A Methodology for Generative Spelling Correction via Natural Spelling Errors Emulation across Multiple Domains and Languages

Nikita Martynov
SberDevices / Moscow
nikita.martynov.98@list.ru

Mark Baushenko
SberDevices / Moscow
MABaushenko@sberbank.ru

Anastasia Kozlova
SberDevices / Moscow
anastasi2510@gmail.com

Katerina Kolomeytseva
SberDevices / Moscow
kolomeytsevak@gmail.com

Aleksandr Abramov
SberDevices / Moscow
andril772@gmail.com

Alena Fenogenova
SberDevices / Moscow
alenush93@gmail.com

Abstract

Large language models excel in text generation and generalization, however they face challenges in text editing tasks, especially in correcting spelling errors and mistyping. In this paper, we present a methodology for generative spelling correction (SC), tested on English and Russian languages and potentially can be extended to any language with minor changes. Our research mainly focuses on exploring natural spelling errors and mistyping in texts and studying how those errors can be emulated in correct sentences to enrich generative models' pre-train procedure effectively. We investigate the effects of emulations in various text domains and examine two spelling corruption techniques: 1) first one mimics human behavior when making a mistake through leveraging statistics of errors from a particular dataset, and 2) second adds the most common spelling errors, keyboard miss clicks, and some heuristics within the texts. We conducted experiments employing various corruption strategies, models' architectures, and sizes in the pre-training and fine-tuning stages and evaluated the models using single-domain and multi-domain test sets. As a practical outcome of our work, we introduce SAGE [1] (Spell checking via Augmentation and Generative distribution Emulation).

1 Introduction

Recent advancements in large language models (LLMs) have shown impressive text generation and language understanding capabilities, evident in benchmarks like SuperGLUE (Wang et al., 2019), GEM (Gehrmann et al., 2021), BigBench (Srivastava et al., 2023) etc. However, these models often encounter challenges when it comes to effectively addressing text editing tasks, particularly automatic correction of misspellings and mistyping. The automatic spelling correction (SC) task is well known, with traditional approaches using rules, dictionaries, or statistical models for spelling error detection and correction. However, the emergence of LLMs and generative techniques has introduced new possibilities and improved the effectiveness of SC.

Thus, this paper addresses the task of automatic generative SC across various domains and proposes the methodology tested on English and Russian languages, which could potentially be extended to any language with minor changes. Our research primarily studies natural orthographic errors, text misspellings, and their emulation during model pre-training. We explore the impact of these emulations on the model's abilities across different domains and model types.

We leverage two different spelling corruption techniques. The first technique applies the statistical analysis of common errors, aiming to mimic natural human behavior when making mistakes. The second technique introduces the most frequent spelling errors, keyboard miss clicks, and a set of heuristics within the texts.

We conduct experiments in both Russian and English languages, employing different corruption strategies and model sizes during pre-training and fine-tuning. As a practical outcome of our work, we introduce SAGE (Spellchecking via Augmentation and Generative distribution Emulation) — a comprehensive library for automatic generative SC. SAGE incorporates various generative models trained with our proposed methodology and includes built-in augmentation techniques. Moreover, we release the data hub within the SAGE project, a valuable Russian language resource consisting of novel open source datasets for spelling.

2 Related work

Spell checking is a fundamental task in natural language processing (NLP) that aims to correct errors and misspellings in text automatically. Multiple approaches, namely rule-based, statistical, and generative SC methods, have been proposed to tackle this task.

[1] https://github.com/ai-forever/sage

Findings of the Association for Computational Linguistics: EACL 2024, pages 138–155

Rule-based spell checking is one of the most common approaches that rely on predefined rules and dictionaries for detecting and rectifying misspelled words. These resources can incorporate algorithmic error models such as Longest Common Subsequence (Taghva and Stofsky, 2001), Levenshtein Distance (Van Delden et al., 2004), or Phonetic Algorithms (Kondrak and Sherif, 2006).

Statistical spell checking approaches employ machine learning algorithms to learn from extensive text corpora. These algorithms can identify common spelling errors and their corresponding corrections. Some examples of statistical approaches include n-gram models (Ahmed et al., 2009), Hidden Markov Models (Stüker et al., 2011), part-of-speech tagging (Vilares et al., 2016) and Noisy Channel Model (Kernighan et al., 1990).

Generative SC is a promising spell checking approach that has shown remarkable results in recent years. Such systems take into account the context, due to the architecture nature of LLMs such as seq2seq Long Short-Term Memory (LSTM) (Evershed and Fitch, 2014), seq2seq Bidirectional LSTM (Zhou et al., 2019), and state-of-the-art transformer models like BERT (Sun and Jiang, 2019), BSpell (Rahman et al., 2022), etc.

The paper (Guo et al., 2019) presents multilingual translation models for paraphrase generation task. M2M100 models (Fan et al., 2020) (Many-to-Many multilingual models) effectively translate source language text into a target language that aligns with the source language. Given the M2M100 models' comprehensive understanding of multiple languages, their utilization in spell checking tasks proves promising. In our research, among other investigations, we explore the suitability of the M2M approach for SC.

Datasets English spell checking research has received significant attention due to widespread English use, which results in the creation of spell checking datasets. Evaluation datasets such as BEA-2019 shared task (Bryant et al., 2019), comprising corpora like FCE (Yannakoudakis et al., 2011), W&I+LOCNESS, Lang-8 (Tajiri et al., 2012), and NUCLE (Dahlmeier et al., 2013), provide valuable resources for assessing spell checking and error correction tasks. NeuSpell (Jayanthi et al., 2020) introduced the BEA60K natural test set and the well-established JFLEG dataset (Napoles et al., 2017), containing only spelling mistakes. Other clean corpora, including the Leipzig Corpora Collection (Biemann et al., 2007) and the Gutenberg corpus (Gerlach and Font-Clos, 2020), offer diverse sources such as news, web content, and books for further exploration in spell checking research.

Among the standard open source datasets for the Russian language is RUSpellRU [2], which emerged after the competition on automatic SC for Russian social media texts (Sorokin et al., 2016). Other open sources include the GitHub Typo Corpus (Hagiwara and Mita, 2019), which contains the Russian section, and the recent work (Martynov et al., 2023), which introduces a multi-domain dataset.

Text corruption methods For training generative SC models, building a parallel corpus is essential. There are several ways to emulate spelling errors or augment the existing datasets. The example is the GEM benchmark and its associated augmentation library NL-Augmenter (Dhole et al., 2023) and the work (Kuznetsov and Urdiales, 2021) with the method for creating artificial typos. For the Russian language, the RuTransform framework (Taktasheva et al., 2022) presents adding noise into data through spelling corruption and (Martynov et al., 2023) proposes augmentation methods.

3 Methodology

In this work, we aim to design models that meet the end users' demands. The broad application areas of SC tools, encompassing various orthographies and styles, pose additional challenges for text editing systems. We decided to enhance the conventional approach of treating standard language as the only correct spelling option.

3.1 Task Formalization

Before defining the SC task, we must establish the *correct spelling* notion we employ in this work. Instead of rigorously normalizing all supposedly erroneous lexemes to the standard language, we propose distinguishing unintentional spelling violations from intentional ones. Plain language, colloquialisms, dialectisms, and abbreviations are examples of the latter. They can express emotions and endow a text with distinct stylistic features. Since the act of intentional violation of spelling can hardly be expressed in terms of strict rules, it seems nearly impossible to distinguish intentional errors automatically. Instead, following (Martynov

[2] `https://www.dialog-21.ru/evaluation/2016/spelling_correction/`

et al., 2023), we use manual labeling and consider a sentence annotated and amended by native experts as correct. Given a correct sentence, any sentence obtained from the correct one by (probably) multiple insertions, deletions, substitutions, or transpositions of characters is considered erroneous. This leads to the following definition of SC task that we use in this paper:

> Let $X = [x_1, ..., x_N] = X_{corr.} \cup X_{incorr.}$, where $x_1, ..., x_N$ is an ordered sequence of lexemes, $X_{corr.} = \{x_i\}_{i=1}^{k}$ is a set of correct lexemes, $X_{incorr.} = \{x_j\}_{j=1}^{p}$ is a set of incorrect lexemes, $p + k = N, p \geq 0, k > 0$, be the sentence that may contain spelling errors. The system M then should produce corresponding sequence (ordered) $Y = [y_1, ..., y_M] = Y_{corr.} \cup Y_{incorr.}, Y_{incorr.} = \emptyset$ so that
>
> 1. Correct lexemes are not modified: $!\exists f : \{x_i\}_{i=1}^{k} \to Y, f-$injective and preserves order and $f(x_i) = x_i$;
> 2. Original style of a sentence X is preserved;
> 3. All the information is fully transferred from X to Y and no new information appears in Y;

Basically, system M only corrects unintentional errors and carries stylistic and factological pallet the same from X to Y.

3.2 Overview

In this paper, we propose a methodology for generative SC, exploring the natural spelling errors across multiple domains and assessing their influence on spell checking quality during pre-training and fine-tuning stages. The method can be summarized as follows:

Corruption step: the paper explores the text corruption techniques using two augmentation algorithms described in Section 3.3.

Generation step: we pre-train the generative models of different sizes and on the extensive synthetic dataset of diverse domains. The error distribution of the synthetic pre-train data is created by emulating the natural distribution of the errors via a statistic-based approach.

Fine-tune step: during the fine-tuning, we investigate the influence of corruption and domains on the final results. The models are evaluated on fixed single-domain and multiple-domain test sets. The experiments involve training the pre-trained models on various training data from single and multiple domains, as well as using the same data corrupted with the two aforementioned augmentation techniques.

The methodology is explored and tested in the Russian and English languages but can be potentially transferred to any language.

3.3 Augmentations Strategies

3.3.1 Heuristic-based spelling corruption

The first strategy represents spelling corruption through exploiting various heuristics, common error statistics, and understanding of implicit mechanics of a language. Nlpaug (Ma, 2019) and NeuSpell (Jayanthi et al., 2020) libraries for English and Augmentex (Martynov et al., 2023) for Russian are notable examples of such strategy. In this work, we choose Augmentex for experiments with Russian LLMs. This library is accompanied with proven effectiveness for the Russian language (Martynov et al., 2023) and provides a flexible interface to its interior methods. Each method is responsible for modeling a specific type of error, including inserting random characters, replacing correctly spelled words with their incorrect counterparts, inserting nearby keyboard characters, and replacing a character with another based on the probability of its erroneous use. Augmentex allows researchers to control the distribution of error noise on word and sentence levels as well. In our experiments, we investigate Augmentex in depth by augmenting fine-tune datasets and studying its impact on models' performance. See details of its configurations used at the augmentation stage in A.3.

3.3.2 Statistic-based spelling corruption

We choose statistic-based spelling corruption (SBSC) from (Martynov et al., 2023) as an attempt to reproduce errors from a particular piece of text. The method mimics human behavior when committing an error by scanning distributions of errors in a given text and then reapplying them on correct sentences. The algorithm requires a parallel corpus of sentence pairs (corrupted_sentence, correct_sentence): it builds a Levenshtein matrix between prefixes of sentences in each pair, then it traverses this matrix back along the main diagonal starting from the bottom right entry. At each step, the algorithm detects the position of an error in a sentence and its corresponding type based on surrounding entries. Our work employs statistic-based spelling corruption to prepare pre-training datasets for both English and Russian generative models. The experiments' results discussed in Section 5.2 suggest SBSC's ability to be transferred to another language other than Russian. We also investigate the capacity of this noising strategy by experimenting with augmentation through spelling corruption while fine-tuning.

3.4 Datasets

For multi-domain spell checking experiments, we developed three distinct data suites.

Golden Test Sets: Fixed datasets, including both single-domain and multiple-domain texts, used for evaluation purposes.

Pre-trained Data: Synthetic data generated to emulate natural and random noise misspellings, employed during the pre-training stage to assess their impact on model performance.

Training Data for fine-tuning: Collected using the same method as the test sets, also corrupted with the proposed augmentation strategies to introduce diverse errors. Used during the fine-tuning stage to explore the impact of the different noises on the model performance across domains.

3.4.1 Golden Test Sets

The datasets for the golden test set are chosen in accordance with the specified criteria. First, *domain variation*: half of the datasets are chosen from different domains to ensure diversity, while the remaining half are from a single domain. This is done separately for English and Russian languages. Another criterion is *spelling orthographic mistakes*: the datasets exclusively comprised mistyping, omitting grammatical or more complex errors of non-native speakers. This focus on spelling errors aligns with the formalization of the task as described in section 3.1.

For the Russian language, we choose four different sets:

RUSpellRU – the single-domain open source dataset for social media texts presented in the Shared Task (Sorokin et al., 2016).

MultidomainGold – the dataset first presented in the paper (Martynov et al., 2023). It's a multi-domain corpus comprising the domains: internet domain presented by the Aranea web-corpus, literature, news, social media, and strategic documents. We followed the methodological criteria of the paper and reproduced the two-stage annotation project via a crowd-sourcing platform Toloka [3]: at the first stage, annotators are asked to correct the mistakes, on the second – to validate the results from the previous step. The statistics and details of the instructions and annotation schema are presented in Appendix A.1 and A.2. Following the annotation methodology, we extend the authors' dataset with two more domains: reviews (the part of the Omnia set (Pisarevskaya and Shavrina, 2022)) and subtitles (the part of the Russian part of the OpenSubtitles set [4]).

GitHubTypoCorpusRu – we take the Russian part of the corpora introduced in work (Hagiwara and Mita, 2019). Additionally, we validate the parallel data of this corpus by the same Toloka project, but only the second step from the methodology.

MedSpellChecker [5] (Pogrebnoi et al., 2023) is a single-domain set of a specific lexicon of the medical domain; the multi-domain set above does not cover that. The set contains the medical texts of anamnesis. The data was verified via a two-stage annotation pipeline as well.

For the English language, we used two sets: **BEA60K** is a multi-domain dataset corpus for spelling mistakes in English.

JHU FLuency-Extended GUG Corpus (JFLEG) is a single domain set, the spelling part. The dataset contains 2K spelling mistakes (6.1% of all tokens) in 1601 sentences.

The test datasets statistics are presented in the Table 5 of the Appendix, the annotation details in Appendix A.2.

3.4.2 Pre-training Data

To prepare pre-training datasets, we take correct samples and then corrupt them employing augmentation strategies described in 3.3. As for correct samples for experiments in Russian, we use twelve gigabytes (12GB) of raw Russian Wikipedia dumps and an open source dataset of transcripted videos in Russian [6] of three and a half million (3.5M) texts. We remove all the sentences with characters other than Russian and English alphabets, digits, and punctuation or under forty characters. We balance

[3] https://toloka.ai/tolokers

[4] https://opus.nlpl.eu/OpenSubtitles-v2016.php

[5] https://github.com/DmitryPogrebnoy/MedSpellChecker/tree/main

[6] https://huggingface.co/datasets/UrukHan/t5-russian-spell_I

both datasets to roughly 3.3 million sentences, resulting in a pre-training corpus of 6.611.990 texts. Then statistic-based spelling corruption is applied. We scan statistics from the train split of RUSpellRU, multiply the number of errors per sentence distribution by ten to ensure we induce a much denser noise in the pre-training corpus than it is in fine-tuning datasets, and apply to the pre-training corpus to get corrupted sentences. As a result, the pre-training dataset is a collection of 6.611.990 text pairs, each consisting of corrupted sentences and corresponding correct sentences.

For pre-training in the English language, we combine clean Leipzig Corpora Collection [7] (News domain) and English Wikipedia dumps, preprocess them the way we applied for Russian and create a parallel corpus using a statistic-based augmentation technique based on a 5k subset of BEA60K. We result in six gigabytes (6GB) of data for pre-training.

3.4.3 Training Data for fine-tuning

As for the datasets for fine-tuning, we use train splits of RUSpellRU and MultidomainGold and a combination of both (details in Table 6 of Appendix). We also employ spelling corruption methods from 3.3 for augmentation purposes in two separate ways. First, we introduce misspellings in erroneous parts of train splits of fine-tuned datasets, inducing more errors without expanding the dataset itself. In the second strategy, we expand train splits of fine-tuned datasets. We obtain correct sentences from a particular dataset, corrupt spelling, and append pairs of corrupted sentences and corresponding correct sentences to the same dataset. In Tables 4 and 10 of Appendix, the first strategy is marked as *Add* and the second as *Concat*.

We do not prepare fine-tuned datasets for the English language since we do not conduct fine-tuning in our experiments.

4 Experiments

We conducted a comprehensive series of experiments involving diverse spelling corruption strategies over the encoder-decoder generative models of different sizes throughout the pre-training and fine-tuning phases as well as zero-shot evaluation of the pre-trained models. The models' statistics are presented in Table 8. We compared performance based on single-domain and multi-domain test sets. Furthermore, we conducted a comparative evaluation of the OpenAI models utilizing different prompts and standard open source models.

4.1 Models

The generative models of different sizes used as pre-trained models in the experiments are the following for the Russian language:

M2M100$_{\mathbf{1.2B}}$ [8] (Fan et al., 2020) M2M100 is a multilingual encoder-decoder (seq-to-seq) model primarily intended for translation tasks proposed by the Meta team. The model contains 1.2B parameters.

M2M100$_{\mathbf{418M}}$ [9] is a 418M parameters model of the M2M100 models family.

Fred-T5 [10] (Full-scale Russian Enhanced Denoisers T5) (Zmitrovich et al., 2023) is a Russian 820M parameters generative model. The model is trained on a mixture of 7 denoisers like UL2 on extensive Russian language corpus (300GB). The model is inspired by the ideas from the work (Tay et al., 2022) and one of the top [11] generative models according to the RussianSuperGLUE benchmark (Shavrina et al., 2020).

In the case of the English language, the utilization of only one pre-trained model was decided due to the considerable environmental impact caused by the training process (see section 6 *Energy Efficiency and Usage* for details).

T5$_{\mathbf{large}}$ [12] is the English encoder-decoder 770M parameters model introduced by Google's AI research team (Raffel et al., 2020).

4.2 Russian experiments

For each of the three models M2M100$_{418M}$, M2M100$_{1.2B}$, FredT5$_{large}$, the performance on the SC task was compared with and without pre-training, and using different training data for fine-tuning.

Pre-training. We use the same data and pre-training scheme for each model. We train our models in sequence-to-sequence manner with corrupted sentence as an input and correct sentence as label with a standard Cross Entropy loss.

We pre-train FredT5$_{large}$ model with a total *batch size* of 64, *AdamW optimizer* (Loshchilov and Hut-

[7] https://corpora.uni-leipzig.de

[8] https://huggingface.co/facebook/m2m100_1.2B

[9] https://huggingface.co/facebook/m2m100_418M

[10] https://huggingface.co/ai-forever/FRED-T5-large

[11] https://russiansuperglue.com/leaderboard/2

[12] https://huggingface.co/t5-large

Model	RUSpellRU			MultidomainGold			MedSpellChecker			GitHubTypoCorpusRu		
	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1
M2M100$_{1.2B}$												
Pre-train (PT.)	59.4	43.3	50.1	56.4	44.8	49.9	63.7	57.8	60.6	45.7	41.4	43.5
No Pre-train	17.8	38.6	24.4	9.7	37.5	15.4	15.6	36.6	21.9	19.4	36.8	25.4
RUSpellRU (+PT.)	82.9	**72.5**	77.3	53.3	57.8	55.5	55.9	57.8	56.9	39.3	41.5	40.4
RUSpellRU	68.8	42.6	52.6	17.9	25.2	21.0	16.3	17.7	17.0	15.1	14.9	15.0
MultidomainGold (+PT.)	84.9	65.0	73.7	62.5	60.9	61.7	76.3	**73.9**	**75.1**	**47.9**	**43.3**	**45.5**
MultidomainGold	75.4	35.7	48.5	46.5	39.9	43.0	69.1	31.0	42.8	27.4	18.6	22.1
RUSpellRU+MDG (+PT.)	**88.8**	71.5	**79.2**	**63.8**	**61.1**	**62.4**	**78.8**	71.4	74.9	47.1	42.9	44.9
RUSpellRU+MDG	81.2	47.4	59.9	45.8	37.0	40.9	71.8	39.1	50.7	26.1	17.4	20.9
M2M100$_{418M}$												
Pre-train (PT.)	57.7	61.2	59.4	32.8	56.3	41.5	23.2	64.5	34.1	27.5	**42.6**	33.4
No Pre-train	10.6	30.4	15.7	6.1	30.4	10.1	6.8	36.1	11.4	12.8	33.2	18.5
RUSpellRU (+PT.)	81.8	63.4	71.4	45.3	55.9	50.0	40.8	52.2	45.8	29.5	36.6	32.7
RUSpellRU	66.5	38.5	48.8	20.9	26.0	23.2	22.3	14.8	17.8	11.4	13.2	12.2
MultidomainGold (+PT.)	81.3	55.4	65.9	57.9	56.5	57.2	**73.5**	**66.0**	**69.5**	40.3	39.2	39.8
MultidomainGold	63.5	31.6	42.2	39.5	34.9	37.0	55.2	32.5	40.9	23.1	15.5	18.5
RUSpellRU+MDG (+PT.)	**87.6**	**64.4**	**74.2**	**60.3**	**56.6**	**58.4**	73.1	62.4	67.3	**42.8**	37.8	**40.2**
RUSpellRU+MDG	74.0	45.2	56.1	39.8	34.4	36.9	59.5	38.4	46.7	24.7	18.0	20.8
FredT5$_{large}$												
Pre-train (PT.)	58.5	42.4	49.2	42.5	42.0	42.2	37.2	51.7	43.3	52.7	41.7	46.6
No Pre-train	1.3	3.4	1.9	1.9	6.0	2.9	0.6	3.2	0.9	2.9	5.7	3.9
RUSpellRU (+PT.)	55.1	73.2	62.9	26.7	55.1	36.0	12.9	49.6	20.4	26.2	40.5	31.8
RUSpellRU	40.7	50.4	45.0	20.5	42.4	27.6	6.9	26.0	11.0	15.2	23.8	18.6
MultidomainGold (+PT.)	67.7	60.2	63.8	**61.7**	60.5	**61.1**	39.5	**60.4**	**47.7**	**69.3**	44.6	**54.3**
MultidomainGold	49.6	39.9	44.2	48.1	43.4	45.6	**43.2**	41.2	42.2	50.8	25.7	34.1
RUSpellRU+MDG (+PT.)	**74.5**	**73.4**	**73.9**	58.3	**63.1**	60.6	37.5	59.3	45.9	61.2	**45.4**	52.1
RUSpellRU+MDG	56.3	56.2	56.3	48.2	48.5	48.3	42.5	42.7	42.6	49.4	26.9	34.8

Table 1: The models' performance in experiments configurations for the Russian language. For each model, the experiments are reported for the raw ($No\ Pre\text{-}train$) model on zero-shot, the pre-train model on zero-shot, the raw model fine-tuned on the specific train set, and the pre-train model ($+PT.$) fine-tuned on the specific train set. Metrics are reported in **Prec**ision / **Rec**all / **F1**-measure format from (Sorokin et al., 2016).

Model	BEA60K					JFLEG				
	Prec.	Rec.	F1	Acc.	Cor. rate	Prec.	Rec.	F1	Acc.	Cor. rate
BERT	65.8	79.6	72.0	**0.98**	0.79	78.5	85.4	81.8	**0.98**	**0.85**
CNN-LSTM	59.7	76.0	66.8	0.96	0.76	76.8	81.1	78.9	**0.98**	0.80
SC-LSTM	61.7	77.1	68.6	0.96	0.77	77.6	82.1	79.8	**0.98**	0.82
Nested-LSTM	63.1	77.7	69.7	0.96	0.77	78.7	82.7	80.6	**0.98**	0.82
SC-LSTM										
+BERT (at input)	66.2	77.5	71.4	**0.98**	0.77	78.1	83.0	80.5	**0.98**	0.83
+BERT (at output)	64.1	76.7	69.8	0.97	0.76	78.3	83.2	80.6	**0.98**	0.83
+ELMO (at input)	62.3	80.4	72.0	0.96	**0.80**	80.6	86.1	83.3	0.98	**0.85**
+ELMO (at input)	60.4	76.5	67.5	0.96	0.77	77.7	82.5	80.0	**0.98**	0.82
gpt-3.5-turbo-0301										
W/O Punctuation	**66.9**	84.1	74.5	0.84	0.77	77.8	88.6	82.9	0.87	0.78
With Punctuation	57.1	83.5	67.8	0.36	0.34	73.3	**87.9**	80.0	0.34	0.32
gpt-4-0314										
W/O Punctuation	68.6	**85.2**	**76.0**	0.84	0.77	77.9	88.3	82.8	0.86	0.77
With Punctuation	58.4	84.5	69.1	0.36	0.35	73.5	87.7	80.0	0.35	0.32
text-davinci-003										
W/O Punctuation	67.8	83.9	75.0	0.83	0.76	76.8	88.5	82.2	0.87	0.78
With Punctuation	57.6	83.3	68.1	0.35	0.34	72.7	**87.9**	79.6	0.34	0.32
T5$_{large}$ (+PT.)	66.5	83.1	73.9	0.83	0.71	**83.4**	84.3	**83.8**	0.74	0.69
T5$_{large}$	2.6	4.7	3.4	0.01	0.0	3.0	4.3	3.6	0.01	0.0

Table 2: The models' performance for the English language on BEA60K and JFLEG datasets. We report the comparative results of our best model ($+PT$), bare T5-large model, OpenAI models and the open source standard solutions for the English language. Metrics are reported in **Prec**ision / **Rec**all / **F1**-measure and **Acc**uracy / **Cor**rection rate formats from (Sorokin et al., 2016) and (Jayanthi et al., 2020) respectively.

Model	RUSpellRU			MultidomainGold			MedSpellChecker			GitHubTypoCorpusRu		
	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1
Yandex.Speller	83.0	59.8	69.5	52.9	51.4	52.2	**80.6**	47.8	60.0	**67.7**	37.5	48.3
JamSpell	42.1	32.8	36.9	25.7	30.6	28.0	24.6	29.7	26.9	49.5	29.9	37.3
Hunspell	31.3	34.9	33.0	16.2	40.1	23.0	10.3	40.2	16.4	28.5	30.7	29.6
gpt-3.5-turbo-0301												
With Punctuation	55.8	75.3	64.1	33.8	72.1	46.0	53.7	66.1	59.3	43.8	57.0	49.6
W/O Punctuation	55.3	75.8	63.9	30.8	70.9	43.0	53.2	67.6	59.6	43.3	56.2	48.9
gpt-4-0314												
With Punctuation	57.0	75.9	65.1	34.0	**73.2**	46.4	54.2	67.7	60.2	44.2	57.4	50.0
W/O Punctuation	56.4	**76.2**	64.8	31.0	72.0	43.3	54.2	69.4	60.9	45.2	**58.2**	51.0
text-davinci-003												
With Punctuation	55.9	75.3	64.2	33.6	72.0	45.8	48.0	66.4	55.7	45.7	57.3	50.9
W/O Punctuation	55.4	75.8	64.0	31.2	71.1	43.4	47.8	68.4	56.3	46.5	58.1	**51.7**
M2M100$_{1.2B}$	**88.8**	71.5	**79.2**	**63.8**	61.1	**62.4**	78.8	**71.4**	**74.9**	47.1	42.9	44.9

Table 3: The results of the models on different golden tests. We report the comparative results of our best model, which is pre-trained *M2M100$_{1.2B}$* fine-tuned on RUSpellRU and MultidomainGold, OpenAI models and the open source standard solutions for the Russian language. Metrics are reported in format **Prec**ision, **Rec**all, **F1**-measure from (Sorokin et al., 2016).

ter, 2017) with an initial *learning rate* of 3e-04 and *linear decay* with no warm-up steps and *weight decay* 0.001 applied to all the parameters but those in LayerNorm (Ba et al., 2016) and biases, and two steps to accumulate gradients for 5 *epochs*. The pre-train procedure took 180 hours on eight Nvidia A100 GPUs.

Both M2M100$_{418M}$ and M2M100$_{1.2B}$ were pre-trained with a total *batch size* of 64, *AdamW optimizer* (Loshchilov and Hutter, 2017) with an initial *learning rate* of 5e-05, *weight decay* of 0.001 applied to all the parameters but those in LayerNorm (Ba et al., 2016) and biases, and *linear decay* for learning rate without warm-up steps. We also used 8 and 2 *gradient accumulation steps* for M2M100$_{418M}$ and M2M100$_{1.2B}$ accordingly. M2M100$_{418M}$ pre-training procedure took five *epochs* and 332 hours on two Nvidia A100 GPUs, and the corresponding procedure for M2M100$_{1.2B}$ lasted for seven *epochs* and 504 hours on eight Nvidia A100 GPUs.

Fine-tuning. We fine-tune pre-trained and non-pre-trained models using one of three sets: $RUSpellRU$, $MultidomainGold(MDG)$, and $RUSpellRU + MDG$. We also use the augmentation strategies for the training data presented in section 3.3 and obtain additional training data to fine-tune the pre-trained models (see section 3.4 Training Data for fine-tuning for details).

We fine-tune models and take the best-performing checkpoint according to the metrics on the corresponding development set. The models' metrics on the development set are presented in the Appendix A.4. We also used the development set to select the optimal hyperparameter values. We use *AdamW optimizer* (Loshchilov and Hutter, 2017) with $\beta_1 = 0.9$, $\beta_2 = 0.99$ and $\epsilon = 1e{-}8$ and a linear learning rate scheduler to fine-tune models. All hyperparameters for fine-tuning models are contained in Appendix A.7.

Model comparison. We compare the performance of fine-tuned models with pre-trained models in a zero-shot setting, Yandex.Speller [13], JamSpell [14], Hunspell [15], and OpenAI [16] models via API (namely, *gpt-3.5-turbo-0301*, *gpt4-0314*, *text-davinci-003*) with different prompts (see Appendix A.6 for the details) using single-domain and multi-domain test sets (see section 3.4 Golden Test Sets for the details).

4.3 English experiments

We pre-train *T5$_{large}$* model as described in 3.4.2 with the following hyperparameters: *batch size* 64, *learning rate* 3e-04 with *linear decay* and no warm-up steps, *weight decay* 0.001 applied analogously as in experiments with the Russian language, 2 *gradient accumulation steps*, 5 *epochs*. Pre-training is done in mixed-precision with data type bfloat16 [17]. The procedure took 360 hours on eight Nvidia A100 GPUs.

We compare the performance of several models on two datasets: BEA60k and JFLEG. The models are as follows: eight NeuSpell models:

[13] https://yandex.ru/dev/speller/
[14] https://github.com/bakwc/JamSpell
[15] https://github.com/hunspell/hunspell
[16] https://chat.openai.com/
[17] https://pytorch.org/docs/stable/generated/torch.Tensor.bfloat16.html

Model	RUSpellRU			MultidomainGold			MedSpellChecker			GitHubTypoCorpusRu		
	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1
M2M100$_{1.2B}$												
Best-of-FT/PT.	**88.8**	72.5	**79.2**	**63.8**	61.1	62.4	**78.8**	73.9	75.1	47.9	43.3	45.5
Augmentex (Add)												
RUSpellRU	70.6	74.0	72.3	46.7	59.0	52.1	48.5	63.2	54.9	40.9	44.7	42.7
MultidomainGold	73.7	67.4	70.4	58.1	62.0	60.0	69.4	74.2	71.7	47.8	47.1	47.5
RUSpellRU+MDG	75.9	75.7	75.8	57.4	**64.8**	60.9	63.3	72.9	67.8	48.0	**48.1**	**48.1**
Augmentex (Concat.)												
RUSpellRU	72.8	75.4	74.0	48.4	60.3	53.7	49.9	63.7	56.0	41.5	45.7	43.5
MultidomainGold	76.7	68.6	72.4	60.8	63.0	61.9	69.4	71.9	70.6	48.4	45.5	46.9
RUSpellRU+MDG	79.3	**76.5**	77.9	59.6	63.6	61.5	68.5	72.1	70.2	48.4	47.0	47.7
SBSC (Add)												
RUSpellRU	79.0	74.2	76.6	52.0	59.2	55.4	53.0	58.8	55.8	37.7	42.7	40.0
MultidomainGold	86.0	60.6	71.1	63.7	63.1	**63.4**	77.4	**75.2**	**76.3**	47.5	41.4	44.2
RUSpellRU+MDG	84.0	74.7	79.1	61.2	64.4	62.8	73.3	72.4	72.8	47.2	43.3	45.2
SBSC (Concat.)												
RUSpellRU	83.3	72.3	77.4	54.0	59.4	56.6	64.7	56.3	60.2	41.7	41.8	41.7
MultidomainGold	82.8	66.3	73.6	63.5	63.3	**63.4**	74.3	71.6	72.9	**48.6**	44.5	46.5
RUSpellRU+MDG	85.9	72.5	78.6	62.5	63.3	62.9	73.9	68.0	70.8	47.7	43.1	45.3
M2M100$_{418M}$												
Best-of-FT/PT.	**87.6**	64.4	**74.2**	**60.3**	56.6	**58.4**	**73.5**	66.0	**69.5**	42.8	42.6	40.2
Augmentex (Add)												
RUSpellRU	60.1	71.2	65.1	35.2	64.1	45.5	24.0	58.6	34.1	28.3	45.8	35.0
MultidomainGold	61.2	66.6	63.8	49.0	61.1	54.4	48.4	**70.1**	57.3	41.0	46.3	43.5
RUSpellRU+MDG	63.1	70.8	66.7	47.4	60.4	53.1	48.6	68.5	56.8	41.3	**47.0**	**44.0**
Augmentex (Concat.)												
RUSpellRU	65.5	**71.3**	68.3	38.0	**64.5**	47.8	28.1	60.1	38.3	29.8	44.4	35.7
MultidomainGold	68.7	64.9	66.7	54.2	60.2	57.0	58.1	66.8	62.1	**42.9**	43.3	43.1
RUSpellRU+MDG	73.1	70.2	71.7	55.0	60.3	57.5	56.1	68.3	61.6	**42.9**	42.8	42.8
SBSC (Add)												
RUSpellRU	75.7	67.5	71.4	43.2	59.9	50.2	36.9	56.0	44.5	31.8	41.5	36.0
MultidomainGold	75.5	61.2	67.6	55.1	57.9	56.5	65.0	67.0	66.0	42.4	42.0	42.2
RUSpellRU+MDG	78.2	67.7	72.6	56.4	59.9	58.1	64.5	67.3	65.8	42.1	40.3	41.2
SBSC (Concat.)												
RUSpellRU	79.5	65.8	72.0	46.4	58.5	51.8	43.8	53.2	48.0	31.4	37.2	34.0
MultidomainGold	75.2	56.5	64.5	55.9	54.0	55.0	64.9	61.4	63.1	42.1	41.2	41.6
RUSpellRU+MDG	83.6	65.6	73.5	58.7	55.4	57.0	66.8	64.5	65.6	42.5	39.0	40.7
FredT5$_{large}$												
Best-of-FT/PT.	74.5	73.4	73.9	61.7	63.1	**61.1**	43.2	60.4	47.7	**69.3**	45.4	54.3
Augmentex (Add)												
RUSpellRU	51.9	74.6	61.2	25.0	57.5	34.9	12.3	51.4	19.8	25.4	43.7	32.2
MultidomainGold	67.4	67.4	67.4	55.8	62.6	59.0	36.6	60.1	45.5	61.4	47.7	53.7
RUSpellRU+MDG	72.0	**77.9**	**74.8**	51.9	**66.6**	58.3	36.5	61.4	45.8	56.7	**49.3**	52.7
Augmentex (Concat.)												
RUSpellRU	53.3	75.6	62.5	26.6	59.2	36.7	12.5	51.7	20.1	26.1	44.0	32.8
MultidomainGold	66.1	67.2	66.7	55.5	65.7	60.2	36.6	64.5	46.7	64.4	47.9	**54.9**
RUSpellRU+MDG	71.1	75.0	73.0	51.1	62.6	56.3	34.9	58.1	43.6	60.3	48.0	53.5
SBSC (Add)												
RUSpellRU	54.5	73.4	62.5	27.1	57.0	36.8	13.0	51.2	20.8	25.9	41.3	31.8
MultidomainGold	73.5	59.3	65.7	61.5	60.5	61.0	**47.6**	57.0	51.9	66.8	44.6	53.5
RUSpellRU+MDG	**77.4**	71.4	74.3	57.8	61.5	59.6	41.6	57.5	48.3	60.1	46.0	52.1
SBSC (Concat.)												
RUSpellRU	55.0	69.8	61.5	26.0	53.5	35.0	12.8	47.1	20.1	27.4	41.3	32.9
MultidomainGold	64.8	63.1	64.0	59.0	62.7	60.8	38.6	**65.2**	48.5	62.6	46.0	53.0
RUSpellRU+MDG	72.4	74.6	73.5	**61.7**	60.2	61.0	42.7	58.6	**49.4**	65.4	46.2	54.1

Table 4: Pre-trained models' performance on test datasets for the Russian language after fine-tuning on augmented datasets. *Augmentex* and *SBSC* represent different methods of augmentation described in 3.3. *Add* and *Concat.* represent different strategies of augmentation described in 3.4 in the section Training Data for fine-tuning. Metrics reported in format **Prec**ision, **Rec**all, **F1** from (Sorokin et al., 2016).

BERT, CNN-LSTM, SC-LSTM, Nested-LSTM, SC-LSTM + BERT at input/output, and SC-LSTM + ELMO at input/output. Additionally, we evaluate OpenAI models via API (namely, *gpt-3.5-turbo-0301*, *gpt4-0314*, *text-davinci-003*) with different prompts: Full, Short, and Cut (see Appendix 9 for the details). Finally, we compare the obtained results on the Full prompt with models from NeuSpell and $T5_{large}$ model.

5 Evaluation

5.1 Metrics

For the evaluation, we use the script from the Dialogue Shared Task (Sorokin et al., 2016).

As a result, the *F1-measure* as the harmonic mean between *Precision* and *Recall* is calculated. *Precision* amounts for the number of correct lexemes the spellchecker system has not altered, while *Recall* reflects the share of appropriately rectified errors. The evaluation script reported all three metrics.

We also evaluated models for the English language with *accuracy* (correct words among all words) and *correction rate* (misspelled tokens corrected), as it was proposed by (Jayanthi et al., 2020).

5.2 Results

Table 1 presents the results of experiments conducted on the Russian language. The findings indicate superior dominance of pre-trained ($+PT.$) models over the bare fine-tuning. Moreover, larger models generally perform better though this trend is only observed for M2M100 models. The Fred-T5 model, despite its larger size compared to the $\text{M2M100}_{\text{418M}}$ model, demonstrates poorer quality on $RuspellRU$ and $MedSpellChecker$ datasets. This difference in performance may be attributed to the multilingual architecture of the M2M100 model. In our experimental setup, we emulated errors in the pre-trained models using the $RuspellRU$ dataset. This may cause the scores of the models on this specific domain to be substantially higher than those obtained on other datasets.

Including corruption strategies (Table 4) during the fine-tuning stage improves scores. This trend persists consistently across different domains. In the case of the heuristic-based approach, *Add* strategy celebrates most of the performance improvements. In contrast, the statistic-based approach manifests equal contribution of both strategies.

Table 3 demonstrates that non-generative models in the Russian language perform comparably to generative OpenAI models, but they are lightweight and more efficient. However, our best M2M100 model configuration significantly outperforms these solutions.

According to Table 2, the pre-trained T5 model shows comparable with OpenAI models results. We emulated the error distribution based on the BEA60K set during pre-training. However, the final evaluation of the JFLEG set is slightly better than the BEA60K.

The Tables 9,11 presented in the Appendix A.4 demonstrate a notable gap in performance between OpenAI models for English and Russian. In English, the results indicate higher performance when punctuation is not considered. Furthermore, three models demonstrate comparable performance across all models, employing more specific prompts shows better results. However, for Russian the *text-davinci-003* model with punctuation performs better. While analyzing the results, we observed that the generated outputs are sensitive to the prompts. The results contain clichés phrases, forcing additional filtering to obtain accurate results. The observed discrepancy can be attributed to the pre-trained nature of the OpenAI models primarily trained on English language data.

6 Conclusion

In this paper, we have presented a novel methodology for generative SC. The approach involves emulating natural spelling errors during large generative model pre-training and has shown state-of-the-art results in addressing text editing tasks. We use two augmentation techniques for text corruption to improve the results. Conducting the experiments in two languages, we have demonstrated the effectiveness of these techniques and the impact of different corruption strategies across different domains. As for the research's practical impact, we proposed the library SAGE for automatic SC, including the Russian data hub, proposed methods, and the family of generative models. The work contributes significantly to the SC field and opens routes for further exploration.

Limitations

The proposed generative methodology of SC and the created models have certain limitations that should be considered:

Decoding strategies and parameters. The choice of the decoding strategy affects the quality of generated texts (Ippolito et al., 2019). However, our current methodology only comprises part of the spectrum of decoding strategies, limiting our evaluation's extent. During the pre-training and fine-tuning stages, the choice of each model's parameters is limited due to the significant computational costs associated with training and processing.

Text Corruptions and data. A limitation of our study is the availability of different data and the variety of specific domains for the training, fine-tuning stages, and annotated data. We tried to address the issue of data diversity by incorporating single-domain and multi-domain datasets in the proposed research. As for data augmentation, the heuristic approach covers only limited augmentation methods.

Context. The SC model may struggle with word context due to the two main factors: 1) the model's context length is constrained (for example, T5 is limited for 512 sequence length); 2) the data used for the fine-tuning is limited to the text's length of the examples in the dataset, which can lead to bad performance on longer texts if the models saw only short ones. We added the domains of various text lengths to address this problem in the Multidomain-Gold set.

Languages. The methodology employed in our study primarily focuses on investigating the applicability of our spell SC methodology within the Russian language, examining its transferability to the English language. The generalizability of the method across diverse language families remains to be determined. We leave these aspects for future work.

Ethics Statement

In our research on generative SC, we prioritize addressing ethical implications and ensuring responsible technology use.

Datasets and Crowdsourcing annotation. Responses of human annotators are collected and stored anonymously, eliminating personally identifiable information. The annotators are warned about potentially sensitive topics in data (e.g., politics, culture, and religion). The average annotation pay rate exceeds the hourly minimum wage in Russia twice. The data are published under an MIT license. We secured access to public datasets, adhering to relevant terms of service and usage policies.

Energy Efficiency and Usage. Training large-scale LLMs consumes significant computational resources and energy, producing substantial carbon emissions. The decision was made to limit the number of pre-trained models employed for English to minimize the ecological footprint of the research. The CO2 emission of pre-training the M2M100 (Fan et al., 2021) and T5 (Raffel et al., 2020) models in our experiments is computed as Equation 1 (Strubell et al., 2019):

$$CO2 = \frac{PUE * kWh * I^{CO2}}{1000} \tag{1}$$

The resulting CO2 emissions are listed below:

1. $M2M100_{1.2B}$ = 87.09 kg;
2. $M2M100_{418M}$ = 57.37 kg;
3. $T5_{large}$ = 62.21 kg;
4. $FredT5_{large}$ = 31.11 kg;

Data centers' power usage effectiveness (PUE) is at most 1.3. Despite the costs, spelling models can efficiently adapt to users' needs, bringing down potential budget costs in modern applications.

Biases. Our datasets reflecting the Internet domain may contain stereotypes and biases similar to the pre-trained models. Risks of misuse in generative LLMs are a well-discussed concern (Weidinger et al., 2021; Bommasani et al., 2021). We recognize the potential for biases in both training data and model predictions. Proper evaluation is crucial to uncover any vulnerabilities in generalizing new data.

Possible Misuse. We are aware that the results of our work could be misused for harmful content. We emphasize that our research should not harm individuals or communities through legislation, censorship, misinformation, or infringing on information access rights. We offer a novel, broadly applicable methodology that is especially valuable for Russian. While it can enhance written communication, ongoing ethical evaluation is crucial to address emerging challenges.

7 Acknowledgements

The authors sincerely thank Alexey Sorokin for providing us with the evaluation script from the Dialogue Shared task. The authors would also like to extend their appreciation to the teams of the authors of the datasets we took for the training and testing parts. We thank Dmitry Pogrebnoy, the author of anamnesis medical data validated and included in our MedSpellChecker set. The authors are grateful for Ibragim Badertdinov's ideas of heuristic-based corrupted method in the texts. The authors would like to thank Denis Kulagin and his "kartaslov" [18] git-project for the data and statistics on typos. The authors are deeply grateful for the valuable contributions of everyone mentioned above. Their efforts played a crucial role in completing this research.

References

Farag Ahmed, Ernesto William De Luca, and Andreas Nürnberger. 2009. Revised n-gram based automatic spelling correction tool to improve retrieval effectiveness. *Polibits*, (40):39–48.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *stat*, 1050:21.

Chris Biemann, Gerhard Heyer, Uwe Quasthoff, and Matthias Richter. 2007. The leipzig corpora collection-monolingual corpora of standard size. *Proceedings of Corpus Linguistic*, 2007.

Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. 2021. On the opportunities and risks of foundation models.

Christopher Bryant, Mariano Felice, Øistein E Andersen, and Ted Briscoe. 2019. The bea-2019 shared task on grammatical error correction. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 52–75.

Daniel Dahlmeier, Hwee Tou Ng, and Siew Mei Wu. 2013. Building a large annotated corpus of learner english: The nus corpus of learner english. In *Proceedings of the eighth workshop on innovative use of NLP for building educational applications*, pages 22–31.

Kaustubh Dhole, Varun Gangal, Sebastian Gehrmann, Aadesh Gupta, Zhenhao Li, Saad Mahamood, Abinaya Mahadiran, Simon Mille, Ashish Shrivastava, Samson Tan, et al. 2023. Nl-augmenter: A framework for task-sensitive natural language augmentation. *Northern European Journal of Language Technology*, 9(1).

[18] https://kartaslov.ru/

John Evershed and Kent Fitch. 2014. Correcting noisy ocr: Context beats confusion. In *Proceedings of the First International Conference on Digital Access to Textual Cultural Heritage*, pages 45–51.

Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Celebi, Guillaume Wenzek, Vishrav Chaudhary, Naman Goyal, Tom Birch, Vitaliy Liptchinsky, Sergey Edunov, Edouard Grave, Michael Auli, and Armand Joulin. 2020. Beyond english-centric multilingual machine translation. *CoRR*, abs/2010.11125.

Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Celebi, Guillaume Wenzek, Vishrav Chaudhary, et al. 2021. Beyond english-centric multilingual machine translation. *The Journal of Machine Learning Research*, 22(1):4839–4886.

Sebastian Gehrmann, Tosin Adewumi, Karmanya Aggarwal, Pawan Sasanka Ammanamanchi, Anuoluwapo Aremu, Antoine Bosselut, Khyathi Raghavi Chandu, Miruna-Adriana Clinciu, Dipanjan Das, Kaustubh Dhole, Wanyu Du, Esin Durmus, Ondřej Dušek, Chris Chinenye Emezue, Varun Gangal, Cristina Garbacea, Tatsunori Hashimoto, Yufang Hou, Yacine Jernite, Harsh Jhamtani, Yangfeng Ji, Shailza Jolly, Mihir Kale, Dhruv Kumar, Faisal Ladhak, Aman Madaan, Mounica Maddela, Khyati Mahajan, Saad Mahamood, Bodhisattwa Prasad Majumder, Pedro Henrique Martins, Angelina McMillan-Major, Simon Mille, Emiel van Miltenburg, Moin Nadeem, Shashi Narayan, Vitaly Nikolaev, Andre Niyongabo Rubungo, Salomey Osei, Ankur Parikh, Laura Perez-Beltrachini, Niranjan Ramesh Rao, Vikas Raunak, Juan Diego Rodriguez, Sashank Santhanam, João Sedoc, Thibault Sellam, Samira Shaikh, Anastasia Shimorina, Marco Antonio Sobrevilla Cabezudo, Hendrik Strobelt, Nishant Subramani, Wei Xu, Diyi Yang, Akhila Yerukola, and Jiawei Zhou. 2021. The GEM benchmark: Natural language generation, its evaluation and metrics. In *Proceedings of the 1st Workshop on Natural Language Generation, Evaluation, and Metrics (GEM 2021)*, pages 96–120, Online. Association for Computational Linguistics.

Martin Gerlach and Francesc Font-Clos. 2020. A standardized project gutenberg corpus for statistical analysis of natural language and quantitative linguistics. *Entropy*, 22(1):126.

Yinpeng Guo, Yi Liao, Xin Jiang, Qing Zhang, Yibo Zhang, and Qun Liu. 2019. Zero-shot paraphrase generation with multilingual language models.

Masato Hagiwara and Masato Mita. 2019. Github typo corpus: A large-scale multilingual dataset of misspellings and grammatical errors. *CoRR*, abs/1911.12893.

Daphne Ippolito, Reno Kriz, João Sedoc, Maria Kustikova, and Chris Callison-Burch. 2019. Comparison of diverse decoding methods from conditional language models. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3752–3762, Florence, Italy. Association for Computational Linguistics.

Sai Muralidhar Jayanthi, Danish Pruthi, and Graham Neubig. 2020. NeuSpell: A neural spelling correction toolkit. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 158–164, Online. Association for Computational Linguistics.

Mark D Kernighan, Kenneth Church, and William A Gale. 1990. A spelling correction program based on a noisy channel model. In *COLING 1990 Volume 2: Papers presented to the 13th International Conference on Computational Linguistics*.

Grzegorz Kondrak and Tarek Sherif. 2006. Evaluation of several phonetic similarity algorithms on the task of cognate identification. In *Proceedings of the Workshop on Linguistic Distances*, pages 43–50.

Alex Kuznetsov and Hector Urdiales. 2021. Spelling correction with denoising transformer.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Edward Ma. 2019. Nlp augmentation. https://github.com/makcedward/nlpaug.

Nikita Martynov, Mark Baushenko, Alexander Abramov, and Alena Fenogenova. 2023. Augmentation methods for spelling corruptions. In *Proceedings of the International Conference “Dialogue*, volume 2023.

Courtney Napoles, Keisuke Sakaguchi, and Joel Tetreault. 2017. Jfleg: A fluency corpus and benchmark for grammatical error correction.

Nikita Pavlichenko, Ivan Stelmakh, and Dmitry Ustalov. 2021. Crowdspeech and vox diy: Benchmark dataset for crowdsourced audio transcription. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1.

Dina Pisarevskaya and Tatiana Shavrina. 2022. Wikiomnia: generative qa corpus on the whole russian wikipedia.

Dmitrii Pogrebnoi, Anastasia Funkner, and Sergey Kovalchuk. 2023. Rumedspellchecker: Correcting spelling errors for natural russian language in electronic health records using machine learning techniques. In *International Conference on Computational Science*, pages 213–227. Springer.

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. *The Journal of Machine Learning Research*, 21(1):5485–5551.

Chowdhury Rafeed Rahman, MD Rahman, Samiha Zakir, Mohammad Rafsan, and Mohammed Eunus Ali. 2022. Bspell: A cnn-blended bert based bengali spell checker.

Tatiana Shavrina, Alena Fenogenova, Emelyanov Anton, Denis Shevelev, Ekaterina Artemova, Valentin Malykh, Vladislav Mikhailov, Maria Tikhonova, Andrey Chertok, and Andrey Evlampiev. 2020. Russiansuperglue: A russian language understanding evaluation benchmark. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4717–4726.

Alexey Sorokin, Alexey Baytin, Irina Galinskaya, and Tatiana Shavrina. 2016. Spellrueval: The first competition on automatic spelling correction for russian. In *Proceedings of the Annual International Conference “Dialogue*, volume 15.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R. Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, Agnieszka Kluska, Aitor Lewkowycz, Akshat Agarwal, Alethea Power, Alex Ray, Alex Warstadt, Alexander W. Kocurek, Ali Safaya, Ali Tazarv, Alice Xiang, Alicia Parrish, Allen Nie, Aman Hussain, Amanda Askell, Amanda Dsouza, Ambrose Slone, Ameet Rahane, Anantharaman S. Iyer, Anders Johan Andreassen, Andrea Madotto, Andrea Santilli, Andreas Stuhlmüller, Andrew M. Dai, Andrew La, Andrew Lampinen, Andy Zou, Angela Jiang, Angelica Chen, Anh Vuong, Animesh Gupta, Anna Gottardi, Antonio Norelli, Anu Venkatesh, Arash Gholamidavoodi, Arfa Tabassum, Arul Menezes, Arun Kirubarajan, Asher Mullokandov, Ashish Sabharwal, Austin Herrick, Avia Efrat, Aykut Erdem, Ayla Karakaş, B. Ryan Roberts, Bao Sheng Loe, Barret Zoph, Bartłomiej Bojanowski, Batuhan Özyurt, Behnam Hedayatnia, Behnam Neyshabur, Benjamin Inden, Benno Stein, Berk Ekmekci, Bill Yuchen Lin, Blake Howald, Bryan Orinion, Cameron Diao, Cameron Dour, Catherine Stinson, Cedrick Argueta, Cesar Ferri, Chandan Singh, Charles Rathkopf, Chenlin Meng, Chitta Baral, Chiyu Wu, Chris Callison-Burch, Christopher Waites, Christian Voigt, Christopher D Manning, Christopher Potts, Cindy Ramirez, Clara E. Rivera, Clemencia Siro, Colin Raffel, Courtney Ashcraft, Cristina Garbacea, Damien Sileo, Dan Garrette, Dan Hendrycks, Dan Kilman, Dan Roth, C. Daniel Freeman, Daniel Khashabi, Daniel Levy, Daniel Moseguí González, Danielle Perszyk, Danny Hernandez, Danqi Chen, Daphne Ippolito, Dar Gilboa, David Dohan, David Drakard, David Jurgens, Debajyoti Datta, Deep Ganguli, Denis Emelin, Denis Kleyko, Deniz Yuret, Derek Chen, Derek Tam, Dieuwke Hupkes, Diganta Misra, Dilyar Buzan, Dimitri Coelho Mollo, Diyi Yang, Dong-Ho Lee, Dylan Schrader, Ekaterina Shutova, Ekin Dogus Cubuk, Elad Segal, Eleanor Hagerman, Elizabeth Barnes, Elizabeth Donoway, Ellie Pavlick, Emanuele Rodolà, Emma Lam, Eric Chu, Eric Tang, Erkut Erdem, Ernie Chang, Ethan A Chi, Ethan Dyer, Ethan Jerzak, Ethan Kim, Eunice Engefu Manyasi, Evgenii Zheltonozhskii, Fanyue Xia, Fatemeh Siar, Fernando Martínez-Plumed, Francesca Happé, Francois Chollet, Frieda Rong, Gaurav Mishra, Genta Indra Winata, Gerard de Melo, Germán Kruszewski, Giambattista Parascandolo, Giorgio Mariani, Gloria Xinyue Wang, Gonzalo Jaimovitch-Lopez, Gregor Betz, Guy Gur-Ari, Hana Galijasevic, Hannah Kim, Hannah Rashkin, Hannaneh Hajishirzi, Harsh Mehta, Hayden Bogar, Henry Francis Anthony Shevlin, Hinrich Schuetze, Hiromu Yakura, Hongming Zhang, Hugh Mee Wong, Ian Ng, Isaac Noble, Jaap Jumelet, Jack Geissinger, Jackson Kernion, Jacob Hilton, Jaehoon Lee, Jaime Fernández Fisac, James B Simon, James Koppel, James Zheng, James Zou, Jan Kocon, Jana Thompson, Janelle Wingfield, Jared Kaplan, Jarema Radom, Jascha Sohl-Dickstein, Jason Phang, Jason Wei, Jason Yosinski, Jekaterina Novikova, Jelle Bosscher, Jennifer Marsh, Jeremy Kim, Jeroen Taal, Jesse Engel, Jesujoba Alabi, Jiacheng Xu, Jiaming Song, Jillian Tang, Joan Waweru, John Burden, John Miller, John U. Balis, Jonathan Batchelder, Jonathan Berant, Jörg Frohberg, Jos Rozen, Jose Hernandez-Orallo, Joseph Boudeman, Joseph Guerr, Joseph Jones, Joshua B. Tenenbaum, Joshua S. Rule, Joyce Chua, Kamil Kanclerz, Karen Livescu, Karl Krauth, Karthik Gopalakrishnan, Katerina Ignatyeva, Katja Markert, Kaustubh Dhole, Kevin Gimpel, Kevin Omondi, Kory Wallace Mathewson, Kristen Chiafullo, Ksenia Shkaruta, Kumar Shridhar, Kyle McDonell, Kyle Richardson, Laria Reynolds, Leo Gao, Li Zhang, Liam Dugan, Lianhui Qin, Lidia Contreras-Ochando, Louis-Philippe Morency, Luca Moschella, Lucas Lam, Lucy Noble, Ludwig Schmidt, Luheng He, Luis Oliveros-Colón, Luke Metz, Lütfi Kerem Senel, Maarten Bosma, Maarten Sap, Maartje Ter Hoeve, Maheen Farooqi, Manaal Faruqui, Mantas Mazeika, Marco Baturan, Marco Marelli, Marco Maru, Maria Jose Ramirez-Quintana, Marie Tolkiehn, Mario Giulianelli, Martha Lewis, Martin Potthast, Matthew L Leavitt, Matthias Hagen, Mátyás Schubert, Medina Orduna Baitemirova, Melody Arnaud, Melvin McElrath, Michael Andrew Yee, Michael Cohen, Michael Gu, Michael Ivanitskiy, Michael Starritt, Michael Strube, Michał Swędrowski, Michele Bevilacqua, Michihiro Yasunaga, Mihir Kale, Mike Cain, Mimee Xu, Mirac Suzgun, Mitch Walker, Mo Tiwari, Mohit Bansal, Moin Aminnaseri, Mor Geva, Mozhdeh Gheini, Mukund Varma T, Nanyun Peng, Nathan Andrew Chi, Nayeon Lee, Neta Gur-Ari Krakover, Nicholas Cameron, Nicholas Roberts, Nick Doiron, Nicole Martinez, Nikita Nangia, Niklas Deckers, Niklas Muennighoff, Nitish Shirish Keskar, Niveditha S. Iyer, Noah Constant, Noah Fiedel, Nuan Wen, Oliver Zhang, Omar Agha, Omar Elbaghdadi, Omer Levy, Owain Evans, Pablo Antonio Moreno Casares, Parth Doshi, Pascale Fung, Paul Pu Liang, Paul Vicol, Pegah Alipoormolabashi, Peiyuan Liao, Percy Liang, Peter W Chang, Peter Eckersley, Phu Mon Htut, Pinyu Hwang, Piotr Miłkowski, Piyush Patil, Pouya Pezeshkpour, Priti Oli, Qiaozhu Mei, Qing Lyu, Qinlang Chen, Rabin Banjade, Rachel Etta Rudolph, Raefer Gabriel, Rahel Habacker, Ramon Risco, Raphaël Millière, Rhythm Garg, Richard Barnes, Rif A. Saurous, Riku Arakawa, Robbe Raymaekers, Robert Frank, Rohan Sikand, Roman Novak, Roman Sitelew, Ronan Le Bras, Rosanne Liu, Rowan Jacobs, Rui Zhang, Russ Salakhutdinov, Ryan Andrew Chi, Seungjae Ryan Lee, Ryan Stovall, Ryan Teehan, Rylan Yang, Sahib Singh, Saif M. Mohammad, Sajant Anand, Sam Dillavou, Sam Shleifer, Sam Wiseman, Samuel Gruetter, Samuel R. Bowman, Samuel Stern Schoenholz, Sanghyun Han, Sanjeev Kwatra, Sarah A. Rous, Sarik Ghazarian, Sayan Ghosh, Sean Casey, Sebastian Bischoff, Sebastian Gehrmann, Sebastian Schuster, Sepideh Sadeghi, Shadi Hamdan, Sharon Zhou, Shashank Srivastava, Sherry Shi, Shikhar Singh, Shima Asaadi, Shixiang Shane Gu, Shubh Pachchigar, Shubham Toshniwal, Shyam Upadhyay, Shyamolima Shammie Debnath, Siamak Shakeri, Simon Thormeyer, Simone Melzi, Siva Reddy, Sneha Priscilla Makini, Soo-Hwan Lee, Spencer Torene, Sriharsha Hatwar, Stanislas Dehaene, Stefan Divic, Stefano Ermon, Stella Biderman, Stephanie Lin, Stephen Prasad, Steven Piantadosi, Stuart Shieber, Summer Misherghi, Svetlana Kiritchenko, Swaroop Mishra, Tal Linzen, Tal Schuster, Tao Li, Tao Yu, Tariq Ali, Tatsunori Hashimoto, Te-Lin Wu, Théo Desbordes, Theodore Rothschild, Thomas Phan, Tianle Wang, Tiberius Nkinyili, Timo Schick, Timofei Kornev, Titus Tunduny, Tobias Gerstenberg, Trenton Chang, Trishala Neeraj, Tushar Khot, Tyler Shultz, Uri Shaham, Vedant Misra, Vera Demberg, Victoria Nyamai, Vikas Raunak, Vinay Venkatesh Ramasesh, vinay uday prabhu, Vishakh Padmakumar, Vivek Srikumar, William Fedus, William Saunders, William Zhang, Wout Vossen, Xiang Ren, Xiaoyu Tong, Xinran Zhao, Xinyi Wu, Xudong Shen, Yadollah Yaghoobzadeh, Yair Lakretz, Yangqiu Song, Yasaman Bahri, Yejin Choi, Yichi Yang, Yiding Hao, Yifu Chen, Yonatan Belinkov, Yu Hou, Yufang Hou, Yuntao Bai, Zachary Seid, Zhuoye Zhao, Zijian Wang, Zijie J. Wang, Zirui Wang, and Ziyi Wu. 2023. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *Transactions on Machine Learning Research*.

Emma Strubell, Ananya Ganesh, and Andrew McCallum. 2019. Energy and policy considerations for deep learning in NLP. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3645–3650, Florence, Italy. Association for Computational Linguistics.

Sebastian Stüker, Johanna Fay, and Kay Berkling. 2011. Towards context-dependent phonetic spelling error correction in children's freely composed text for diagnostic and pedagogical purposes. In *Twelfth annual conference of the international speech communication association*.

Yifu Sun and Haoming Jiang. 2019. Contextual text denoising with masked language model. In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, pages 286–290.

Kazem Taghva and Eric Stofsky. 2001. Ocrspell: an interactive spelling correction system for ocr errors in text. *International Journal on Document Analysis and Recognition*, 3(3):125–137.

Toshikazu Tajiri, Mamoru Komachi, and Yuji Matsumoto. 2012. Tense and aspect error correction for esl learners using global context. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 198–202.

Ekaterina Taktasheva, Tatiana Shavrina, Alena Fenogenova, Denis Shevelev, Nadezhda Katricheva, Maria Tikhonova, Albina Akhmetgareeva, Oleg Zinkevich, Anastasiia Bashmakova, Svetlana Iordanskaia, Alena Spiridonova, Valentina Kurenshchikova, Ekaterina Artemova, and Vladislav Mikhailov. 2022. TAPE: Assessing few-shot Russian language understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 2472–2497, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Yi Tay, Mostafa Dehghani, Vinh Q Tran, Xavier Garcia, Jason Wei, Xuezhi Wang, Hyung Won Chung, Dara Bahri, Tal Schuster, Steven Zheng, et al. 2022. Ul2: Unifying language learning paradigms. In *The Eleventh International Conference on Learning Representations*.

Sebastian Van Delden, David Bracewell, and Fernando Gomez. 2004. Supervised and unsupervised automatic spelling correction algorithms. In *Proceedings of the 2004 IEEE International Conference on Information Reuse and Integration, 2004. IRI 2004.*, pages 530–535. IEEE.

Jesús Vilares, Miguel A Alonso, Yerai Doval, and Manuel Vilares. 2016. Studying the effect and treatment of misspelled queries in cross-language information retrieval. *Information Processing & Management*, 52(4):646–657.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. *Advances in neural information processing systems*, 32.

Laura Weidinger, John Mellor, Maribeth Rauh, Conor Griffin, Jonathan Uesato, Po-Sen Huang, Myra Cheng, Mia Glaese, Borja Balle, Atoosa Kasirzadeh, et al. 2021. Ethical and social risks of harm from language models.

Helen Yannakoudakis, Ted Briscoe, and Ben Medlock. 2011. A new dataset and method for automatically grading esol texts. In *Proceedings of the 49th annual meeting of the association for computational linguistics: human language technologies*, pages 180–189.

Yingbo Zhou, Utkarsh Porwal, and Roberto Konow. 2019. Spelling correction as a foreign language.

Dmitry Zmitrovich, Alexander Abramov, Andrey Kalmykov, Maria Tikhonova, Ekaterina Taktasheva, Danil Astafurov, Mark Baushenko, Artem Snegirev, Tatiana Shavrina, Sergey Markov, et al. 2023. A family of pretrained transformer language models for russian. *arXiv preprint arXiv:2309.10931*.

A Appendix

A.1 Data

The information of the collected data for the train set and expansion of the gold sets are presented in Tables 6 and 5.

Datasets	1S-A	2S-A	Size	Length
Web (Aranea)	+	+	756	133.8
Literature	+	+	260	194.3
News	+	+	245	278.7
Social media	+	+	200	149.6
Strategic Doc	+	+	250	182.9
Reviews	+	+	586	678.9
OpenSubtitles	+	+	1810	44.2
RUSpellRU	-	-	2008	87
GitHubTypoCorpusRu	-	+	868	156
MedSpellChecker	+	+	1054	135
BEA60k	-	-	63044	79.1
JFLEG	-	-	1601	109

Table 5: The test golden sets statistics. The sizes of the test sets parts in the number of examples (mostly sentences). $1S-A$ represents if the dataset was validated on the first annotation step. $2S-A$ represents if the dataset was validated on the second annotation step. $Length$ is the average number of symbols in one dataset's example.

Datasets	1S-A	2S-A	Size	Length
Web (Aranea)	+	+	386	108.4
News	+	+	361	268.1
Social media	+	+	430	163.9
OpenSubtitles	+	+	1810	45.3
Reviews	+	+	584	689.1
RUSpellRU	-	-	2000	77.9

Table 6: The train sets statistics. The sizes of the train sets parts in the number of examples (primarily sentences). $1S-A$ represents if the dataset was validated on the first annotation step. $2S-A$ represents if the dataset was validated on the second annotation step. $Length$ is the average number of symbols in one dataset's example.

A.2 Annotation

For the extension of the gold test set and the MultidomainGold train part, we use the two-stage annotation setups via a crowd-sourcing platform Toloka[19] (Pavlichenko et al., 2021) similarly to the work (Martynov et al., 2023):

1. **Data gathering stage**: the texts with possible mistakes are provided, and the annotators are asked to write the sentence correctly;
2. **Validation stage**: the pair of sentences (source and its corresponding correction from the previous stage) are provided, and the annotators are asked to check if the correction is right.

[19] https://toloka.ai/tolokers

The annotation costs and the details for the created sets in the current work are presented in Table 7.

Params	S1.Tr	S2.Tr	S1.Te	S2.Te
IAA	82.06	85.20	82.34	91.78
Total	720$	451$	732$	947$
Overlap	3	3	3	3
N_T	7	7	8	8
N_{page}	4	5	4	5
N_C	50	46	50	46
N_U	12	10	10	9
ART	102	71	95	60

Table 7: Details on the data collection projects for the Golden Test sets and the Train MultidomainGold for both parts of the annotation pipeline ($S1.Tr$ is the first annotation stage of train set; $S2.Te$ is the second annotation step of the test set respectively). **IAA** refers to the average IAA confidence scores, %. IAA of the first step is calculated as the expected value of annotators' support of the most popular correction over all labeled texts. IAA of the second step is calculated as an average value of confidence scores overall labeled texts. **Total** is the total cost of the annotation project. **Overlap** is the number of votes per example. N_T is the number of training tasks. N_{page} denotes the number of examples per page. N_C is the number of control examples. N_U is the number of users who annotated the tasks. **ART** means the average response time in seconds.

Model	Speed	Size	Params
$\text{M2M100}_{\text{1.2B}}$	175.73	4.96	1.2B
$\text{M2M100}_{\text{418}}$	326.16	1.94	418M
$\text{Fred-T5}_{\text{large}}$	177.12	3.28	820M
T5_{large}	190.96	2.95	770M

Table 8: The Models' statistics. $Speed$ is the speed of the model on inference on a single Nvidia A100 in symbols per second. $Params$ represents the number of the models' parameters. $Size$ is the size of the models' checkpoint weights in GB.

A.3 Augmentation strategies details

In the diverse array of settings available within Augmentex, customization options include the percentage of phrase changes, the maximum and minimum

Prompt	gpt-3.5-turbo-0301						gpt-4-0314						text-davinci-003					
	BEA60K			JFLEG			BEA60K			JFLEG			BEA60K			JFLEG		
	Pr.	Rec.	F1	Pr.	Rec.	F1	Pr.	Rec.	F1	Pr.	Rec.	F1	Pr.	Rec.	F1	Pr.	Rec.	F1
Full Prompt																		
W/O Punctuation	**66.9**	84.1	**74.5**	**77.8**	88.6	**82.9**	**68.7**	85.3	**76.1**	**77.9**	88.3	**82.8**	**67.7**	84.0	**75.0**	**76.8**	88.5	**82.2**
With Punctuation	57.1	83.5	67.8	73.3	87.9	80.0	58.6	84.5	69.2	73.5	87.7	80.0	57.6	83.3	68.1	72.7	87.9	79.6
Short Prompt																		
W/O Punctuation	38.7	**86.3**	53.5	43.5	**89.5**	58.6	39.0	**85.5**	53.5	39.5	**90.3**	55.0	38.6	**86.5**	53.4	40.1	**90.5**	55.6
With Punctuation	34.4	85.5	49.0	41.9	89.0	57.0	34.7	84.9	49.2	37.9	89.7	53.3	34.7	85.9	49.4	38.6	90.0	54.0
Cut Prompt																		
W/O Punctuation	22.6	80.3	35.3	20.5	80.8	32.7	22.7	80.2	35.4	21.5	83.7	34.3	22.3	80.2	34.9	21.1	83.1	33.7
With Punctuation	20.6	79.6	32.8	19.9	79.9	31.9	20.8	79.5	33.0	20.8	82.9	33.3	20.4	80.1	32.6	20.7	82.5	33.1

Table 9: OpenAI models' performance on SC tasks in English. $W/OPunctuation$ and $WithPunctuation$ reflect the absence and presence of punctuation in the sentence, respectively. Metrics are reported in format **Prec**ision, **Rec**all, **F1**-measure from (Sorokin et al., 2016).

number of errors, and the proportion of phrases eligible for modifications. Among its various augmentation strategies, we choose the word-level approach (replacing the symbols with a probability of their mistaken use) and the sentence-level approach (substituting words with frequent incorrect alternatives). We configured the first setup with the parameters: aug_rate=0.1, min_aug=1, max_aug=3, mult_num=5, action="orfo" and aug_prob=0.7, and the second: aug_rate=0.6, min_aug=1, max_aug=5, action="replace" and aug_prob=0.7.

A.4 Experiments evaluation results

The evaluation of all the experiments discussed in the section 4 that are not covered in the main text are presented in the Tables 9, 11. The evaluation on development sets during the training is presented in Table 10.

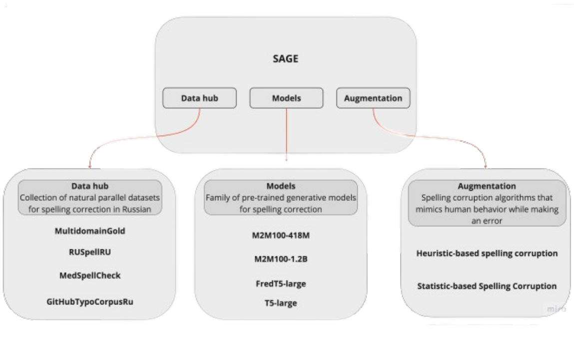

Figure 1: The architecture overview of the SAGE library.

A.5 SAGE library

As the practical result of the introduced methodology, we present SAGE [20] (Spell checking via Augmentation and Generative distribution Emulation). The library consists of three parts: data hub, augmentation strategies, and the family of the models. The architecture is presented on a Figure 1. The data hub includes the whole collection of natural parallel datasets for SC in Russian that were created within the frame of our research. The family of pre-trained generative models for SC involves all the best models trained during the current research for the Russian and English languages. The models are assessed with the inference code from the HuggingFace library [21] and the evaluation script. The last part is the augmentation methods included in SAGE. The statistic-based approach is presented for emulating the user's parallel corpus distribution and provides the emulation algorithm on new data. The heuristic-based approach is presented for producing the noise via different strategies on a word and sentence level in the non-labeled text data.

A.6 OpenAI models prompts experiments

We conduct experiments 9, 11 varying different prompts OpenAI models to evaluate their performance on Golden test sets in Russian and English. For both English and Russian sets, we try three types of prompts: 1) **Cut prompt** for Russian: "Perepishi tekst bez orfograficheskih, grammaticheskih oshibok i opechatok, sohranjaja ishodnyj stil' teksta, punktuaciju, ne raskryvaja abbreviatur i ne izmenjaja korrektnyj tekst:"; for English: "Correct spelling and grammar in the following text:". 2) **Short prompt** for Russian: "Perepishi tekst bez orfograficheskih, grammaticheskih oshibok i opechatok, sohranjaja ishodnyj stil' teksta, punktuaciju, ne raskryvaja abbreviatur i ne izmenjaja korrektnyj tekst:"; for English: "Correct spelling

[20] https://github.com/ai-forever/sage

[21] https://github.com/huggingface/transformers

	M2M100$_{1.2B}$			M2M100$_{418M}$			FredT5$_{large}$		
	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1
Fine-tuning									
without Pre-training									
RUSpellRU	70.8	53.1	60.6	70.5	50.0	58.5	35.6	58.2	44.2
MultidomainGold	40.0	41.2	40.6	34.7	40.5	37.4	51.3	52.8	52.1
RUSpellRU+MDG	51.9	45.6	48.5	46.7	45.8	46.3	48.5	57.0	52.4
with Pre-training									
RUSpellRU	**88.5**	**82.7**	**85.5**	**80.2**	**72.5**	**76.1**	46.7	**80.1**	59.0
MultidomainGold	60.2	67.8	63.8	52.5	59.8	55.9	62.1	69.8	65.7
RUSpellRU+MDG	72.2	73.6	72.9	64.2	64.2	64.2	**62.9**	75.7	**68.7**
Augmentations									
Augmentex (Add)									
RUSpellRU	82.7	82.7	82.7	66.1	76.5	70.9	44.7	78.1	56.9
MultidomainGold	58.3	68.8	63.1	44.2	63.3	52.1	56.7	70.1	62.7
RUSpellRU+MDG	67.5	78.5	72.6	53.1	71.3	60.9	56.6	77.3	65.4
Augmentex (Concat.)									
RUSpellRU	82.7	82.7	82.7	71.2	78.1	74.5	46.4	**81.6**	59.2
MultidomainGold	58.8	69.8	63.8	48.3	61.8	54.2	54.1	73.1	62.2
RUSpellRU+MDG	68.7	76.9	72.6	56.7	68.0	61.9	56.7	76.3	65.0
SBSC (Add)									
RUSpellRU	**88.6**	83.2	**85.8**	77.5	**79.1**	**78.3**	46.3	78.6	58.2
MultidomainGold	57.5	68.8	62.6	50.3	63.1	56.0	63.5	72.8	67.8
RUSpellRU+MDG	69.8	76.9	73.2	59.4	69.8	64.2	63.3	76.7	69.3
SBSC (Concat.)									
RUSpellRU	86.8	**84.2**	85.5	**79.7**	76.0	77.8	45.2	78.6	57.4
MultidomainGold	59.8	69.1	64.1	51.1	60.5	55.4	61.2	71.7	66.1
RUSpellRU+MDG	68.4	76.5	72.2	62.5	65.8	64.1	**66.0**	76.7	**71.0**

Table 10: The evaluation of models' configurations with fine-tuning and the augmentations on dev sets. Metrics are reported in format **Prec**ision, **Rec**all, **F1**-measure from (Sorokin et al., 2016)

Prompt	gpt-3.5-turbo-0301		gpt-4-0314		text-davinci-003	
	W/O Punctuation	With Punctuation	W/O Punctuation	With Punctuation	W/O Punctuation	With Punctuation
Full Prompt						
RUSpellRU	55.3 / **75.8** / 63.9	**55.8** / 75.3 / **64.1**	56.4 / **76.2** / 64.8	**57.0** / 75.9 / **65.1**	55.4 / **75.8** / 64.0	**55.9** / 75.3 / **64.2**
MultidomainGold	30.8 / 70.9 / 43.0	**33.8 / 72.1 / 46.0**	31.0 / 72.0 / 43.3	**34.0 / 73.2 / 46.4**	31.2 / 71.1 / 43.4	**33.6 / 72.0 / 45.8**
MedSpellChecker	53.2 / 67.6 / **59.6**	**53.7** / 66.1 / 59.3	**54.2** / 69.4 / **60.9**	**54.2** / 67.7 / 60.2	47.8 / 68.4 / **56.3**	**48.0** / 66.4 / 55.7
GitHubTypoCorpusRu	**44.5 / 58.1 / 50.4**	43.8 / 57.0 / 49.6	**45.2 / 58.2 / 51.0**	44.2 / 57.4 / 50.0	**46.5 / 58.1 / 51.7**	45.7 / 57.3 / 50.9
Short Prompt						
RUSpellRU	23.1 / 63.9 / 34.0	23.8 / 63.5 / 34.7	22.3 / 60.7 / 32.7	23.2 / 60.5 / 33.6	24.3 / 63.5 / 35.2	25.2 / 63.6 / 36.1
MultidomainGold	12.7 / 54.4 / 20.6	15.0 / 55.8 / 23.6	13.5 / 55.6 / 21.7	15.4 / 55.9 / 24.1	13.8 / 56.5 / 22.2	16.1 / 57.7 / 25.2
MedSpellChecker	30.7 / 76.1 / 43.8	29.2 / **77.9** / 42.5	29.0 / **78.6** / 42.4	30.6 / 76.9 / 43.8	29.8 / 76.4 / 42.9	28.4 / **77.9** / 41.7
GitHubTypoCorpusRu	18.4 / 45.8 / 26.3	18.8 / 46.9 / 26.9	17.1 / 46.0 / 25.0	17.7 / 47.1 / 25.7	19.7 / 47.1 / 27.8	20.1 / 47.1 / 28.2
Cut Prompt						
RUSpellRU	37.9 / 70.3 / 49.3	38.8 / 70.1 / 50.0	35.6 / 64.1 / 45.8	36.4 / 64.0 / 46.4	37.0 / 69.5 / 48.3	37.9 / 69.4 / 49.0
MultidomainGold	7.2 / 46.4 / 12.5	7.5 / 49.1 / 13.1	10.5 / 62.1 / 18.0	7.6 / 46.3 / 13.0	10.6 / 60.6 / 18.0	12.3 / 62.0 / 20.6
MedSpellChecker	5.5 / 52.2 / 10.0	5.3 / 56.3 / 9.7	4.7 / 49.7 / 8.6	5.6 / 51.9 / 10.2	5.9 / 59.9 / 10.8	6.5 / 57.6 / 11.7
GitHubTypoCorpusRu	17.0 / 50.4 / 25.4	17.2 / 50.3 / 25.7	18.0 / 52.7 / 26.8	18.4 / 53.5 / 27.4	18.7 / 53.0 / 27.7	18.6 / 53.3 / 27.6

Table 11: OpenAI models' performance on SC task in Russian. $W/OPunctuation$ and $WithPunctuation$ reflect the absence and presence of punctuation in the sentence, respectively. Metrics are reported in format **Prec**ision, **Rec**all, **F1**-measure from (Sorokin et al., 2016).

and grammar in the following text: . Do not provide any interpretation of your answer.". 3) **Full Prompt** for Russian: "Perepishi tekst bez orfograficheskih, grammaticheskih oshibok i opechatok, sohranjaja ishodnyj stil' teksta, punktuaciju, ne raskryvaja abbreviatur, ne izmenjaja korrektnyj tekst. Napishi tol'ko pravil'nyj otvet bez dopolnitel'nyh ob"jasnenij."; for English: "Rewrite text without spelling errors, grammatical errors, and typos, preserve the original text style and punctuation, do not open abbreviations, and do not change the correct text. Do not provide any interpretation of your answer.".

A.7 Hyperparameters

Model	Hyperparameters				
	learning rate	weight decay	warm-up steps	batch size	epochs
M2M100$_{1.2B}$					
Fine-tuning					
RUSpellRU	8.62e-5	0.0288	5	16	7
MultidomainGold	4.96e-5	0.0135	5	16	8
RUSpellRU+MDG	6.48e-5	0.0416	10	16	7
Pr. + Fine-tuning					
RUSpellRU	8.62e-5	0.0288	5	16	7
MultidomainGold	4.96e-5	0.0135	5	16	8
RUSpellRU+MDG	6.48e-5	0.0416	10	16	7
Augmentex					
RUSpellRU	2e-5	0.01	0	8	7
MultidomainGold	2e-5	0.01	0	4	7
RUSpellRU+MDG	2e-5	0.01	0	4	7
SBSC					
RUSpellRU	8.62e-5	0.0288	5	16	7
MultidomainGold	4.96e-5	0.0135	5	16	8
RUSpellRU+MDG	6.48e-5	0.0416	10	16	7
M2M100$_{418M}$					
Fine-tuning					
RUSpellRU	4.56e-5	0.0493	5	16	7
MultidomainGold	3.39e-5	0.0182	7	16	7
RUSpellRU+MDG	2.66e-5	0.0314	15	8	7
Pr. + Fine-tuning					
RUSpellRU	4.56e-5	0.0493	5	16	7
MultidomainGold	3.39e-5	0.0182	7	16	7
RUSpellRU+MDG	2.66e-5	0.0314	15	8	7
Augmentex					
RUSpellRU	2e-5	0.01	0	16	7
MultidomainGold	2e-5	0.01	0	8	7
RUSpellRU+MDG	2e-5	0.01	0	8	7
SBSC					
RUSpellRU	4.56e-5	0.0493	5	16	7
MultidomainGold	3.39e-5	0.0182	7	16	7
RUSpellRU+MDG	2.66e-5	0.0314	15	8	7
FredT5$_{large}$					
Fine-tuning					
RUSpellRU	2e-4	0.01	0	8	10
MultidomainGold	2e-4	0.01	0	8	10
RUSpellRU+MDG	2e-4	0.01	0	8	8
Pr. + Fine-tuning					
RUSpellRU	2e-4	0.01	0	8	10
MultidomainGold	2e-4	0.01	0	8	10
RUSpellRU+MDG	2e-4	0.01	0	8	8
Augmentex					
RUSpellRU	2e-4	0.01	0	8	10
MultidomainGold	2e-4	0.01	0	8	10
RUSpellRU+MDG	2e-4	0.01	0	8	8
SBSC					
RUSpellRU	2e-4	0.01	0	8	10
MultidomainGold	2e-4	0.01	0	8	10
RUSpellRU+MDG	2e-4	0.01	0	8	8

Table 12: The hyperparameters of models' configurations (pre-trained, fine-tuning, augmentation).