RuMedSpellchecker: correcting spelling errors for natural Russian language in electronic health records using machine learning techniques

Dmitrii Pogrebnoi[0000−0001−6392−8445], Anastasia Funkner[0000−0002−4596−6293], and Sergey Kovalchuk[0000−0001−8828−4615]

ITMO University, Saint Petersburg, Russia
pogrebnoy.inc@gmail.com, funkner.anastasia@gmail.com, sergey.v.kovalchuk@gmail.com

Abstract. The incredible advances in machine learning have created a variety of predictive and decision-making medical models that greatly improve the efficacy of treatment and improve the quality of care. In healthcare, such models are often based on electronic health records (EHRs). The quality of this models depends on the quality of the EHRs, which are usually presented as plain unstructured text. Such records often contain spelling errors, which reduce the quality of intelligent systems based on them. In this paper we present a method and tool for correcting spelling errors in medical texts in Russian. By combining the Symmetrical Deletion algorithm and a finely tuned BERT model to correct spelling errors, the tool can improve the quality of original medical texts without significant cost. We have evaluated the correction precision and performance of the presented tool and compared it with other popular spelling error correction tools that support Russian language. Experiments have shown that the presented approach and tool are 7% superior to existing open-source tools for automatically correcting spelling errors in Russian medical texts. The proposed tool and its source code are available on GitHub[1] and pip[2] repositories.

Keywords: spell checking · text correction · BERT · transformers · Russian natural language · electronic health records

1 Introduction

The integration of machine learning techniques in the field of healthcare has revolutionized the way treatments are administered and care is provided to patients. Predictive and decision-making models based on electronic health records help to take into account patient specifics more accurately and better adjust treatment for each patient. One important problem with EHRs, is that they are

[1] https://github.com/DmitryPogrebnoy/MedSpellChecker
[2] https://pypi.org/project/medspellchecker

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36024-4_16

usually presented as unstructured text and contain spelling errors. According to a study by Toutanova et al. [17], spelling errors occur mainly for two reasons. The first reason is mainly related to the person himself and is that the writer may not know exactly how to spell a word correctly and therefore make mistakes. The second reason has to do with technology and is due to poor-quality typing devices, which can also lead to spelling errors. Such mistakes in EHRs can confuse machine-learning-based medical systems and reduce their quality and effectiveness of recommendations. Therefore to overcome this problem, we need a spelling error correction tool that will accurately correct errors and can improve model accuracy without additional cost.

There are many different open source tools for correcting spelling errors. However, most of them support English and a few other popular languages, while the number of such tools is much smaller for other, less common languages. In addition, medical texts differ significantly from general texts. They contain many medical terms that are almost never found in ordinary texts and require special processing. Correction of a valid word can change a sentence and lead to unpredictable results. Therefore special tools for medical texts aimed at correcting spelling errors in such texts are essential.

There are no known special open source tools for correcting Russian medical texts. And this is to be expected, since there are very few open datasets with medical texts in Russian. Nevertheless, there are several open source general purpose tools for correcting spelling errors in Russian texts.

In this paper, we present a method and a tool for correcting spelling errors specifically in Russian medical texts. The new tool is based on a combination of the Symmetric Deletion [6] algorithm for generating edit candidates and a finely tuned BERT-based [5] machine learning model for ranking candidates and selecting the most appropriate.

2 Related Works

There are a number of research papers devoted to different approaches to the processing of medical texts. However, most of them are mainly devoted to texts in English.

The study by Yifan Peng et al. [12] found that fine-tuning the BERT model on medical notes and PubMed articles outperformed most existing state-of-the-art models and demonstrating the effectiveness of fine-tuning language models for specific domains. The study by Jinhyuk Lee et al. [8] presents a BioBERT model that is fine-tuned to a biomedical corpus in English. Results of experiments have shown that the derived model outperforms the basic BERT model in various text mining tasks of medical texts. In another related paper by Emily Alsentzer et al. [2], the authors presents the ClinicalBERT model for English language. This model is finely tuned on clinical data and outperforms the basic BERT and BioBERT models in almost any type of task.

The processing of medical texts in Russian is much less well researched. This is mainly because there is not yet a significant amount of medical data collected

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36024-4_16

for the Russian language that is comparable with the existing Russian-language medical data corpus. In spite of this, corpora are gradually forming and this field of processing Russian medical texts is actively developing.

The paper by Alexander Yalunin et al. [18] in Sberbank AI lab is one such study. The authors of the paper ideationally replicated Jinhyuk Lee et al. work with the BioBERT model. They fine-tuned several different BERT models on an open corpus of medical texts in Russian and compared the metrics of the new models on specific domain tasks. The evaluation results showed that the fine-tuned BERT models also outperformed the basic BERT models even though the fine-tuning corpus was smaller than in the BioBERT case.

The study by Ksenia Balabaeva et al. [3] is devoted specifically to correcting spelling errors in Russian medical texts. This work uses a combination of Damerau-Levenstein [4] distance to generate editing candidates and Word2vec [9] or FastText [7] embeddings to rank and select the best candidate. The evaluation showed quite strong results, but there is still room for improvement.

There have been impressive advances in the application of BERT models to various tasks in medical text processing. In this paper we present a new method and its implementation as a tool for correcting spelling errors in medical texts in Russian. We also compared the precision metrics and performance of the developed tool with other popular open source tools for spelling error correction in Russian.

3 Methods

3.1 Types of spelling errors

There are many different views on what errors should be corrected by a text correction tool. Alexey Sorokin et al. [15] in their study presented the results of the first competition on automatic text correction in Russian. Seven teams participated in the contest. Their task was to create tools that had to most accurately correct texts that contained predefined kinds of errors. In addition to syntactic errors, the texts contained grammatical and cognitive errors, as well as some other kinds of errors.

Medical texts are extremely sensitive to errors. An incorrectly corrected word, or worse, a corrected valid word, can greatly affect the meaning of a sentence and lead to unpredictable consequences. The more kinds of errors the tool tries to correct, the more cases when the tool can work false positive and change a valid word. To start gradually and reduce at first the number of false positive cases we consider only the kind of spelling errors. Developed spellchecker supports correction of six types of spelling errors. Four types of errors are related only to letters and two more types of errors related to spaces. Examples of supported spelling error types are shown in Fig. 1.

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36024-4_16

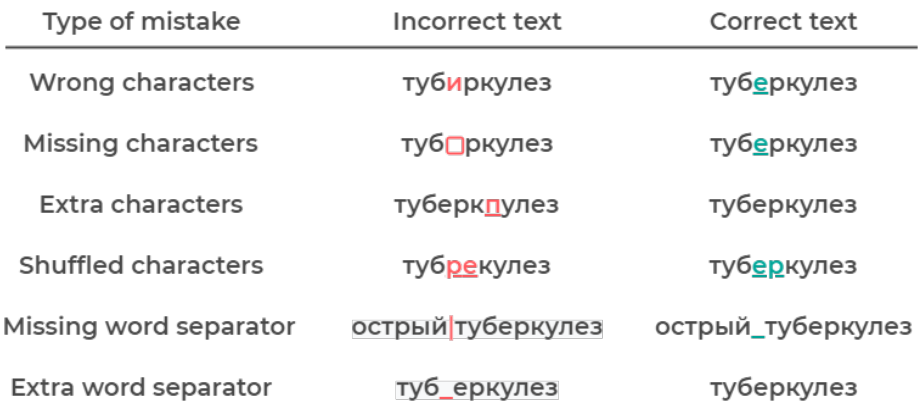

Fig. 1. Supported types of spelling errors by the spellchecking tool. As an example, the Russian words «tuberculosis» and «acute tuberculosis».

3.2 Datasets

Developing an accurate and efficient tool for correcting spelling errors in Russian-language medical texts is impossible without collecting enough data to train machine learning models and testing the tool.

The availability of open sources of Russian medical texts and data sets is extremely limited. This is mainly due to the complexity of collecting and processing sensitive data, since they can potentially contain personal data. In this paper, we use two open medical cases in Russian, as well as two closed datasets.

The summary of the datasets used is shown in Table 1.

Table 1. Summary information on the Russian medical datasets used.

Dataset Name	Number of Records	Avg Tokens in Record
RuMedNLI [11]	14717	8
RuMedPrimeData [16]	15250	31
Almazov Center	2355	42
Russian Academy of Sciences [14]	161	863

The first public datasets is RuMedNLI: A Russian Natural Language Inference Dataset For The Clinical Domain [11] by Pavel Blinov et al. This dataset is a manually translated from English MedNLI [13] dataset and contains 14717 medical records. Another public dataset is RuMedPrimeData [16] by Starovoytova Elena et al. This dataset contains 15250 medical anamneses of SSMU hospital visitors.

One of the private datasets is dataset with patients' medical anamneses which is provided by the Institute of Artificial Intelligence Problems of the Research Institute of the Russian Academy of Sciences (Russian Academy of Sciences) [14].

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36024-4_16

This dataset contains 161 large fragments of patients' anamneses. In addition, a closed corpus of anamneses provided by the Almazov National Medical Research Center (Almazov Center) was also used. This dataset contains 2355 patient anamneses for the period from 2010 to 2015.

Each of the datasets was pre-processed. All texts were converted to lower case and then lemmatised using the pymorphy2[3] tool. Such harsh pre-processing was done because of the extremely limited datasets. The lemmatisation helped to get rid of words in different forms and to use words in their initial form. This increased the number of example sentences for a particular word, which contributed to a better fine-tuning of the language models.

After preprocessing, all four datasets were combined into one. In total, all datasets together contain 30,737 medical records in Russian, which takes about 10.25 Mb.

3.3 Algorithm

The text correction algorithm takes raw, unprocessed text as input and returns a corrected text. The algorithm uses the Damerau-Levenstein [4] edit distance to generate candidates for correcting an invalid word. This algorithm generates candidates with an edit distance of one by default, but can potentially be extended to greater values. A special pre-calculated Symmetric Deletion index is used to optimize the computation of the edit distance and to speed up the generation of edit candidates. The diagram of the spellchecking process is shown in the Figure 2.

The process is arranged as follows. First of all, the medical text is divided into tokens. Next, a couple of conditions are checked. Whether there are any non-Russian letters in the token and whether the word is a name. To check if a word is a name, we check if the word does not contain any capital letters, or if it is at the beginning of a sentence. If at least one condition is not fulfilled, the token is considered as not valid for correction and gets into the final result as it is. Otherwise, the token is reduced to lowercase letters, and the lemmatized form of the token and information about the form of the original word is retained in the internal representation. After that, it is checked whether the token or its lemmatised form is included in the dictionary of correct words. If it is included, then such a token gets into the final text as it is. Otherwise, a list of candidates is generated to replace the incorrect word. Then this list is ranked by a special language model and the most suitable candidate gets into the final text. The best candidate is transformed into the required form and case of the original word. This happens with every token. At the end, the corrected tokens are assembled into the final text.

The precision of the corrected algorithm depends mainly on the completeness and purity of the dictionary with correct words, and on how well the language model correctly ranks the edit candidates.

[3] `https://github.com/pymorphy2/pymorphy2`

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36024-4_16

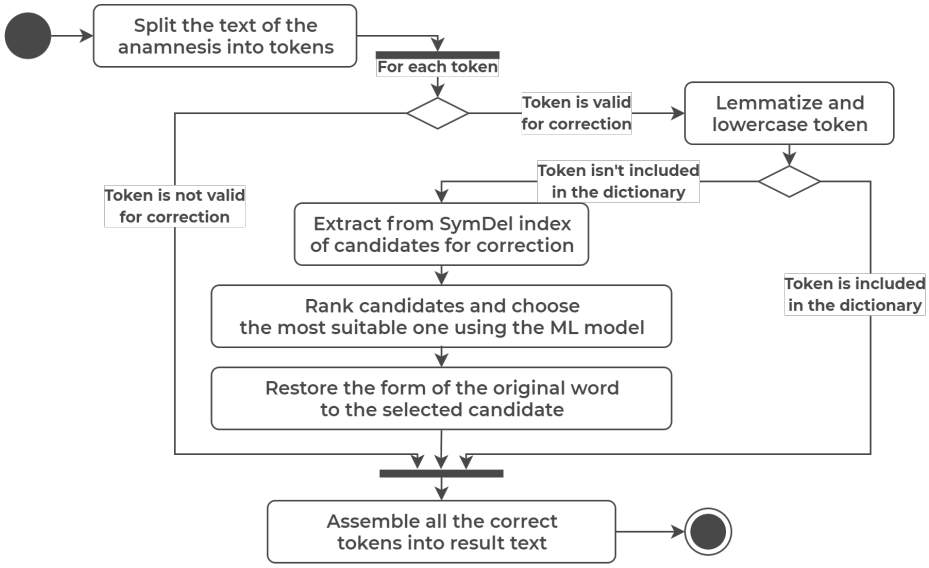

Fig. 2. Examples of supported types of spelling errors by the spellchecking tool.

3.4 Implementation

The new tool for correcting the spelling of medical texts is intended to work only with Russian text and is written in Python. The tool consists of seven components. The architecture of the tool is shown in the Figure 3.

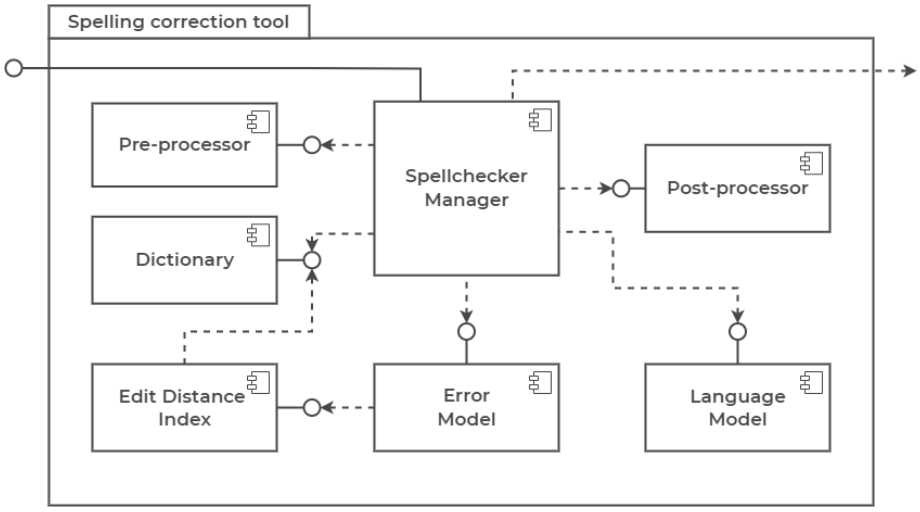

Fig. 3. Architecture of the new spelling correction tool.

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36024-4_16

The Pre-processor component is responsible for splitting the text into separate words, removing punctuation and capitalisation, and determining the lemmatised form and form information of the original word. In contrast, the Post-processor component restores the form of the original words to the corrected words and reassembles the entire text from the individual words.

The Dictionary component contains a dictionary of correct words and is used to quickly determine whether a word is correct or requires correction. This dictionary contains only lemmatised words from the Aspell Russian dictionary and several medical dictionaries. The final dictionary contains 214629 words in the primary form.

Error Model component is responsible for generating a list of candidates for correcting an incorrect word. The error model uses the Damerau-Levenstein edit distance to create a list of candidates. Generation of the candidate list is a computationally intensive operation. The Edit Distance Index component, which contains a special index, is used to significantly speed up the operation and improve the overall performance of the tool.

Language Model is responsible for ranking the editing candidates and choosing the most suitable one to replace the incorrect word. A finely tuned machine learning models based on the BERT architecture is used to rank candidates for Russian medical texts. The main advantage of this approach is that the model is able to take into account the context around the incorrect word when ranking candidates, which improves the quality of ranking and accuracy of correction.

3.5 Language Model

In this paper, three different BERT base models were fine-tuned on a collected dataset of Russian medical texts. Fine-tuning of the models took place using the transformers, datasets and accelerate libraries from the Hugging Faces platform. A common Fill Mask task was used to fine-tune the models. All hyperparameters for fine-tuning all models are the same and are shown in Table 2

Table 2. Hyperparameters of fine-tuning of all models.

Parameter	Value
Train epoch	5
Learning rate	0.00005
Weight decay	0.01
FP16 training	True
Gradient accumulation steps	256
Per device train batch size	1
Per device eval batch size	1
Gradient checkpointing	True

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36024-4_16

The first base model is pre-trained ruRoBERTa-large[4] model from Sberbank AI. This model was chosen because it is one of the larger and more efficient models of the BERT architecture. This model is publicly available and pre-trained for the Russian language, which makes it easy to use for various applications in Russian. The fine-tuned model is published on Hugging Face as DmitryPogrebnoy/MedRuRobertaLarge[5].

The second basic model was the distilbert-base-multilingual-cased model[6]. This model is slightly inferior in efficiency to the previous model, but has a much smaller size and better performance. This model supports multiple languages including Russian. For the task of ranking editing candidates in Russian, all other languages are superfluous. To reduce the size of the model, support for unnecessary languages was discarded using the approach described in the paper by Amine A. et al. [1]. But unlike the approach in the paper, all tokens not containing Russian letters except for special characters, numbers and punctuation marks were discarded. This halves the size of the model. The converted model is published on the Hugging Face service as DmitryPogrebnoy/distilbert-base-russian-cased[7]. The fine-tuned model is published on Hugging Face as DmitryPogrebnoy/MedDistilBertBaseRuCased[8].

The third basic model was the cointegrated/rubert-tiny2[9] model. This model only supports the Russian language and is based on the BERT architecture. The main feature that distinguishes this model from the previous two is that it is much smaller in size. It is half the size of the MedDistilBertBaseRuCased model. The fine-tuned model is published on Hugging Face as DmitryPogrebnoy/MedRuBertTiny2[10].

As a result, the Language Model component contains three different BERT-based models fine-tuned to rank a edit candidate. With several models of different sizes, it is possible to use this spellchecker tool on a variety of technical hardware, from a high-performance server with the MedRuRobertaLarge model to an common personal computer with the MedRuBertTiny2 model.

The Language Model component can easily be extended by adding candidate rankers based on other machine learning models. It is enough to implement the necessary interface and everything will work out of the box. As an example, we have added rankers based on the RuBioBERT and RuBioRoBERTa models from the paper by Alexander Yalunin et al. [18].

3.6 Python package

The developed spellchecker was assembled in a Python package and uploaded to the official pip repository. The package with the new tool is called med-

[4] https://huggingface.co/sberbank-ai/ruRoberta-large
[5] https://huggingface.co/DmitryPogrebnoy/MedRuRobertaLarge
[6] https://huggingface.co/distilbert-base-multilingual-cased
[7] https://huggingface.co/DmitryPogrebnoy/distilbert-base-russian-cased
[8] https://huggingface.co/DmitryPogrebnoy/MedDistilBertBaseRuCased
[9] https://huggingface.co/cointegrated/rubert-tiny2
[10] https://huggingface.co/DmitryPogrebnoy/MedRuBertTiny2

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36024-4_16

Listing 1.1. An example of using a new tool to correct an error in the sentence "The patient has been diagnosed with a heart attack"

```
1  from medspellchecker.tool.medspellchecker \
2      import MedSpellchecker
3  from medspellchecker.tool.distilbert_candidate_ranker \
4      import RuDistilBertCandidateRanker
5
6  candidate_ranker = RuDistilBertCandidateRanker()
7  spellchecker = MedSpellchecker(candidate_ranker)
8  fixed_text = spellchecker.fix_text(
9    "У больного диагностирован инфркт"
10 )
11
12 print(fixed_text)
```

spellchecker[11]. In addition to the source code and necessary classes, this package also contains a dictionary of correct words. Apart from the dictionary, the package does not contain any other additional datasets or models, which keeps the size of the package manageable.

The medspellchecker package does not contain any parts or compiled fragments of ranking models. They are all downloaded automatically when needed from the public repository on Hugging Faces. An internet connection is required to use the tool, at least for the first time. Once the model has been downloaded and cached, an internet connection is not required.

An example of the use of a package with the developed tool is shown in Listing 1.1.

The first two lines import the main class of the package. Lines 3 and 4 then import one of the three available classes to rank the edit candidates. Line 6 creates an instance of the candidate ranking class based on the fine-tuned DistilBert model. Line 7 creates an instance of the class to correct spelling errors. On line 8, the *fix_text* method is called, which takes the raw text and returns the corrected text. Finally line 12 prints the corrected result, which looks like «У больного диагностирован инфаркт». In this way, the package can be used to correct spelling errors in Russian medical texts in a few lines. The package also allows you to extend the ranking classes and add your own custom ones.

4 Experiments

There are several open source tools that can automatically correct spelling errors in words. Most of these tools focus mainly on English text, while the other languages fall by the wayside and cannot boast such a rich range of tools. Nevertheless, there are several tools that support the Russian language. However, none of them is intended for medical texts.

[11] `https://pypi.org/project/medspellchecker`

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36024-4_16

We have evaluated the precision and performance of the tool, and compared it to other open source tools for correcting spelling errors in Russian texts.

The following parameters were identified for comparison.

- Error precision. This is the percentage of words with an error that the tool correctly corrected relative to all incorrect words.
- Lexical precision. This is the percentage of correct words without mistakes that the tool does not correct relative to all correct words.
- Overall precision. This is the average of error precision and lexical precision.
- Performance. Average number of words processed per second.

Precision metrics and tool performance were evaluated on two different datasets. The first dataset contains correct and incorrect words without their context. This set contains 100 entries for each of the first four error types, 900 entries for the fifth error type and 1000 for the sixth error type shown in Figure 1. Thus this test dataset contains a total of 2300 records. Each record contains a pair of words. The first word contains a particular type of error and the second word is its correct version.

The second dataset also contains 100 entries for each of the first four error types, 900 entries for the fifth error type and 1000 for the sixth error type shown in Figure 1. Thus this test dataset contains a total of 2300 records. This set does not contain single words, but words with context. Each record consists of three parts. The first part is a coherent passage of 10 words, one of which is written with an error of a certain type. The second part is the same 10 words, but fully correct. The third part is the number of the incorrect word, which is used to calculate the error and lexical precision.

All words and passages in the test datasets are collected from various Russian medical texts and anamneses.

Thus we have two test datasets with medical data. The first dataset allows us to evaluate how good the tool performs in correcting single words, and the second dataset allows us to calculate the quality of word correction in a coherent context.

In order to better compare the tools was also calculated overall precision, which is the average of lexical and error precision. This metric allows estimating the quality of spelling error correction in general. However, it should be noted that there are usually more correct words in a text than there are incorrect words, therefore lexical precision is more important than error precision.

In addition to the quality of correction of spelling errors, it is also necessary to take into account the time in which these errors are corrected. Of course, the faster the tool works, the better. The performance test was conducted on a computer on Ubuntu 20.04 with 24 GB RAM, Intel Core i5-10210U CPU @ 1.60GHz * 8 and NVIDIA Tesla V100.

5 Results and Discussion

Seven open source tools for correcting spelling errors in Russian texts were chosen for the experiments. For each tool, there are Python wrappers for easy handling

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36024-4_16

from Python code. The selected tools and their Python wrappers are shown in the Table 3.

Table 3. Open source tools and their Python wrappers for correcting spelling errors in Russian chosen for experiments. Full links to Github repositories require «https://www.github.com/» prefix.

Tool name	Wrapper name	Wrapper GitHub repository
Aspell	Aspell-python	WojciechMula/aspell-python
Hunspell	PyHunspell	blatinier/pyhunspell
Enchant	PyEnchant	pyenchant/pyenchant
LanguageTool	LanguageTool-python	jxmorris12/language_tool_python
Peter Norvig's spellchecker [10]	PySpellChecker	barrust/pyspellchecker
Symspell	SymspellPy	mammothb/symspellpy
Jumspell	Jumspell	bakwc/JamSpell

In addition to the existing tools, the experiment was also conducted with RuMedSpellchecker in ten different configurations, depending on the language model and the type of processor used. The first six configurations used models fine-tuned as part of this paper. The remaining four configurations used the RuBioBERT and RuBioRoBERTa models, which were obtained in the paper by Alexander Yalunin et al. [18] in Sberbank AI lab.

An example of correcting spelling errors with selected tools is shown in Table 4. The misspelled sentence is «У больногодиагностирован инфркт и туберкулз». The correct result of the misspelled sentence is «у больного диагностирован инфаркт и туберкулез». The case of letters is not taken into account. In this example, RuMedSpellchecker tool was run in CPU mode, as the quality and result of the fix is independent of which computing mode is used.

As you can see even such a relatively simple example causes problems with the correction. However, the Aspell-python and LanguageTool-python tools properly corrected the example sentence. The other existing tools made mistakes in endings, prepositions, or missing a space between words. The new tool corrected the example preposition with only two of the language models. The other three had incorrect results.

The results of the example sentence corrections cannot be used to evaluate the precision of the corrections and the performance of the tools. The following are the results of the single word corrections experiment and the word with context corrections experiment.

The result of the experiment with single word corrections is shown in Table 5. In addition to the tool names and four metric columns, the table contains a column with the CPU or GPU type of processor. GPUs can significantly improve the performance of spelling correction tools, but only those that support it. Unfortunately, none of the existing evaluated tools support GPU computing.

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36024-4_16

Table 4. The results of fixing the «У больногодиагностирован инфркт и туберкулз» sentence by the various tools selected.

Tool name	Result
Aspell-python	**У больного диагностирован инфаркт и туберкулез**
PyHunspell	Уф больного диагностирован инфаркт аи туберкул
PyEnchant	И больного диагностирован инфаркт аи туберкул
LanguageTool-python	**У больного диагностирован инфаркт и туберкулез**
PySpellChecker	У больногодиагностирован инфаркт и туберкулез
SymspellPy	и больногодиагностирован инфаркт и туберкулез
SymspellPy (compound mode)	у больного диагностирования инфаркт и туберкулез
Jumspell	У больногодиагностирован инфркт и туберкулз
RuMedSpellchecker (MedRuRobertaLarge)	У больного диагностирован инфект и туберкулз
RuMedSpellchecker (MedDistilBertBase)	**У больного диагностирован инфаркт и туберкулез**
RuMedSpellchecker (MedRuBertTiny2)	**У больного диагностирован инфаркт и туберкулез**
RuMedSpellchecker (RuBioBERT)	У больного диагностирован инфркт и туберкулз
RuMedSpellchecker (RuBioRoBERTa)	У больного диагностирован инфект и туберкулз

Table 5. Comparison of spelling correction tools in Russian medical texts and a new tool in the single word correction test.

Tool name	Processor type	Error precision	Lexical precision	Overall precision	Average words per second
Aspell-python	CPU	**0.86**	0.859	**0.859**	283.7
PyHunspell	CPU	0.812	0.539	0.675	9.4
PyEnchant	CPU	0.829	0.541	0.685	20
LanguageTool-python	CPU	0.762	0.904	0.833	25.1
PySpellChecker	CPU	0.354	0.86	0.607	3.4
SymspellPy	CPU	0.399	0.813	0.606	**9702.8**
SymspellPy (compound mode)	CPU	0.465	0.512	0.489	672
Jumspell	CPU	0.267	0.947	0.607	2552.1
RuMedSpellchecker (MedRuRobertaLarge)	CPU	0.715	**0.991**	0.853	2.1
	GPU				5.9
RuMedSpellchecker (MedDistilBertBaseRuCased)	CPU	0.701	**0.991**	0.846	12.7
	GPU				39.7
RuMedSpellchecker (MedRuBertTiny2)	CPU	0.681	**0.991**	0.836	24.2
	GPU				79.1
RuMedSpellchecker (RuBioRoBERTa)	CPU	0.695	**0.991**	0.843	2.2
	GPU				5.8
RuMedSpellchecker (RuBioBERT)	CPU	0.683	**0.991**	0.837	8.3
	GPU				20.1

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36024-4_16

The Aspell-python tool performed best in the single-word correction test in terms of error precision and overall precision. The new tool, on the other hand, was quite average in error precision. However, the tool was the best in lexical precision, which also influenced the overall precision and made it close to the best. This is probably due to the additionally extended vocabulary of valid words. In terms of performance, SymSpellPy showed the best result, while Jumspell also showed a high value. As expected, the performance of the new tool depends on the model and processor type used. The smaller the model, the faster the tool runs, but the precision is slightly reduced. Also the tool with our fine-tuned models shows slightly better precision metrics than with RuBioBERT and RuBioRoBERTa respectively. Overall, it can be said that the performance of the tool is average compared to competitors.

The results of the precision metrics of the new tool in the one-word correction test were not as high. However, the main feature of the language models used in the tool is to take into account the context around the word being corrected. Therefore, the new tool revealed itself in the test with word correction in context. The result of the experiment with word correction in context is shown in Table 6.

Table 6. Comparison of spelling correction tools in Russian medical texts and a new tool in the test of correcting a words with context.

Tool name	Processor type	Error precision	Lexical precision	Overall precision	Average words per second
Aspell-python	CPU	0.731	0.93	0.831	357.3
PyHunspell	CPU	0.706	0.719	0.713	11.8
PyEnchant	CPU	0.721	0.719	0.72	24.3
LanguageTool-python	CPU	0.727	0.942	0.835	43.6
PySpellChecker	CPU	0.304	0.868	0.586	6.7
SymspellPy	CPU	0.37	0.913	0.642	**26060.2**
SymspellPy (compound mode)	CPU	0.483	0.804	0.643	1604.2
Jumspell	CPU	0.307	0.969	0.638	4322.3
RuMedSpellchecker (MedRuRobertaLarge)	CPU	**0.792**	0.984	**0.888**	8.9
	GPU				29.1
RuMedSpellchecker (MedDistilBertBaseRuCased)	CPU	0.765	**0.99**	0.878	45.5
	GPU				153.8
RuMedSpellchecker (MedRuBertTiny2)	CPU	0.742	0.987	0.865	127.3
	GPU				356.2
RuMedSpellchecker (RuBioRoBERTa)	CPU	0.738	0.987	0.863	9.1
	GPU				31.3
RuMedSpellchecker (RuBioBERT)	CPU	0.715	0.988	0.852	30.7
	GPU				95.6

RuMedSpellchecker outperformed the competition in all precision metrics in the test of correcting words with context. In addition, in this test, the new

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36024-4_16

tool showed significantly better performance than in the previous test. However, SymSpellPy was still the performance leader.

The new tool with the largest language model, MedRuRobertaLarge, scored higher in the error precision metric than the other models. However, lexical precision was higher with the MedDistilBertBaseRuCased model. This can probably be explained by the very limited dataset for fine-tuning large models. Also, due to the limited dataset, the generalization ability of the models may be biased. For example, for other types of medical texts the tool may perform worse. Despite this it is very likely that with more medical texts to train, the fine-tuning of models will be better and the precision metrics of the tool will also be even higher.

Nevertheless, the results show that the new approach for correcting medical texts in Russian is effective and outperforms the competition by 7% in overall precision. Moreover, this approach can be used not only to correct medical texts. This algorithm can be adapted to other specific domains and languages with limited available datasets.

6 Conclusion

We presented a new method for correcting medical texts in Russian and a tool that implements it. Experiments have shown that the new tool is slightly inferior to existing tools when correcting single words, but outperforms them by 7% in overall precision when correcting words with context. The achieved result can be improved by more medical data in Russian for better fine-tuning of language models.

The presented method can be used not only for correction of medical texts in Russian. The algorithm can be adapted to correct spelling errors in texts in other low-resource languages. In addition, the presented approach can also be applied not only to medical texts, but also to texts of other specific domains with a limited set of available data.

Acknowledgments

This work was supported by the Ministry of Science and Higher Education of Russian Federation, goszadanie no. 2019-1339.

References

1. Abdaoui, A., Pradel, C., Sigel, G.: Load what you need: Smaller versions of mutililingual BERT. In: Proceedings of SustaiNLP: Workshop on Simple and Efficient Natural Language Processing. pp. 119–123. Association for Computational Linguistics, Online (Nov 2020). https://doi.org/10.18653/v1/2020.sustainlp-1.16
2. Alsentzer, E., Murphy, J., Boag, W., Weng, W.H., Jindi, D., Naumann, T., McDermott, M.: Publicly available clinical BERT embeddings. In: Proceedings

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36024-4_16

of the 2nd Clinical Natural Language Processing Workshop. pp. 72–78. Association for Computational Linguistics, Minneapolis, Minnesota, USA (Jun 2019). https://doi.org/10.18653/v1/W19-1909
3. Balabaeva, K., Funkner, A., Kovalchuk, S.: Automated spelling correction for clinical text mining in russian. Studies in health technology and informatics **270**, 43–47 (06 2020). https://doi.org/10.3233/SHTI200119
4. Damerau, F.J.: A technique for computer detection and correction of spelling errors. Commun. ACM **7**(3), 171–176 (mar 1964). https://doi.org/10.1145/363958.363994
5. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding (2018), `https://arxiv.org/abs/1810.04805`
6. Github repository of symspell tool (2018), `https://github.com/wolfgarbe/SymSpell`
7. Joulin, A., Grave, E., Bojanowski, P., Mikolov, T.: Bag of tricks for efficient text classification (2016), `https://arxiv.org/abs/1607.01759`
8. Lee, J., Yoon, W., Kim, S., Kim, D., Kim, S., So, C.H., Kang, J.: BioBERT: a pre-trained biomedical language representation model for biomedical text mining. Bioinformatics **36**(4), 1234–1240 (sep 2019). https://doi.org/10.1093/bioinformatics/btz682
9. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space (2013), `https://arxiv.org/abs/1301.3781`
10. Norvig, P.: Peter norvig's blog post about a simple spell checking algorithm (2007), `https://norvig.com/spell-correct.html`
11. Pavel, B., Aleksandr, N., Galina, Z., Arina, R., Vladimir, K., Chaitanya, S.: Rumednli: A russian natural language inference dataset for the clinical domain (2022). https://doi.org/http://doi.org/10.13026/gxzd-cf80
12. Peng, Y., Yan, S., Lu, Z.: Transfer learning in biomedical natural language processing: An evaluation of bert and elmo on ten benchmarking datasets (2019), `https://arxiv.org/abs/1906.05474`
13. Romanov, A., Shivade, C.: Lessons from natural language inference in the clinical domain (2018), `https://arxiv.org/abs/1808.06752`
14. Shelmanov, A.O., Smirnov, I.V., Vishneva, E.A.: Information extraction from clinical texts in russian. Computational Linguistics and Intellectual Technologies: Papers from the Annual International Conference "Dialogue" **270**, 537–549 (2015)
15. Sorokin, A., Baytin, A., Galinskaya, I., Rykunova, E., Shavrina, T.: Spellrueval : the first competition on automatic spelling correction for russian (2016), `https://www.dialog-21.ru/media/3427/sorokinaaetal.pdf`
16. Starovoitova, E., Kulakov, E., Fedosenko, S., Shmyrina, A., Kirillova, N., Vinokurova, D., Balaganskaya, M.: RuMedPrimeData (2021). https://doi.org/10.5281/zenodo.5765873
17. Toutanova, K., Moore, R.C.: Pronunciation modeling for improved spelling correction. In: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics. p. 144–151. ACL '02, Association for Computational Linguistics, USA (2002). https://doi.org/10.3115/1073083.1073109
18. Yalunin, A., Nesterov, A., Umerenkov, D.: Rubioroberta: a pre-trained biomedical language model for russian language biomedical text mining (2022), `https://arxiv.org/abs/2204.03951`

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36024-4_16