Computational Linguistics and Intellectual Technologies:
Proceedings of the International Conference "Dialogue 2025"

April 23–25, 2025

An adaptation of sequence labeling approach to Grammatical Error Correction for Russian language

Regina Nasyrova
MSU Institute for AI
Moscow, Lomonosov avenue, 27, corp. 1
r.nasyrova at iai.msu.ru

Alexey Sorokin
MSU Institute for AI; Yandex
a.sorokin at iai.msu.ru

Abstract

We propose a realization of sequence labeling approach to grammatical error correction of Russian. We discuss the difficulties of method adaptation and the modifications we made in order to deal with Russian morphology. Our model outperforms SOTA single model approaches on GERA corpus, achieving the F0.5 score of 69.9%, and shows solid performance on other corpora as well.

Keywords: Grammatical Error Correction, sequence labeling, language models finetuning
DOI: 10.28995/2075-7182-2025-23-XX-XX

Адаптация метода разметки последовательности для исправления грамматических ошибок в русскоязычных текстах

Аннотация

В работе предлагается адаптация метода исправления грамматических ошибок, основанного на разметке последовательности, к русскому языку. Мы подробно описываем сделанные модификации, позволяющие учитывать сложную морфологию русского языка в задаче исправления грамматических ошибок. На корпусе GERA наш метод показывает качество на уровне передовых подходов, достигая F0.5-меры в 67%.

Ключевые слова: Исправление грамматических ошибок, разметка последовательности, языковые модели

1 Introduction

Grammatical Error Correction is the task of converting a source text to its correct variant without any grammatical errors, namely punctuation, orthographic, syntactic, lexical and other mistakes. As any text-to-text task, it is naturally treated as "translation" from the language of ungrammatical texts to the language of grammatical ones. Consequently, standard models for machine translation (MT), such as Transformer, can be used for GEC task without adaptation. These models are trained on large corpora of parallel data, containing pairs of source sentences and their corrected versions.

Despite being fruitful and successful, this approach does not take into account the crucial difference between GEC and machine translation: in case of MT source and target texts are not superficially related. These texts may even use different alphabets. Although the source and the target languages may have some related words, the word order usually undergoes significant changes during translation. However, the correspondence between initial texts and target texts in GEC is less arbitrary. Most of the words remain the same during the correction and the ones subject to modification often do not change their

positions. Though sometimes grammar correction requires complete rewriting of the sentence, albeit such cases are relatively rare.

Moreover, single word edits are also restricted. For example, in case of morphological errors the correct word form belongs to the same lexeme and may be selected from the finite list of the source word inflections. Likewise, typo corrections usually belong to the finite set of dictionary words on Levenshtein distance 1 from the source word. Word deletions and insertions primarily involve closed parts of speech, such as prepositions, determiners and punctuation marks. Given all of this, the ability of sequence-to-sequence models to generate arbitrary texts is redundant during GEC task and may even be detrimental due to the hallucinations that change the meaning of the original text.

Due to these considerations, it might be beneficial to formalize GEC as sequence labeling task as opposed to sequence transduction task. Instead of generating the target text, the sequence labeling model predicts individual word edits that transform the original sequence of words into the correct one. This approach was proposed in the seminal GECTOR paper(Omelianchuk et al., 2020) for the English language, achieving the state-of-the-art performance at the time of publication (2020). In addition to its high quality, the GECTOR approach has other benefits: sequence labeling is much faster than sequence transduction and requires less data to converge during training. It is also more interpretable than the usual sequence rewriting as individual edit operations correspond to common error patterns, such as choosing a wrong word form or an incorrect preposition.

Unfortunately, this interpretability does not come for free: the more complex is the morphology of the language, the more labour is required to design the label system reflecting it. Because of this, we do not know any equivalents of GECTOR for other languages than English except Chinese. We fill this gap by creating a GECTOR-like model for Russian and show its competitive performance on Russian GEC benchmarks.

2 Related Work

The task of GEC has long been perceived as either a classification task or an instance of machine translation. The first approach entails the training of classifiers to predict the most likely correction out of a confusion set for the particular error type, for example, incorrect usage of determiners or prepositions (Dahlmeier and Ng, 2012; Rozovskaya and Roth, 2019). The second approach defines GEC as the task of translating a sentence from the grammatically "incorrect language" to the grammatically "correct" one (Náplava and Straka, 2019; Grundkiewicz et al., 2019). Both solutions have their advantages and limitations. While classifiers achieve high performance on the error types they have been trained for, they are unable to correct other types of mistakes. MT models, on the other hand, are able to handle various types of errors, but require much more training data and computational resources, than classifiers.

Another way of managing grammatical errors is to assign a transformation label to each token in a sentence, so that after all transformations are done, the correct version of the sentence is obtained (Omelianchuk et al., 2020; Mesham et al., 2023). This approach takes into account the fact that GEC does not change most tokens in sentences, unlike ordinary MT.

Recent approaches involve Large Language Models (LLMs). In (Kaneko and Okazaki, 2023) develop the idea of GEC saving most tokens in the sentence by making large language models generate edit spans and corrections instead of the whole target sentence:

```
Source sentence: Through thousands of years.
Target sentence: Through the thousands of years.
Target response: (1,1, ‘‘the’’)
```

One of the pointed out limitations is that the models are unable to generate responses in the given format without instruction tuning, i.e. in zero-shot or few-shot settings. However, this method allows to optimize inference time and cost.

In (Omelianchuk et al., 2024) LLMs in a zero-shot setting, finetuned LLMs, sequence-to-sequence models and edit-based models are compared on the GEC task and it is noted that, on the one hand, LLMs have a higher recall due to the greater creativity, which, on the other hand, leads to hypercorrection effects. They observe no obvious leader among GEC models of different type. While LLMs have the

highest recall, the highest precision is achieved by a much smaller model from (Sorokin, 2022). This model is a two-staged edit-based pipeline, consisting of edit generator and edit reranker, and presents current SOTA results for the Russian language. As GEC models of various type differ in strengths and weaknesses, they may be complementary, that is why, the authors of (Omelianchuk et al., 2024) propose their ensembling. As a result, a simple majority voting of best single-model systems yields even better results for English. They assume that majority voting enables to minimize noisy edits that are inconsistent among models, while elevating reliable edits, which makes it a fruitful approach to GEC.

3 Model description

3.1 GECTOR

In this subsection we briefly describe the original GECTOR model. For more information, we recommend to follow the source paper (Omelianchuk et al., 2020). In this paper authors consider grammatical error correction as prediction of word-level edits, corresponding to insertion, deletion and replacement operations (KEEP stands for keeping the source word unchanged). REPLACE_X corresponds to replacing current word with another word X, APPEND_X – to insertion of X after the current word and DELETE – to word deletion. In addition to these *basic transformations*, task-specific *g-transformations* are introduced. They include noun number and verb form change, for example, when a tag $VERB_FORM_VBZ_VBN is predicted, it means changing the 3rd person form of a verb, e. g. *pushes*, to its past tense form *pushed*. These transformations are implemented using a morphological dictionary. Besides inflection labels, several other operations are used, such as changing word capitalization and merging two adjacent forms. The described label system cannot handle changing a word and adding another token (e. g., a comma) after it simultaneously. This issue is solved using iterative approach, when the model output is passed again as input until only KEEP labels are predicted.

The proposed labeling allows to formalize GEC as sequence labeling. Consequently, the task is solved by finetuning any pretrained encoder on the task of label prediction. This allowed (Omelianchuk et al., 2020) to reach SOTA level for English in 2020. Moreover, sequence labeling is much cheaper than encoder-decoder architectures for training and inference.

3.2 Label inventory for Russian

The advantages mentioned above make GECTOR a promising approach for other languages as well. However, the only language GECTOR was adapted to is Chinese (Zhang et al., 2022). The key problem is morphological complexity: for a language with a large number of grammatical categories the number of g-transformations grows exponentially. Also, designing a set of labels for a morphologically complex language requires additional effort besides pure enumeration.

We refer to the Figure 1 for the description of label extraction. To implement it, we developed an algorithm of linguistic alignment, which is a modification of Levenshtein distance algorithm that has penalties for different lemmas and parts of speech and also accounts for merged-separate-hyphenated spelling of words. In order to obtain lemmas, parts of speech and morphological features, Spacy(Honnibal et al., 2020) was used.

In the English GECTOR model, a relatively large label set of 5000 operations is used. The majority of them consists of REPLACE_X labels corresponding, in particular, to spelling errors. To reduce vocabulary size and make model learning easier, we predict a dedicated SPELL tag for spelling errors. Their corrections are generated in the postprocessing phase, see Subsection 3.4 below.

3.3 Model architecture changes

As was already mentioned, the original GECTOR model cannot handle word modification and inserting another word after it in one step, decomposing edit process to multiple phases, see Table 1. In preliminary experiments we found the iterative editing suboptimal and decided to implement word insertion via INSERT operations applied to spaces between words, not APPEND tags. Our scheme is illustrated in the Figure 2. More precisely, we modified the conventional token classification task so that labels would be

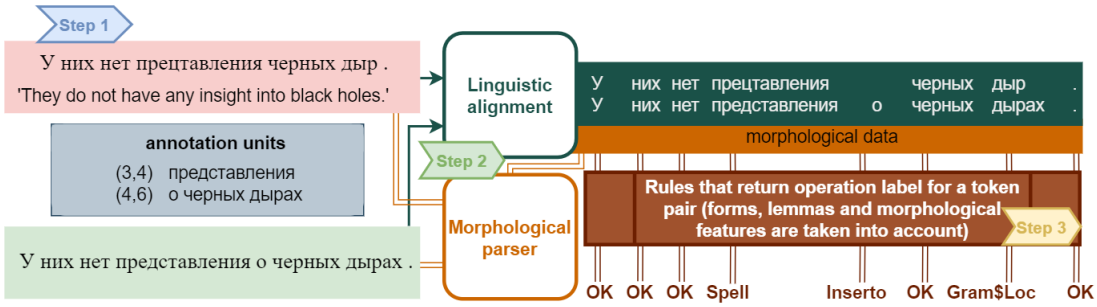

Figure 1: Our preprocessing pipeline. 1. Collecting a grammatical variant of source sentence, using error indices and corrections from annotation units. Source sentence is highlighted with light red, while target sentence – with light green. 2. Both sentences are passed through the morphological parser and linguistic alignment algorithm. As a result, pairs of corresponding tokens are gathered (word columns highlighted with emerald) as well as their morphological features and lemmas. 3. Adopting the information collected during the step 2, rules assign each token in the source text an operation label, so that if all operations are implemented, the source text would be transformed into the target sentence. E.g. in the given sentence only three non-KEEP operations are required: correcting a spelling error in *prectavleniya*, inserting *o* after it and changing case of noun *dyr* to locative. N.B. KEEP is replaced with OK in the figure for illustrative purposes.

Iter.	Source	Labels	Result
1	CLS Boy fall the floor	APPEND_The LOWER VBD KEEP KEEP	The boy fell the floor
1	CLS The boy fell the floor	KEEP KEEP KEEP APPEND_on KEEP KEEP	The boy fell on the floor

Table 1: An example of iterative GECToR labeling and corresponding sentence edits.

predicted not only for subtokens[1], but also for spaces between them. Several decisions had to be made for it to be possible.

Firstly, determining how to represent tokens and spaces. It is not evident, at first glance, whether using the first or the last subtoken of tokens would be the optimal way to represent them in GEC, as various error types may be encountered both in the beginning and in the end of the word form, e.g. spelling errors are frequently made within the stem, whereas grammatical errors primarily affect inflections. For implementation considerations, we decided to use the embeddings of first subtokens as the representations of tokens. As for the spaces between the tokens, we chose as their representation the average of the immediate preceding and following embeddings.

Secondly, finding a convenient way of implementing this approach. We adopted the following strategy: after the tokenization, two numeral masks are created. The process is reflected as step 2 in Figure 2: light yellow mask (left-mask or LM) and light purple mask (right-mask or RM). They have the same length of $2n + 1$, where n is a number of tokens in a source sentence. It accounts for all tokens, spaces after them and a space in the beginning as an insertion may be there as well. Numbers in dark green font represent spaces, whereas others (in dark brown font) – tokens. LM contains indices of first subtokens of tokens and of spaces' immediate preceding subtokens. RM consists of the former and of spaces' immediate following subtokens. For each of the $2n + 1$ spaces and tokens, a pair of left index and right index would become available: for tokens they would be expressed by the same number, whereas for spaces – by the indices of surrounding left and right subtokens. Afterwards, when a tokenized sentence is passed

[1]We use *subtokens* for units after the tokenization, as they may represent parts of tokens – symbols, word forms or punctuation marks.

through an encoder (ruRoberta-large[2] in our case) and subtoken embeddings are obtained (step 3), masks are used to select only the embeddings of corresponding subtokens, consequently, there are two sets of embeddings: for subtokens 1) from LM and 2) from RM, which are then being averaged (step 4). As a result, $2n + 1$ embeddings are extracted, every second one corresponds to the token in a source text, others – to the spaces for insertions. Token embeddings are first subtoken embeddings, while space embeddings are the averages of surrounding subtokens' embeddings.

Thirdly, our preliminary research showed that models tend to confuse labels for spaces with labels for tokens, that is why another modification was added. We decided to add trainable embeddings of token type, representing spaces or tokens, and combine them (step 5) with subtoken embeddings from the previous step, effectively solving the issue.

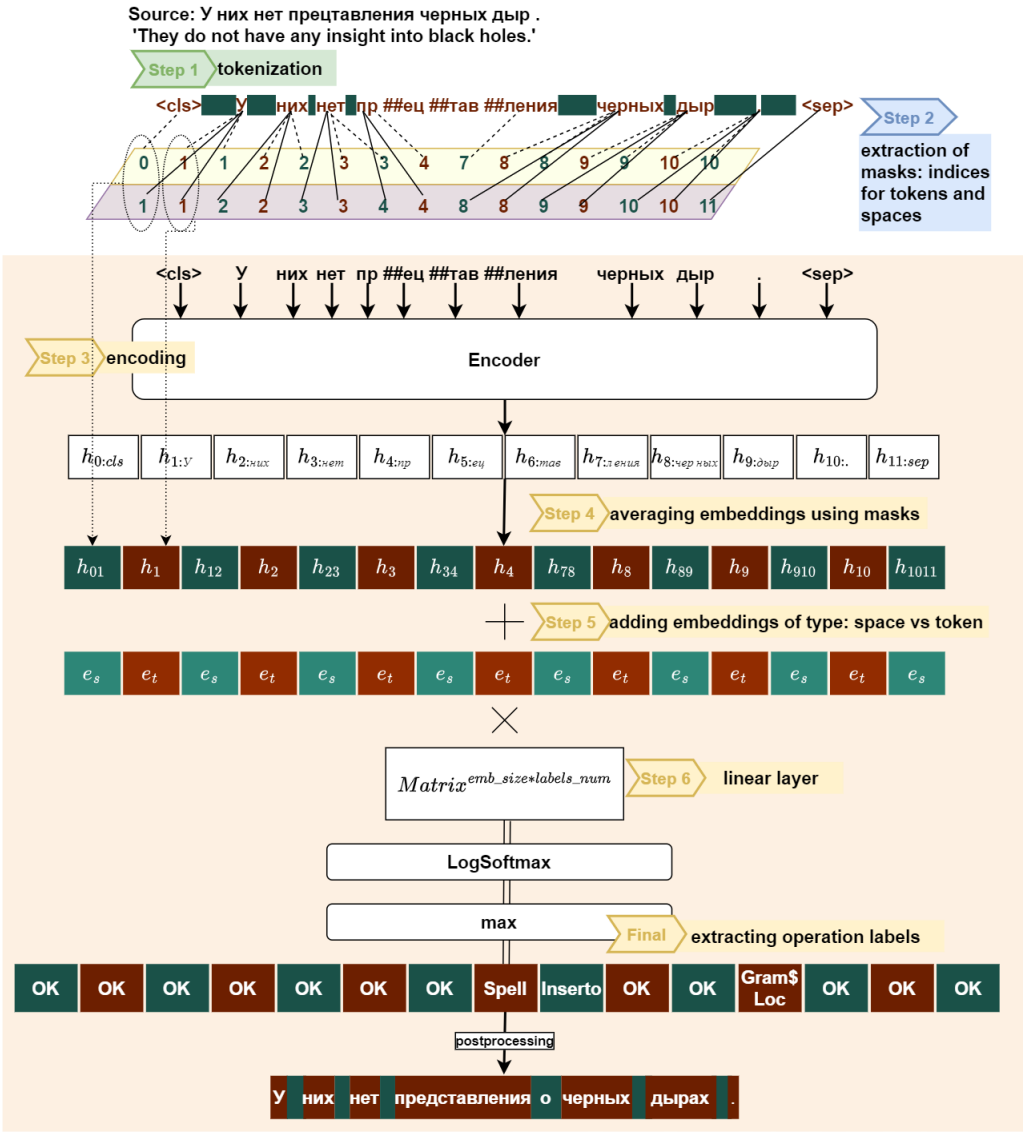

Figure 2: Our model pipeline.

3.4 Edit postprocessing

After predicting the labels, the corresponding output words are inferred. For grammatical labels we utilize the pymorphy2 library(Korobov, 2015) and its *inflect* method that allows to predict any inflected form of a word given the morphological features of the inflected word. In order to apply this function,

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

“Дорогая модель, тебе будут даны слова с опечатками, в скобках будет указано предложение, в котором они встретились. Пожалуйста, выведи исправления этих слов в том же порядке, но без предложения в скобках и каких-либо комментариев, начиная со слова "Ответ:".”

Figure 3: The prompt for spelling errors correction.

we manually convert Conll-U morphological labels predicted by the parser to the Pymorphy format.

For spelling labels we use the external API, namely YandexGPT. We replace the words, preliminarily labeled with SPELL by the SPELL token and pass both source and the tagged sentence using the prompt given in Figure3. We decided to use a large language model instead of local spellcheckers since one needs to select among several possible corrections and traditional models do not provide such possibility.

4 Model evaluation

4.1 Data for training and evaluation

Five existing Russian GEC datasets were used in the experiments: RULEC-GEC(Rozovskaya and Roth, 2019), RU-Lang8(Trinh and Rozovskaya, 2021), GERA(Sorokin and Nasyrova, 2024), RLC-GEC and RLC-Crowd((Kosakin et al., 2024)).

- RULEC-GEC is a subset of the RULEC Corpus(Alsufieva et al., 2012) that contains essays of 12 learners of Russian as a foreign language and 5 heritage speakers.
- RU-Lang8 is the Russian learner subset of Lang-8 Corpus(Mizumoto et al., 2012), which includes small texts produced by speakers of more than 34 languages. Only validation and test samples of RU-Lang8 were manually re-annotated, while training data remains noisy, so the usage of this corpus in our experiments is reduced to these partitions.
- GERA is based on Russian middle school essays, representing the only source of Russian native speakers’ errors.
- RLC-GEC and RLC-Crowd are derived from the Russian Learner Corpus (RLC)(Rakhilina et al., 2016), consisting of texts written by college and university learners of the Russian language from different countries. The former dataset is the subset of RLC which contains annotated corrections, whereas the latter consists of crowdsourced annotations.

Datasets vary greatly in error distribution and size, see Table 2 and Table 3, respectfully. While spelling errors are the most prominent in RULEC-GEC and RU-Lang8, in GERA corrections of punctuation form the largest share. The RLC dataset is the only one that has lexical choice errors as most common. Additionally, unlike the “heritage subset” of RULEC-GEC, RU-Lang8 and GERA, punctuation errors do not occur among the 4 most common error types in the RLC dataset, which, in ins turn, has a much larger fraction of syntactic errors than other corpora. However, the main difference between the datasets lies in the order of the most frequent errors, while the categories mainly remain the same: spelling, lexical choice and noun case errors.

RULEC-GEC (learners)	RULEC-GEC (heritage)	RU-Lang8	GERA	RLC dataset
Spell (18.6)	Spell (42.4)	Spell (19.2)	Punct (42.5)	Lex. (19.7)
Noun:Case (14.0)	Punct (22.9)	Noun:Case (12.6)	Spell (23.6)	Spell (15.8)
Lex. (13.3)	Noun:Case (7.8)	Lex. (11.6)	Lex (13.6)	Syntax (13.8)
Lack (8.9)	Lex. (5.5)	Punct (10.3)	Noun:Case (5.1)	Noun:Case (8.3)

Table 2: Top-4 most common errors in Russian GEC datasets. The data for the first three columns is obtained from (Trinh and Rozovskaya, 2021), statistics for GERA and the RLC dataset are adopted from (Sorokin and Nasyrova, 2024) and (Kosakin et al., 2024). “Lex.” stands for lexical choice errors.

	RULEC-GEC	RU-Lang8	GERA	RLC Dataset
Size (sentences)	12,480	4,412	6,681	31,519 (GEC), 34,150 (Crowd)

Table 3: Quantitative comparison of Russian GEC datasets.

Based on this comparison, we assumed that the datasets would be complementary for the model training, that is, the combination of GERA with RULEC-GEC and RU-Lang8 would be beneficial for the correction of spelling and punctuation errors, while adding RLC datasets to them would facilitate the correction of lexical and grammatical errors and improve the correction of misspellings even further.

We evaluate our model on the test partitions of RULEC-GEC, RU-Lang8 and GERA. Firstly, we train the models on the concatenation of the first three datasets and synthetic samples, containing either 50K, 234K or 1M samples. Afterwards we finetune the model on the dataset in question until convergence and select the best checkpoint according to the metrics on the validation set. Synthetic data is obtained using random corruptions such as word insertion, replacement or deletion, changing the word form, switching adjacent words etc., following (Sorokin, 2022). The frequencies of different errors mimics their distribution in the training subsets of the evaluation datasets.

Moreover, we conducted several experiments, using RLC-GEC and RLC-Crowd datasets in addition to RULEC-GEC, RU-Lang8, GERA and synthetic data during the pretraining stage. Then we repeated the same finetuning, as was described above.

4.2 Model comparison

We compare our model with several methods. The first approach is supervised finetuning of large language models. For comparison we select two 7B models: the well-known multilingual Qwen2.5-7B Instruct model and T-Lite 1.0 that was obtained by further training of Qwen2.5-7B on Russian data. For all the models we use the same training procedure as for GECTOR. We pretrain the models on the concatenation of the datasets using learning rate of 1e-5 and batch size of 32 and finetune them on the dataset in question with batch size 1e-6. All other training parameters are set to default.

We also include in our comparison state-of-the-art methods from the previous works, such as the Transformer encoder-decoder model(Náplava and Straka, 2019) and reranking approach of (Sorokin, 2022). The latter method performs in two phases, creating a candidate pool of errors on the first stage and reranking the generated errors using a binary classifier on the second. The candidates are generated either using rules or applying a generative ruGPT model. All the results of these methods are taken from the corresponding papers. We measure model quality using M2scorer(Dahlmeier et al., 2013) and report F0.5 score as the main metric. Evaluation results are provided in Table 4.

Model	RULEC-GEC	RU-Lang8	GERA
Transformer	63.3/27.5/50.2[1]	55.3/28.5/46.5[2]	NA
ruGPT	65.7/25.4/51.3[3]	NA	66.5/ 28.6/52.6[4]
ruGPT+rerank	73.7/27.3/55.0[3]	NA	78.4/44.4/68.0[4]
Qwen 7B	60.2/32.6/51.5	60.2/36.7/53.4	74.3/48.2/67.1
T-lite	61.0/35.2/53.2	62.5/40.4/56.3	76.3/49.4/68.8
GECTOR synth50K	66.3/18.5/43.7	57.5/25.6/45.1	72.6/46.0/65.1
GECTOR synth1M	61.6/22.5/45.7	59,0/28.8/48.8	75.6/46.5/67.2
rules+ranker	66.5/28.6/52.6[3]	70.5/29.1/54.8[4]	86.9/42.9/71.6[4]
GECTOR synth234K+RLC	68.3/22.6/48.7	62.9/31.3/52.3	78.2/49.1/69.9

Table 4: Results on RULEC-GEC, RU-Lang8 and GERA datasets. Our models are GECTOR synth50K, GECTOR synth1M and GECTOR synth234K+RLC. Each cell contains precision, recall and F0.5 values. The sources are [1] –(Náplava and Straka, 2019), [2] –(Trinh and Rozovskaya, 2021), [3] –(Sorokin, 2022), [4] –(Sorokin and Nasyrova, 2024)

We observe that GECTOR pretrained on 234K synthetic samples and RLC datasets (apart from RULEC-GEC, RU-Lang8 and GERA) is our most effective solution, outperforming the variations pre-

trained on less or more synthetic data without the addition of RLC, which proves the vital role of large amounts of diverse natural data for the success of this approach. GECTOR synth234K+RLC demonstrates impressive results on GERA, performing better than large language models, despite having 20 times less parameters, yet falls 1.7 points behind the SOTA Russian GEC model (rules+ranker), mostly due to a colossal gap in precision.

However, as for the other two datasets, it performs much worse as compared to both large models and the two-staged pipeline, mainly suffering from poor recall. We think there are two possible explanations: firstly, most of errors corrected by GECTOR can be generated by a rule-based generator. Therefore, the performance of these two methods is also correlated. As shown in (Sorokin and Nasyrova, 2024), the rule-based generator outperforms the GPT-based one on GERA, consequently, the error distribution of this corpus is the most suitable for the sequence labeling approach. We assume it is caused by the large number of punctuation errors in GERA. Secondly, the inflection correction stage of GECTOR strongly depends on the accuracy of morphological analysis. Since RULEC-GEC and RU-Lang8 consist of L2 texts, that is, learner errors which considerably vary, the quality of taggers trained on standard corpora significantly decreases when they are applied to out-of-domain texts.

In Table 5 we compare the most frequent error categories in GERA and RULEC-GEC and the quality of our models on them, as these are the datasets on which our model performs the best and the worst. The main difference between the two corpora lies in the lower frequency of punctuation errors and the higher frequency of lexical and grammatical errors in RULEC-GEC as opposed to GERA. On RULEC-GEC we also observe a significant decrease both in precision and recall for punctuation and case mistakes and suppose that the main reason for this is inconsistent annotation of such errors in RULEC-GEC. GECTOR expectedly struggles with lexical errors since due to the structure of vocabulary the model can insert only the most frequent words, making free rewriting impossible in principle.

Category	Corpus	without RLC			with RLC			fraction
		P	R	F0.5	P	R	F0.5	
PUNCT	GERA	79.0	69.2	76.8	75.5	70.4	74.4	39.2
	RULEC	57.1	4.4	16.7	55.1	7.8	24.9	10.4
S:ORTH	GERA	83.0	56.4	75.9	84.6	56.4	76.9	22.2
	RULEC	73.8	53.9	68.7	72.7	45.2	64.8	19.7
L:OTHER	GERA	25.0	4.5	13.1	18.2	3.9	10.5	14.2
	RULEC	37.8	4.1	14.3	35.1	3.2	11.8	23.5
G:NOUN:CASE	GERA	68.8	43.8	63.2	71.8	40.6	62.2	6.3
	RULEC	74.5	40.5	63.8	73.6	42.6	64.3	13.5
S:LETTER:CASE	GERA	89.5	61.8	82.1	78.6	60.0	74.0	5.0
	RULEC	12.1	26.7	13.6	13.3	26.7	14.8	0.3
LACK	GERA	18.2	4.9	11.8	14.3	4.9	10.3	3.7
	RULEC	29.4	2.0	7.8	42.6	4.5	15.9	9.6

Table 5: Results of GECTOR synth1M on GERA and RULEC-GEC corpora for main error categories.

In Table 5 we also compare two variants of our model: the one pretraining on all 4 Russian GEC corpora and the variant without RLC pretraining. The results of comparison are indecisive: on GERA adding high-quality RLC data leads to consistent improvement over all error categories, while the effect on RULEC is less pronounced. Additionally, as training on RLC provides better improvement than adding more synthetic data, we conclude that GECTOR requires substantial amounts of natural data, not only synthetics. Without variable natural errors, its label vocabulary becomes too small to cover all possible mistakes, especially in lexics. Comparing to the English model of (Omelianchuk et al., 2020), their label vocabulary was an order of magnitude larger including about 5000 purely lexical operations, such as inserting or replacing with a particular word. Our research implies that several tens of sentences are not enough to learn such errors on Russian material.

5 Conclusion

We developed a sequence tagging approach to grammatical error correction of Russian, presenting novel methodology of preprocession and model architecture for Russian GEC. We investigated different pretraining setups depending on the size of synthetic data and adopting datasets with distinct error distributions. We found the combination of diverse natural data coming from both learners and native speakers of Russian with medium-sized synthetic data to be the most fruitful one. Our method outperforms larger language models on the GERA dataset but falls behind generative models on two other corpora that require more extensive rewriting. However, our approach can be viable in situations when efficiency is more important than quality and more strict control achieved by tagging is desirable. As a future direction, we plan to combine tagging approach with larger language models, analogous to (Kaneko and Okazaki, 2023).

References

Anna A Alsufieva, Olesya V Kisselev, and Sandra G Freels. 2012. Results 2012: Using flagship data to develop a russian learner corpus of academic writing. *Russian Language Journal*, 62(1):6.

Daniel Dahlmeier and Hwee Tou Ng. 2012. A beam-search decoder for grammatical error correction. // *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, P 568–578.

Daniel Dahlmeier, Hwee Tou Ng, and Siew Mei Wu. 2013. Building a large annotated corpus of learner English: The NUS corpus of learner English. // Joel Tetreault, Jill Burstein, and Claudia Leacock, *Proceedings of the Eighth Workshop on Innovative Use of NLP for Building Educational Applications*, P 22–31, Atlanta, Georgia, June. Association for Computational Linguistics.

Roman Grundkiewicz, Marcin Junczys-Dowmunt, and Kenneth Heafield. 2019. Neural grammatical error correction systems with unsupervised pre-training on synthetic data. // Helen Yannakoudakis, Ekaterina Kochmar, Claudia Leacock, Nitin Madnani, Ildikó Pilán, and Torsten Zesch, *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, P 252–263, Florence, Italy, August. Association for Computational Linguistics.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength natural language processing in Python.

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

Mikhail Korobov. 2015. Morphological analyzer and generator for russian and ukrainian languages. // *Analysis of Images, Social Networks and Texts: 4th International Conference, AIST 2015, Yekaterinburg, Russia, April 9–11, 2015, Revised Selected Papers 4*, P 320–332. Springer.

Daniil Kosakin, Sergei Obiedkov, Ivan Smirnov, Ekaterina Rakhilina, Anastasia Vyrenkova, and Ekaterina Zalivina. 2024. Russian learner corpus: Towards error-cause annotation for L2 Russian. // Nicoletta Calzolari, Min-Yen Kan, Veronique Hoste, Alessandro Lenci, Sakriani Sakti, and Nianwen Xue, *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, P 14240–14258, Torino, Italia, May. ELRA and ICCL.

Stuart Mesham, Christopher Bryant, Marek Rei, and Zheng Yuan. 2023. An extended sequence tagging vocabulary for grammatical error correction. // Andreas Vlachos and Isabelle Augenstein, *Findings of the Association for Computational Linguistics: EACL 2023*, P 1608–1619, Dubrovnik, Croatia, May. Association for Computational Linguistics.

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. // Martin Kay and Christian Boitet, *Proceedings of COLING 2012: Posters*, P 863–872, Mumbai, India, December. The COLING 2012 Organizing Committee.

Jakub Náplava and Milan Straka. 2019. Grammatical error correction in low-resource scenarios. // Wei Xu, Alan Ritter, Tim Baldwin, and Afshin Rahimi, *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, P 346–356, Hong Kong, China, November. Association for Computational Linguistics.

Kostiantyn Omelianchuk, Vitaliy Atrasevych, Artem Chernodub, and Oleksandr Skurzhanskyi. 2020. Gector–grammatical error correction: Tag, not rewrite. // *Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications*, P 163–170.

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*.

Ekaterina Rakhilina, Anastasia Vyrenkova, Elmira Mustakimova, Alina Ladygina, and Ivan Smirnov. 2016. Building a learner corpus for Russian. // Elena Volodina, Gintarė Grigonytė, Ildikó Pilán, Kristina Nilsson Björkenstam, and Lars Borin, *Proceedings of the joint workshop on NLP for Computer Assisted Language Learning and NLP for Language Acquisition*, P 66–75, Umeå, Sweden, November. LiU Electronic Press.

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.

Alexey Sorokin and Regina Nasyrova. 2024. Gera: a corpus of russian school texts annotated for grammatical error correction. // *Proceedings of The 12th International Conference on Analysis of Images, Social Networks and Texts, to appear*.

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

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

Yue Zhang, Zhenghua Li, Zuyi Bao, Jiacheng Li, Bo Zhang, Chen Li, Fei Huang, and Min Zhang. 2022. Mucgec: a multi-reference multi-source evaluation dataset for chinese grammatical error correction. // *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, P 3118–3130.