ISSN 0361-7688, Programming and Computer Software, 2024, Vol. 50, No. 4, pp. 315–321. © Pleiades Publishing, Ltd., 2024.
Russian Text © The Author(s), 2024, published in Programmirovanie, 2024, Vol. 50, No. 4.

RuGECToR: Rule-Based Neural Network Model for Russian Language Grammatical Error Correction

I. A. Khabutdinov[a,b,*] (ORCID: 0009-0007-3828-2487), **A. V. Chashchin**[b,**] (ORCID: 0009-0009-8789-4396), **A. V. Grabovoy**[a,b,***] (ORCID: 0000-0002-4031-0025), **A. S. Kildyakov**[b,****] (ORCID: 0009-0005-9591-0983), and **U. V. Chekhovich**[b,c,*****] (ORCID: 0000-0002-5204-5484)

[a] *Moscow Institute of Physics and Technology (National Research University) Dolgoprudny, Moscow oblast, 141701 Russia*

[b] *Antiplagiat, Moscow, 117105 Russia*

[c] *Federal Research Center "Computer Science and Control" of the Russian Academy of Sciences, Moscow, 119333 Russia*

**e-mail: khabutdinov@ap-team.ru*

***e-mail: chashchin@ap-team.ru*

****e-mail: grabovoy@ap-team.ru*

*****e-mail: kildyakov@ap-team.ru*

******e-mail: chehovich@ap-team.ru*

Received February 21, 2024; revised March 18, 2024; accepted March 18, 2024

Abstract—Grammatical error correction is one of the core natural language processing tasks. Presently, the open-source state-of-the-art sequence tagging for English is the GECToR model. For Russian, this problem does not have equally effective solutions due to the lack of annotated datasets, which motivated the current research. In this paper, we describe the process of creating a synthetic dataset and training the model on it. The GECToR architecture is adapted for the Russian language, and it is called RuGECToR. This architecture is chosen because, unlike the sequence-to-sequence approach, it is easy to interpret and does not require a lot of training data. The aim is to train the model in such a way that it generalizes the morphological properties of the language rather than adapts to a specific training sample. The presented model achieves the quality of **82.5** in the metric $\mathbf{F_{0.5}}$ on synthetic data and **22.2** on the RULEC dataset, which was not used at the training stage.

Keywords: natural language processing, grammatical error correction, sequence tagging, generation of datasets, fine-tuning

DOI: 10.1134/S0361768824700129

1. INTRODUCTION

In the modern world, the automatic correction of grammatical errors (GEC) is an important problem [1–3] due to the increase in the amount of text data [4]. Manually checking large texts is a time-consuming task. In addition, the solution to this problem has specific applications: checking essays [5], correcting messages on social networks [6], correction of texts written by foreign students [7], and search for text borrowings [8]. The latter task is explained by the fact that grammatical errors are one of the ways to bypass the text borrowing search system [9]. To train models for searching text borrowings, texts of scientific papers containing a small number of errors are used. Users may deliberately make many errors, thereby affecting the model's predictions. This leads to the emergence of adversarial attacks on the natural language processing model [10]. Most research in the field of GEC is focused on the English language [11]. For many other languages, including Russian, the number of studies is much smaller [12]. It is especially worth noting the complex morphology of the Russian language, which complicates the GEC.

Currently, there are two most effective approaches to GEC for the English language—Sequence-to-Sequence (Seq2Seq) [13, 14] and Sequence Tagging (ST) [15, 16]. In the Seq2Seq approach, the source sequence is a sentence with errors and the target sequence is a sentence without errors. This approach to GEC works well but has low speed and poor interpretability, since additional functionality must be used to determine the type of error. The models based on the ST approach do not have such drawbacks: it is not necessary to completely generate a sentence without errors—it is sufficient to just mark the errors in the original sentence. In addition, they are easy to interpret, since they solve the classification problem for each token matching it with the desired rule from a given dictionary of correction rules.

In [12], the authors compare these approaches to GEC in texts in Russian. The results show that, on a small amount of annotated data, methods based on the ST approach significantly outperform methods based on the Seq2Seq approach. To compare these methods, the authors present the RULEC dataset containing sentences with annotated grammatical errors. This dataset consists of foreign students' essays.

To solve the GEC problem in English, the most effective ST model is GECToR [15]. This model consists of an encoder based on the transformer architecture [17] and a two-headed classifier. The first head predicts the presence of an error, and the second one predicts the corresponding rule for correcting the error. The corresponding rule is then applied to each token in the sequence for correction. If a token does not contain errors, then the "$KEEP" rule is assigned to this token, which leaves this token unchanged. Training the GECToR model consists of three stages:

(1) training on synthetic data;

(2) fine-tuning on errorful corpus of data;

(3) fine-tuning on a combination of errorful and error-free corpuses of data.

In this paper, we propose an algorithm for generating a synthetic dataset containing texts in Russian for the first stage of training. The GECToR model is adapted into the RuGECToR model for GEC in Russian texts using generated synthetic data and data from open sources. This model does not require a large amount of training data unlike models based on the Seq2Seq architecture and shows competitive quality for English texts. The aim of this study is to train a model capable of generalizing the morphological properties of the Russian language rather than to adjust to the training sample.

2. STATEMENT OF THE GRAMMATICAL ERROR CORRECTION PROBLEM

A set of pairs in which each pair consists of a sentence s_i and the corresponding ground truth correction t_i

$$S = \{(s_i, t_i)\}_{i=1}^{N},$$

where N is the number of pairs, is given. In this work, we assume that each sentence is represented by a sequence of tokens $s_i = \{x_1, x_2, \ldots, x_{n_i}\}$, and each ground truth correction is represented by a sequence of correction rules $t_i = \{y_1, y_2, \ldots, y_{n_i}\}$, where n_i is the length of the ith sentence. By the term *token*, we mean a word; therefore, the decomposition of a sentence into tokens is carried out at the level of words. Each correction rule is an element of a given dictionary $y_j \in t_i, j = 1, \ldots, n_i$. Compilation of the dictionary is described below.

The aim is to find a function T for constructing a map from the set of tokens s_i to the set of correction rules t_i

$$T : s_i \mapsto t_i \in \{0, 1, \ldots, k\}^{n_i},$$

where k is the size of the dictionary of correction rules.

3. THE PROPOSED APPROACH TO GRAMMATICAL ERROR CORRECTION

The input of the GECToR model is a sentence that needs to be corrected. The sentence is then decomposed into a sequence of tokens. Thus, the task of the GEC is reduced to finding a map to match each token with the corresponding rule from the dictionary of correction rules. A dictionary consisting of 5183 rules was built for the Russian language; the rules will be described below in detail.

This statement of the problem has a disadvantage: each token is mapped to only one rule from the dictionary, but sometimes more changes are required to correct the error. To overcome this disadvantage, it is proposed to use an iterative correction: the proposal obtained as a result of the model operation is again fed to the model input. Thus, the model predicts rules for a new sequence of tokens. The rules are designed in such a way that a sequence of tokens with errors can be converted into a sequence of tokens without errors in a finite number of iterations. The following adaptations of the GECToR model for the Russian language were made:

(1) The RuGECToR was trained using two stages:

(i) Training on synthetic errorful corpus of data.

(ii) Fine-tuning on a combination of synthetic errorful and error free corpuses of data.

(2) The stage of applying the model to correct errors was changed using the pymorphy2 library [18].

(3) A dictionary for error correction was compiled.

(4) The Multilingual BERT [19] was used as an encoder.

4. DESCRIPTION OF THE DATASET

To generate synthetic errors at the first stage of training, sentences were taken from the Russian-language Wikipedia [20] and school essays [21]. For the second stage of training, school essays [21] and literary texts [22] were used. Specialized errors were generated for the following parts of speech: verb, adjective, noun, pronoun, participle, and numeral. Before the error generation process begins, the positions of the tokens for the specified parts of speech in all sentences are numbered.

The error generation process is carried out as follows:

Table 1. An example of generating errors in a sentence

Original sentence	Sentence with errors	Correction rules
'Человеческий', **'дух'**, 'непостижимо', 'могуч', ',', 'и', 'убить', 'его', 'в', **'человеке'**, 'почти', 'невозможно', '.'	**'Человеческий-дух'**, 'непостижимо', 'могуч', ',', 'и', 'убить', 'его', 'в', **'чел'**, **'овеке'**, 'почти', 'невозможно', '.'	**$TRANSFORM_SPLIT_HYPHEN**, $KEEP, $KEEP, $KEEP, $KEEP, $KEEP, $KEEP, $KEEP, **$MERGE_SPACE**, $KEEP, $KEEP, $KEEP

Table 2. Training dataset information

Dataset	Number of sentences	Fraction of sentences containing errors	Training stage
Wikipedia + essays	10000000	≈100%	I
Literary works + essays	1000000	≈50%	II

(1) A sentence is randomly chosen, and for each token an error corresponding to the token's part of speech is randomly generated;

(2) Next, each token is assigned a rule from the dictionary of correction rules that should be used to correct the error.

The distribution of errors is close to the uniform one. Table 1 illustrates the process of error generation in a sentence.

Table 2 shows that, for the first stage of training, 10000000 sentences were used, where each sentence contains arbitrary types of errors except punctuation. For the second stage of training, we used 1000000 sentences: 500000 sentences did not contain errors, 250000 sentences contained all kinds of errors except punctuation, and the remaining 250000 sentences contained only punctuation errors. For the second stage of training, literary works were added and Wikipedia was excluded in order to make the model more robust to differences between datasets [23]. School essays and a test subset of the RULEC dataset were used as test datasets.

In [15], the authors divide the rules applied to the source tokens $\{x_1, x_2, \ldots, x_{n_i}\}$ to obtain the target sentence into two types—basic and grammatical rules. In our study, this division was retained. The grammatical rules were selected in such a way that the dictionary of correction rules covers the set of rules of the Russian language. The basic rules perform the most common editing operations at the token level: keeping the current token x_i unchanged—the rule $\$KEEP$, deleting the current token x_i—the rule $\$DELETE$, adding a new token t_1 after the current token x_i—the rule $\$APPEND\_t_1$, and replacing the current token x_i with another token t_2—the rule $\$REPLACE\_t_2$. To generate the rules, 2500 most commonly used Russian words in terms of the Juilland coefficient [24] were used. For each word w_i, the corresponding rules $\$APPEND\_w_i$ and $\$REPLACE\_w_i$ were added.

The grammar rules perform operations that are specific to a particular occurrence. For the Russian language, the following rules are used, which apply directly to the token: changing tense, case, gender, person, and number. Rules for common errors, e.g., spelling “ться”/“тся” and “при”/“пре” are added. Rules that are independent of language are also used: combining two words using a space or hyphen, dividing a hyphenated word into two parts, and changing the capitalization of the first letter of a word.

Consider an example of a correction rule for correcting “красивая” by “красивый.” It is required to convert the adjective from feminine to masculine. Initially, the rules for grammatical transformations were created as follows:

$$\$TRANSFORM\_ADJF\_GEND\_femn\_masc$$

Such rules contain detailed information about

(1) the part of speech the word belongs to,

(2) the grammatical feature that needs to be changed;

(3) the grammeme of the original and corrected word.

We also decided to combine the rules for different parts of speech excluding the name of the part of speech. In addition, it was decided to refuse to indicate the grammeme for the original (incorrect) word, since only information about the new grammeme is needed to correct it. In this study, it is assumed that the model itself will learn to match rules according to the parts of speech. For example, a verb and an adjective can change gender, but a noun cannot. So, the final rules used in our model are as follows:

$$\$TRANSFORM\_GEND\_masc$$

This modification was made in order to reduce the size of the dictionary and increase the number of

Table 3. Examples of sentence corrections made by RuGECToR

Correction rules	Examples of sentences
$KEEP	Я хочу играться
$TRANSFORM_ТЬСЯ/ТСЯ	Я хочу (игратся → играться)
$KEEP	Мы не успели дораспеределиться
$TRANSFORM ТЬСЯ/ТСЯ	Мы не успели (дораспеределится → дораспеределиться)
$TRANSFORM_ПРЕ/ПРИ, $APPEND_ли	Можешь (ли) (приобразовать → преобразовать) это выражение?
$MERGE_SPACE	(Рас скажи → Расскажи) о себе
$TRANSFORM_GEND_femn	Она очень (красивый → красивая)
$MERGE_HYPHEN	(Красно зеленый → Красно-зеленый) цвет
$KEEP	Красный, зеленый цвета
$DELETE	Гектор (плохо → ∅) работает
$KEEP	Гектор отлично работает

Table 4. Example of iterative correction

# of iteration	Correction of a sentence	Position of token to be corrected
Original sentence	Можешь приобразовать это выражение?	0
First iteration	Можешь преобразовать это выражение?	2
Second iteration	Можешь ли преобразовать это выражение?	1

examples for each rule in the training dataset. This achieves a compromise between the generalization ability and the model size, since as the number of rules increases, the size of the classification layer increases.

5. EXPERIMENTS

The multilingual BERT was chosen as a pretrained encoder based on the transformer architecture. The model was trained in two stages each of which lasted 50 epochs. At the first stage of training, the batch size was 32 and at the second stage it was 16. Adam [25] with the parameter $\mathrm{lr} = 10^{-5}$ was chosen as the parameter optimizer.

5.1. Example of Sentence Correction

Table 3 shows examples of error correction using the RuGECToR model in human-written sentences. This is the result of applying the appropriate rules to the input tokens $\{x_1, x_2, \ldots, x_{n_i}\}$ during one iteration. As a result of the model work, the word "дораспределится" was corrected in accordance with the rules of the Russian language although this word was not represented in the training data set.

5.2. Example of Iterative Correction

Table 4 shows examples of iterative error correction. The model predicts the rule $*TRANSFORM_ПРЕ_ПРИ* in the first iteration and the rule $*APPEND_ли* in the second iteration. Even though the sentence is correct already after the first iteration, it reads better with the particle "ли."

5.3. Example of Correcting Real-Life Essays

In this subsection, the result of applying the RuGECToR model to real-life essays is studied. Only the sentences in which the model detected at least one error were considered. Table 5 shows examples of such sentences. The model is not perfect at correcting errors, but it was trained on synthetically generated data and is still able to find errors in real sentences.

6. QUALITY METRICS

The performance of the models was compared using synthetic and real data. To assess quality, the metrics described in [11] were used:

$$R = \frac{\sum_{i=1}^{N} |g_i \cap e_i|}{\sum_{i=1}^{N} |g_i|}, \quad P = \frac{\sum_{i=1}^{N} |g_i \cap e_i|}{\sum_{i=1}^{N} |e_i|},$$

$$F_{0.5} = \frac{(1 + 0.5^2) \cdot R \cdot P}{R + 0.5^2 \cdot P},$$

where N is the number of sentences, e_i is the set of corrections predicted by the model for the sentence s_i, and g_i is the set of ground truth corrections. The intersection of g_i and e_i is defined as

Table 5. Example of correcting real-life essays

Original sentence	Corrected sentence
Таким образом любовь к Родине крайне положительно влияет на человека дарит вдохновение в творчестве и **силы** в борьбе	Таким образом любовь к Родине крайне положительно влияет на **жизнь** человека дарит вдохновение в творчестве и **сил** в борьбе
В этом случае межнациональный конфликт подразумевает конфликт между этническими общностями, обычно проживающих поблизости государствах.	В этом случае межнациональный конфликт подразумевает конфликт между этническими общностями, обычно проживающих поблизости **в** государствах
Автор повествует о бедной семье, в которой **пока** единственный ребенок – старший сын	Автор повествует о бедной семье, в которой единственный ребенок – старший сын
Этот случай показывает, что для родителей не может быть счастья большего, чем радость ребенка, а исполнение мечт и **вера** во что-то чудесное, не позволяет осквернить несчастьям и ненастью светлую душу ребенка	Этот случай показывает, что для родителей не может быть счастья большего, чем радость ребенка, а исполнение мечт и **веры** во что-то чудесное, не позволяет осквернить несчастьям и ненастью светлую душу ребенка

$$g_i \cap e_i = \{e \in e_i | \exists g \in g_i : e = g\}.$$

6.1. Synthetic Data

For the synthetic test dataset, 10 thousand sentences containing errors were generated. The results obtained on the synthetic dataset are shown in Table 6. It can be seen that the metrics R and $F_{0.5}$ at the second stage of training are lower than at the first one. This can be explained by two reasons:

(1) 50% of sentences used at the second stage of training were error free. The accuracy improved, but the number of covered tokens decreased.

(2) At the second stage of training, additional sentences from literary works were used that were not included in the corpus of sentences used at the first stage. This improved the generalization ability of the model due to adding data from another collection.

6.1. Real Data

As real test data, we used the RULEC dataset. The results are shown in Table 7. The model reaches a value of 22.2 in the $F_{0.5}$ metric. This value is higher than the result demonstrated by the basic approach on the RULEC dataset despite the fact that our model was not trained on it. Thus, the proposed model has good generalization ability and is not retrained for a specific dataset. Furthermore, our model performs better on synthetic test data than on real data. This could be expected since the synthetic training and synthetic test datasets have similar distributions, while RULEC is very different from the synthetic training dataset. Table 7 also shows that the generalization ability of the model increases at the second stage of training. On the RULEC dataset, the performance of the model after the second stage is significantly higher than after the first one.

Table 6. Model performance on a synthetic test dataset

Model	Training stage	P	R	$F_{0.5}$
RuGECToR	I	88.4	67.1	83.1
RuGECToR	II	88.5	65.1	82.5

7. CONCLUSIONS

The problem of grammatical error correction has been well studied for the English language, but has not been sufficiently studied for morphologically rich languages due to the complexity of error correction and the small amount of annotated data. This study presents an effective and interpretable grammatical error correction model for the Russian language. For this purpose, a dictionary of correction rules for the Russian language was compiled and our own synthetic dataset was generated, which consists of sentences containing errors. The model application phase is then modified to use these rules. The most effective Sequence Tagging model for the English language—GECToR—was taken as the architecture of the model. Our model achieved 82.5 on synthetic data and 22.2 on real data in the metric $F_{0.5}$. This model outperforms the basic model on a dataset on which it was not trained.

In future, we plan to expand the dictionary of correction rules and adjust the model parameters on other datasets. Another direction for further research is the use of various encoders based on the transformer architecture for the Russian language and their ensemble.

Table 7. Comparison of models on the RULEC dataset

Model	Training dataset	P	R	$F_{0.5}$
Classifiers (learner)	RULEC	22.6	4.8	12.9
Classifiers (minimal sup.)	RULEC	38.0	7.5	21.0
MT	RULEC	30.6	2.9	10.6
RuGECToR	Synthetic (I stage)	23.6	5.6	14.3
RuGECToR	Synthetic (II stage)	40.8	7.9	22.2

FUNDING

This work was supported by ongoing institutional funding. No additional grants to carry out or direct this particular research were obtained.

CONFLICT OF INTEREST

The authors declare that they have no conflicts of interest.

REFERENCES

1. Rozovskaya, A. and Roth, D., "Grammatical error correction: Machine translation and classifiers," *Proc. of the 54th Annual Meeting of the Association for Computational Linguistics,* Vol. 1, Berlin: Association for Computational Linguistics, 2016.
2. Yuan, Z., Stahlberg, F., Rei, M., Byrne, B., and Yannakoudakis, H., "Neural and FST-based approaches to grammatical error correction," *Proc. of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications,* Florence, Association for Computational Linguistics, 2019, pp. 228–239.
3. Bryant, C. and Ng, H.T., "How far are we from fully automatic high quality grammatical error correction?," Association for Computational Linguistics, 2015.
4. Rajput, D., "Review on recent developments in frequent itemset based document clustering, its research trends and applications," *Int. J. Data Anal. Tech. Strategies,* 2019, vol. 11, pp. 176–195.
5. Flickinger, D. and Yu, J., "Toward more precision in correction of grammatical errors," *Proc. of the Seventeenth Conference on Computational Natural Language Learning: Shared Task,* Sofia, Association for Computational Linguistics, 2013, pp. 68–73.
6. Yuan, X., Pham, D., Davidson, S., and Yu, Z., "Eracond: error annotated conversational dialog dataset for grammatical error correction," *Proc. of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies,* Seattle, Association for Computational Linguistics, 2022.
7. Sie Yuen Lee, J. and Seneff, S., "An analysis of grammatical errors in non-native speech in English," *IEEE Spoken Language Technology Workshop,* 2008, pp. 89–92.
8. Zhuravlev, K., Rudakov, K., Inyakin, A., Kirsanov, A., Lisitsa, A., Nikitov, G., Peskov, N., Yaminov, R., and Chekhovich, Y., "The system of recognition of intelligent text reuse "antiplagiat," *Mathematical Methods of Pattern Recognition: 12th All-Russian Conference,* MAKS PRESS, 2005, pp. 329–332.
9. Keck, C.M., "How do university students attempt to avoid plagiarism? A grammatical analysis of undergraduate paraphrasing strategies," *Writing Pedagogy,* 2010, vol. 2, pp. 193–222.
10. Zhang, W.E., Sheng, Q.Z., Alhazmi, A., and Li, C., "Adversarial attacks on deep learning models in natural language processing: A survey," *ACM Trans. Intell. Syst. Technol,* 2020, vol. 11, no. 3.
11. Ng, H.T., Wu, S.M., Briscoe, T., Hadiwinoto, C., and Susanto, R.H., and Bryantc, C., "The CoNLL-2014 shared task on grammatical error correction," *Proc. of the Eighteenth Conference on Computational Natural Language Learning: Shared Task,* Baltimore: Association for Computational Linguistics, June, 2014.
12. Rozovskaya, A. and Roth, D., "Grammar error correction in morphologically rich languages: the case of Russian," *Trans. Assoc. Comput. Linguistics,* 2019, vol. 7, pp. 1–17.
13. Rothe, S., Mallinson, J., Malmi, E., Krause, S., and Severyn, A., "A simple recipe for multilingual grammatical error correction, *Proc. of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing,* Association for Computational Linguistics, 2021, Vol. 2, pp 702–707.
14. Grundkiewicz, R., Junczys-Dowmunt, M., and Heafield, K., "Neural grammatical error correction systems with unsupervised pre-training on synthetic data," *Proc. of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications,* Florence, Association for Computational Linguistics, 2019, pp. 252–263.
15. Omelianchuk, K., Atrasevych, V., Chernodub, A., and Skurzhanskyi, O., "GECTtoR— grammatical error correction: Tag, not rewrite," *Proc. of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications,* Seattle, Association for Computational Linguistics, 2020.
16. Malmi, E., Krause, S., Rothe, S., Mirylenka, D., and Severyn, A., "Encode, tag, realize: High-precision text editing," *Proc. of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP),* Hong Kong, Association for Computational Linguistics, 2019, pp. 5054–5065.
17. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A., Kaiser, L.U., and Polosukhin I., "Attention is all you need," *Advances in Neural Information Processing Systems,* Guyon, I., Luxburg, U.V.,

Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., Eds., Curran Associates, Inc., 2017, Vol. 30.

18. Korobov, M., "Morphological analyzer and Generator for Russian and Ukrainian Languages," *Analysis of Images, Social Networks and Texts,* Ser. *Communications in Computer and Information Science,* Khachay, M. Konstantinova, N., Panchenko, A., Ignatov, D. and Labunets, V., Eds., Springer, 2015, Vol. 542, pp. 320–332.
19. Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K., BERT: Pre-training of deep bidirectional transformers for language understanding, Proc. of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2019, Minneapolis, Association for Computational Linguistics, Vol. 1, pp. 4171–4186.
20. Vrandecic, D. and Krotzsch, M., Wikidata: a free collaborative knowledgebase, *Commun. ACM,* 2014, vol. 57, no. 10, pp. 78–85.
21. Open source collection of school essays. www.kritika24.ru. Accessed November 7, 2022.
22. Open source collection of literary works. https://proza.ru. Accessed November 7, 2022.
23. Trinh, V.A. and Rozovskaya, A., "New dataset and strong baselines for the grammatical error correction of Russian," *Findings of the Association for Computational Linguistics: ACL-IJCNLP,* Association for Computational Linguistics, 2021, pp. 4103–4111.
24. Lyashevskaya, O. and Sharov, S., *Frequency Dictionary of the Modern Russian Language (Based on the Materials of the National Corpus of the Russian Language)*, Moscow: Azbukovnik, 2009 [in Russian].
25. Kingma, D.P. and Ba, J., *Adam: A method for Stochastic Optimization,* 2017.

Translated by A. Klimontovich