Санкт-Петербургский государственный университет

Кафедра информационно-аналитических систем

Группа 21.Б10-мм

# Автоматическая детекция грамматических ошибок с помощью машинного обучения

***Копань Артём Юрьевич***

Отчёт по учебной практике
в форме «Сравнение»

Научный руководитель:
ассистент кафедры ИАС Г.А. Чернышев

Санкт-Петербург
2024

# Оглавление

# Введение

Каждый год множество студентов пишет учебные практики и выпускные квалификационные работы. Одной из проблем при написании текстов отчётов по ним является соблюдение норм и правил русского языка. Однако проверка большого числа работ на наличие грамматических, орфографических и пунктуационных ошибок весьма трудоёмка, поэтому желательна автоматизация этого процесса.

Контроль орфографии не представляет из себя большой проблемы, так как давно существуют средства автоматической проверки орфографии. При написании учебных работ студенты могут пользоваться автоматической проверкой орфографии в редакторах LaTeX, таких как Overleaf, или специальными плагинами для настольных редакторов LaTeX. Эти инструменты, как правило, сверяют каждое написанное слово по отдельности со своим словарём, и поэтому достаточно надёжны, если речь идёт об ошибках, допущенных в отдельных словах.

Что же касается грамматических ошибок, то при их выявлении возникают сложности. Русский язык отличается большим количеством правил и исключений из них, и его сложнее формализовать, чем, например, английский. Отсюда следует сложность разработки инструментов для исправления грамматических ошибок в русском языке. Распространённые инструменты, такие как Microsoft Word, показывают низкую эффективность для русского языка.

Данная работа посвящена реализации инструмента для обнаружения грамматических ошибок, который будет работать в рамках системы MAP. *Mundane Assignment Police* (MAP) [10] — это веб-сервис, предназначенный для автоматической проверки текстов учебных практик и ВКР на соответствие принятому стилю оформления и отсутствие стилистических ошибок. В этом семестре было решено добавить в сервис возможность обнаруживать также грамматические ошибки, в частности, ошибки согласования. Разрабатываемый инструмент должен будет осуществлять автоматическое обнаружение ошибок согласования, то есть употребление неправильной формы слова в предложении по ро-

ду, числу, падежу.

Из-за сложности данной задачи было решено разбить работу на два семестра. В первом семестре требуется провести обзор существующих подходов к данной проблеме. Во втором — реализовать алгоритм, решающий поставленную задачу.

# 1. Постановка задачи

В ходе работы над учебной практикой были поставлены следующие задачи:

- изучить существующие работы, в которых рассматривается решение задачи автоматической коррекции грамматических ошибок;
- провести обзор найденных работ;
- провести обзор существующих инструментов для решения данной задачи;
- провести обзор существующих датасетов для задачи автоматической коррекции грамматических ошибок;
- сформировать датасет для решения поставленной задачи;
- обучить модели машинного обучения на полученных данных;
- провести экспериментальное исследование и сравнить результативность этих моделей.

Ввиду обширности задачи было решено реализовать первые четыре подзадачи в первом семестре, а непосредственно обучение и сравнение моделей машинного обучения — во втором.

# 2. Обзор

Исправление грамматических ошибок — довольно широкая исследовательская область, освещённая во многих работах. При этом больше всего исследований по этой теме проведено для английского языка, для других же языков их гораздо меньше. При этом задача определения ошибок согласования гораздо меньше освещена, так как она актуальна не для всех языков (например, в английском согласования существительных, прилагательных и причастий как такового нет). Для русского языка не было найдено специализированных работ по данной теме, только статьи по более общей задаче — исправлению грамматических ошибок.

## 2.1. Научные работы

### 2.1.1. A Language Model for Grammatical Error Correction in L2 Russian

Так, в работе "A Language Model for Grammatical Error Correction in L2 Russian" [9] авторы решают задачу коррекции грамматических ошибок в русском языке для иностранцев, изучающих русский язык (L2 speakers). При этом фокус делается именно на письменном языке. В статье предлагается архитектура языковой модели для нахождения и исправления различных грамматических ошибок. В качестве базовой модели используется Yandex.Speller[1]. Этот инструмент позволяет находить ошибки в текстах на русском, украинском и английском языках и использует библиотеку для машинного обучения Catboost[2], которая реализует алгоритм классификации на основе дерева решений с применением градиентного бустинга. Данные для обучения взяты из Национального корпуса русского языка[3], в частности, из его части — Газетного корпуса. Для измерения точности лингвистической модели исполь-

[1] https://yandex.ru/dev/speller
[2] https://catboost.ai
[3] https://ruscorpora.ru/

зовался корпус RULEC-GEC[4]. Авторы используют итеративный метод исправления ошибок, то есть каждая ошибка исправляется независимо от других. Каждый этап алгоритма ответственен за определённый тип ошибок, на выходе этап возвращает ряд частично исправленных версий одного и того же предложения. Отбирается пять наиболее вероятных исправленных предложений. Расчёт вероятности исправления производится с помощью трёхграммной модели Кнезера-Нея (Kneser-Ney), обученной на Газетном корпусе с KenLM [13]. Предложенный авторами алгоритм решает следующие задачи:

- исправление орфографических ошибок (это распространённая и хорошо изученная задача, и она не будет рассматриваться в рамках нашей работы);
- добавление пропущенной запятой перед союзами и выбор корректной формы предлога “о”/“об” (решается с помощью простых правил);
- замена неправильно употреблённого предлога с помощью ruBERT (например, “в”/“во”, “с”/“со”);
- исправление ошибок согласования подлежащего-сказуемого и прилагательного-существительного по роду, числу, падежу. Эта задача и будет рассматриваться в данной работе. Для нахождения таких ошибок из Газетного корпуса извлекаются биграммы и триграммы. Для каждого предложения корпуса проводится POS-теггинг, строится дерево разбора и извлекаются все возможные цепочки из двух или трёх слов, одно из которых является существительным. Это слово сохраняется, а зависимые слова заменяются соответствующими грамматическими тегами.

Затем полученное решение сравнивается с другими алгоритмами, решающими ту же задачу. Авторам удалось достичь значения метрики

[4] https://github.com/arozovskaya/RULEC-GEC

$F_{0.5} = 0.4141$. Здесь

$$F_{0.5} = \frac{1.25 \cdot Precision \cdot Recall}{0.25 \cdot Precision + Recall}.$$

### 2.1.2. Automatic detection and correction of context-dependent dt-mistakes using neural networks

В работе “Automatic detection and correction of context-dependent dt-mistakes using neural networks” [3] авторы решают похожую проблему для голландского языка. Глаголам в голландском языке присваивается разное склонение в зависимости от их грамматической роли и позиции в предложении. Это приводит к возникновению одной из самых распространённых орфографических ошибок в голландском языке, которая обычно называется dt-ошибкой.

Авторы используют для предсказания представление основы глагола и представление контекста. Контекст включает информацию о подлежащем, времени глагола и положении глагола по отношению к подлежащему. Далее к векторным представлениям основы и контекста применяется механизм внимания (Attention) и SoftMax. Путём конкатенации этих векторов получается вектор суффиксного представления, который и используется при обучении нейронной сети, которая относит глагол к одному из классов.

В качестве данных для обучения использовалась голландская часть Параллельного корпуса слушаний Европейского парламента [8]. В части предложений были искусственно сделаны ошибки, чтобы получить негативные примеры для обучения. Для тестирования использовалось несколько наборов данных, не связанных с обучающим набором (основанных на тестах на правописание, по 20 примеров в каждом):

- “Nooit meer dt-fouten” от газеты “De Standaard”;
- языковой тест HBO по правописанию глаголов от издательского дома “Uitgeverij Pak”;
- тест по голландскому языку, также от “Uitgeverij Pak”.

После этого были проведены три эксперимента:

- влияние орфографических ошибок на PoS-теггер;
- тестирование версии модели, не учитывающей контекст;
- тестирование разных версий модели с учётом контекста.

Последний эксперимент наиболее интересен. В нём авторам удалось добиться $F_1 = 97.85\%$ при использовании модели, которая применяет к представлению контекста механизм внимания и затем использует модель BiLSTM, а основа глагола представляется как word+char.

Наконец, полученная модель сравнивается с другими существующими решениями:

- Microsoft Word (2013 Professional Plus, Office 365 Desktop, Office 365 Online);
- Schrijfassistent — инструмент для проверки грамматики, стиля письма и орфографии для голландского языка;
- languagetool.org — аналогичный open-source инструмент, использующий правила для коррекции dt-ошибок;
- valkuil.net — инструмент, использующий фиксированное окно контекста из четырёх слов для коррекции dt-ошибок.

Предложенный авторами подход лидирует в этом сравнении, обгоняя конкурентов на всех предложенных датасетах. Так, на датасете HBO он даёт $F_1 = 57.14\%$, в то же время для MS Word $F_1 = 11.76\%$.

Таким образом, нам стоит попробовать использовать в своей работе BiLSTM, похожее представление слов и использовать контекст.

### 2.1.3. A Multilayer Convolutional Encoder-Decoder Neural Network for Grammatical Error Correction

В работе “A Multilayer Convolutional Encoder-Decoder Neural Network for Grammatical Error Correction” [5] рассматривается применение мно-

гослойной свёрточной нейронной сети для решения задачи исправления грамматических ошибок для английского языка.

Нейронная сеть имеет архитектуру “encoder-decoder”. Encoder (кодировщик) в данном случае кодирует потенциально ошибочные предложения в векторном пространстве, а decoder (декодировщик) на основе этого представления строит исправленное предложение. Для представления предложений используются эмбеддинги (embeddings). Encoder состоит из семи слоёв, которые последовательно действуют на входные данные операциями умножения и сложения векторов, свёртки и gated linear units (GLU). Decoder также состоит из семи слоёв. Эти слои также применяют ко входным эмбеддингам операции свёртки и GLU, а также механизм внимания.

Эта модель была предобучена на большом корпусе англоязычных текстов. Для создания векторных представлений слов использовался метод word2vec.

Модель строит исправленное предложение, оценивая вероятности появления отдельных слов. На каждом шаге генерируется несколько вариантов, и в конце берётся наилучший вариант предложения (с самой высокой суммарной вероятностью).

Для обучения оценки эффективности модели использовались датасеты Lang-8 [7] и NUCLE [6] (всего 1.3 миллиона пар предложений, 18.05 миллионов пар слов). Предобучение производилось на корпусе английской Википедии (1.78 миллиардов слов). Также использовалось подмножество корпуса Common Crawl (94 миллиарда слов) для обучения лингвистической модели для повторной оценки. Применение вышеописанной модели вместе с ансамблированием и лингвистической моделью, обученной на Common Crawl, показало результат $F_{0.5} = 54.13$.

### 2.1.4. Grammatical Error Correction in Low-Resource Scenarios

В работе “Grammatical Error Correction in Low-Resource Scenarios” [11] рассматривается применение модели глубокого обучения на основе архитектуры “трансформер” [2]. Также авторы собрали датасет для зада-

Таблица 1: Результаты трансформера для разных языков

| Язык | Pr | Rec | $F_{0.5}$ |
|---|---|---|---|
| Русский | 63.26 | 27.50 | 50.20 |
| Чешский | 83.75 | 68.48 | 80.17 |
| Немецкий | 78.21 | 59.94 | 73.71 |

Таблица 2: Сравнение моделей на основе CNN и трансформера

| **Модель** | **RULEC** | | | **RU-Lang8** | | |
|---|---|---|---|---|---|---|
| | Pr | Rec | $F_{0.5}$ | Pr | Rec | $F_{0.5}$ |
| CNN | 55.8 | 26.6 | 45.7 | 57.9 | 26.8 | **47.0** |
| Transformer | 59.1 | 26.1 | **47.2** | - | - | - |
| Transformer + dev | 63.3 | 27.5 | **50.2** | 55.3 | 28.5 | 46.5 |

чи GEC для чешского языка. Разработанная модель оценивалась для чешского, немецкого и русского языков.

Для русского языка использовался датасет RULEC-GEC, для чешского — AKCES-GEC. Этот датасет был создан на основе данных Czech Language Acquisition Corpora (AKCES), Корпусов изучения чешского языка. AKCES включает несколько лингвистических корпусов, собранных на основе письменных и устных работ изучающих чешский язык в качестве первого или второго языка. Для немецкого языка используется GEC-корпус на основе текстов Википедии [4].

Результаты предобученной, а затем точно настроенной модели на датасете RULEC-GEC для русского, чешского и немецкого языков показаны в таблице 1.

Две вышеописанные модели глубокого обучения были протестированы на датасетах RULEC-GEC и RU-Lang8 [14]. Результаты сравнения отражены в таблице 2. В строке “Transformer + dev” отражены результаты трансформера после точной настройки на данных RULEC (train + dev). Можно заметить, что на датасете RULEC трансформер уверенно опережает свёрточную нейронную сеть, а на RU-Lang8 наоборот, отстаёт. По всей видимости, это связано с различным характером текстов, которые использовались для построения этих наборов данных, учитывая, что тонкая настройка трансформера производилась именно на RULEC.

## 2.2. Инструменты

### 2.2.1. LanguageTool

LanguageTool[5] — это многофункциональный инструмент для проверки грамматики, орфографии, стиля и синтаксиса. Он поддерживает 30 языков, в том числе русский.

В LanguageTool используются правила, без использования машинного обучения. Правила оформляются в XML и имеют следующий вид:

```
<rule id="BED_ENGLISH"
name="Possible typo 'bed/bat(bad) English/...'">
    <pattern>
      <marker>
        <token regexp="yes">bed|bat</token>
      </marker>
      <token regexp="yes">English|attitude</token>
    </pattern>
    <message>Did you mean
    <suggestion>bad</suggestion>?</message>
    <example correction="bad">Sorry for my
    <marker>bed</marker> English.</example>
</rule>
```

То есть пишется предложение или словосочетание с ошибкой, и как её надо исправлять.

Недостатком инструмента является малое количество правил для русского языка, и, как следствие, многие ошибки не обнаруживаются, в том числе ошибки согласования. Так, если ввести в LanguageTool предложение “*Предложенные* авторами подход лидирует в этом сравнении” (ошибка согласования: множественное число вместо единственного), то ошибка не будет обнаружена (по состоянию на 05.01.2024).

[5] https://languagetool.org/

Таблица 3: Эффективность Яндекс.Спеллера на датасете RULEC-GEC

| **Тип правил** | $F_{0.5}$ |
|---|---|
| Постановка запятых | 37.37% |
| Выбор подходящего предлога | 37.89% |
| Выбор подходящего предлога (ruBERT) | 38.51% |
| Control and agreement errors | 39.03% |

### 2.2.2. Яндекс.Спеллер

Яндекс.Спеллер[6] — это инструмент для исправления орфографических и грамматических ошибок. Доступны русский, украинский и английский языки.

Для нахождения ошибок в Яндекс.Спеллере используются технологии машинного обучения, а именно метод дерева решений с использованием градиентного бустинга, реализованный с помощью библиотеки CatBoost[7].

В целом, Яндекс.Спеллер показывает неплохую эффективность в задаче поиска грамматических ошибок для русского языка. Так, в рассмотренной выше статье [9] приводятся результаты замеров эффективности чистого Яндекс.Спеллера на датасете RULEC-GEC (см. таблицу 3).

Яндекс.Спеллер работает с ошибками согласования эффективнее, чем LanguageTool, однако обнаруживает их не всегда. Так, в предложении "*Предложенные* авторами подход лидирует в этом сравнении" он так же не находит ошибки (по состоянию на 05.01.2024). Однако он хорошо справляется с более простыми случаями.

## 2.3. Наборы данных

### 2.3.1. RULEC-GEC

Авторами статьи [12] был разработан и построен датасет RULEC-GEC[8], который использовался в работе [9]. Этот набор данных содер-

[6] https://yandex.ru/dev/speller/

[7] https://catboost.ai

[8] https://github.com/arozovskaya/RULEC-GEC

жит набор предложений, извлечённых из Корпуса академического письма для учащихся, изучающих русский язык (Russian Learner Corpus of Academic Writing, RULEC) [1], который состоит из эссе и статей, написанных в университетских условиях в США студентами, изучающими русский язык как иностранный, и носителями наследия (heritage speakers — те, кто вырос в США, но знакомился с русским языком дома). Предложения исправляются носителями русского языка, а каждой ошибке присваивается категория. Весь корпус RULEC состоит из 560 тысяч токенов (предложений); набор данных RULEC-GEC содержит 206 258 токенов, 13 047 из них ошибочны и исправлены. RULEC доступен бесплатно для исследовательского использования.

В RULEC-GEC представлены следующие типы ошибок:

- Орфография;
- Существительное: падеж, число, род, др.;
- Лексика: замена;
- Пунктуация;
- Вставка;
- Замена;
- Удаление;
- Прилагательное: падеж, род, число, др.;
- Предлог;
- Лексика: морф.;
- Глагол: число/лицо, вид, залог, время, др.;
- Местоимение;
- Союз.

Таблица 4: Некоторые типы ошибок в корпусе RU-Lang8

| Тип ошибки | Пример |
|---|---|
| выбор подходящего слова | предлагает → утверждает |
| лишнее слово | был → ∅ |
| выбор правильного предлога | в → из |
| выбор правильной формы слова | вдохновлённым → вдохновенной |
| неправильный падеж (сущ.) | специалисты → специалистам |
| неправильный падеж (прил.) | главная → главную |
| неправильное число/лицо/род (гл.) | живут → живёт |
| неправильный вид (гл.) | чувствовала → почувствовала |
| неправильный залог (гл.) | продолжала → продолжалась |

В RULEC присутствуют ошибки согласования существительных и прилагательных по роду, числу, падежу, но у него есть существенный недостаток: используются тексты не носителей русского языка, а иностранцев, которые изучают его.

### 2.3.2. RU-Lang8

Датасет RU-Lang8 был представлен в работе [14]. Он был собран с использованием данных из корпуса Lang-8 [7]. Lang-8 — это набор данных с сайта для изучения иностранных языков[9]. Он содержит данные от изучающих различные иностранные языки и слабо аннотирован (частичные исправления вносятся волонтёрами, но они довольно сильно зашумлены). Корпус Lang-8 состоит из пар вида `(исходное предложение, исправленное предложение)`. В то время как подкорпус английского языка содержит более 30 миллионов токенов (пар предложений), подкорпус изучающих русский язык невелик и содержит около 633 тысяч токенов. Авторы статьи выбрали 51 575 токенов и дополнительно исправили их вручную. Эти пары предложений были случайным образом разделены на разделы обучения, разработки и тестирования.

Затем примеры были размечены по типам ошибок (см. таблицу 4).

Корпус RU-Lang8 отличается от RULEC тем, что последний состоит из эссе, написанных в университетских условиях в контролируемой

[9] https://lang-8.com

среде, тогда как данные Lang-8 собирались онлайн; большинство текстов на RU-Lang8 представляют собой короткие абзацы или вопросы, задаваемые учащимися. Кроме того, в RULEC первый язык авторов текстов — английский, а в RU-Lang8 разнообразие первых языков авторов больше.

Недостатком этого датасета является то, что его нет в открытом доступе.

## 2.4. Обсуждение

По итогам проведённого обзора научных работ, существующих инструментов и датасетов было решено разработать собственный инструмент для решения задачи автоматического поиска ошибок согласования в русском языке с использованием нейронных сетей. Предполагается, что это позволит существенно увеличить эффективность решения данной задачи по сравнению с использованием лингвистических правил и классических алгоритмов машинного обучения (например, метода деревьев решений с использованием градиентного бустинга, который применяется в Яндекс.Спеллере).

В силу того, что наборы данных RULEC-GEC и RU-Lang8 не очень хорошо отражают ошибки носителей русского языка, а также включают много примеров ошибок, которые не представляют интереса в рамках решения данной задачи, было решено разработать собственный датасет на основе доступных корпусов текстов. Возможно, стоит предобучить нейронную сеть на корпусе RULEC, а затем провести точную настройку на собственном датасете.

# Заключение

В ходе работы над учебной практикой были достигнуты следующие результаты:

- Проведён обзор нескольких существующих решений задачи автоматической коррекции грамматических ошибок (GEC);
- Найдены инструменты, решающие данную задачу для русского языка, проведён их обзор;
- Найдены датасеты для задачи GEC, проведён их обзор, выделены преимущества и недостатки.

Учебная практика имеет форму сравнения, поэтому не предполагает написания кода на данном этапе работы.

# Список литературы


[1] Alsufieva Anna, Kisselev Olesya, and Freels Sandra. Results 2012: Using Flagship Data to Develop a Russian Learner Corpus of Academic Writing // Russian Language Journal. — 2012. — Vol. 62. — P. 79–105.

[2] Vaswani Ashish, Shazeer Noam, Parmar Niki, Uszkoreit Jakob, Jones Llion, Gomez Aidan N., Kaiser Lukasz, and Polosukhin Illia. Attention Is All You Need // CoRR. — 2017. — Vol. abs/1706.03762. — arXiv : 1706.03762.

[3] Heyman Geert, Vulić Ivan, Laevaert Yannick, and Moens Marie-Francine. Automatic detection and correction of context-dependent dt-mistakes using neural networks // Computational Linguistics in the Netherlands Journal. — 2018. — Dec. — Vol. 8. — P. 49–65. — Access mode: https://clinjournal.org/clinj/article/view/79.

[4] Boyd Adriane. Using Wikipedia Edits in Low Resource Grammatical Error Correction // Proceedings of the 2018 EMNLP Workshop W-NUT: The 4th Workshop on Noisy User-generated Text / ed. by Xu Wei, Ritter Alan, Baldwin Tim, and Rahimi Afshin. — Brussels, Belgium : Association for Computational Linguistics. — 2018. — Nov. — P. 79–84. — Access mode: https://aclanthology.org/W18-6111.

[5] Chollampatt Shamil and Ng Hwee Tou. A Multilayer Convolutional Encoder-Decoder Neural Network for Grammatical Error Correction // Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018 / ed. by McIlraith Sheila A. and Weinberger Kilian Q. — AAAI Press. — 2018. — P. 5755–5762. — Access mode: https://doi.org/10.1609/aaai.v32i1.12069.

[6] Dahlmeier Daniel, Ng Hwee Tou, and Wu Siew Mei. Building a Large Annotated Corpus of Learner English: The NUS Corpus of Learner English // Proceedings of the Eighth Workshop on Innovative Use of NLP for Building Educational Applications / ed. by Tetreault Joel, Burstein Jill, and Leacock Claudia. — Atlanta, Georgia : Association for Computational Linguistics. — 2013. — June. — P. 22–31. — Access mode: https://aclanthology.org/W13-1703.

[7] Mizumoto Tomoya, Hayashibe Yuta, Komachi Mamoru, Nagata Masaaki, and Matsumoto Yuji. The Effect of Learner Corpus Size in Grammatical Error Correction of ESL Writings // Proceedings of COLING 2012: Posters / ed. by Kay Martin and Boitet Christian. — Mumbai, India : The COLING 2012 Organizing Committee. — 2012. — Dec. — P. 863–872. — Access mode: https://aclanthology.org/C12-2084.

[8] Koehn Philipp. Europarl: A Parallel Corpus for Statistical Machine Translation // Proceedings of Machine Translation Summit X: Papers. — Phuket, Thailand. — 2005. — Sep. 13-15. — P. 79–86. — Access mode: https://aclanthology.org/2005.mtsummit-papers.11.

[9] Remnev Nikita, Obiedkov Sergei, Rakhilina Ekaterina V., Smirnov Ivan, and Vyrenkova Anastasia. A Language Model for Grammatical Error Correction in L2 Russian // CoRR. — 2023. — Vol. abs/2307.01609. — arXiv : 2307.01609.

[10] Mundane Assignment Police (MAP). — https://github.com/Darderion/map.

[11] Náplava Jakub and Straka Milan. Grammatical Error Correction in Low-Resource Scenarios // Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019) / ed. by Xu Wei, Ritter Alan, Baldwin Tim, and Rahimi Afshin. — Hong Kong, China : Association for Computational Linguistics. — 2019. — Nov. — P. 346–356. — Access mode: https://aclanthology.org/D19-5545.

[12] Rozovskaya Alla and Roth Dan. Grammar Error Correction in Morphologically Rich Languages: The Case of Russian // Transactions of the Association for Computational Linguistics. — 2019. — Vol. 7. — P. 1–17. — Access mode: https://aclanthology.org/Q19-1001.

[13] Heafield Kenneth, Pouzyrevsky Ivan, Clark Jonathan H., and Koehn Philipp. Scalable Modified Kneser-Ney Language Model Estimation // Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers) / ed. by Schuetze Hinrich, Fung Pascale, and Poesio Massimo. — Sofia, Bulgaria : Association for Computational Linguistics. — 2013. — Aug. — P. 690–696. — Access mode: https://aclanthology.org/P13-2121.

[14] Trinh Viet Anh and Rozovskaya Alla. New Dataset and Strong Baselines for the Grammatical Error Correction of Russian // Findings of the Association for Computational Linguistics: ACL/IJCNLP 2021, Online Event, August 1-6, 2021 / ed. by Zong Chengqing, Xia Fei, Li Wenjie, and Navigli Roberto. — Association for Computational Linguistics. — 2021. — Vol. ACL/IJCNLP 2021 of Findings of ACL. — P. 4103–4111. — Access mode: https://doi.org/10.18653/v1/2021.findings-acl.359.