Machine learning technology for correcting electronic medical texts in Russian

Student: Dmitry Pogrebnoy, J42332c

Supervisor: Sergey Kovalchuk, PhD

Electronic medical records

- Many medical models based on patients' medical records
- Quality of models depends mainly on the quality of source texts
- Patient data is a plain text with many spelling errors
- Spelling errors greatly reduce the quality of the final models
- Fixing such errors will improve the quality of the medical models

Goal and tasks

Goal: Design a method and implement an automatic spelling correction tool for medical texts in Russian.

Tasks:

- Perform an overview of the Russian medical texts correction.
- Analyze existing solutions for correcting Russian texts.
- Design a new method for correcting spelling errors.
- Design the architecture and implement a new spelling correction tool.
- Conduct approbation of the developed tool.
- Compare results of the developed tool and existing ones.

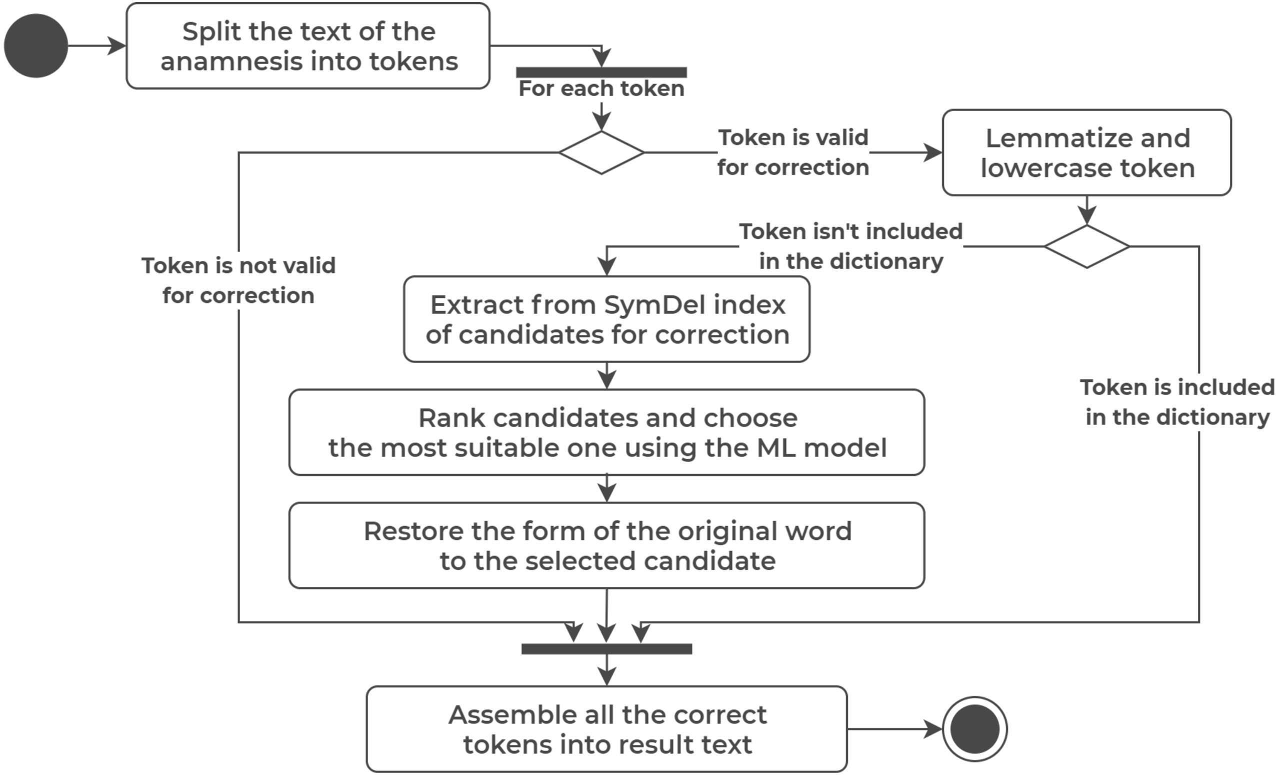
Spelling errors

Type of mistake	Incorrect text	Correct text
Wrong characters	тубиркулез	туберкулез
Missing characters	тубркулез	туберкулез
Extra characters	туберкплулез	туберкулез
Shuffled characters	тубрекулез	туберкулез
Missing word separator	острыйтуберкулез	острый_туберкулез
Extra word separator	туб_еркулез	туберкулез

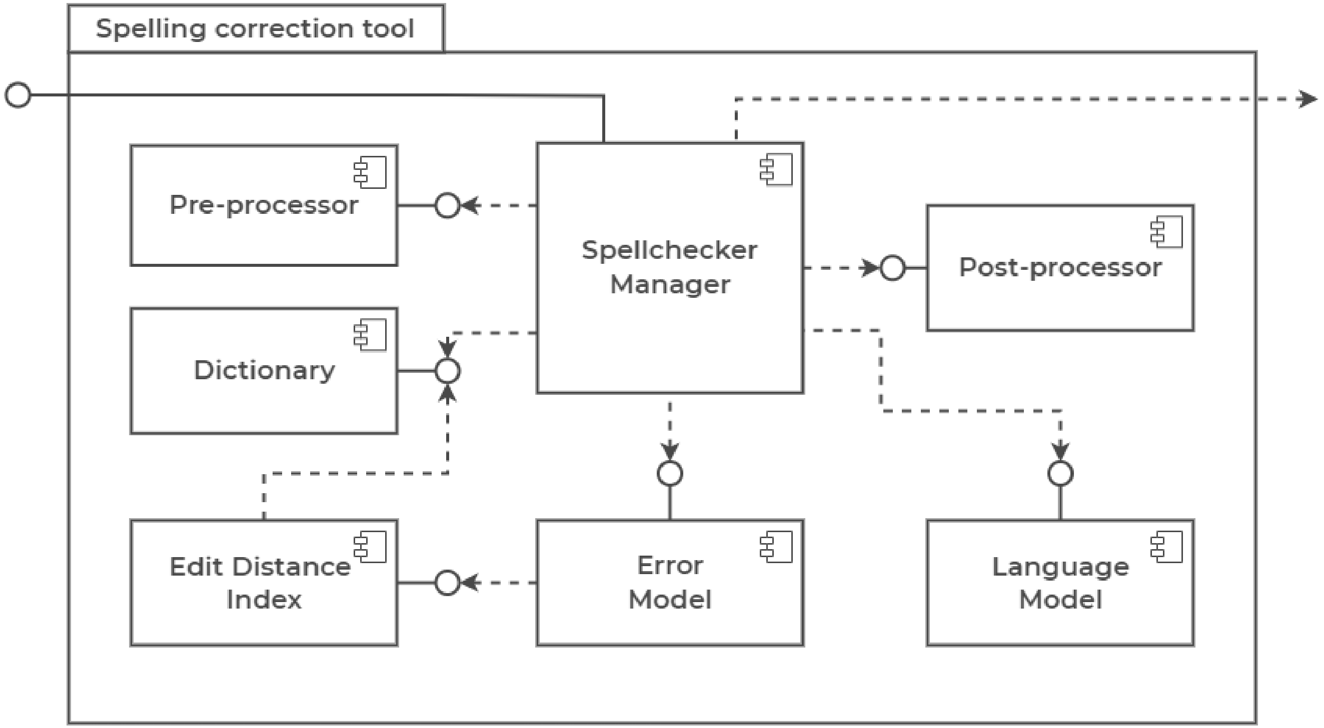
Existing tools

- There are several Russian open source tools
  - Aspell
  - Hunspell
  - Enchant
  - LanguageTool
  - Symspell
  - Jumspell
- Not one is intended for medical texts
- Not one uses advanced language models

Spelling correction process



Tool architecture



Anamneses dataset

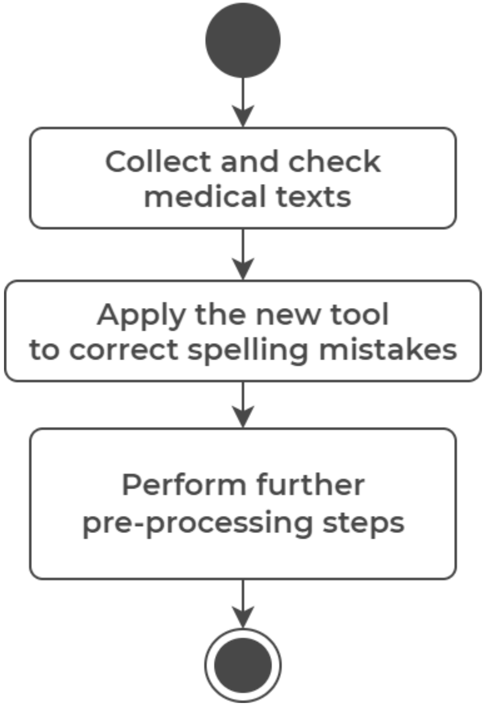
- Public datasets
  - RuMedNLI – 14716 records
  - RuMedPrimeData – 15249 records
- Private datasets
  - Almazov National Medical Research Center – 2355 records
  - Research Institute of the Russian Academy of Sciences – 161 records
- All datasets were pre-processed and assembled into final one
  - Tokenization and lemmatization
  - Stop words filtering

Fine-tune BERT models

- sberbank-ai/ruRoberta-large → MedRuRobertaLarge
  - Size – 1.4 Gb
- distilbert-base-multilang-cased → MedDistilBertBaseRuCased
  - Converted from multilang to Russian model
  - Size - 217 Mb
- cointegrated/rubert-tiny2 → MedRuBertTiny2
  - Size – 117 Mb
- All models are published on the Hugging Face repository
- RuBioBERT and RuBioBERTa were adapted for the tool

Method of tool use

- Only for correction of medical texts in Russian
- Preferably use for medical anamneses
- Use in the preprocessing pipelines
- Use before any other preprocessing steps
- Make sure everything is okay afterwards



Example of correction

Tool	Corrected Result
Original	тревожное расстройство (золофт) и атопичекий дерматит
Aspell-python	тревожное расстройство золота и тапочкой дерматит
PyHunspell	тревожное расстройство золото аи топический дерматит
PyEnchant	тревожное расстройство золото аи атипический дерматит
LanguageTool-python	тревожное расстройство (золото) и утопический дерматит
PySpellChecker	тревожно расстройство (золофт) и атопичекий дерматит
SymspellPy	тревожное расстройство (золофт) и утопический гематит
Jumspell	тревожное расстройство (золофт) и атопичекий дерматит
Tool (MedDistilBERT)	**тревожное расстройство (золофт) и атопический дерматит**

Word tests internals

- Single test - error and lexical precision
  - 2700 test samples
- Context test - error and lexical precision
  - 2700 test samples
  - 10 words in each sample
  - One of ten words is incorrect, other words are correct
  - Same incorrect words as in single test
- Test on real anamnesis
  - 100 real anamnesis from Almazov dataset
  - Count the correct and unnecessary corrections

Single word test

Tool	Error precision	Lexical precision	Average precision	Average words per second
Aspell-python	**0.86**	0.859	**0.859**	283.7
PyHunspell	0.812	0.539	0.675	9.4
PyEnchant	0.829	0.541	0.685	20
LanguageTool-python	0.762	0.904	0.833	25.1
PySpellChecker	0.354	0.86	0.607	3.4
SymspellPy	0.399	0.813	0.606	**9702.8**
Jumspell	0.267	0.947	0.607	2552.1
Tool (CPU, MedDistilBERT)	0.701	**0.991**	0.846	12.7
Tool (GPU, MedDistilBERT)				39.7

Context word test

Tool	Error precision	Lexical precision	Average precision	Average words per second
Aspell-python	0.739	0.93	0.835	357.3
PyHunspell	0.706	0.719	0.713	11.8
PyEnchant	0.721	0.719	0.72	24.3
LanguageTool-python	0.727	0.942	0.835	43.6
PySpellChecker	0.304	0.868	0.586	6.7
SymspellPy	0.37	0.913	0.642	**26060.2**
Jumspell	0.307	0.969	0.638	4322.3
Tool (CPU, MedDistilBERT)	**0.765**	**0.99**	**0.878**	45.5
Tool (GPU, MedDistilBERT)				153.8

Test on real anamnesis

Tool	Correct fixes	Unnecessary fixes	Fixes ratio
Aspell-python	20	171	0.105
PyHunspell	–	–	–
PyEnchant	–	–	–
LanguageTool-python	**21**	135	0.135
PySpellChecker	–	–	–
SymspellPy	–	–	–
Jumspell	–	–	–
Tool (MedRoBERTa)	19	**6**	**0.76**
Tool (MedDistilBERT)	18	9	0.667
Tool (MedBertTiny2)	17	11	0.607

Python package

- Assembled the pip python package
- Package contains
  - Source code
  - Dictionary with correct words
  - No models included
- Models are loaded dynamically as needed
- Published package name – medspellchecker

Conclusion

- Overview of the Russian medical texts correction is performed.
- Existing solutions for correction of Russian texts are analyzed.
- The new method of correcting spelling errors is designed.
- The new spelling correction tool is designed and implemented.
- The approbation of the developed tool is conducted.
- Results of the developed tool and existing ones are compared.

The paper was accepted for the ICCS 2023 conference.

Links

GitHub project
github.com/DmitryPogrebnoy/MedSpellChecker

Fine-tuned models
huggingface.co/DmitryPogrebnoy

Pip package
pypi.org/project/medspellchecker

Thank you for your attention!

ITsMOre than a UNIVERSITY

Why was Python chosen to implement the algorithm and the tool?

- Main things about Python
  - high development velocity
  - large number of libraries for all needs

- Performance-critical operations are delegated to high-performance libraries
  - Edit distance calculation – **editdistpy** package
  - Model inference – **transformers** and **accelerate** packages

- Effect of Python on performance is negligible

What purpose does the tool use the Damerau-Levenstein editing distance for?

- Edit distance allows to limit the number of words for which the language model inference is computed
- Compute only for a small subset of words, not for the whole dictionary
- The Damerau-Levenstein distance naturally reflects the basic types of spelling errors
- In this way acceptable performance is achieved

Metrics

- **Error precision** – the ratio of the number of correctly corrected words to the total number of incorrect words
- **Lexical precision** – the ratio of the number of unchanged modified words to the total number of correct words
- **Average precision** – the average of error precision and lexical precision
- **Performance** – the number of words processed by the tool per second

- **Correct fixes** - the number of correctly fixed errors
- **Unnecessary fixes** - the number of correct words corrected
- **Fixes ratio** - the ratio of the correct fixes metric to the unnecessary fixes