Автоматизация сверки и устранения дубликатов в персональных данных

Введение в проблематику

1. Что такое дубликаты?
2. Почему они возникают?
3. Какие сферы деятельности это затрагивает?
4. Зачем нужно устранять дубликаты?
5. Почему нельзя положиться только на номер паспорта?

Постановка задачи

1. Повышение качества данных
2. Разработка алгоритм поиска дубликатов
3. Применение на реальных данных
4. Оценка результатов алгоритма
5. Разработка прототипа библиотеки для интеграции решения в МИАЦ

Существующие решения

1. Сервис “Dadata.ru”
2. Сервис и библиотека “dedupe.io”
3. Сервис “Мастер адресов”
4. Решение “Индекс пациентов” от компании Нетрика

Обзор данных

	Фамилия	Имя	Отчество	Дата рождения
Всего значений	34406	34405	31995	34406
Уникальные значения	16304	1315	2006	16508
Пропущенные значения	0	1	2411	0

Табл.1. Статистика по данным.

Алгоритм поиска дубликатов

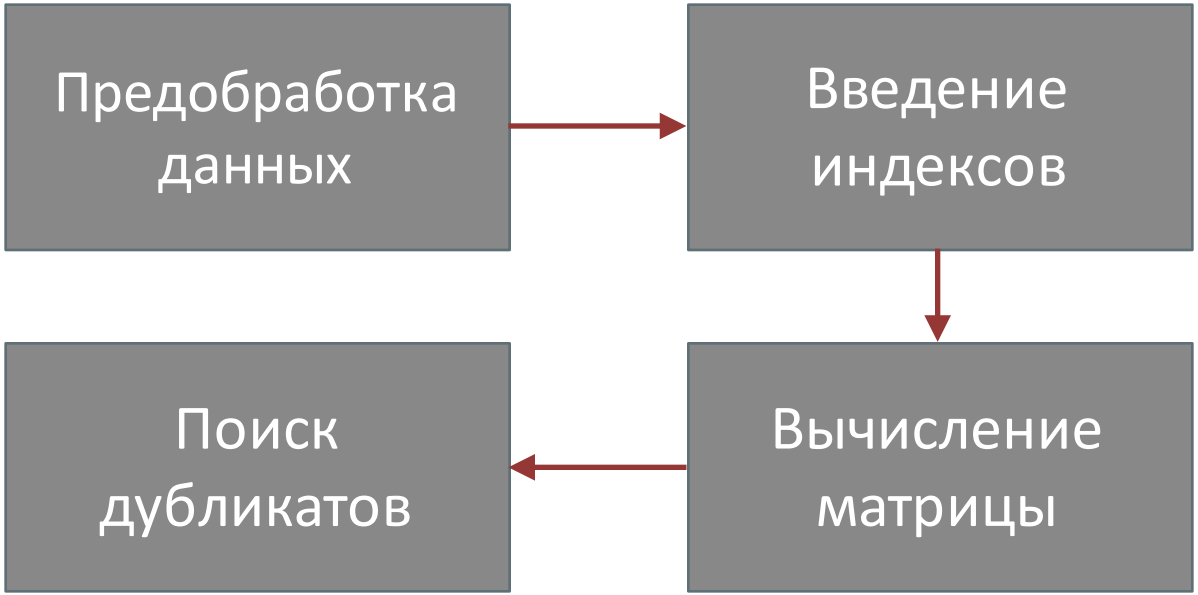


Рис.1. Схема алгоритма поиска дубликатов

Типы ошибок и методы их исправления

Типы ошибок

1. Ошибки на уровне поля
    a. Опечатки
    b. Различные стандарты написания
2. Ошибки на уровне записи
    a. Пропущенные значения
    b. Несоответствие поля и значения

Методы исправления

1. Регулярные выражения
2. Эвристические правила
3. Полуавтоматическое исправление

Алгоритм поиска дубликатов: индексирование

Задача: уменьшить количество рассматриваемых записей.

Решение: выделить группы потенциальных дубликатов.

Критерий: на основе подстроки или доли общих букв.

index — ассоциативный массив, в котором:

ключ — номер записи,

значение — список потенциальных дубликатов.

Алгоритм поиска дубликатов: вычисление матрицы

Матрица расстояний:

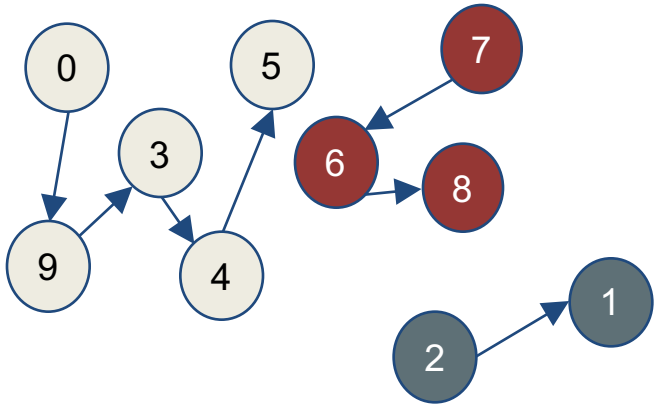
$$x^k_{ij} = dist(row_i[k], row_j[k]), \quad (1)$$

$$dist \text{ — редакционное расстояние}, \quad (2)$$

$$k \in [1, count_{fields}], \ i \in [1, count_{rows}], \ j \in index[i]. \quad (3)$$

Алгоритм поиска дубликатов

Рекурсивный алгоритм



Линейный алгоритм

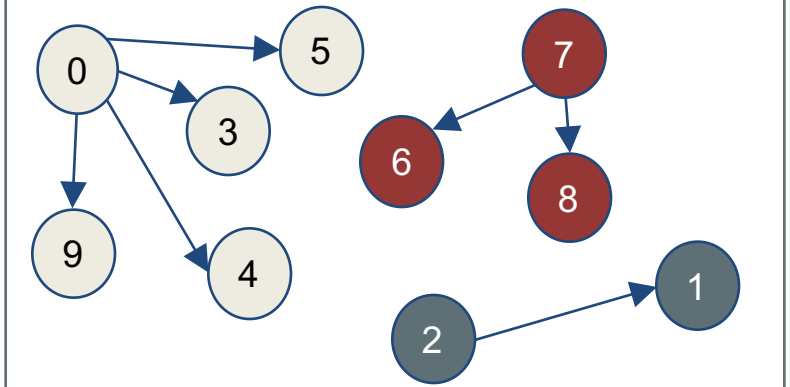


Рис.2. Варианты алгоритма поиска дубликатов

Устранение дубликатов: подходы

1. Оставить наиболее полную запись
2. Оставить самую свежую запись
3. Объединить все записи в одну
4. Предоставить выбор эксперту

Полученные результаты

	Тестовая выборка	Реальные данные
Всего записей	1000	34406
Уникальные записи	990	33987
Кластеры по 1 записи	981	33575
Кластеры по 2 записи	8	405
Кластеры по 3 записи	1	7
Точность	0.998	Не известно

Табл.2. Полученные результаты. Алгоритм поиска: линейный.
Пороговые значения: 0.8, 0.8, 0.8, 0.9.

Выводы

- Решить задачу без участия человека нельзя.
- Идеи, на которых построен алгоритм, заслуживают дальнейшего развития.
- Разработан набор методов для очистки данных.
- Реализован алгоритм поиска дубликатов.
- Реализован прототип библиотеки для применения решения в сторонних проектах.

Спасибо за внимание

Исходный код: https://github.com/KirovVerst/record_linkage