostly binary (whether to use the comma or not) and rely on separate tokens, while the lexical rules are more vague and usually deal with more options.

When training the embedder, we use the retrieval quality as an intrinsic quality metric: the more often the embedder retrieves examples that belong to the same rule, the better it is. We observe that this internal metric correlates well with error correction quality, as shown in Table 6.

We provide illustrative examples of retrieved samples together with corresponding model outputs in Figures 3 and 4.

Category	Qwen2.5-7B			YandexGPT5-Pro		
	P	R	F0.5	P	R	F0.5
Grammar	50.0	36.5	46.6	86.3	69.8	82.4
Lexis	46.7	22.6	38.5	85.0	54.8	76.6
Punct.	66.2	53.6	63.0	85.7	83.3	85.2
Spelling	55.2	44.9	52.8	80.9	77.4	80.2

Table 5: Per-category scores of 5-shot learning, GECTOR+FT retriever for Qwen2.5-7B and YandexGPT5-Pro models.

5.2 Results for other corpora

The results on the introduced LORuGEC corpus prove the utility of our approach on a rule-oriented corpus. We wonder whether GECTOR-based demonstration selection improves results for general GEC corpora as well. To verify it, we compare three types of few-shot example selection (random, GECTOR and GECTOR+FT) on three available corpora: RULEC-GEC, RU-Lang8 and GERA. The results for the first two corpora are provided in Table 7, the results for GERA are in Table 13.

We again observe the advantage of GECTOR-based examples over random samples. Finetuning of GECTOR retriever on LORuGEC data does not have a clear positive effect probably due to the difference in error distribution between corpora. Due to larger sizes of these corpora, few-shot learning is not able to outperform full finetuning, but demon-

Retriever	acc.	top-5 recall	Qwen2.5-7B F0.5		YandexGPT5-Pro F0.5	
			1-shot	5-shot	1-shot	5-shot
random	2.3	10.3	40.0	42.4	76.7	83.0
GECTOR	31.7	49.3	46.5	51.2	79.9	81.4
GECTOR+FT	55.9	72.2	49.5	56.1	81.6	82.3

Table 6: Correlation between retrieval and GEC metrics for different retrevers. Accuracy is the percentage of cases when the most closest example belongs to the same rule and recall-5 – the fraction of cases when such examples occur among top 5 closest examples.

	Qwen-2.5 7B Instruct						YandexGPT-5 Lite 8B Instruct					
Setup	RULEC-GEC			RU-Lang8			RULEC-GEC			RU-Lang8		
	P	R	F0.5	P	R	F0.5	P	R	F0.5	P	R	F0.5
zero-shot	38.2	39.3	38.4	48.9	39.2	46.6	41.7	42.6	41.9	53.8	41.9	50.9
random, 1-shot	40.7	*37.8*	40.1	50.4	37.1	47.1	43.5	41.9	43.2	55.1	42.5	52.0
random, 5-shot	42.4	37.9	41.4	51.6	38.3	48.2	43.7	45.1	44.0	55.4	47.5	53.6
gector, 1-shot	*41.8*	37.6	*40.9*	*53.7*	*38.8*	*49.8*	45.0	*42.5*	44.5	56.9	43.5	53.6
gector, 5-shot	43.9	37.1	42.4	*55.4*	40.2	51.5	46.0	45.4	45.9	*57.2*	*48.3*	*55.2*
gector+FT, 1-shot	41.7	37.2	40.7	52.6	38.1	48.8	*45.4*	42.2	*44.7*	*57.1*	*43.7*	*53.8*
gector+FT, 5-shot	*44.7*	**38.1**	*43.2*	55.3	**40.7**	*51.6*	*46.1*	**45.8**	*46.0*	56.0	47.7	54.1
finetuning	**52.2**	31.2	**46.0**	**61.7**	37.2	**54.5**	**57.3**	38.9	**52.4**	**66.3**	**48.5**	**61.8**
prev. SOTA	70.5	29.1	54.8[2]	73.7	27.3	55.0[1]	70.5	29.1	54.8[2]	73.7	27.3	55.0[1]

Table 7: Comparison of different few-shot example selection methods on RULEC-GEC and RU-Lang8 corpora. The best metric inside the same approach (e.g., 1-shot) is presented in italics and the best overall metric – in bold. [1] refers to Sorokin (2022) and [2] to Sorokin and Nasyrova (2025)

strates higher recall in 3 of 4 experiments.

We also apply our approach to English BEA corpus, see Appendix G.3 for details. There GECTOR-based example selection leads to a small (about 1.5% F0.5 score) but consistent improvement.

6 Discussion and conclusions

In our study we make two principal contributions:

1. We release a new LORuGEC corpus, which differs from existing Russian GEC corpora in data sources, difficulty and typology of errors and, most importantly, the presence of rule labels. This annotation makes our corpus more suitable for L1 educational applications, such as school writing assistants.

2. We compare several methods of in-context learning on our data and discover that retrieval-based demonstration selection significantly outperforms random choice. The retrieval leverages the encoder-based GECTOR model. Contrastive finetuning of this encoder to predict rule labels further improves correction quality.

Since our data has a distinct error distribution, we also check the second result on other corpora. We observe that GECTOR-based in-context examples retrieval is beneficial over random selection. This confirms that our approach effectively works for general GEC data, at least for Russian.

As a future work, we plan to extend our corpus in terms of size and errors number. We have already collected a small pool of sentences with multiple errors, which require additional verification. To reduce annotation burden, we also experimented with example generation. We found that LLM may effectively generate 5 examples to the required rules: 23 out of 25 samples were correct, however, they were shorter and less variable than the manually collected ones, thus further investigation is needed.

We also believe our approach to be viable in domains where task-induced similarity differs from surface meaning similarity. For example, in code retrieval similar programs are not the ones using the same variable names but the ones using the same algorithms. So we hope to investigate the usefulness of our approach in other fields.

7 Limitations

1. As for any LLM-based method, our results are prompt-dependent. In particular, our prompts were optimized towards YandexGPT models and might be suboptimal for the models from other families or for later versions of YandexGPT. However, we did not find any major differences in results when slightly modifying the prompt.

2. For now we evaluate our approach only on Russian and the results may differ for other languages. However, the approach itself has no language-specific details.

3. The LORuGEC corpus is rather small in size compared to other GEC corpora, thus the result may change after collecting more analogous data. We addressed this question in the Conclusions section.

4. Though in principle contrastive tuning works with multiple example labels, we were unable to successfully extend our approach to the multilabel case.

8 Ethics considerations

Our work is based on Large Language Models. We acknowledge that such models might be used in a harmful or malicious manner, however, we utilize them only for scientific purposes. Nevertheless, if a retrieved fewshot sample includes an unsafe generation, that may bias the model towards undesirable behaviour. Thus generalizing our method to datasets containing such examples requires additional precautions.

All of the students who participated in the creation of the dataset earned credit hours as a result. The students were informed about the goals of the work and gave their content for dataset publication.

Acknowledgments

References

Anna Alsufieva, Olesya Kisselev, and Sandra Freels. 2012. Results 2012: Using flagship data to develop a Russian learner corpus of academic writing. *Russian Language Journal*, 62:79–105.

Svetlana Berezina and Nikolaj Borisov. 2017. *Russkij yazyk v sxemax i tablicax (in Russian)*. Eksmo, Moskva.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

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.

Steven Coyne, Keisuke Sakaguchi, Diana Galvan-Sosa, Michael Zock, and Kentaro Inui. 2023. Analyzing the Performance of GPT-3.5 and GPT-4 in Grammatical Error Correction. *arXiv preprint arXiv:2303.14342*.

Daniel Dahlmeier, Hwee Tou Ng, and Siew Mei Wu. 2013. Building a large annotated corpus of learner English: The NUS corpus of learner English. In *Proceedings of the Eighth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 22–31, Atlanta, Georgia. Association for Computational Linguistics.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Shengfeng Ye, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wanjia Zhao, Wen Liu,

Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanhong Xu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

Tao Fang, Shu Yang, Kaixin Lan, Derek F Wong, Jinpeng Hu, Lidia S Chao, and Yue Zhang. 2023. Is ChatGPT a Highly Fluent Grammatical Error Correction System? A Comprehensive Evaluation. *arXiv preprint arXiv:2304.01746*.

Pavel Graschenkov, Lada Pasko, Kseniia Studenikina, and Mikhail Tikhomirov. 2024. Russian parametric corpus ruparam (in Russian). *Scientific and Technical Journal of Information Technologies, Mechanics and Optics*, 24(6):991–998.

Matias Jentoft and David Samuel. 2023. NoCoLA: The norwegian corpus of linguistic acceptability. In *Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa)*, pages 610–617.

Masahiro Kaneko and Naoaki Okazaki. 2023. Reducing sequence length by predicting edit spans with large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 10017–10029.

Elena Kochneva. 1983. *Slovar' sochetaemosti slov russkogo yazyka (in Russian)*. Russkij yazyk, Moskva.

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. *arXiv preprint arXiv:2305.18156*.

Katerina Margatina, Timo Schick, Nikolaos Aletras, and Jane Dwivedi-Yu. 2023. Active learning principles for in-context learning with large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5011–5034, Singapore. Association for Computational Linguistics.

AI Meta. 2024. Introducing meta llama 3: The most capable openly available llm to date. *Meta AI*, 2(5):6.

Tomoya Mizumoto, Yuta Hayashibe, Mamoru Komachi, Masaaki Nagata, and Yuji Matsumoto. 2012. The effect of learner corpus size in grammatical error correction of esl writings. In *Proceedings of COLING 2012: Posters*, pages 863–872.

Harsha Nori, Yin Tat Lee, Sheng Zhang, Dean Carignan, Richard Edgar, Nicolo Fusi, Nicholas King, Jonathan Larson, Yuanzhi Li, Weishung Liu, et al. 2023. Can generalist foundation models outcompete special-purpose tuning? case study in medicine. *arXiv preprint arXiv:2311.16452*.

Kostiantyn Omelianchuk, Vitaliy Atrasevych, Artem Chernodub, and Oleksandr Skurzhanskyi. 2020. GECToR – grammatical error correction: Tag, not rewrite. In *Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 163–170, Seattle, WA, USA → Online. Association for Computational Linguistics.

Kostiantyn Omelianchuk, Andrii Liubonko, Oleksandr Skurzhanskyi, Artem Chernodub, Oleksandr Korniienko, and Igor Samokhin. 2024. Pillars of grammatical error correction: Comprehensive inspection of contemporary approaches in the era of large language models. *arXiv preprint arXiv:2404.14914*.

OpenAI. 2023. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Amin Robatian, Mohammad Hajipour, Mohammad Reza Peyghan, Fatemeh Rajabi, Sajjad Amini, Shahrokh Ghaemmaghami, and Iman Gholampour. 2025. GEC-RAG: Improving Generative Error Correction via Retrieval-Augmented Generation for Automatic Speech Recognition Systems. *arXiv preprint arXiv:2501.10734*.

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.

Ditmar Rozental'. 1997. *Spravochnik po pravopisaniyu i stilistike (in Russian)*. Komplekt, SPB.

Alla Rozovskaya and Dan Roth. 2019. Grammar error correction in morphologically rich languages: The

case of Russian. *Transactions of the Association for Computational Linguistics*, 7:1–17.

Ohad Rubin, Jonathan Herzig, and Jonathan Berant. 2022. Learning to retrieve prompts for in-context learning. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2655–2671.

Elena Simakova. 2016. *Russkij yazyk: Novyj polnyj spravochnik dlya podgotovki k EGE' (in Russian)*. AST:Astrel', Moskva.

Alexey Sorokin. 2022. Improved grammatical error correction by ranking elementary edits. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11416–11429.

Alexey Sorokin and Regina Nasyrova. 2025. Gera: A corpus of russian school texts annotated for grammatical error correction. In *Analysis of Images, Social Networks and Texts*, pages 148–163, Cham. Springer Nature Switzerland.

Ekaterina Taktasheva, Maxim Bazhukov, Kirill Koncha, Alena Fenogenova, Ekaterina Artemova, and Vladislav Mikhailov. 2024. Rublimp: Russian benchmark of linguistic minimal pairs. *arXiv preprint arXiv:2406.19232*.

Chenming Tang, Fanyi Qu, and Yunfang Wu. 2024. Ungrammatical-syntax-based in-context example selection for grammatical error correction. *arXiv preprint arXiv:2403.19283*.

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. Association for Computational Linguistics.

Nina Valgina, Nataliya Es'kova, Ol'ga Ivanova, Svetlana Kuz'mina, Vladimir Lopatin, and Lyudmila Chel'cova. 2009. *Pravila russkoj orfografii i punktuacii. Polnyj akademicheskij spravochnik (in Russian)*. AST, Moskva.

Nina Valgina, Ditmar Rozental', and Margarita Fomina. 2002. *Sovremennyj russkij yazyk: Uchebnik (in Russian)*. Logos, Moskva.

Elena Volodina, Yousuf Ali Mohammed, and Julia Klezl. 2021. DaLAJ – a dataset for linguistic acceptability judgments for Swedish. In *Proceedings of the 10th Workshop on NLP for Computer Assisted Language Learning*, pages 28–37, Online. LiU Electronic Press.

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2024. Multilingual e5 text embeddings: A technical report. *arXiv preprint arXiv:2402.05672*.

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman. 2020. BLiMP: A benchmark of linguistic minimal pairs for English. In *Proceedings of the Society for Computation in Linguistics 2020*, pages 409–410, New York, New York. Association for Computational Linguistics.

Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. 2019. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641.

Haoran Wu, Wenxuan Wang, Yuxuan Wan, Wenxiang Jiao, and Michael Lyu. 2023. ChatGPT or Grammarly? Evaluating ChatGPT on Grammatical Error Correction Benchmark. *arXiv preprint arXiv:2303.13648*.

Xin Xu, Yue Liu, Panupong Pasupat, Mehran Kazemi, et al. 2024. In-context learning with retrieved demonstrations for language models: A survey. *arXiv preprint arXiv:2401.11624*.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. 2024. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*.

A Annotation Instruction

Выберите грамматический справочник по русскому языку, затем составьте набор правил.

Для каждого правила найдите 15 примеров (предложений). Предложения должны быть из разных источников и желательно не из художественной литературы. Примеры также не должны быть тривиальными.

Добавьте в предложения нарушения той нормы, которую Вы исследуете. Если есть несколько способов допустить ошибку в правиле, отразите это в собранных примерах.

Для каждого правила протестируйте YandexGPT 3 Pro на его примерах. Если модель не справилась хотя бы в одном примере, то проанализируйте, что отличает сложные предложения, и соберите еще 5-10 сложных примеров.

(Select a reference book for Russian, after that choose the rules for consideration.

For each rule find 15 example sentences that are preferably from different sources and not trivial, avoid using examples from fiction.

Add errors to the sentences based on the rule under consideration. If there are several ways of making a mistake in a rule, this should be reflected in the collected set of sentences for it.

For each rule test the YandexGPT 3 Pro on its sentences. If there are any imperfections in the

model's corrections, analyse what distinguishes complicated sentences and gather 5-10 more complex examples.)

B Educational sources of the rules

- High school Unified State Exam preparation books: (Berezina and Borisov, 2017) (Simakova, 2016)
- Academic handbook on spelling and punctuation: (Valgina et al., 2009), http://orthographia.ru/
- Handbook on the contemporary Russian language: (Valgina et al., 2002), https://pedlib.ru/Books/6/0262/
- Handbook on spelling and stylistics: (Rozental', 1997), https://rosental-book.ru/
- Dictionary of Russian collocations: (Kochneva, 1983)
- Educational web-sources: https://orfogrammka.ru/, https://gramota.ru/biblioteka/spravochniki/, http://old-rozental.ru/, https://grammatika-rus.ru/, https://licey.net/free/4-russkii_yazyk/, https://www.yaklass.ru/p/russky-yazik/

C Rules of Russian grammar in LORuGEC

- **Grammar**
  - 1 Incorrect expression of government
  - 2 Declension of cardinal numerals
  - 3 Declension of numerals *poltora* ('one and a half.NOM'), *poltory* ('one and a half.GEN'), *poltorasta* ('a hundred and fifty.NOM')
  - 4 Agreement between the participle and the word it defines
- **Punctuation**
  - 5 Commas in idiomatic expressions
  - 6 Commas between homogeneous subordinate clauses
  - 7 Commas between subordinate and main clauses
  - 8 Commas between the two conjunctions
  - 9-11 Commas before the conjunction *kak* ('as'): 3 instances
  - 12 Sentences with homogeneous parts
  - 13 Converbs after conjunctions
  - 14 Clauses related to the personal pronoun
  - 15 Clauses that are distant from the word they define
  - 16 Punctuation in meaningful (indecomposable) expressions
  - 17 Linking words and constructions
  - 18 Recurring conjunctions
  - 19 Dashes in sentences with no conjunctions
  - 20 Dashes between the subject and the predicate
  - 21 Dashes in case of appositions
- **Semantics**
  - 22 Collocations
  - 23 Pleonasms
- **Spelling**
  - 24 *n* and *nn* in the suffixes of adjectives
  - 25 Vowels in the suffixes of participles
  - 26 Noun suffixes *on'k*, *en'k*
  - 27 Suffixes *ic*, *ec* in neuter nouns
  - 28 Suffixes *ek*, *ik*
  - 29 Adjective suffixes *insk*, *ensk*
  - 30 Prefixes *pre* and *pri*
  - 31 *y* and *i* after prefixes
  - 32 Vowels after *c*
  - 33 Vowels after sibilants
  - 34 Separating soft and hard signs
  - 35 Hyphens as part of written equivalents of complex words
  - 36 Joint, separate or hyphenated spelling of adverbs
  - 37 Compound adjectives
  - 38 Particle *taki* ('still')
  - 39 *zato* ('at least')
  - 40 *ottogo* ('that is why')
  - 41 *prichyom* and *pritom* ('moreover')
  - 42 *takzhe* ('also')
  - 43 *chtoby* ('to')
  - 44 *pol-* ('half')
  - 45 *ne* (negative particle) with verbs
  - 46 *ne* with adjectives
  - 47 *ne* with participles
  - 48 *ne* with nouns

Complexity. As may be observed in the figure 1, the largest percentages of collected complex rules occur among punctuation and semantics.

D Details on LORuGEC format

The dataset consists of rules, their definitions, information on their complexity for the YandexGPT model, pairs of corresponding tokenized[13] grammatical and ungrammatical sentences (see Table 8). There is some additional information, representing grammar sections which rules pertain to, sources of rules as well as indication of the subset for each sentence (validation or test, see more in the next section). There are few sentences in the dataset that do not contain any errors (see column *Correct source sentences* in Table 1), because it is also crucial to verify if models are prone to hypercorrection. These sentences are also marked with metadata. We also present our data in .M2, which is a conventional GEC format.

An example from LORuGEC in the first format type may be seen in the Table 8.

The same sentence, but expressed in the .M2-standard:

```
S Иванова , как художника , я совсем не знаю .
A 1 2|||None|||||REQUIRED|||-NONE-|||0
A 4 5|||None|||||REQUIRED|||-NONE-|||0
```

According to the .M2-standard, the source text is denoted with `S`, while the corresponding edits are prefixed with `A`. Each edit consists of the error span, error type, correction, if the edit is optional or required, additional remarks and annotator ID, yet we do not make use of error types. The given annotation demonstrates the requirement to delete two commas in the sentence.

E Model hyperparameters

E.1 Model prompt

Our final prompt for grammatical error correction of Russian texts is given in Figure 2.

E.2 Training hyperparameters

We train the model with Huggingface Transformers Trainer using the hyperparameters from Table 9 for all experiments. When two values are given, the first value is used for training from scratch, and the second – for finetuning from a checkpoint that was already trained on a larger general GEC corpus.

We also made the following model-specific changes:

1. Llama-8B-Instruct is tuned using $\text{learning\_rate} = 3\text{e}{-6}$ and YandexGPT5-Lite using $\text{learning\_rate} = 1\text{e}{-6}$. For both these models we use $\text{max\_grad\_norm} = 0.3$.

2. LORA finetuning is performed with $\text{learning\_rate} = 1\text{e}{-4}$ and physical batch size 4.

F Retriever training

F.1 GECTOR pretraining

Since the morphological features of English and Russian differ significantly, we reimplement the GECTOR preprocessing by ourselves. The sets of G-labels correspond to combinations of morphological features, e.g., the label NOUN,Nom+Plur corresponds to putting the noun into Plural number and Nominative case, keeping other morphological features intact. When the corpus is converted into the pairs of word sequences and their edit labels, we implement training using standard HuggingFace Transformer instruments for sequence labeling. We omit the decoder as we do not need the exact surface transformations predicted by the GECTOR model, but only its labels and hidden states.

We train the GECTOR model on the concatenation of RULEC-GEC, ru-Lang8, GERA and 1 million sentences with synthetic errors. When generating synthetic data, we use the Russian subset of Oscar corpus[14] as source and introduce artificial errors simulating the error distribution of three mentioned corpora. The model is initialized from ruRoberta-large[15] model, the hyperparameters of training are given in Table 10.

F.2 Contrastive tuning

As mentioned in Subsection 4.3, we tune the retriever on the task of rule label prediction using contrastive learning. The tuning is performed on the validation set of LORuGEC. The training objective is a standard triplet loss

$$L(h, h^+, h^-) = \max\left(\frac{\rho(h, h^+) - \rho(h, h^-) + \alpha}{t}, 0\right),$$

where ρ is the distance function (e.g., cosine), α is the margin and t is the temperature. Here h^+

[13] We made use of NLTK Tokenizer: https://www.nltk.org/api/nltk.tokenize.html.

[14] https://huggingface.co/datasets/oscar-corpus/OSCAR-2109

[15] https://huggingface.co/ai-forever/ruRoberta-large

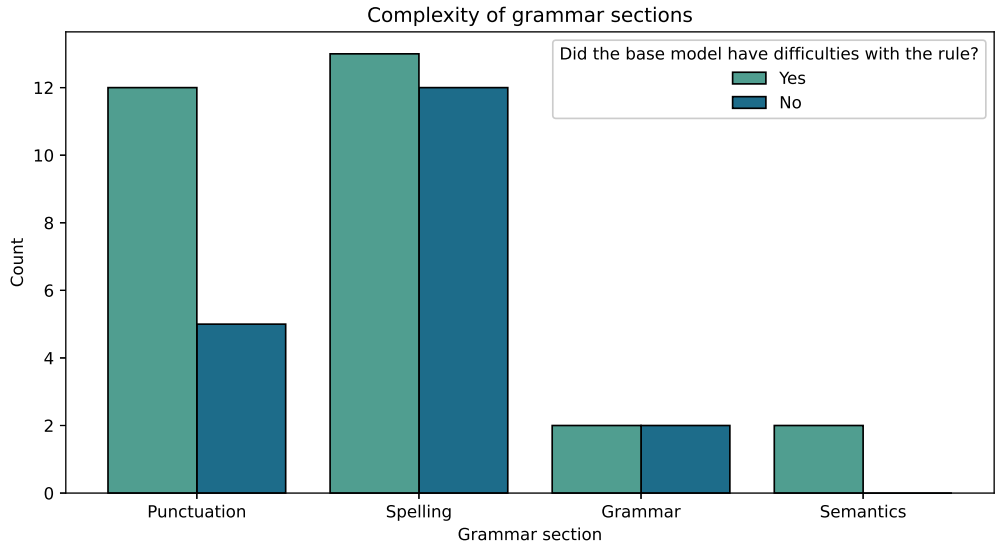

Figure 1: Complexity of different grammar sections is expressed by the number of complex rules for the YandexGPT3 Pro model. We considered the rule to be difficult if the model failed to correct some of its sentences (see 3.1).

The rule	**Did the base model have difficulties with the rule?**	**Initial sentence**	**Correct sentence**
Запятая перед союзом “как”: 2 случай (Commas before the conjunction *kak* ‘as’: second case)	Нет (No)	Иванова , как художника , я совсем не знаю . (I don’t know Ivanov at all , as an artist.)	Иванова как художника я совсем не знаю . (I don’t know Ivanov at all as an artist.)

Table 8: An example of a rule from the dataset with English translation. Additional metadata and other sentences for this rule are omitted for illustrative purposes.

Parameter	value
GPU	A100 80B
num GPUs	1
epochs	3/5
physical batch size	1
batch size	32
learning rate	1e−5/1e−6
max_grad_norm	1.0
optimizer	adafactor
scheduler	triangular
warmup	0.1
weight decay	0.01
precision	fp16
gradient checkpointing	yes

Table 9: Hyperparameters used for 7B/8B language models finetuning.

Parameter	Value
Epochs	3
Batch size	32
Learning rate	1e-5
Optimizer	AdamW
Scheduler	Triangluar
Warmup	0.1

Table 10: Hyperparameters of GECTOR encoder training

is the closest example with the same class label and h^- is the closest example with incorrect label. We represent each sentence with up to 3 hidden states of the most probable error positions in it, provided their probability exceeds the threshold θ. When there is no such position, only the most probable position is extracted. If $\mathcal{H}(s)$ in the set of all hidden states corresponding to a sentence s,

Дорогая языковая модель, после "Исходное предложение" тебе будет дано предложение на русском языке, которое может содержать орфографические, пунктуационные, грамматические и речевые ошибки. Выведи, пожалуйста, только корректный вариант данного предложения, не давая никаких комментариев и не выделяя никаких символов. Твоя задача – минимально изменить текст, не меняй слова и знаки препинания, которые и так правильные.
(*Dear language model, after "The initial sentence" you'll be given a sentence in Russian which may contain spelling, punctuation, grammatical and speech errors. Print, please, only the correct version of this sentence without giving any comments and highlighting any symbols. Your task is to minimally edit the text, don't change the words and punctuation marks that are already correct.*)

Figure 2: Prompt for correction of Russian text. The English translation is given in brackets.

the distance between two sentences is the minimal distance between its state representations:

$$\rho(s, s') = \min_{h \in \mathcal{H}(s), h' \in \mathcal{H}(s')} \rho(h, h').$$

We collect training triples at the beginning of each epoch. For each sentence we search for its nearest neighbours using approximate nearest neighbour (ANN) search with cosine distance. We implement ANN search using Faiss. After processing all the batches we recalculate the hidden representations and update the vector storage. The hyperparameters of contrastive fine-tuning are given in Table 11.

Parameter	Value
Epochs	10
Batch size	8
Learning rate	1e-5
Optimizer	AdamW
Scheduler	Triangluar
Warmup	0.1

Table 11: Hyperparameters of GECTOR encoder training

G Additional results

G.1 Additional results on LORuGEC

Here we evaluate two more models on LORuGEC, repeating the setup of Section 5. We select Llama3-8B-Instruct[16](Meta, 2024) as a medium-size open-source model and GPT4o-2024-05-13(OpenAI, 2023) as a large open-source model. The results are provided in Table 12. The models follow the same pattern as the Qwen2.5-7B and YandexGPT models (see Table 4) with GECTOR+FT being the best few-shot selection method. That means that our approach works both for strong closed-source models and comparably weaker open-source models with limited knowledge of Russian. Interestingly, the GPT4o model almost reaches the level of YandexGPT5-Pro, providing additional evidence that huge language models trained on large amounts of different texts, e.g. educational ones, may not only memorize the rules encountered in these texts, but also apply them to similar language material.

G.2 Additional results on GERA

The comparison of different few-shot selection method on GERA is provided in Table 13.

G.3 Results for English

We evaluate our approach on English, using the development subset of W&I corpus(Bryant et al., 2019), also known as BEA-2019, as our evaluation corpus. We follow the setup of the previous subsection using Qwen2.5-7B Instruct as an open-source model and GPT4o-05-13(OpenAI, 2023) as the closed-source one. The main difference with LORuGEC experiments is the absence of analogous rule-type-annotated corpus for English. Therefore, we cannot readily adapt the contrastive tuning stage. We tried to replace the rule labels with the ERRANT edit types, however, most of the sentences contained several errors of different types. We attempted to train the encoder on the subset of single-error sentences but the approach was not successful.

The generative LLM is finetuned on the W&I corpus training set. For encoder training we utilize a larger cLang-8 corpus (Rothe et al., 2021), corpus parameters are given in Table 14. Note that we don't use BEA-2019 for GECTOR encoder training

[16] https://huggingface.co/meta-llama/Meta-Llama-3-8B-Instruct

Setup	Llama3-8B-Instruct			GPT4o-2024-05-13		
	P	R	F0.5	P	R	F0.5
zero-shot	24.0	30.3	25.1	65.6	68.6	66.2
1-shot, random	30.1	32.8	30.6	71.8	69.5	71.4
5-shot, random	32.1	30.2	31.7	75.3	70.3	74.3
1-shot, GECTOR	30.5	34.9	31.3	72.8	73.2	72.9
5-shot, GECTOR	37.6	37.7	37.6	76.2	74.8	75.9
1-shot, GECTOR+FT	*32.7*	*36.7*	*33.4*	*74.8*	*75.9*	*75.0*
5-shot, GECTOR+FT	*42.7*	**42.9**	*42.7*	**79.6**	**77.9**	**79.2**
ext. finetuning	39.5	14.7	29.6		NA	
ext.+LORuGEC finetuning	**58.6**	33.6	**51.0**		NA	
LORuGEC LORA finetuning	48.8	*36.5*	45.7		NA	

Table 12: Comparison of different LLMs on the LORuGEC test set in zero-shot, few-shot and finetuning modes. Ext. finetuning refers to training on the concatenation of other Russian GEC corpora. The best metric inside the same approach (e.g., 1-shot) is presented in italics and the best overall metric – in bold.

Setup	Qwen-2.5 7B Instruct			YandexGPT-5 Lite 8B Instruct		
	P	R	F0.5	P	R	F0.5
zero-shot	50.3	43.7	48.8	70.8	53.3	66.5
random, 1-shot	58.6	41.4	54.1	76.8	52.2	70.2
random, 5-shot	59.9	40.2	54.5	73.8	56.0	69.4
gector, 1-shot	*58.9*	*44.0*	*55.1*	*77.7*	*54.8*	*71.7*
gector, 5-shot	*65.0*	46.6	*60.2*	*75.4*	58.6	*71.3*
gector+FT, 1-shot	58.6	*44.0*	55.0	76.1	*54.8*	70.7
gector+FT, 5-shot	62.6	**47.9**	59.0	74.8	*58.8*	70.9
finetuning	**75.8**	45.9	**67.1**	**78.0**	**59.0**	**73.3**

Table 13: Comparison of different few-shot example selection methods on GERA. The best metric inside the same approach (e.g., 1-shot) is presented in italics and the best overall metric – in bold.

to simulate the case when large in-domain training corpus is not available.

Corpus	Size	Usage
BEA-2019 train	34308	Training
cLang-8	2372119	encoder training
BEA-2019 dev	4384	Testing

Table 14: GEC corpora used for experiments on English.

Results on BEA-2019 development set are available in Table 15. Here we use ERRANT-3.0(Bryant et al., 2017) to obtain evaluation metrics. Comparing them to the results of the previous subsection, we observe the following:

1. Again, retriever-based selection of demonstration samples produces small but stable improvements. These improvements are stable across models and the number of few-shot examples.
2. However, the difference with baseline is smaller than for LORuGEC. In particular, the achieved enhancements are not sufficient to reach the level of finetuned model. We hypothesize that the reason for this is the larger size of training corpus in case of English that allows the finetuned model to achieve larger improvements over the zero-shot version.

H Implementation of our Approach

We present several responses of the YandexGPT-lite model to the sentences from LORuGEC, which were obtained using different kinds of 1-shot demonstration: random, GECTOR-based or finetuned GECTOR-based, see Figures 3 and 4.

Method	few-shot method	k	Qwen2.5-7B			GPT4o-05-13		
Zero-shot	–	0	36.2	43.4	37.5	34.2	**52.6**	36.8
few-shot	random	1	37.9	42.8	38.8	35.7	51.5	38.0
few-shot	random	5	38.4	43.6	39.4	37.2	49.0	39.1
few-shot	GECTOR	1	39.1	44.4	40.1	37.2	52.0	39.4
few-shot	GECTOR	5	40.0	46.0	41.1	**39.4**	51.5	**41.4**
LLM external	–	0	**53.4**	**48.8**	**52.4**	NA	NA	NA

Table 15: Comparison of different few-shot example selection methods on BEA development set.

Source sentence	Врачи буквально обомлели оттого, что увидели.	The doctors literally were stunned **by what** they saw.
Target sentence	Врачи буквально обомлели от того, что увидели.	(*от того* 'by what' in this context must be written separately.)
Random 1-shot demonstration		
Исходное предложение: Куда запропастилась моя подружонька? Исправленное предложение: Куда запропастилась моя подруженька?		The initial sentence: Where is my girl-friend? The correct sentence: Where is my girl-friend? (The word *подруженька* 'girl-friend' contained a spelling error in the initial sentence.)
LLM Response	Врачи буквально обомлели оттого, что увидели.	(The model did not correct the sentence.)
GECTOR 1-shot demonstration		
Исходное предложение: Родители были счастливы от того, что Владик передумал. Исправленное предложение: Родители были счастливы оттого, что Владик передумал.		The initial sentence: Parents were happy because Vladik changed his mind. The correct sentence: Parents were happy because Vladik changed his mind. (The word *оттого* 'because' must be written in one word in this context.)
LLM Response	Врачи буквально обомлели оттого, что увидели.	(The model did not correct the sentence.)
GECTOR-FT 1-shot demonstration		
Исходное предложение: Я расстроился оттого, что увидел. Исправленное предложение: Я расстроился от того, что увидел.		The initial sentence: I was upset by what I saw. The correct sentence: I was upset by what I saw. (*от того* 'by what' in this context must be written separately.)
LLM Response	Врачи буквально обомлели от того , что увидели .	**(The model successfully corrected the sentence.)**
Conclusion	Only the finetuned GECTOR was able to obtain the sentence with the same preposition and pronoun *от того* 'by what' and the same context in which it must be written separately, not in one word, as opposed to the demonstration chosen by the basic GECTOR. Random selection had a spelling error in it which did not at all resemble the target error. Consequently, LLM was able to correct the sentence only with the GECTOR-FT demonstration.	

Figure 3: Implementation of our approach on the sentence from LORuGEC using YandexGPT5-Lite model. Incorrect parts are marked with red, corrected parts are marked with green for illustrative purposes. There were no highlights in experiments. In the second column we also present English translations of the sentence and demonstrations as well as comments to them in brackets. The same holds for Figure 4.

Source sentence	Кажется, это сон, и я сплю.	It seems, it's a dream, and I'm dreaming.
Target sentence	Кажется, это сон и я сплю.	(Кажется'It seems' is a part of the sentence that is related to both clauses *это сон*'it's a dream' and *я сплю*'I'm dreaming' which are connected by the conjunction *и*'and', that is why there must not be any commas between the clauses before the conjunction.)
Random 1-shot demonstration		
Исходное предложение: Вы можете подумать, что вас это некасается и даже рассмеяться.. Исправленное предложение: Вы можете подумать, что вас это не касается и даже рассмеяться..		The initial sentence: You may think, that it does not concern you and even laugh.. The correct sentence: You may think, that it does not concern you and even laugh.. (In Russian negative particle *не* must be written separately from the verb, so *не касается*'does not concern' must not be written in one word.)
LLM Response	Кажется, это сон, и я сплю.	(The model did not correct the sentence.)
GECTOR 1-shot demonstration		
Исходное предложение: В это время раскрылась дверь поместья, и вышел начальник дозора. Исправленное предложение: В это время раскрылась дверь поместья и вышел начальник дозора.		The initial sentence: At that moment, the door of the manor opened, and the head of the watch came out. The correct sentence: At that moment, the door of the manor opened and the head of the watch came out. (*В это время*'at that moment' denotes the time for both the opening of the door (*раскрылась дверь поместья*) and the arrival of the head of the watch (*вышел начальник дозора*), so there should not be any commas before the conjunction *и*'and' which connects these two clauses.)
LLM Response	Кажется, это сон и я сплю.	**(The model successfully corrected the sentence.)**
GECTOR-FT 1-shot demonstration		
Исходное предложение: Самгин понимал, что говорит плохо, и что слова его не доходят до неё. Исправленное предложение: Самгин понимал, что говорит плохо и что слова его не доходят до неё.		The initial sentence: Samgin knew that he was speaking badly, and that his words were not reaching her. The correct sentence: Samgin knew that he was speaking badly and that his words were not reaching her. (*Самгин понимал*'Samgin knew' about both facts: that he was speaking badly (*что говорит плохо*) and that his words were not reaching her (*что слова его не доходят до неё*), so there must be no comma before the conjunction *и*'and' that connects these clauses.)
LLM Response	Кажется, это сон и я сплю.	**(The model successfully corrected the sentence)**
Conclusion	Both GECTOR-based models selected demonstrations that follow the punctuation pattern of the source sentence. These demonstrations allowed the LLM to effectively correct the sentence, unlike the randomly selected sentence which had to do with incorrect spelling.	

Figure 4: Implementation of our approach on another sentence from LORuGEC.