Исправление грамматических ошибок в домене низкоресурсных языков

Хабутдинов Ильдар Айратович

Научный руководитель: к.ф.-м.н. А. В. Грабовой
Московский Физико-Технический Институт

20 июня 2025 г.

Задача исправления грамматических ошибок

Проблема

Интерпретируемое автоматическое исправление текстовых последовательностей.

Задача

Исправление грамматических ошибок в домене низкоресурсных языков.

Метод решения

Метод инъективного отображения из множества произвольных символьных последовательностей в множество наперед заданных целевых последовательностей.

Предложенный метод решения задачи основан на сведении задачи нахождения последовательности корректирующих преобразований к задаче поиска оптимального редакционного предписания между исходной и целевой последовательностями.

Визуализация постановки задачи

Исходное предложение

Вчера мы сходили в парк и увидели там много птицы, они была красивые и поют громко.

Целевое предложение

Вчера мы сходили в парк и увидели там много птиц, они были красивыми и пели громко.

Исправление ошибок на уровне слов

“Вчера”, “мы”, “сходили”, “в”, “парк”, “и”, “увидели”, “там”, “много”, “птицы”, “,” , “они”, “была”, “красивые”, “и”, “поют”, “громко”, “.”

Отображение ф → **$TRANSFORM_FORM_accs** **$REPLACE_были** **$REPLACE_красивыми** **$TRANSFORM_TIME_past**

“Вчера”, “мы”, “сходили”, “в”, “парк”, “и”, “увидели”, “там”, “много”, “птиц”, “,” , “они”, “были”, “красивыми”, “и”, “пели”, “громко”, “.”

Множество корректирующих преобразований: {$KEEP, $DELETE, $APPEND_word, $REPLACE_word, $TRANSFORM_TIME, $TRANSFORM_GEND, ...}

Словарь корректирующих преобразований: {$KEEP, $DELETE, $APPEND_опоздание, ..., $REPLACE_сдаваться, ..., $TRANSFORM_TIME_pres, $TRANSFORM_GEND_masc, ...}

Задача построения корректирующих преобразований

Заданы:

- Алфавит Σ
- Язык Σ^*: множество всех возможных строк над Σ, включая пустую строку ε
- Словарь токенов: $V \subseteq \Sigma^*$

Задан токенизатор $T : s \to \{x_1, x_2, ..., x_n\}, s \in \Sigma^*, x_j \in V$
Множество символьных послед. с разбиением на послед. токенов длины n_i,

$$\mathcal{X} = \{s_i | s_i = \{x_1, x_2, ..., x_{n_i}\} | x_j \in V, 1 \leq j \leq n_i\}_{i=0}^{N},$$

Множество целевых последовательностей с разбиением длины m_i :

$$\mathcal{Y} = \{t_i | t_i = \{y_1, y_2, ..., y_{m_i}\} | y_j \in V, 1 \leq j \leq m_i\}_{i=0}^{N}$$

Множество корректирующих преобразований $\mathcal{F}$, $f \in \mathcal{F} : V \to V$
На основе заданного множества необходимо построить словарь K корректирующих преобразований размера p

$$K = \{f_i : a_i \to b_i | f_i \in \mathcal{F}, a_i, b_i \in V\}_{i=0}^{p}$$

Требуется найти отображение $\phi : \{x_1, x_2, ..., x_{n_i}\} \to \{f_1, f_2, ..., f_{n_i}\}$

$$\{f_1, f_2, ..., f_{n_i}\} : \{f_1, f_2, ..., f_{n_i}\} \circ \{x_1, x_2, ..., x_{n_i}\} \to \{y_1, y_2, ..., y_{m_i}\}, k_j \in K$$

Символом $\circ$ обозначено применение корр. преобразования f_j к токену x_j.

Корректирующие преобразования для русского языка

- Алфавит Σ_{ru} — это множество из 33 букв русского алфавита: $\Sigma_{ru} = \{А, Б, В, ..., Э, Ю, Я\}$
- Язык Σ^*_{ru}: множество всех возможных строк над Σ, включая пустую строку ε
- В качестве словаря токенов взят словарь русских слов $V_{ru} \subseteq \Sigma^*_{ru}$
- Задан токенизатор для русского языка T_{ru}

Строится словарь корректирующих преобразований K[1], который состоит из элементов:

1. Преобразования APPEND и REPLACE с использованием 2500 самых частых русских слов на основе коэффициента Жуайна;
2. Преобразования DELETE и KEEP для удаления слова и сохранения слова соответственно;
3. преобразования, общие для всех слов: слияние и разделение слов, изменение регистра букв и т.д.;
4. специальные преобразования для изменения соответствующих частей речи: изменение падежа, рода, лица, времени, числа;
5. преобразования для исправления распространенных ошибок в глаголах: ться/тся и при/пре.

[1] **Khabutdinov, I.A., Chashchin, A.V., Grabovoy, A.V. et al.** RuGECToR: Rule-Based Neural Network Model for Russian Language Grammatical Error Correction.

Автоматическое построение корректирующих преобразований

Проблемы ручной разработки корректирующих преобразований при переходе к произвольному языку:

- Необходимо определение множества корректирующих преобразований
- необходимо составление словаря корректирующих преобразований;
- необходимо осуществить лингвистический анализ грамматических правил данного языка;

Решение: введение обобщенного множества корректирующих преобразований, переход к исправлению ошибок и разбиению символьных последовательностей на уровне подслов.

Обобщенные корректирующие преобразования

Заданы:

- Произвольный алфавит Σ_{any}
- Произвольный язык Σ^*_{any}: множество всех возможных строк над Σ_{any}, включая пустую строку ε
- Словарь токенов: $V_{\text{any}} \subseteq \Sigma^*_{\text{any}}$
- Токенизатор T_{any}

Условие достижимости. Заданы произвольные конечные символьные последовательности s и t. Токенизатор A удовлетворяет условию достижимости, если любую последовательность $A(t)$ можно получить из любой другой последовательности $A(s)$ за конечное число операций вставки, удаления и замены.

Если пара $(V_{\text{any}}, T_{\text{any}})$ удовлетворяет **условию достижимости**, то минимальное достаточное множество корректирующих преобразований $\mathcal{F}$ состоит из элементов:

1. KEEP — оставить токен без изменения
2. REPLACE_t — заменить токен x_j произвольным токеном t
3. DELETE — удалить токен x_j
4. APPEND_t — добавить токен t после токена x_j

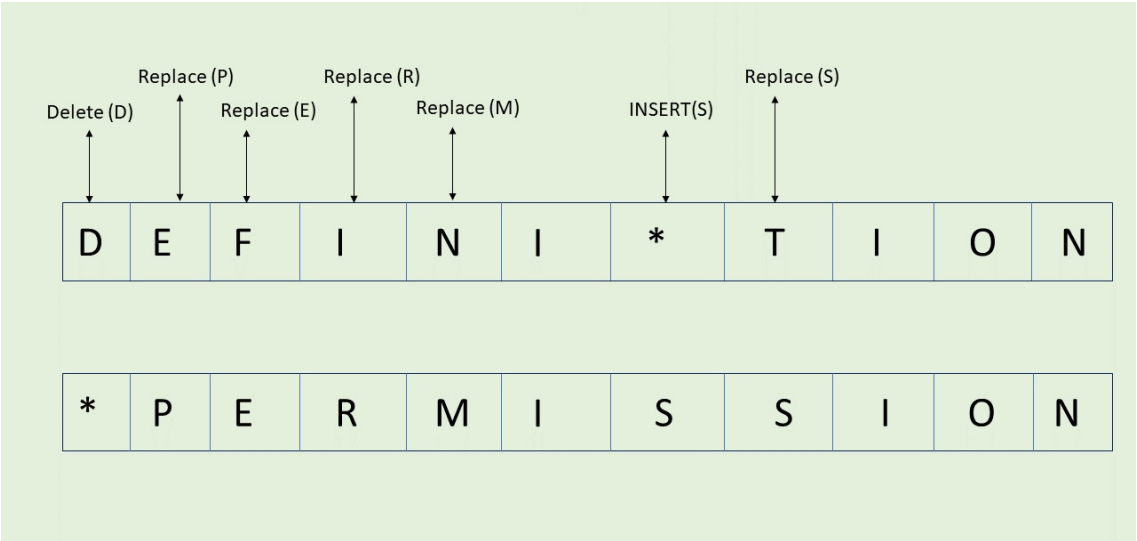
Замечание: T_{ru} не удовлетворяет условию достижимости

Определение редакционного предписания

Опр. Редакционное предписание — это последовательность корретирующих преобразований, необходимых для получения целевой последовательности из исходной, имеющая минимальное количество операций вставки, замены и удаления.

Пусть $D_{i,j}$ — это расстояние редактирования между префиксами $s[0..i]$ и $t[0..j]$ длины i и j. Где $D_{0,j} = 0$ и $D_{i,0} = 0$. Остальные значения определяются рекуррентным соотношением:

$$D_{i,j} = \begin{cases} D_{i-1,j-1}, s[i] = t[j] \\ 1 + min\{D_{i-1,j}, D_{i,j-1}, D_{i-1,j-1}\}, else \end{cases}$$

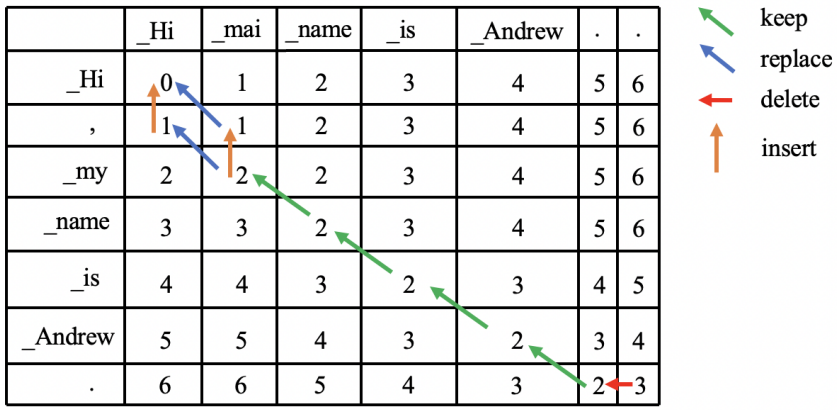


Граф подзадач для редакционного предписания

Граф подзадач, где каждая вершина — пара индексов (i, j), ребра графа — возможные корректирующие преобразования между последовательностями s и t:

- Если $s[i] = t[j]$, то переход по диагонали без изменений — операция *keep*.
- Если символы $s[i]$ и $t[j]$ различны, то возможны следующие операции:
  Замена (replace): переход по диагонали $(i-1, j-1)$,
  Вставка (append): переход из $(i, j-1)$,
  Удаление (delete): переход из $(i-1, j)$.

Утв. Количество возможных редакционных предписаний равно количеству путей в графе подзадач, имеющих минимальную стоимость.



	_Hi	_mai	_name	_is	_Andrew	.	.
_Hi	0	1	2	3	4	5	6
,	1	1	2	3	4	5	6
_my	2	2	2	3	4	5	6
_name	3	3	2	3	4	5	6
_is	4	4	3	2	3	4	5
_Andrew	5	5	4	3	2	3	4
.	6	6	5	4	3	2	3

Редакционные предписания — {delete . ; insert _my ; replace _mai with ,} и {delete . ; insert , ; replace _mai with _my}

Поиск оптимального редакционного предписания

Обозначим $EP_k = \{e_1, e_2, ..., e_{o_k}\}$ множество редакционных предписаний для пары $(s_k,\ t_k)$, где o_k — количество редакционных предписаний для k-й пары.

Для нахождения оптимального редакционного предписания в k-ой паре, мы учитываем сходство исходных и целевых токенов для коррект. преобр. replace.

Пусть $R_l \subset e_l, l \in \{1, 2, ..., o_k\}$ - множество всех правил replace мощностью p_l для произвольного редакционного предписания e_l:

$$R_l = \{replace\_t_{1i}\_t_{2i}\}_{i=0}^{p_l},$$

где исходный токенов $t_{1i} \in s_k$, целевой токен $t_{2i} \in t_k$.

Введем сходство токенов как:

$$\sigma_l = \sum_{i=0}^{p_l} LevenshteinDist(t_{1i}\_t_{2i})$$

где *LevenshteinDist* - функция, вычисляющая расстояние Левенштейна между t_{1_i} и t_{2_i} на уровне символов внутри токенов.

Утв. Редакционное предписание $e_l^* : l = argmin\{\sigma_1, \sigma_2, ..., \sigma_{o_k}\}$ является оптимальным.

Преимущества предложенного метода

Преимущества данного подхода:

1. Данный метод предложен для целого класса токенизаторов, для которых выполнено условие достижимости;
2. нет необходимости в ручной разработке словаря грамматических правил, следовательно, может быть обобщено на любой низкоресурсный язык;
3. множество корректирующих преобразований не зависит от языка;
4. нет необходимости в ручной разработке словаря корректирующих правил K, корректирующие преобразования строятся путем нахождения редакционного предписания.

Эксперимент с корр. преобразованиями русского языка

Система	Этап обучения	P	R	$F_{0.5}$
RuGECToR	1	88.4	**67.1**	**83.1**
RuGECToR	2	**88.5**	65.1	82.5

Таблица: Оценка работы модели на синтетическом тестовом наборе данных

Система	Обучающие данные	P	R	$F_{0.5}$
Classifiers (learner)	RULEC	22.6	4.8	12.9
Classifiers (min sup.)	RULEC	38.0	7.5	21.0
MT	RULEC	30.6	2.9	10.6
RuGECToR	Synthetic (1 stage)	23.6	5.6	14.3
RuGECToR	Synthetic (2 stages)	**40.8**	**7.9**	**22.2**

Таблица: Сравнение работы моделей на наборе данных RULEC

Модель достигает $F_{0.5} = 22.2$. Этот результат выше бейзлайна, несмотря на то, что модель не обучалась на этом наборе данных.

Эксперимент с обобщенными корр. преобразованиями

Dataset	**#sents**		**Training**
	Subword-level	Word-level	**stage**
PIE-synthetic	9,000,000	9,000,000	I
Lang-8	787,613	947,344	II
NUCLE	51,929	56,958	II
FCE	25,968	34,490	II
W&I+LOCNESS	21,828	34,304	II, III

Таблица: Обучающие наборы данных для каждого этапа с соответствующим количеством предложений для модели на уровне слов и модели на уровне подслов.

Model	**CoNLL-2014 (test)**			**BEA-2019 (test)**		
	P	R	$F_{0.5}$	P	R	$F_{0.5}$
GECToR (subword-level + XLNet)	72.3	40.4	62.4	70.5	41.6	61.9
GECToR (word-level + BERT)	72.1	**42.0**	63.0	71.5	**55.7**	67.6
GECToR (word-level + RoBERTa)	73.9	41.5	64.0	77.2	55.1	71.5
GECToR (word-level + XLNet)	**77.5**	40.1	**65.3**	**79.2**	53.9	**72.4**

Таблица: Сравнение работы модели на уровне подслов с моделями на уровне слов.

Выносится на защиту

- Предложено множество корректирующих преобразований для русского языка.
- Модель RuGECToR, аппроксимирующая последовательность корректирующих преобразований.
- Предложено множество обобщенных корректирующих преобразований.
- Доказано, что для предложенного множества выполняется условие достижимости.
- Модель GECToR, аппроксимирующая последовательность корректирующих преобразований на уровне подслов.

Список работ автора по теме диссертации

Публикации

1. **Khabutdinov, I.A., Chashchin, A.V., Grabovoy, A.V. et al.** RuGECToR: Rule-Based Neural Network Model for Russian Language Grammatical Error Correction. Program Comput Soft 50, 315–321 (2024). https://doi.org/10.1134/S0361768824700129
2. **K. Varlamova, I. Khabutdinov and A. Grabovoy**, "Automatic Spelling Correction for Russian: Multiple Error Approach,"2023 Ivannikov Ispras Open Conference (ISPRAS), Moscow, Russian Federation, 2023, pp. 169-175, doi: 10.1109/ISPRAS60948.2023.10508161.
3. **Gritsai, German & Khabutdinov, Ildar & Grabovoy, Andrey**. (2024). Multi-head Span-based Detector for AI-generated Fragments in Scientific Papers. 220-225. 10.18653/v1/2024.sdp-1.21.
4. **K. Grashchenkov, A. Grabovoy and I. Khabutdinov**, "A Method of Multilingual Summarization For Scientific Documents,"2022 Ivannikov Ispras Open Conference (ISPRAS), Moscow, Russian Federation, 2022, pp. 24-30, doi: 10.1109/ISPRAS57371.2022.10076852.
5. **Gritsai, German & Voznyuk, Anastasia & Khabutdinov, Ildar & Grabovoy, Andrey.** (2024). Advacheck at GenAI Detection Task 1: AI Detection Powered by Domain-Aware Multi-Tasking. 10.48550/arXiv.2411.11736.
6. **Khabutdinov, I.A., Krinitskiy, M.A. Belikov, R.A.** Identifying Cetacean Mammals in High-Resolution Optical Imagery Using Anomaly Detection Approach Employing Machine Learning Models. Moscow Univ. Phys. 78 (Suppl 1), S149–S156 (2023). https://doi.org/10.3103/S0027134923070147

Выступления с докладом

1. RuGECToR: нейросетевая модель на основе правил для исправления грамматических ошибок на русском языке «Открытая конференция ИСП РАН», 2022.
2. Multi-head Span-based Detector for AI-generated Fragments in Scientific Papers, SDP@ACL, 2024.
3. Анализ работы BERT-подобных моделей в задачах классификации грамматических ошибок на русском языке «65-я научная конференция МФТИ», 2023.