LORuGEC: the Linguistically Oriented Rule-annotated corpus for Grammatical Error Correction of Russian

Anonymous ACL submission

Abstract

We release LORuGEC – the first rule-annotated corpus for Russian Grammatical Error Correction. The sentences in it are accompanied with the grammar rules governing their spelling. In total, we collected 48 rules with 348 sentences for validation and 612 for testing. LORuGEC occurs to be challenging for open-source LLMs: the best F0.5-score is achieved by Qwen2.5-7B and is only 44%. The closed YandexGPT4 Pro model achieves the score of 73%. By using a rule-informed retriever for few-shot example selection, we improve these scores up to 56% for Qwen and 80% for YandexGPT.

1 Introduction

Since the first works on Grammatical Error Correction (GEC), its primary application was for second language acquisition, due to the fact that people tend to make multiple errors, when studying a foreign language. That is why most of the corpora for GEC is based on foreign learners' texts or contain a mix of second (L2) and first (L1) language data. For example, in the case of English only the LOCNESS Corpus(Bryant et al., 2019) is based on L1 essays, while NUCLE(Dahlmeier et al., 2013) and Cambridge English Write&Improve Corpus (W&I)(Bryant et al., 2019) include only L2 data. The same holds for Russian, where both RULEC-GEC(Rozovskaya and Roth, 2019) and RU-Lang8(Trinh and Rozovskaya, 2021) consist of L2 and heritage data and only the recent GERA(Sorokin and Nasyrova, 2024)[1] is based on the native speakers' school texts.

As was observed multiple times, L2 and L1 texts differ by the error distribution(Bryant et al., 2019; Flachs et al., 2020; Sorokin and Nasyrova, 2024). However, another factor affecting the complexity of grammatical errors is the source of data. Most of the time, free-form essays serve as source texts for GEC corpora. While writing them, people tend to select expressions they are more confident in, reducing the risk of making grammatical errors, so some complex constructions become underrepresented in GEC corpora. Thus, the models trained on such data have limited ability to correct complex errors, restricting their educational usefulness.

This observation was verified empirically by training large language models (LLMs) on the GEC task using existing Russian corpora, such as GERA and RU-Lang8. We found that after such training LLMs improve mostly precision, while the change in recall is either less notable or even negative. Generally speaking, LLMs become more strict and less creative, which indicates that they excel in fixing "familiar" types of errors, which they were trained on, but mostly refrain from correcting other types – performing even worse than its basic version applied in a zero-shot mode.

Due to these considerations, our initial goal was to study the ability of large language models to correct complex grammatical errors. The current work primarily describes the first and the main part of this study – data collection. Our approach is case-oriented: we form a list of complex rules, using grammar handbooks as sources of data and then ask the annotators to collect sentence examples whose spelling is guided by these rules.

Eventually, we apply several models and approaches to the test sample of our data. Unsurprisingly, finetuning occurs to be suboptimal, as opposed to the few-shot approach which yields the best results. We also briefly compare several methods of few-shot example selection. We hope that our data will become useful both for NLP and educational purposes. We make it freely available[2].

[1] https://github.com/ReginaNasyrova/GERA

[2] To appear in the final version of the paper. The corpus is available in supplementary data.

2 Related work

2.1 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, 2024). 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[3] speakers. The authors comprised a list of 23 error type labels that cover (morpho)syntactic, lexical and spelling errors, and sentences were annotated according to them by the two annotators with linguistic backgrounds.

The RU-Lang8 Dataset constitutes a subset of the Lang-8 Corpus(Mizumoto et al., 2012) learner corpus, which is based on the language learning website[4], rather than university essays, that is why most texts in it appear 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.

2.2 Linguistically motivated data for GEC

GEC corpora are conventionally based on real-world learner data, not a predefined error taxonomy. A partial example of error-driven approach was a work of Volodina et al. (2021), where the four principal error types from the existing dataset 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. Actually, one may even convert a dataset of minimal pairs, like BLIMP, to GEC-like format by using the ungrammatical element of the pair as the source and the grammatical one – as the target, that is precisely the approach 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).

[3]People who were exposed to the language at home, but grew up in places where other languages prevail.

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

3 LORuGEC

We aimed at creating a Russian GEC dataset that would be more challenging for large language models and more linguistically-oriented, than existing corpora. Errors in it reflect specific rules of Russian grammar that are harder for models to obtain only by training on the masked or next token prediction tasks, for several reasons. Firstly, training data may lack diversity, hence, e.g., models struggle with correcting spelling errors in words with uncommon prefixes.

Secondly, in order to implement some of the selected rules, it is not enough to have profound knowledge of tokens and their co-occurrence, they also require deeper understanding of semantics. Some of the challenges that models face are with common particles that may be written in one word or separately with the next token, depending on their meaning in a context (see Example (1)), or commas that may be omitted in the sentences with several clauses (where commas, as a general rule, must be), if they have a common semantic component, for example, expressing place or time (Example (2)).

3.1 Data

Rules of Russian grammar as well as examples to them were selected manually from the following grammar reference books, their electronic versions and educational websites (see more details on data extraction in the next section):

- 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/

(1) a. Он пошел **не** смотря вниз.

Он пошел не смотря вниз.
He went not looking down

'He went not looking down'

b. Он пошел **не**смотря на предупреждение.

Он пошел несмотря на предупреждение.
He went not looking at warning

'He went despite the warning.'

(2) a. Они заполняли форму**,** и им приходило уведомление.

Они заполняли форму и им приходило уведомление
They filled form and to them came notification

'They filled the form, and a notification came to them.'

b. **Ранее** они заполняли форму и им приходило уведомление.

Ранее они заполняли форму и им приходило уведомление
Earlier they filled form and to them came notification

'Earlier they filled the form and [earlier] a notification came to them.'

(3) Дети**,** гуляя по парку**,** ели мороженое.

Дети гуляя по парку ели мороженое
Kids walking around park ate ice cream

'Kids ate ice cream, [while] walking around the park.'

- 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/

3.2 Collection and Annotation

Data was extracted and annotated by three bachelor students with a linguistic background, who are also Russian native speakers, then it was verified by the two principle annotators. The annotators were given the following instruction and task description (we refer to Appendix E for the exact instruction):

- Select a source/sources of rules (see the list of chosen educational handbooks and websites in the section above), then choose several rules from different grammar sections: punctuation, spelling, grammar[5] and semantics.
- Find or construct 15 examples for each of the selected rules. As there is no available information on large language models' training data, to reduce the risk of compromising the dataset, several precautions have to be taken:
  - Preferably, choose sentences from different sources.
  - Avoid using quotations from fiction.
  - Refrain from selecting commonplace examples.
- Corrupt a fragment of each sentence which has to do with the rule it was selected for. If there may be a number of ways to make a mistake in a rule, it should be taken into account, while transforming the sentences for this rule. For instance, in Russian converbial clauses in the middle of the sentence must be marked with commas on both sides – Example (3) – so there are at least three ways of making a mistake: by overlooking the first comma, the second one or both.

[5]The word *grammar* is polysemous, in GEC all kinds of errors in a text, except for the factual ones, are considered to be *grammatical errors*. Yet there are also specifically *grammatical errors*, which have to do with grammatical categories, e.g. wrong choice of number.

Our goal is to include diverse examples of errors for each rule, since it would more precisely reflect the set of possible mistakes in a text.

- For each rule test the YandexGPT3 Pro[6] model on its sentences. If there are any imperfections in the way the model has corrected them, analyse, what makes these sentences different from the others and gather 5-10 more challenging examples.

3.3 Data Format

The dataset consists of rules, their definitions, information on their complexity for the YandexGPT model, pairs of corresponding tokenized[7] grammatical and ungrammatical sentences (see Table 1). 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 2), 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. 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:

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

3.4 Rules Description and Statistics

We gathered 48 rules from 4 grammar sections. The majority of them represent punctuation and spelling:

- **Grammar**
  - 1 Incorrect expression of government
  - 2 Declension of cardinal numerals
  - 3 Declension of numerals *poltora*, *poltory*, *poltorasta*
  - 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*: 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

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

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

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*: 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 1: An example of a rule from the dataset with English translation. Additional metadata and other sentences for this rule are omitted.

38 Particle *taki*
39 *zato*
40 *ottogo*
41 *prichyom* and *pritom*
42 *takzhe*
43 *chtoby*
44 *pol-*
45 *ne* with verbs
46 *ne* with adjectives
47 *ne* with participles
48 *ne* with nouns

Our research during the annotation showed that 29 out of 48 collected rules were challenging for the YandexGPT3 Pro. As may be observed in the Figure 1, the largest percentages of collected complex rules occur among punctuation and semantics. This partly proves our hypothesis that rules which require the understanding of semantics pose a more serious challenge to LLMs.

We collected 960 pairs of sentences, 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 (see Figure 2).

Consequently, the size of the test subset is twice as large as the size of the validation one (see Table 2). Additionaly, unlike the latter, only the test subset includes initially correct sentences (for hypercorrection considerations). In both samples, however, two thirds of the sentences come from complex rules.

3.5 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, 2024), our data differs in several aspects:

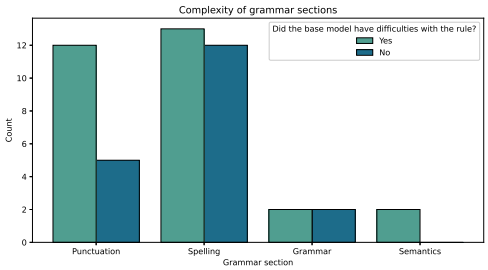

Figure 1: Complexity of different grammar sections is expressed by the number of complex rules for the YandexGPT3 Pro model. We considered the rule to be difficult if the model failed to correct some of its sentences (see 3.2).

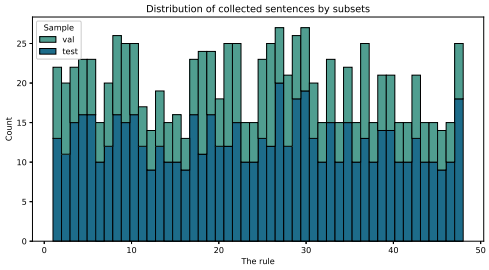

Figure 2: Distribution of sentences for each rule among validation and test samples.

- 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 purposely created for evaluation purposes, not for training. Therefore, it has no training subset and is much smaller than other corpora (see Table 3). On the other hand, almost all sentences in our corpus contain errors and are supposed to be challenging in contrast to other GEC data.

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 2: 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 3: Quantitative comparison of GEC datasets for Russian.

corpus	P	R	F0.5	uncov., %
RULEC-GEC	58.5	31.8	50.1	42.0
RU-Lang8	62.8	37.7	55.4	48.8
GERA	72.9	47.2	65.7	33.7
LORuGEC	50.8	17.3	36.7	21.9

Table 4: Comparison of GEC model performance and difficult fraction (uncov., %) for different Russian GEC corpora. The model is Qwen2.5-7B finetuned on concatenation of Russian 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 the Table 4, LORuGEC has the lowest percentage of errors, whose corrections cannot be generated by a rule-based generator. This generator was designed to cover all the pattern-based corrections, such as punctuation errors, word form changes, deletion, insertion or replacement of closed word categories (prepositions, conjunctions and pronouns), spelling errors, etc. Despite this, the GEC model finetuned on the concatenation of 3 Russian GEC corpora (see Section 4 for details) has much lower scores on LORuGEC than on other corpora. It implies that on LORuGEC GEC models struggle with discriminating between correct and incorrect edits, not with generating the suggestions.

4 Model evaluation

In the first series of our experiments we evaluate several open-source models as well as the closed YandexGPT model[8]. Between the open-source models we select the multilingual Qwen2.5-3B Instruct[9], Qwen2.5-7B Instruct[10] (Yang et al., 2024) and the T-Lite 7B model[11], which is also based on Qwen. We selected these models among other variants as during preliminary experiments they showed a decent ability to correct grammatical errors in a zero-shot mode.

As it 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.

For all the models we report results of 0-shot, 1-shot, 3-shot and 5-shot runs obtained with the prompt given in Appendix A. The demonstrations for few-shot are selected at random. We also evaluate finetuned versions of open-source models in zero-shot mode. Since the validation part of our corpus is too small to use it for training, we tune the models on the concatenation of available Russian GEC corpora: RULEC-GEC, Ru-Lang8 and GERA.

As it 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 first result of our work is the difference between closed-source and open-source models (see Table 5). A partial explanation is the larger size of YandexGPT Pro model, however, the Lite model also clearly outperforms the open-source models. We have two possible explanations: first, many examples are taken from the school textbooks

[8] https://yandex.cloud/ru/docs/foundation-models/concepts/yandexgpt/models, assessed 20th January, 2025.

[9] https://huggingface.co/Qwen/Qwen2.5-3B-Instruct

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

[11] https://huggingface.co/t-tech/T-lite-it-1.0qwen

Model	0-shot			1-shot			3-shot			5-shot			FT		
	P	R	F0.5	P	R	F0.5	P	R	F0.5	P	R	F0.5	P	R	F0.5
Qwen-3B	29.1	31.6	29.6	29.6	29.0	29.5	32.7	30.2	32.2	46.3	36.9	44.1	45.1	18.6	35.1
Qwen-7B	38.5	37.3	38.2	43.3	36.1	41.7	46.4	36.5	44.0	46.3	36.9	44.1	50.8	17.3	36.7
T-Lite	31.7	45.4	33.8	37.9	43.7	38.9	40.5	44.6	41.3	41.8	44.5	42.3	54.1	22.4	**42.1**
YaGPT4 Lite	63.5	64.2	63.7	67.9	66.4	67.6	67.8	64.6	67.2	67.0	65.5	66.7		NA	
YaGPT4 Pro	68.2	68.2	**68.2**	72.0	66.6	**70.8**	74.9	67.6	**73.3**	79.7	69.1	**73.5**		NA	

Table 5: Comparison of different LLMs on the test set in zero-shot, few-shot and finetuning (FT) mode.

that likely were in the training data of Yandex models. Second, open-source LLMs are aligned on "creative" instruction-following tasks that require the model to significantly rewrite the input text(Ouyang et al., 2022; Taori et al., 2023), such as making the text more engaging or more formal. This makes large language models prone to overcorrection and hallucination.

As for the open-source LLMs, we also observe a clear difference between the behaviour of the basic and finetuned models. The finetuned models follow the pattern of traditional Russian GEC models based on smaller LLMs or Transformer networks as their precision is much higher than recall(Sorokin, 2022). On the contrast, pretrained LLMs have moderate recall but poor precision compared to earlier results on other corpora, implying that a large fraction of their edits is unnecessary. The manual analysis demonstrated that the pretrained models (in particular, T-Lite) tend to overcorrect, not only correcting evident ungrammatical constructions, but also trying to improve text fluency or make it more "standardized". We suppose the alignment procedure of modern instruction-tuned LLMs to be the reason, since traditional alignment datasets contain a significant fraction of text editing tasks, which require more extensive rewriting, than GEC. Additionally, the T-Lite model often does not follow the prompt precisely, adding superfluous explanations or comments, but they are avoided with few-shot demonstrations.

To unveil the potential of few-shot learning and additional corpora information, we perform a second series of experiments. During it, we try to make the demonstrations related to the input text in order to provide more relevant few-shot examples. The simplest solution could be to select the demonstrations from the same rule subset as the sentence under consideration. However, for test sentences rule labels could potentially not be available. Since the rule set is open and no corpus can cover all of the grammatical rules of the language, training a rule classifier is also not a complete solution.

Our demonstration selection method consists in training a sentence embedder and retrieving the closest neighbours of the test example using the embedding similarity. To do so, we finetune an encoder model using triplet loss, selecting as positive samples sentences from the same rule subset, the procedure is described in more details in Appendix B. Finetuning is performed on the validation set of our corpus. We initialize the embedder from the analogue of GECTOR model(Omelianchuk et al., 2020) trained on Russian grammatical error data. To evaluate the embedder we compare the results of few-shot learning using random and retrieved examples, results are available in Table 6.

We observe that similarity-based retrieval consistently outperforms the random one. Since the retriever was trained on rule annotation from our corpus, it proves the utility of rule labeling not only for linguistic, but also for practical purposes.

5 Conclusion

We created a linguistically-oriented evaluation corpus for Grammatical Error Correction of Russian. It occurs to be challenging to current open-source models both in zero-shot mode or after finetuning on other Russian GEC data. However, the closed YandexGPT Pro4 model yields much higher scores, achieving the F0.5 score of 68% in zero-shot mode and 73% with 5-shot.

Since our corpus is additionally equipped with rule type information, we also show the utility of this annotation by training an encoder to assign similar vectors for examples with analogous mistakes. Using the trained encoder to select similar examples, we improve the quality of 5-shot error correction up to 80%. We hope our study will shed additional light on the role of linguistic information in grammatical error correction.

Model	Selection	1-shot			3-shot			5-shot		
		P	R	F0.5	P	R	F0.5	P	R	F0.5
Qwen-7B	random	43.3	36.1	41.7	46.4	36.5	44.0	46.3	36.9	44.1
	retrieval	48.2	45.2	47.6	56.4	52.4	55.6	57.1	53.0	56.3
YandexGPT4 Lite	random	67.9	66.4	67.6	67.8	64.6	67.2	67.0	65.5	66.7
	retrieval	72.9	69.8	72.3	75.0	71.0	74.1	75.8	68.5	74.2
YandexGPT4 Pro	random	72.0	66.6	70.8	74.9	67.6	73.3	79.7	69.1	73.5
	retrieval	78.0	72.1	**76.7**	81.0	72.1	**79.1**	81.5	73.0	**80.0**

Table 6: Comparison of naive (*random*) and similarity-based (*retrieval*) few-shot example selection methods.

6 Limitations

- First, despite its representativeness, our corpus does not cover all possible rules of Russian grammar, so the performance may differ for other rules.
- Our corpus does not reflect the distribution of error types in natural language. By design, it reflects more the theoretical knowledge of Russian grammar rules than the practical performance of the models.
- Most of examples in our corpus have exactly one error of a given type, for sentences with two or more errors model performance may drop.
- Though our corpus was created to be challenging for the YandexGPT3 model, the next generation of LLMs shows much higher performance on it, achieving the F0.5-score of 80%. So the notion of "difficulty" evidently changes with the progress of LLMs.
- We utilize the closed-source Yandex GPT model. Evaluation results may change slightly with new version releases. In particular, as models become stronger, the difference between different methods will diminish.

7 Ethical considerations

- All of the students who participated in the creation of the dataset earned credit hours as a result and were preliminary informed that afterwards it would be made publicly available and consented to it.
- Although we aimed at collecting sentences which LLMs did not encounter in the training data (see 3.2), we cannot fully guarantee this as training datasets are not accessible.
- In order to make our evaluations reproducible, we use zero temperature and present the hyperparameters' values and prompts (see Appendices A, C).

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.

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.

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.

Simon Flachs, Ophélie Lacroix, Helen Yannakoudakis, Marek Rei, and Anders Søgaard. 2020. Grammatical error correction in low error density domains: A new benchmark and analyses. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8467–8478, Online. Association for Computational Linguistics.

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.

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

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.

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.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

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.

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, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Alexey Sorokin and Regina Nasyrova. 2024. GERA: a corpus of Russian school texts annotated for Grammatical Error Correction. In *Proceedings of The 12th International Conference on Analysis of Images, Social Networks and Texts, Springer LNCS Vol. 15419, to appear*.

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*.

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`.

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.

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.

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 Prompt for LLMs

We prompted the LLMs with the following instruction:

Ты – квалифицированный редактор текстов, прекрасно знающий правила русского языка. Тебе будет дан текст на русском языке. Твоя задача – исправить все пунктуационные, орфографические, грамматические и речевые ошибки в приведённом тексте, не меняя его смысл.

Твой ответ должен включать в себя только исправленный текст. Ни в коем случае не пиши никаких комментариев или пояснений.

(*You are a qualified text editor who knows the rules of Russian perfectly well. You will be given a text in Russian. Your task is to correct all punctuation, spelling, grammar and speech errors in the following text without changing its meaning.*

Your answer should include only the corrected text. In any case, do not write any comments or explanations.)

B Retriever tuning details

Here we briefly describe our retriever tuning procedure. As it is studied in a separate submitted paper, we do not dive into details but describe it briefly.

The goal of retriever tuning is to assign similar vector representations to texts from the same class. It is trained on triples of the form $\langle h, h^+, h^- \rangle$, where h is the current sample, h^+ is the positive anchor belonging to the same class and h^- is the negative anchor from another class. We try to minimize the distance between h and h^+ and maximize the distance between h and h^- by minimizing the triplet loss

$$loss(h, h^+, h^-) = \max(\rho(h, h^+) - \rho(h, h^-) + \alpha, 0),$$

where $\alpha = 0.1$ is the margin and ρ is the distance function (cosine similarity).

Since GECTOR uses its hidden states, not the [CLS] token to predict edit labels, we represent a text t by the set $\mathcal{T}(t)$ of encoder outputs corresponding to its 3 most probable error positions. In the formula above $\rho(h, h^+)$ is actually a minimum of all the distances $\rho(h_i, h_j^+)$, $h_i \in \mathcal{T}(t), h_j \in \mathcal{T}(t^+)$. We observe that even without contrastive tuning the closest neighbours of GECTOR hidden states belong to the text of the same corpus. The goal of contrastive tuning is to strengthen this property.

On each epoch we simply iterate over all training data samples and for each sample h select the closest state representing a text from the same class as h^+ and the closest state representing a text from another class as h^-. In contrastive learning terms, the triples are composed from a hard positive and a hard negative. We perform such tuning for 10 epochs.

C LLM hyperparameters

We train the model with Huggingface Transformers Trainer using the hyperparameters from Table 7 for all experiments.

During inference we process the input text using the model tokenizer chat template. When performing few-shot, demonstration samples are added as user messages and their corrections – as assistant messages.

Parameter	value
GPU	A100 80B
num GPUs	1
epochs	3
physical batch size	1 (4 for 3B model)
batch size	32
learning rate	1e−5
max_grad_norm	1.0
optimizer	adafactor
scheduler	triangular
warmup	0.1
weight decay	0.01
precision	fp16
gradient checkpointing	yes

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

Model	P	R	F0.5
Qwen2.5-3B-Instruct[14]	29.1	31.6	29.6
Qwen2.5-7B-Instruct[15]	38.5	37.3	38.2
T-Lite[16]	31.7	45.4	33.8
Llama3-8B-Instruct[17]	27.5	29.6	27.9

Table 8: Results on different models in zero-shot mode on LORuGEC test set.

D Additional fewshot results

During model selection we compared several models, results are provided in Table 8. We also tried to evaluate several other models available in the Russian NLP community, such as ruGPT3.5-13B[12] and Vikhr[13]. However, they were very bad in following instructions, in particular, we didn't manage to prevent them from generating additional content and hallucinating.

E 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

[12] https://huggingface.co/ai-forever/ruGPT-3.5-13B

[13] https://huggingface.co/Vikhrmodels/Vikhr-Nemo-12B-Instruct-R-21-09-24

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.)