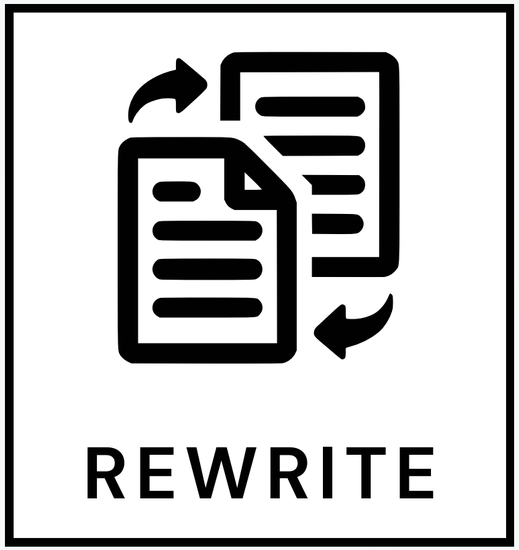
СБЕР

КРИНЖЕНЕРЫ

TALENT CASE CONTEST 2023

ПОСТАНОВКА ЗАДАЧИ

- Дан датасет, содержащий строки на русском языке.
- Запрещено использовать нейронные сети и готовые модели машинного обучения.
- Необходимо реализовать не менее пяти алгоритмов для выявления рерайта.



ЭТАПЫ РЕШЕНИЯ

АНАЛИЗ

- Произвести анализ датасета
- Проанализировать основные приёмы рерайтинга
- Проанализировать существующие решения

ОБРАБОТКА

- Привести данные к практичному представлению (форматировать и структурировать)

РЕАЛИЗАЦИЯ

- Реализовать и применить алгоритмы сравнения строк
- Разработать алгоритм группировки строк, используя алгоритмы сравнения строк

ОЦЕНКА

- Провести оценку результатов работы разработанных решений

ДАТАСЕТ

Анализ

Обнаруженные способы рерайтинга

- Перестановка слов
- Добавление(удаление) слов
- Орфографические ошибки
- Синонимичные конструкции

Дополнительные сложности

- Строки, отличающиеся только местоимениями
- Строки, которые отличаются на 1 слово

Примечание

Каждая строка в датасете может иметь от 0 до 2 рерайтов

Обработка

Все строки из датасета прошли следующую обработку

- Приведение строк к нижнему регистру
- Удаление знаков препинания
- Замена “ё” на “е”
- Удаление лишних пробелов

ИНДЕКСЫ

Вычисление схожести на основе сравнения множеств токенов

РАССТОЯНИЯ

Расчет количества операций для преобразования одной строки в другую

ВЕКТОРИЗАЦИЯ

Преобразование строк в числовой вектор и применение метрик

ИИ

Использование моделей машинного обучения

СУЩЕСТВУЮЩИЕ РЕШЕНИЯ

Найденные алгоритмы сравнения строк могут быть эффективны только в определенных сценариях использования, но оказываются бесполезными или недостаточно точными в других случаях.

НАШИ РЕШЕНИЯ

1. Фильтрация
2. Общее покрытие
3. Ансамбль
4. Итеративная кластеризация
5. Частотный анализ

РЕШЕНИЕ 1

ФИЛЬТРАЦИЯ

ОПИСАНИЕ

ИДЕЯ РЕШЕНИЯ ЗАКЛЮЧАЕТСЯ В ПОЭТАПНОМ ОТБОРЕ ГРУПП, Т.Е. ФИЛЬТРАЦИИ ДАТАСЕТА, ПРИ ПОМОЩИ ПЕРЕЧИСЛЕННЫХ АЛГОРИТМОВ.

ИСПОЛЬЗУЕМЫЕ АЛГОРИТМЫ

- РАССТОЯНИЕ КАРЛОВСКОГО (TS = 0.89, CP = TRUE)
- СРАВНЕНИЕ МНОЖЕСТВ СЛОВ (TS = 0.99, CP = FALSE)
- РАССТОЯНИЕ ХЭММИНГА (TS = 0.9, CP = FALSE)
- КОСИНУСНОЕ СХОДСТВО С ВЕКТОРИЗАЦИЕЙ TF-IDF (TS = 0.7, CP = TRUE)
- СХОДСТВО ДЖАРО-ВИНКЛЕРА (TS = 0.9, CP = TRUE)
- СРАВНЕНИЕ НАБОРОВ МЕСТОИМЕНИЙ

*TS – пороговое значение метрики. CP – необходимость сравнения наборов местоимений.

РЕШЕНИЕ 2

ОБЩЕЕ ПОКРЫТИЕ

ОПИСАНИЕ

КОМБИНАЦИЯ АЛГОРИТМОВ НЕЧЕТКОГО СРАВНЕНИЯ СТРОК, УДОВЛЕТВОРЯЮЩИХ СЛЕДУЮЩИМ УСЛОВИЯМ:

- АЛГОРИТМ МОЖЕТ ОБНАРУЖИТЬ НЕКОТОРЫЕ РЕРАЙТЫ
- АЛГОРИТМ НЕ ПОСЧИТАЕТ СИЛЬНО ОТЛИЧАЮЩИЕСЯ СТРОКИ ЗА РЕРАЙТ

ИСПОЛЬЗУЕМЫЕ АЛГОРИТМЫ

- КОСИНУСНОЕ РАССТОЯНИЕ (TS = 0.9)
- РАССТОЯНИЕ ДАМЕРАУ-ЛЕВЕНШТЕЙНА (TS = 0.9)
- L2-РАССТОЯНИЕ (TS = 0.9)
- N-ГРАММЫ + МЕТРИКА ИНДЕКС ТВЕРСКОГО (TS = 0.9)
- РАССТОЯНИЕ ХЭММИНГА (TS = 0.9)
- СРАВНЕНИЕ НАБОРОВ МЕСТОИМЕНИЙ

*TS – пороговое значение метрики.

РЕШЕНИЕ 3

АНСАМБЛЬ

ОПИСАНИЕ

РЕШЕНИЕ ОСНОВАНО НА МЕТОДЕ ГОЛОСОВАНИЯ, ГДЕ КАЖДЫЙ АЛГОРИТМ ГОЛОСУЕТ ЗА ОДИН ИЗ ОТВЕТОВ И КАЖДЫЙ АЛГОРИТМ ИМЕЕТ ВЕС ГОЛОСА

ИСПОЛЬЗУЕМЫЕ АЛГОРИТМЫ

- КОСИНУСНОЕ СХОДСТВО С ВЕКТОРИЗАЦИЕЙ ПОДСЧЕТОМ
- КОСИНУСНОЕ СХОДСТВО С ВЕКТОРИЗАЦИЕЙ TF-IDF
- РАССТОЯНИЕ ЛЕВЕНШТЕЙНА
- РАССТОЯНИЕ ДАМЕРАУ-ЛЕВЕНШТЕЙНА
- МАНХЭТТЕНСКОЕ РАССТОЯНИЕ
- ЕВКЛИДОВО РАССТОЯНИЕ
- АЛГОРИТМ ВЫРАВНИВАНИЯ ПОСЛЕДОВАТЕЛЬНОСТИ В СОЧЕТАНИИ С РАССТОЯНИЕМ ХЭММИНГА
- РАССТОЯНИЕ ХЭММИНГА
- РАССТОЯНИЕ ХЭЛЛИНГЕРА
- ИНДЕКС ЖАККАРДА
- РАССТОЯНИЕ ДЖАРО-ВИНКЛЕРА
- ДИВЕРГЕНЦИЯ ДЖЕНСЕНА-ШЕННОНА
- РАССТОЯНИЕ КАРЛОВСКОГО
- АЛГОРИТМ МАЙЕРСА
- N-ГРАММЫ
- N-ГРАММЫ В СОЧЕТАНИИ С ИНДЕКСОМ ТВЕРСКОГО
- КОЭФФИЦИЕНТ СЁРЕНСЕНА
- СРАВНЕНИЕ МНОЖЕСТВ СЛОВ
- СРАВНЕНИЕ НАБОРОВ МЕСТОИМЕНИЙ

РЕШЕНИЕ 4

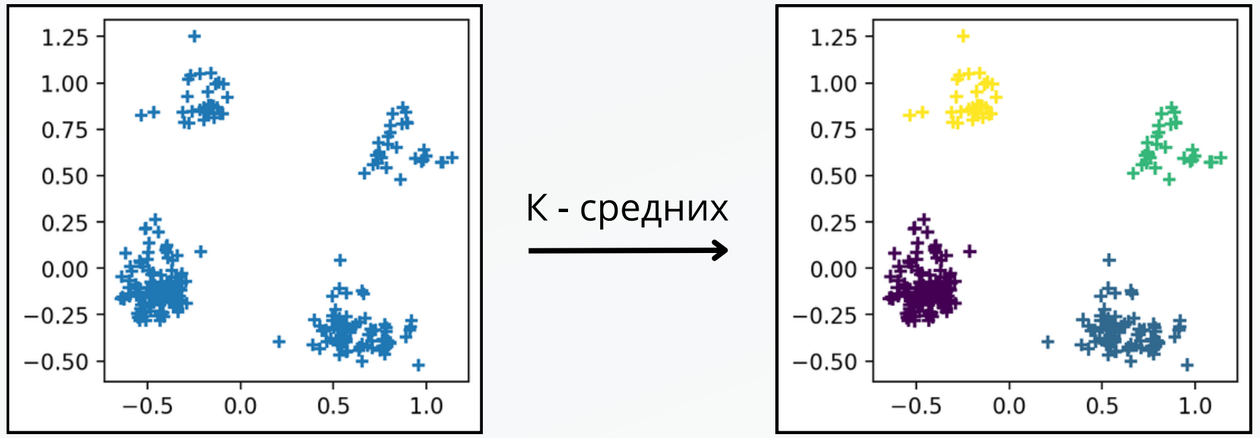
ИТЕРАТИВНАЯ КЛАСТЕРИЗАЦИЯ

ОПИСАНИЕ

РЕШЕНИЕ ОСНОВАНО НА ВЕКТОРИЗАЦИИ ДАННЫХ С ЦЕЛЬЮ ОБРАБОТКИ ПОЛУЧЕННОГО ДАТАСЕТА МЕТОДОМ КЛАСТЕРИЗАЦИИ

ИСПОЛЬЗУЕМЫЕ АЛГОРИТМЫ

- **МЕТОД КЛАСТЕРИЗАЦИИ К-СРЕДНИХ**
- **РАСТОЯНИЕ КАРЛОВСКОГО**



РЕШЕНИЕ 5

ЧАСТОТНЫЙ АНАЛИЗ

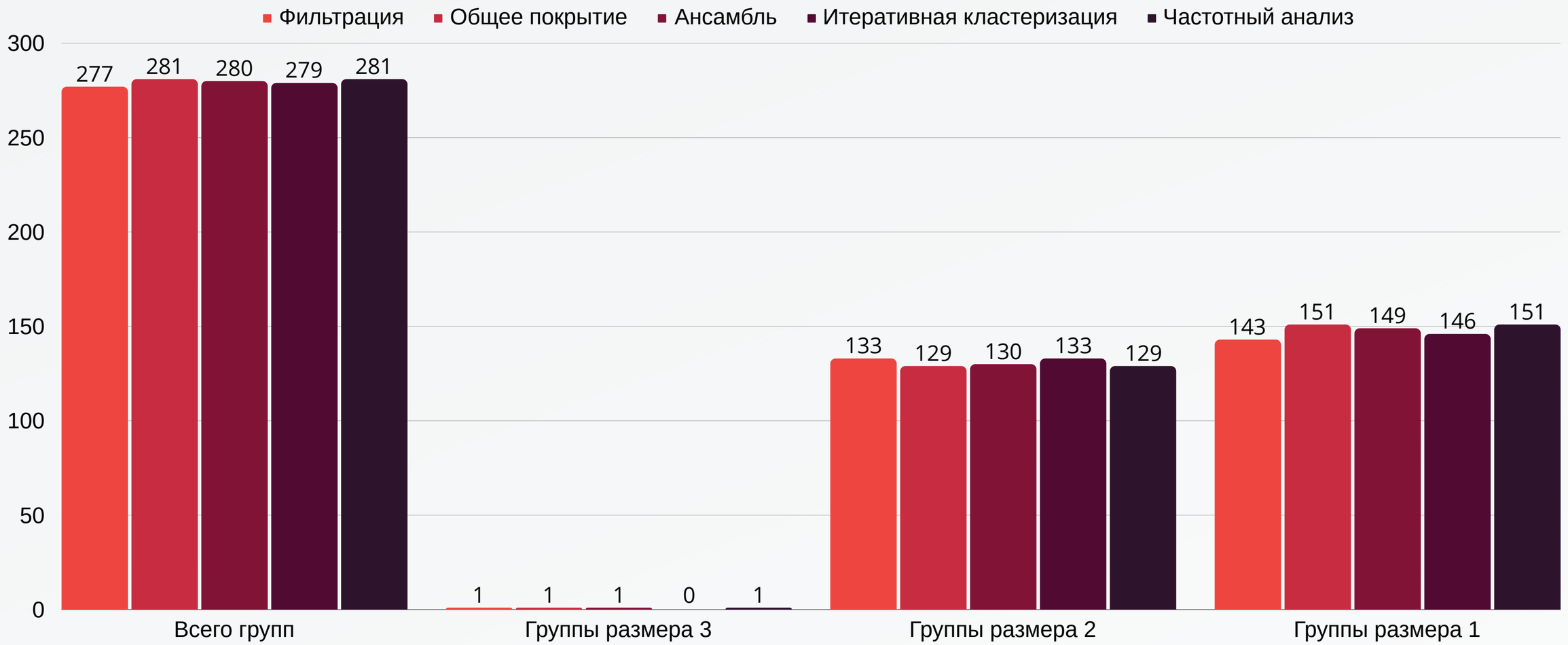
ОПИСАНИЕ

РЕШЕНИЕ ОСНОВАНО НА ИСПРАВЛЕНИИ ВСЕХ ОПЕЧАТОК В ДАТАСЕТЕ ЗА СЧЕТ ПОДБОРА БЛИЖАЙШЕГО ИЗВЕСТНОГО КОРРЕКТНОГО СЛОВА В BK-ДЕРЕВЕ, А ТАКЖЕ ПРОВЕРКЕ СМЫСЛОВЫХ ПАТТЕРНОВ, ТАКИХ КАК ПРОВЕРКА МЕСТОИМЕНИЙ И ОТРИЦАНИЙ.

ИСПОЛЬЗУЕМЫЕ АЛГОРИТМЫ

- РАССТОЯНИЕ ЛЕВЕНШТЕЙНА
- BK-ДЕРЕВЬЯ
- ОФОГРАФИЯ НА ОСНОВЕ ЧАСТОТЫ ПОЯВЛЕНИЯ СЛОВА
- ПРОВЕРКА НАБОРА СЛОВ
- ПРОВЕРКА НАБОРОВ МЕСТОИМЕНИЙ
- ПРОВЕРКА ОТРИЦАНИЯ

СРАВНЕНИЕ РЕЗУЛЬТАТОВ



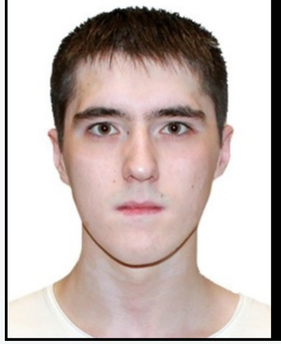
ВЫВОДЫ

1) Найдены 4 способа рерайта
2) Найдены 4 группы существующих решений
3) Сформировано 5 систем алгоритмов
4) Создан структурированный и наглядный репозиторий проекта

НАША КОМАНДА

Руденко
Юрий

JS / TS
React / Next
Frontend Developer



Галимуллин
Данис

C++ / Kotlin / Python
Data Scientist
Android Developer

Симоненко
Никита

C/C++
Rust enjoyer
System Programmer

Новиков
Михаил

C/C++ / Python
All enjoyer
Python Developer

Гусев
Иван

Java / Kotlin
C++ enjoyer
Backend Developer

Q&A