Rubic2: Ensemble Model for Russian Lemmatization

Ilia Afanasev[1] **Anna Glazkova**[2,3] **Olga Lyashevskaya**[4,5]
Dmitry Morozov[3,6] **Ivan Smal**[3,7] **Natalia Vlasova**[8]

[1]MTS AI LLC [2]University of Tyumen [3]Russian National Corpus
[4]HSE University [5]Vinogradov Russian Language Institute RAS
[6]The Artificial Intelligence Research Center of Novosibirsk State University
[7]Novosibirsk State University [8]A. K. Ailamazyan Program Systems Institute RAS

Abstract

Pre-trained language models have significantly advanced natural language processing (NLP), particularly in analyzing languages with complex morphological structures. This study addresses lemmatization for the Russian language, the errors in which can critically affect the performance of information retrieval, question answering, and other tasks. We present the results of experiments on generative lemmatization using pre-trained language models. Our findings demonstrate that combining generative models with the existing solutions allows achieving performance that surpasses current results for the lemmatization of Russian. This paper also introduces Rubic2, a new ensemble approach that combines the generative BART-base model, fine-tuned on a manually annotated data set of 2.1 million tokens, with the neural model called Rubic which is currently used for morphological annotation and lemmatization in the Russian National Corpus. Extensive experiments show that Rubic2 outperforms current solutions for the lemmatization of Russian, offering superior results across various text domains and contributing to advancements in NLP applications.

1 Introduction

Lemmatization, the process of linking an inflected word form to its normal form, is essential for associating word variations with lexical resources. Lemmatization is crucial for morphologically rich languages because it reduces complex inflected forms to their base forms, facilitating better analysis and processing of text (Figure 1). This standardization improves information retrieval and enhances understanding in various natural language processing (NLP) tasks (Lyashevskaya et al., 2020; Sorokin et al., 2017).

This work focuses on the lemmatization of Russian words. Despite the availability of effective solutions for this task, several challenges remain for existing models. Previous studies (Kotelnikov et al., 2018; Lyashevskaya et al., 2023, 2020) indicated some difficulties, including lemmatizing words in old spelling, certain modern vocabulary, proper names, and abbreviations. These shortcomings and the rapid development of NLP tools highlight the need for new solutions for lemmatizing Russian-language texts.

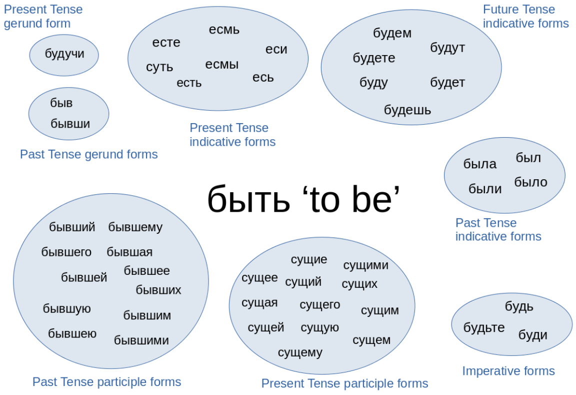

Figure 1: Fragment of the indicative paradigm of the verb быть ‘to be’.

This study explored approaches to generative lemmatization based on pre-trained language models. Preliminary experiments included the comparison of several models for lemmatization in a generative manner. Then, we fine-tuned the BART-base model selected based on the results of preliminary experiments on the manually labeled data set containing 2.1M tokens. This model was compared to existing solutions for lemmatizing Russian, and ensemble approaches were also tested. Finally, we presented Rubic2, a neural ensemble solution for lemmatizing Russian texts that demonstrated state-of-the-art results across various domains by achieving accuracy scores ranging from 82.87% to 99.69%.

The main contributions of this work can be summarized as follows.

Proceedings of the 10th Workshop on Slavic Natural Language Processing (Slavic NLP 2025), pages 157–170
July 31, 2025 ©2025 Association for Computational Linguistics

- We explored generative approaches to lemmatization using pre-trained language models, focusing on the Russian language. By combining generative models with existing solutions, the research achieved performance improvements over current state-of-the-art methods in Russian lemmatization.

- We introduced Rubic2, a new ensemble model that integrates the generative BART-base model with an existing neural model, Rubic, which is used for morphological annotation and lemmatization in the Russian National Corpus.

The rest of the paper is structured in the following way. Section 2 contains a brief review of related work on Russian text lemmatization and generative lemmatization approaches. Section 3 describes the data used for training and testing. Section 4 presents the experimental setup and results. In Section 5, we provide an error analysis. Section 6 concludes this paper.

2 Related Work

2.1 Approaches to Russian Texts Lemmatization

Lemmatization is a crucial task for morphologically rich languages. Over the past two decades, NLP tools for Russian have seen significant advancements, largely thanks to developing tools for morphological parsing and conducting shared tasks which involve texts from various sources. A number of morphological taggers have been developed for Russian including language-specific tools like MyStem (Segalovich, 2003) and PyMorphy2 (Korobov, 2015), as well as multilingual models trained on Russian data such as UDpipe (Straka et al., 2016) and Stanza (Qi et al., 2020). According to the experimental results in (Akhmetov et al., 2020; Kotelnikov et al., 2018), language-specific PyMorphy2 and MyStem show high-quality lemmatization performance across various corpora. The morphological parser MyStem is a console application compatible with various operation systems. It relies on Zaliznyak's dictionary (Zaliznyak, 1977) and can propose a hypothesis for unknown words by finding the closest matches in the vocabulary. MyStem uses context to disambiguate homonyms, evaluate hypotheses, and supports user dictionaries. PyMorphy2 utilizes the OpenCorpora project dictionary (Bocharov et al., 2011) and predicts hypotheses for unknown words using rules and calculates conditional probabilities for all analysis outcomes. Additionally, PyMorphy2 can generate word forms based on grammatical features.

There have been several competitions on lemmatizing Russian in recent years, covering texts from various genres and domains. RU-EVAL, the first shared task on Russian part-of-speech (POS) tagging, lemmatization, and morphological analysis of texts from various domains, including news, technical, and fiction texts (Astaf'eva et al., 2010), took place in 2010, achieving 98% accuracy for lemmatization and 97.3% for POS tagging. At the MorphoRuEval-2017 shared task (Sorokin et al., 2017), the models achieved 97.11% accuracy in predicting morphological features and 96.91% in lemmatization on a diverse data set containing a large number of social media texts. From 2016 to 2019, Russian morphology was also highlighted in SIGMORPHON (Cotterell et al., 2018), with the best result reaching 94.4% accuracy in word inflection within context using the data set based on Wiktionary.

At GramEval-2020 (Lyashevskaya et al., 2020), the best result (ranging from 78.3% to 98% across different domains) was achieved by qbic (Anastasyev, 2020). This neural model combines RuBERT (Kuratov and Arkhipov, 2019) embeddings with morphological data from PyMorphy2, using a BiLSTM network to obtain word encodings. After obtaining word embeddings, three classifiers are applied, each dedicated to a specific task: morphology tagging, lemmatization, and syntax dependency parsing. The lemmatization process occurs in two stages: first, the classifier assigns a specific rule to each token, and then the rule is applied. Each lemmatization rule defines the length of the suffix to be removed and a substring to be added. The system determines a set of 1000 to 2000 lemmatization rules based on the training data. Using this architecture, the Rubic model (Lyashevskaya et al., 2023) was developed for tagging in the Russian National Corpus (Savchuk et al., 2024). It features an improved lemmatization approach that utilizes information from the part-of-speech tagging module and advanced post-processing techniques.

2.2 Generative Lemmatization

Over the past decade, lemmatization systems leverage attentional sequence-to-sequence neural architectures to convert inflected word forms into lemmas at the character level. For instance, Bergmanis and Goldwater (2018) introduced a context-sensitive approach by incorporating surrounding characters without relying on morphological or POS tags. The paper (Pütz et al., 2018) proposed morphologically-informed neural sequence-to-sequence architecture for lemmatization. The results presented in these works showed comparable quality to rule-based baselines.

Recent studies applied a transformer-based model to lemmatizing texts. Over the past few years, generative lemmatization has gained significant attention from the academic community, with current developments demonstrating highly promising results. The proposed models typically take an inflected word form as input, along with POS tags, morphological features, or contextual information. Some research focuses on ancient and historical languages. For instance, during the shared tasks (Dereza et al., 2024; Sprugnoli et al., 2022) models for lemmatization using T5 and ByT5 were proposed (Riemenschneider and Krahn, 2024; Wróbel and Nowak, 2022). These models utilized the word form along with POS tags. In (Riemenschneider and Frank, 2023), T5 was applied for lemmatization using the full sentence context of the word form without receiving or predicting POS tags or morphological features. Dorkin and Sirts (2023) proposed an encoder-decoder architecture for Estonian language lemmatization with several additional morphological features based on a character-level transformer. The BART-large model was applied to lemmatizing Russian texts (Lyashevskaya et al., 2023) and showed promising results compared with a BERT-based lemmatization rules classifier.

Previous studies have shown that transformer-based models perform well in lemmatizing texts across different languages. Several studies have focused on generative lemmatization, which transforms word forms into their lemmas. While there have been initial attempts to use generative models for lemmatizing Russian texts, their full potential remains unclear. There is also a need to evaluate the performance of ensembles combining existing and new models to address their limitations. This requires large-scale experiments on a diverse data set. This study aims to address these research gaps.

3 Data

The experiments were performed using a diverse collection of text samples that included a wide range of genres, text types, domains, time periods, and orthographic variations. The text collection mainly consists of the texts from the Russian National Corpus (RNC)[1] (Savchuk et al., 2024) and the Taiga corpus[2] (Shavrina and Shapovalova, 2017). All source data are freely available to researchers in the Universal Dependencies format[3]. The collected data was manually reannotated by experts in Russian morphology and syntax according to the Russian UD-Ext scheme (Lyashevskaya, 2019). All data is presented in the CONLL-U format. Table 1 reports on the structure of the available training, development, and test data sets.

Based on these data, we have merged the following test sets: *RNC*, which includes a collection of texts from RNC in modern spelling; *RNC*$_{+XVIII}$, which extends RNC with texts written in diverse pre-Soviet orthographies; and the Taiga sets, which consist of test data from the GramEval-2020 shared task and a new set named *CAPS*[4] that includes Soviet telegrams and advertisement texts and contains a large number of words written in capital letters. Thus, the following sets can serve as a base for register-specific evaluation: *fiction*, *news*, *poetry*, social media (*social*), wikipedia (*wiki*), *CAPS*. The size of the small test sets is kept to compare the current and previous results. The resulting test sets are presented in Table 2.

4 Experiments and Results

4.1 Preliminary Experiments

During preliminary experiments, we assessed the ability of generative models to perform lemmatization of Russian words. For this, we selected three types of models:

- BART (Lewis et al., 2020), a transformer-based denoising autoencoder for pre-training a seq2seq model. We used BART-base[5] and

[1] https://ruscorpora.ru/en

[2] https://tatianashavrina.github.io/taiga_site

[3] https://universaldependencies.org/: UD_RussianSynTagRus, UD_RussianTaiga, UD_Russian-Poetry treebanks

[4] Kohen's kappa was utilized to evaluate the agreement between two experts. κ=.932 indicates a reliable quality of the lemma assignment. Most mismatches can be related to the theoretically-plausible "equivalent interpretations" in context (eg. adjective vs. participle, reflexive vs. passive verb, pluralia tantum lemma vs. lemma in singular).

[5] https://huggingface.co/facebook/bart-base

Data set	Train size	Dev size	Test size	Varieties	Shared task
SynTagRus2.8-UDext	1.5M	–	–	fiction, popular science, journalism, news, Wikipedia	GramEval-2020 (1.1M in train; 400K: new data)
Taiga-UDext	.2M	10K	10K	social media, YouTube comments, Q&A, reviews; poetry & prosaic fiction; news	GramEval-2020 (the 17th century texts and SynTagRus excluded)
prose-XX	.1M	.2M	.1M	modern fiction & nonfiction	–
prose-XIX	49K	42K	19K	the 19th c. drama, fiction, & nonfiction	–
old-orthography	93K	9K	15K	pre-Soviet spelling	–
RNC$_{+XVIII}$	75K	4K	7K	18th c. non-standard spelling	–
poetry	35K	–	1K	RNC Poetry corpus	test: GramEval-2020; train: new data
newspapers-XXI	12K	10K	14K	RNC Media & Main corpus (journalism, announcements)	–
CAPS	–	–	1K	telegrams & advertisement	
GramEval-2020	–	–	6K	fiction, news, social, poetry, wiki	GramEval-2020
Total size	2.1M	256K	170K		

Table 1: Data used for training and evaluation, size in *tokens*.

Test sets			
RNC sets		GramEval-2020 (Taiga)	
RNC	142K	fiction	1.1K
RNC$_{+XVIII}$	22K	news	1.3K
CAPS	1.0K	poetry	1.0K
		social	1.1K
		wiki	1.5K

Table 2: Size of test sets, *tokens*.

`BART-large`[6] with 139M and 406M parameters respectively.

- `mBART-50`[7] (Tang et al., 2021), a machine translation sequence-to-sequence model that uses the same baseline architecture as that of multilingual BART (Liu et al., 2020), 680M parameters. mBART-50 was trained on more than 50 languages with a combination of span masking and sentence shuffling.

- `ruT5` (Zmitrovich et al., 2024), a Russian-language text-to-text transformer pre-trained on a corpus including Russian texts from various publicly available resources, which represent diverse domains. The architecture and training procedure are similar to T5 (Raffel et al., 2020). We used two model configurations: `ruT5-base`[8] and `ruT5-large`[9] with 222M and 737M parameters respectively.

The summary of the model architecture configuration including the number of layers and attention heads, the hidden layer dimension, and other characteristics is presented in Table 6 (Appendix A).

For the preliminary experiments, we used a sample of 10K random lemmas from the training set. The test was conducted on a sample of 5K lemmas from the RNC$_{+XVIII}$ test set. Each model was fine-tuned for 20 epochs with a maximum sequence length of 512 tokens. The learning rate was 1e-5 for ruT5 and 4e-5 for BART and mBART.

The model input was presented as the word form with a POS tag and a set of morphological features. The output was the lemma of the word. Additionally, for the model that demonstrated the best performance, we assessed the effectiveness of the use of the word form's context. We considered two variations of using context: full context and a context window of one word. If the context window included the beginning or end of a sentence, they were marked with the tokens BEGIN and END, respectively (see Table 3). Following (Lyashevskaya et al., 2020), we used the lemmatization accuracy metric that represents a standard accuracy metric, disregarding letter capitalization and $\mathrm{e}/\ddot{\mathrm{e}}$ choice.

Table 4 and Figure 2 show the results on the test sample. The highest results in the table are highlighted. The best score using the standard input was achieved by the BART-base model (95.7%). A similar result was demonstrated by the BART-large model (95.62%). mBART, ruT5-base, and ruT5-large showed lower results (89.4%, 91.38%, and 88.74% respectively). The use of context did not

[6] https://huggingface.co/facebook/bart-large
[7] https://huggingface.co/facebook/mbart-large-50
[8] https://huggingface.co/ai-forever/ruT5-base
[9] https://huggingface.co/ai-forever/ruT5-large

Context	Деревня осталась позади за буграми. [*The village was left behind over the hills.*]	
	Example 1	
Form	позади	
Lemma (output)	позади [*behind*]	
Standard Input	**+ Full Context**	**+ Context (Window Size = 1)**
позади ADV Degree:Pos	позади ADV Degree:Pos Деревня осталась позади за буграми.	позади ADV Degree:Pos осталась позади за
	Example 2	
Form	буграми	
Lemma (output)	бугор [*the hill*]	
Standard Input	**+ Full Context**	**+ Context (Window Size = 1)**
буграми NOUN Animacy:Inan Case:Ins Gender:Masc Number:Plur	буграми NOUN Animacy:Inan Case:Ins Gender:Masc Number:Plur Деревня осталась позади за буграми.	буграми NOUN Animacy:Inan Case:Ins Gender:Masc Number:Plur за буграми.
	Example 3	
Form	.	
Lemma (output)	.	
Standard Input	**+ Full Context**	**+ Context (Window Size = 1)**
. PUNCT	. PUNCT Деревня осталась позади за буграми.	. PUNCT буграми. END

Table 3: Examples of input and output formats.

Model	**Accuracy, %**
Standard input	
BART-base	95.70
BART-large	95.62
mBART	89.40
ruT5-base	91.38
ruT5-large	88.74
BART-base$_{+full\ context}$	95.52
BART-base$_{+context\ (window\ size = 1)}$	95.70

Table 4: Results of preliminary experiments (20 epochs).

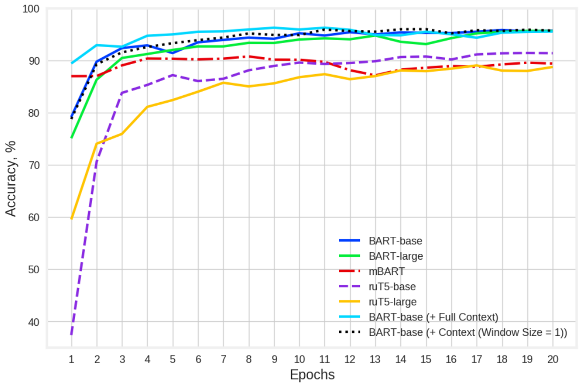

Figure 2: Accuracy scores on the test sample (preliminary experiments).

significantly improve the performance of the BART-base model. The use of full context led to a slight decrease in results, while using a context window had no major impact on the results. Based on the results of preliminary experiments, the BART-base model with the input consisted of the word form with a POS tag and morphological features was chosen for further research.

4.2 BART-based Model

Following the preliminary experiment results, the BART-base model was selected for further analysis, with the input consisting of the word form along with a POS tag and morphological features. We fine-tuned BART-base for 40 epochs on the full train set of 2.1M tokens described in Section 3. For training, we used POS tags and morphological features annotated by experts.

During the development and test phases, POS tags and morphological features were extracted using the Rubic model (Lyashevskaya et al., 2023), as raw texts in subsequent lemmatization do not have expert annotations. The fine-tuned model was evaluated on the development set to select the best-performing version. The highest result on the development set (98.37%) was obtained after 31 epochs (Figure 4, Appendix A). Then, the selected model was evaluated on the test sets and compared with several baselines.

To achieve higher performance on the test set, we applied a set of heuristics to post-process the output of the BART-base model. For instance, if a word was a singular noun in the nominative case, which corresponded to the base form of a noun in Russian, the lemma should have matched the word form. The full list of heuristics is shown in Table 7 (Appendix A). The set of heuristics was proposed by the RNC linguists based on the analysis of frequency dictionaries and texts from RNC. The examples of errors of the BART model that can be corrected using heuristics are presented in Table 8 (Appendix A).

4.3 Baselines

As baselines, we adopted four rule-based and supervised approaches to lemmatizing Russian texts:

- PyMorphy2 (Korobov, 2015) and MyStem (Segalovich, 2003), rule-based morphological analyzers. Both analyzers return all possible lemmatization options for a given word ordered by frequency of occurrence, so we evaluated two lemmatization strategies. In the first one, we used only the first lemmatization option. In the second one, we considered all possible lemmatization options. If any of these options matched the gold lemma, the generated lemma was deemed correct. The second strategy allowed us to assess the theoretical potential of the method to produce correct lemmas. However, in practice, this strategy is not applicable.

- Stanza (Qi et al., 2020), a Python natural language analysis package. Stanza's lemmatizer is designed as a combination of dictionary-based and neural seq2seq lemmatizers.

- Rubic (Lyashevskaya et al., 2023), a neural network algorithm consisting of three steps. First, it generates word embeddings by combining RuBERT (Kuratov and Arkhipov, 2019) embeddings with morphological data from PyMorphy2. Second, these embeddings are processed through a BiLSTM network to obtain word encodings. Finally, three classifiers predict lemmata, morphological information, and dependency tree of the sentence; notably, the lemmatization classifier also relies on the output of the morphological classifier and is further refined with language-specific heuristics. Currently, Rubic is used in the RNC for morphological annotation and lemmatization.

We have considered alternative options to use as a baseline, for example, a simple recurrent neural network (RNN) (Cho et al., 2014). However, it showed extremely low results on the material, when compared with the fine-tuned language models. In addition, the lexical diversity of the material made impossible using the dictionary-based postprocessing heuristics that proved to be useful for other Slavic languages (Afanasev and Lyashevskaya, 2024). Part-of-speech tags sometimes actually worsen performance, which also speaks against using RNN as a part of the Rubic pipeline. Crucially, we intended to use the most robust pipeline possible, and RNN did not meet the criteria, given its previous results for Slavic languages and our preliminary experiments, which, for brevity and clarity, we do not report in the paper.

Among the multilingual models trained on Russian data, we selected Stanza, based on its superior performance compared to other models, particularly UDPipe (Straka et al., 2016), as demonstrated in previous studies (Afanasev, 2023; Afanasev and Lyashevskaya, 2024).

4.4 Experimental Results

BART-base was evaluated on the test sets and compared with the baselines. Performance scores in terms of accuracy score are presented in Table 5. The lines BART$_{postproc}$ and BART show the results for the fine-tuned BART-base with and without post-processing. The asterisk (*) marks PyMorphy2 and MyStem considering all possible lemmatization options. The highest score for each test set is underlined.

The results demonstrated that in most cases neural models outperformed the results of rule-based approaches, even when using the version with all possible lemmatization options. This demonstrated that the complexity of natural language texts and the variety of word forms require more sophisticated approaches for their processing. The best results among the baselines was obtained by Rubic. BART$_{postproc}$ outperformed Rubic on the RNC (+0.12%), RNC$_{+XVIII}$ (+0.11%), news (+0.77%), poetry (+0.58%), and CAPS (+0.49%) test sets. Rubic achieved better scores on social and wiki texts (+0.72% and +0.86% respectively). Both models showed equal performance on the fiction domain (99.22%).

The results showed that no single model was superior across all domains. At the same time, a quick empirical error analysis revealed that different models exhibited different types of errors. Based on this observation, we explored a two-step ensemble learning approach to combine the outputs of the two models that demonstrated the best performance on the test sets: BART$_{postproc}$ and Rubic. In the first step, we compiled a dictionary of letter combinations that do not occur in correct Russian lemmas. In the second step, we checked the lemmas generated by the basic model using the dictionary. If a lemma produced by the basic model contained any of these combinations or special symbols that were not present in the word form, the generated lemma was marked as incorrectly generated. For incorrectly generated lemmas, the lemma obtained from the supporting model was used. We evaluated two ensemble configurations. In the first configuration, BART$_{postproc}$ was the basic model and Rubic was the supporting model (`BART` → `Rubic`), and in the second configuration, their roles were reversed (`Rubic` → `BART`).

The use of ensembles allowed us to achieve better results for most domains. The highest accuracy was achieved using BART → Rubic on the RNC (99.05%), RNC$_{+XVIII}$ (98.85%), fiction (99.39%), and news (99.69%) test sets. For the poetry test set, the scores obtained by BART-base (99.23%) did not improve while using ensembles. For social and wiki texts, the results of Rubic (98.31% and 97.6% respectively) remained unsurpassed. For CAPS, the best score in our experiments was achieved by PyMorphy considering all possible lemmatization options (83.25%). The second-best result was shown by BART → Rubic (82.87%).

In our experiments, the best performance across most test sets was demonstrated by the BART → Rubic ensemble, which we named `Rubic2`. The final workflow of Rubic2 is shown in Figure 3. The pipeline begins with the extraction of morphological features from the input text using the Rubic model. These features (POS tags and morphological attributes) are then passed to the BART-based lemmatizer, which generates a lemma for each token based on the word form and its morphological context. Simultaneously, the same word form is processed by the Rubic lemmatization component, which relies on a combination of RuBERT-based embeddings and morphological analysis using PyMorphy2. These embeddings incorporate both contextual information from the transformer model and rule-based morphological tags, providing an alternative lemma candidate. The outputs of both models are then passed to the Merger block. This component compares the two lemmas using a set of predefined heuristics.

Our findings suggest that using ensemble learning to combine the outputs of different neural network models improves lemmatization performance across texts from various domains. By incorporating a dictionary of impossible letter combinations, we are able to identify errors in generative lemmatization and replace incorrectly generated lemmas with those produced by a neural model based on a different approach. This enables us to combine the strengths of both models while mitigating their weaknesses. In the next section, we examine common errors of both models and analyze which types of errors our approach successfully addresses and which challenges still warrant further investigation.

5 Error Analysis

In this section, we compared the common errors of the BART-base and Rubic models to better understand strengths and limitations of Rubic2.

Since BART-base is a generative model, its errors are primarily associated with the hallucination of symbols and the reproduction of stereotypical patterns in lemmas. The model performs worse with homonymous forms, uninflected words, and words with rare alternations (e.g., веки (word form) → веко (generated lemma), gold lemma - век, 'a century', a homonymous form for веко, 'an eyelid'; леди → ледя, gold lemma - леди, 'a lady'). Errors in words with the prefix пол- are also frequently observed (полдеревни → полдеревня, gold lemma - полдеревни, 'a half of the village'). In nouns ending in -нье, BART usually replaces -нье with -ние (увещанья → увещание, gold lemma - увещанье, 'an exhortation'). Other frequent errors made by BART are related to the lemmatization of numerals (13.32 → 133232, gold lemma - 13.32), hashtags (#life → #de, gold lemma - #life), and user mentions (@Zhirinovskiy → @Zhovovskiy, gold lemma - @Zhirinovskiy). In some instances, BART-base generates unexpected characters (анан�с) and replaces repeated vowels (пепельница → пепэльница, gold lemma - пепельница, 'an ashtray').

In general, Rubic's errors are related to the incorrect selection of lemmatization rules, leading to excessive or insufficient deletion of symbols. In some cases, Rubic does not remove the accent mark (e.g., напыще́нный → напыще́нный, gold

Model	RNC	RNC$_{+XVIII}$	fiction	news	poetry	social	wiki	CAPS
BART	98.72	98.48	99.13	99.62	99.04	96.79	94.99	81.22
BART$_{postproc}$	98.91	98.66	99.22	99.62	99.23	97.59	95.74	81.61
Rubic	98.79	98.55	99.22	98.85	98.65	98.31	97.60	81.12
Stanza	93.79	92.85	97.65	97.32	93.27	95.10	89.70	59.24
MyStem	91.60	91.29	91.64	91.65	90.67	91.98	89.56	78.90
PyMorphy2	91.23	90.42	95.21	94.10	94.23	93.76	89.35	80.15
MyStem*	94.07	93.78	94.25	93.79	91.63	95.10	90.18	81.32
PyMorphy2*	94.08	92.59	97.56	95.79	95.96	96.97	90.25	83.25
BART → Rubic (Rubic2)	99.05	98.85	99.39	99.69	99.23	97.86	96.36	82.87
Rubic → BART	98.84	98.60	99.22	98.93	98.94	98.31	97.60	81.12

Table 5: Lemmatization accuracy scores, %. * – considering all possible lemmatization options.

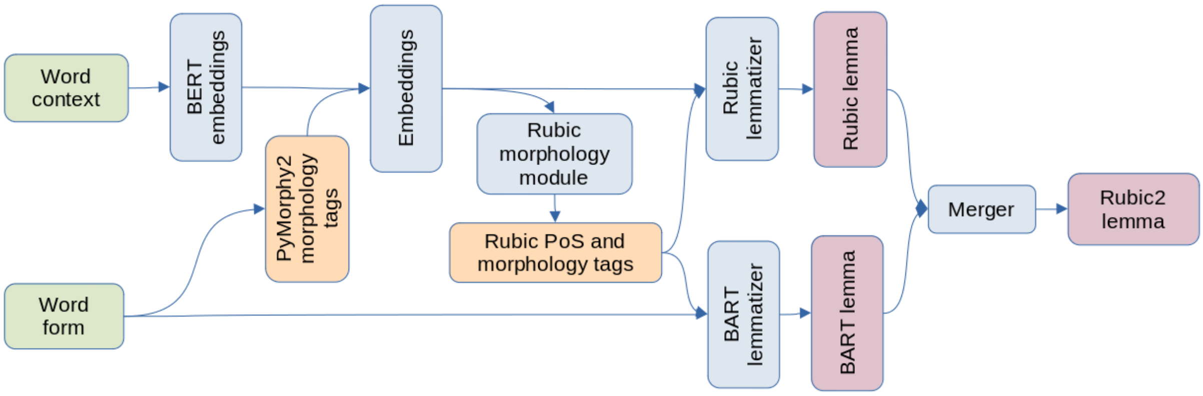

Figure 3: The Rubic2 pipeline.

lemma - напыщенный, ‘pompous’). Rubic often produces errors with nouns ending in -й, altering the word stem or adding suffixes (поцелуй → поцелок, gold lemma - поцелуй, ‘a kiss’; ручей → ручень, gold lemma - ручей, ‘a stream’). Additionally, in several instances, Rubic incorrectly lemmatizes adverbs and prepositions by completely replacing the original word form with new words (краше → хорошо, gold lemma - красиво, ‘beautifully’; промеж → много, gold lemma - промеж, ‘between’). Finally, Rubic performs worse in lemmatizing verbs and proper nouns compared to BART-base.

Both Rubic and BART-base struggle with the lemmatization of abbreviations (often shortened forms with full stop placed after an initial letter or several letters), reflexive verbs, and the words the stems in which end with a soft consonant. Plurale Tantum words and word forms in the plural are also occasionally incorrectly lemmatized. The models also demonstrate more lemmatization errors with nouns that contain fleeting vowels; however, such errors are more typical for Rubic. Some errors are related to the lemmatization of adjectives ending in -ой and -ый. The models exhibit errors when lemmatizing distorted word forms, with Rubic making such mistakes much more frequently. Examples of word form types that are challenging for both models are shown in Table 9 (Appendix A). The presented cases pose a challenge for the further improvement of Rubic2.

6 Conclusion

This paper addresses lemmatization for the Russian language. Our study integrates generative lemmatization and current effective neural models for lemmatizing Russian to address the limitations of both approaches and leverage the advancements in NLP and pre-trained language models. Extensive experiments reveal that the Rubic2 ensemble model presented in this paper shows high performance on various domains ranging from 82.87% to 99.69% in terms of the accuracy score. Given the importance of lemmatization for morphologically rich languages and the effectiveness of the proposed methodology, we believe our work makes a significant contribution to the field.

The current study is limited by the complex format of the input data. Generative lemmatization requires extracting the morphological features with the Rubic model, thus making lemmatization performance reliant on the accuracy of these predictions. Additionally, the model makes errors in certain challenging cases, such as the lemmatization of abbreviations or foreign words. This limitation could be addressed by using a more advanced contextual analysis of word forms. Another direction for future research is to create a linguistic-informed model generating only valid lemmas according to language rules, without needing heuristics for post-processing.

7 Limitations

We identified the following limitations of our study.

Significance of the Results: In this work, we proposed an ensemble model for Russian lemmatization that outperformed the previous state-of-the-art Rubic model on most test sets. The improvement ranged from 0.17 to 1.75%. A conducted bootstrap analysis (N = 1000 resamples) using all test data showed the Rubic mean accuracy of 98.42% (95% CI: [98.36%, 98.49%], min = 98.33%, max = 98.54%) and the Rubic2 mean accuracy of 98.67% (95% CI: [98.62%, 98.73%], min = 98.58%, max = 98.79%) (Figure 5, Appendix A). The difference was statistically significant ($p < 0.01$) in accordance with the Wilcoxon test. The obtained gain is substantial in absolute terms when applying Rubic2 to annotate large text corpora. Moreover, the achieved improvements helped correct several typical errors of the Rubic model (see Section 5).

Dataset Issues: Although the training and testing data cover a wide range of domains, some domains are either absent or underrepresented (Table 1). This may lead to a bias in model fine-tuning towards more common categories.

The input format used for fine-tuning BART is of limited suitability for homonym recognition. The task of lemmatizing homonyms, which is particularly relevant for Russian texts (Lyashevskaya et al., 2011), requires further investigation.

Runtime: We observe a longer lemma generation time compared to the Rubic model (see Table 10, Appendix A). This is due to the larger number of model parameters, its ensemble architecture, and resource constraints. In this work, accuracy is the top priority, as Rubic2 is intended for annotating large text corpora ($\approx 10^8 - 10^9$ tokens). This process involves a one-time annotation of the data, followed by multiple uses of the results.

The duration of one epoch of fine-tuning the BART-base model on the full training set was approximately 4.5 hours using an NVIDIA RTX 4090 GPU and an AMD Ryzen 9 7900X processor. We estimate the total time spent on preparing the final model to be 270 GPU hours.

Usage of Instruction-based LLMs: In this work, we focused on encoder-decoder architectures and did not consider instruction-based LLMs. This limitation is due to our future plans for applying the model to annotate large text corpora, as well as resource constraints. Despite a significant increase in runtime when using BART, Rubic2 allows annotation to be performed within a reasonably limited time. For example, annotating all test data ($\approx$ 300K tokens) takes about eleven minutes in real time. Our preliminary experiments have shown that using instruction-based LLMs would require significantly more time.

Acknowledgments

The article was prepared with the support of the Center for Technologies for Society of the Yandex Cloud platform.

References

Ilia Afanasev. 2023. The use of Khislavichi lect morphological tagging to determine its position in the East Slavic group. In Tenth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial 2023), pages 174–186, Dubrovnik, Croatia. Association for Computational Linguistics.

Ilia Afanasev and Olga Lyashevskaya. 2024. String Similarity Measures for Evaluating the Lemmatisation in Old Church Slavonic, pages 13 – 35. Brill, Leiden, The Netherlands.

Iskander Akhmetov, Alexander Krassovitskiy, Irina Ualiyeva, Alexander F Gelbukh, and Rustam Mussabayev. 2020. An open-source lemmatizer for Russian language based on tree regression models. Research in Computing Science, 149(3):147–153.

DG Anastasyev. 2020. Exploring pretrained models for joint morpho-syntactic parsing of Russian. In Computational Linguistics and Intellectual Technologies, pages 1–12, Moscow, Russia.

I Astaf'eva, A Bonch-Osmolovskaya, A Garejshina, Ju Grishina, V D'jachkov, M Ionov, A Koroleva, M Kudrinsky, A Lityagina, E Luchina, et al. 2010. NLP evaluation: Russian morphological parsers. In Proceedings of Dialog Conference, Moscow, Russia.

Toms Bergmanis and Sharon Goldwater. 2018. Context sensitive neural lemmatization with Lematus. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), pages 1391–1400, New Orleans, Louisiana. Association for Computational Linguistics.

V Bocharov, S Bichineva, D Granovsky, N Ostapuk, and M Stepanova. 2011. Quality assurance tools in the OpenCorpora project. In Computational Linguistics and Intellectual Technologies, pages 101–109, Moscow, Russia.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 1724–1734, Doha, Qatar. Association for Computational Linguistics.

Ryan Cotterell, Christo Kirov, John Sylak-Glassman, Géraldine Walther, Ekaterina Vylomova, Arya D McCarthy, Katharina Kann, Sabrina J Mielke, Garrett Nicolai, Miikka Silfverberg, et al. 2018. The CoNLL–SIGMORPHON 2018 shared task: Universal morphological reinflection. In Proceedings of the CoNLL–SIGMORPHON 2018 Shared Task: Universal Morphological Reinflection, pages 1–27, Brussels. Association for Computational Linguistics.

Oksana Dereza, Adrian Doyle, Priya Rani, Atul Kr. Ojha, Pádraic Moran, and John McCrae. 2024. Findings of the SIGTYP 2024 shared task on word embedding evaluation for ancient and historical languages. In Proceedings of the 6th Workshop on Research in Computational Linguistic Typology and Multilingual NLP, pages 160–172, St. Julian's, Malta. Association for Computational Linguistics.

Aleksei Dorkin and Kairit Sirts. 2023. Comparison of current approaches to lemmatization: A case study in Estonian. In Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa), pages 280–285, Tórshavn, Faroe Islands. University of Tartu Library.

Mikhail Korobov. 2015. Morphological analyzer and generator for Russian and Ukrainian languages. In Analysis of Images, Social Networks and Texts: 4th International Conference, AIST 2015, Yekaterinburg, Russia, April 9–11, 2015, Revised Selected Papers 4, pages 320–332. Springer.

Evgeny Kotelnikov, Elena Razova, and Irina Fishcheva. 2018. A close look at Russian morphological parsers: which one is the best? In Artificial Intelligence and Natural Language: 6th Conference, AINL 2017, St. Petersburg, Russia, September 20–23, 2017, Revised Selected Papers 6, pages 131–142. Springer.

Y Kuratov and M Arkhipov. 2019. Adaptation of deep bidirectional multilingual transformers for Russian language. In Komp'juternaja Lingvistika i Intellektual'nye Tehnologii, pages 333–339, Moscow, Russia.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 7871–7880, Online. Association for Computational Linguistics.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation. Transactions of the Association for Computational Linguistics, 8:726–742.

Olga Lyashevskaya, Ivan Afanasev, Sergey Rebrikov, Yulia Shishkina, Elvira Suleymanova, I Trofinov, and Natalia Vlasova. 2023. Disambiguation in context in the Russian National Corpus: 20 years later. In Proceedings of International Conference "Dialogue", pages 1–12, Moscow, Russia.

Olga Lyashevskaya, Olga Mitrofanova, Maria Grachkova, Sergey Romanov, Anastasia Shimorina, and Alexandra Shurygina. 2011. Automatic word sense disambiguation and construction identification based on corpus multilevel annotation. In Text, Speech and Dialogue: 14th International Conference, TSD 2011, Pilsen, Czech Republic, September 1-5, 2011. Proceedings 14, pages 80–90. Springer.

ON Lyashevskaya. 2019. A reusable tagset for the morphologically rich language in change: A case of middle Russian. In Komp'juternaja Lingvistika i Intellektual'nye Tehnologii, pages 422–434, Moscow, Russia.

ON Lyashevskaya, TO Shavrina, IV Trofimov, NA Vlasova, et al. 2020. GramEval 2020 shared task: Russian full morphology and universal dependencies parsing. In Proceedings of the International Conference Dialogue, volume 2020, pages 553–569, Moscow, Russia.

Tobias Pütz, Daniël De Kok, Sebastian Pütz, and Erhard Hinrichs. 2018. Seq2seq or perceptrons for robust lemmatization. an empirical examination. In Proceedings of the 17th international workshop on treebanks and linguistic theories (TLT 2018), pages 193–207, Oslo University, Norway. Linköping University Electronic Press Linköping, Sweden.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D Manning. 2020. Stanza: A Python Natural Language Processing Toolkit for Many Human Languages. In Proceedings of the 58th Annual Meeting of the Association for

Computational Linguistics: System Demonstrations, pages 101–108, Online. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. Journal of machine learning research, 21(140):1–67.

Frederick Riemenschneider and Anette Frank. 2023. Exploring large language models for classical philology. In Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 15181–15199, Toronto, Canada. Association for Computational Linguistics.

Frederick Riemenschneider and Kevin Krahn. 2024. Heidelberg-boston @ SIGTYP 2024 shared task: Enhancing low-resource language analysis with character-aware hierarchical transformers. In Proceedings of the 6th Workshop on Research in Computational Linguistic Typology and Multilingual NLP, pages 131–141, St. Julian's, Malta. Association for Computational Linguistics.

Svetlana O Savchuk, Timofey Arkhangelskiy, Anastasiya A Bonch-Osmolovskaya, Ol'ga V Donina, Yuliya N Kuznetsova, Ol'ga N Lyashevskaya, Boris V Orekhov, and Mariya V Podryadchikova. 2024. Russian national corpus 2.0: New opportunities and development prospects. Voprosy Jazykoznanija, (2):7–34.

Ilya Segalovich. 2003. A fast morphological algorithm with unknown word guessing induced by a dictionary for a web search engine. Proceedings of the International Conference on Machine Learning; Models, Technologies and Applications. MLMTA'03, 2003:273.

Tatiana Shavrina and Olga Shapovalova. 2017. To the methodology of corpus construction for machine learning: "Taiga" syntax tree corpus and parser. In Proceedings of "CORPORA-2017" International Conference, pages 78–84.

A Sorokin, V Bocharov, S Alexeeva, D Granovsky, T Shavrina, O Lyashevskaya, K Droganova, and A Fenogenova. 2017. MorphoRuEval-2017: An evaluation track for the automatic morphological analysis methods for Russian. In Computational Linguistics and Intellectual Technologies, pages 297–313, Moscow, Russia.

Rachele Sprugnoli, Marco Passarotti, Flavio Massimiliano Cecchini, Margherita Fantoli, and Giovanni Moretti. 2022. Overview of the EvaLatin 2022 evaluation campaign. In Proceedings of the Second Workshop on Language Technologies for Historical and Ancient Languages, pages 183–188, Marseille, France. European Language Resources Association.

Milan Straka, Jan Hajic, and Jana Straková. 2016. UDPipe: trainable pipeline for processing CoNLL-U files performing tokenization, morphological analysis, pos tagging and parsing. In Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16), pages 4290–4297, Portorož, Slovenia. European Language Resources Association (ELRA).

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

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pages 38–45, Online. Association for Computational Linguistics.

Krzysztof Wróbel and Krzysztof Nowak. 2022. Transformer-based part-of-speech tagging and lemmatization for Latin. In Proceedings of the second workshop on language technologies for historical and ancient languages, pages 193–197, Marseille, France. European Language Resources Association.

A Zaliznyak. 1977. Grammaticheskij slovar' russkogo jazyka (Russian Grammar Dictionary). Nauka, Moscow.

Dmitry Zmitrovich, Aleksandr Abramov, Andrey Kalmykov, Vitaly Kadulin, Maria Tikhonova, Ekaterina Taktasheva, Danil Astafurov, Mark Baushenko, Artem Snegirev, Tatiana Shavrina, Sergei S. Markov, Vladislav Mikhailov, and Alena Fenogenova. 2024. A family of pretrained transformer language models for Russian. In Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024), pages 507–524, Torino, Italia. ELRA and ICCL.

A Additional Figures and Tables

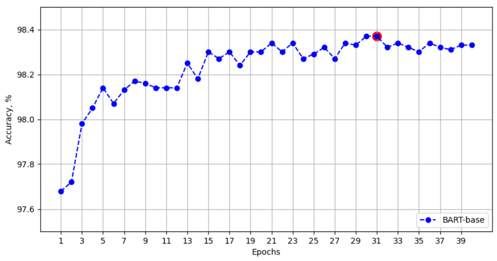

Figure 4: Accuracy scores on the development set.

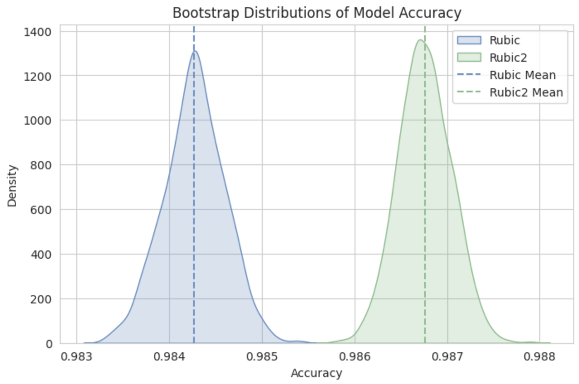

Figure 5: Bootstrap distribution of Rubic and Rubic2.

B Potential Risks

The primary intended purpose of Rubic2 is the annotation of large ($\approx 10^8 - 10^9$ tokens) corpora of Russian-language texts for further linguistic analysis. The final lemmatization results are not guaranteed to be fully accurate, as indicated by the accuracy findings presented (see Section 4). The evaluation of the model performance is limited to the domains covered in this study (see Section 3).

C Scientific Artifacts

We list the licenses of the scientific artifacts used in this paper: PyMorphy2 (MIT license), MyStem (license agreement[10]), Stanza (Apache license 2.0), Rubic (license agreement[11]), BART (Apache license 2.0), mBART-50 (MIT license), ruT5 (MIT license), Huggingface Transformers (Wolf et al., 2020) (Apache License 2.0), SimpleTransformers[12] (Apache license 2.0).

The data set used in this study was previously employed in other research on lemmatizing Russian words (Lyashevskaya et al., 2023). The pre-annotated data set was provided by the Russian National Corpus (Savchuk et al., 2024) for use exclusively for scientific purposes under a license agreement of RNC[13]. It does not contain personal data but may include a small number of examples of obscene or offensive vocabulary.

We ensured that our use of existing artifacts aligns with their intended purpose as specified by their original creators.

[10] https://yandex.ru/legal/mystem

[11] https://ruscorpora.ru/en/page/license-neuro

[12] https://simpletransformers.ai/

[13] https://ruscorpora.ru/en

Model	**Architecture**	**Params**	**Data source**	**Layers**	**Tokenizer**	**Heads**	**Hidden**
BART-base	Encoder-decoder	139M	BookCorpus, Stories, Wikipedia, CC News, OpenWebText	12	BPE, 50×10^3	16	768
BART-large		406M		24		16	1024
mBART		680M	CC25, XLMR	24		16	1024
ruT5-base		222M	Wikipedia, C4, news, Librusec, OpenSubtitles	12	Sentence-Piece, 32×10^3	12	768
ruT5-large		737M		24		16	1024

Table 6: Model overview. The Tokenizer column includes the tokenization method and the vocabulary size.

Features and values	**Comment**	**Performance impact for the RNC test set**	**Result**
Foreign = Yes	Foreign word	↑0.1355%	Lemma:=Wordform
NOUN and Case = Nom and Number = Sing	Noun in the singular form in the nominative case	↑0.0303%	
VERB VerbForm = Inf	Verb in the infinitive form	↑0.1331%	
PUNCT	Punctuation	↑0.012%	
SYM	Symbol	↑0.004%	
(NUM or ANUM) and not Wordform.isalpha()	Cardinal or ordinal numeral, and not all characters are alphabetic letters	↑0.0007%	

Table 7: Set of heuristics to post-process the output of BART-base.

Model input	**Gold lemma**	**Generated lemma (raw output)**
Qeexo X Foreign:Yes	*Qeexo*	*oeeo*
математика NOUN Animacy:Inan Case:Nom Gender:Fem Number:Sing	математика (mathematics)	матема� ика
кружиться VERB Aspect:Imp Transit:Intr VerbForm:Inf Voice:Act	кружиться (to spin)	кружить (to make spin)
.... PUNCT	....	....
*** SYM	***	%
187 NUM NumForm:Digit NumType:Card	*187*	*Top*
1:1 ANUM NumForm:Digit NumType:Card	*1:1*	*111*

Table 8: Examples of the errors corrected by heuristics.

Form	**Gold lemma**	**BART-base**	**Rubic**	**Comment**
макс.	максимум (a maximum)	МАКС.	максийскикиЙск	Abbreviation
ккал	килокалория (a kilocalorie)	ккаилограмм	ккал	
бежим	бежать (to run)	бежать	бежаться	Reflexive and non-reflexive verb forms
вспоминается	вспоминаться (to be remembered)	вспоминать	вспоминаться	
Трансваале	Трансвааль (Transvaal)	трансвааль	Трансваал	Word forms with a stem ending in a soft consonant. Proper nouns.
отеле	отель (a hotel)	отел	отель	
валенках	валенок (a felt boot)	валенки	валенок	Word forms in plural
цветы	цветок (a flower)	цветок	цветы	
когтями	коготь (a claw)	когт	коготь	Words with fleeting vowels
паренька	паренек (a lad)	пареньень	паренько	
хворые	хворый (sick)	хворый	хворой	Adjectives ending in -ой and -ый
роковые	роковой (fatal)	роковый	роковой	

Table 9: Examples of lemmatization errors. Generated lemmas that do not match the gold lemma are underlined (case insensitive).

Model	**Speed (tokens per second)**
BART-base	707
Rubic	1416

Table 10: Average lemmatization speed. The model results were obtained using an NVIDIA RTX 4090 GPU and an AMD Ryzen 9 7900X processor.