П. А. Арсенович, В. В. Малайчук

СРАВНИТЕЛЬНЫЙ АНАЛИЗ АЛГОРИТМОВ НЕЧЕТКОГО ПОИСКА ДЛЯ ОЧИСТКИ ДАННЫХ ОТ ДУБЛИКАТОВ

Проводится сравнительный анализ алгоритмов нечеткого поиска, применяемых для задачи очистки данных и выявления скрытых дубликатов. Рассмотрены метрические методы сопоставления строк: классическое расстояние Левенштейна, алгоритм Джаро-Винклера и комбинированный подход Token Sort.

Введение

В больших бизнес-системах данные поступают из множества источников (операционные базы, CRM, ERP, внешние данные и т. д.), что нередко приводит к проблемам качества данных. Наличие дубликатов, опечаток и несоответствий форматов в базах данных приводит к искажению аналитики, снижению эффективности маркетинговых кампаний и прямым финансовым потерям.

Согласно исследованиям в области управления информационными ресурсами, «грязные» данные могут приводить к потере до 15–25% операционной прибыли компаний. Проблема избыточности и противоречивости записей особенно остро проявляется в задачах формирования единого профиля клиента, где дублирование информации препятствует получению достоверной статистической отчетности. В условиях цифровизации бизнес-процессов автоматизация интеллектуального поиска и объединения схожих объектов становится критически важным этапом построения надежных хранилищ данных.

При анализе качества данных принято выделять явные и неявные дубликаты. Явные дубликаты представляют собой точные копии строк в таблицах. Проблема поиска и устранения таких дубликатов уже решена на уровне систем управления базами данных и легко устраняется встроенными средствами SQL (например, с использованием оконных функций или группировок) [1].

Основную же исследовательскую сложность представляют именно неявные дубликаты. Задача выявления и объединения записей, относящихся к одной и той же сущности, осложняется человеческим фактором: опечатками при вводе, перестановкой слов и использованием различных регистров [2, 5].

I. Теоретические сведения

Для решения задачи сопоставления строк в работе рассматриваются три популярных метрических алгоритма:

– Расстояние Левенштейна. Классическая метрика, определяющая минимальное количество односимвольных операций (вставка, удаление, замена), необходимых для превращения одной строки в другую.

– Сходство Джаро-Винклера. Модификация метрики Джаро, которая предназначена преимущественно для коротких строк (например, имен собственных). Алгоритм дает больший вес строкам, у которых совпадает начальная часть (префикс) [3].

– Метод Token Sort. Комбинированный подход, реализующий многоэтапную нормализацию данных. Перед вычислением редакционного расстояния выполняется предварительная обработка: приведение строки к нижнему регистру и удаление знаков препинания. Затем строка разбивается на отдельные слова (токены), которые сортируются в алфавитном порядке и объединяются обратно в единую последовательность. Итоговый коэффициент сходства рассчитывается на базе расстояния Левенштейна между нормализованными строками, что позволяет эффективно нивелировать проблему перестановки слов местами (например, при сравнении ФИО или адресов).

Для оценки качества работы алгоритмов используются метрики матрицы ошибок:

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F_1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$$

где TP – истинно-положительные решения, FP – ложно-положительные решения (ошибочное объединение разных записей), FN – ложно-отрицательные решения (пропущенный дубликат).

II. Практическая часть

Для проведения эксперимента использовался язык программирования Python и библиотеки `pandas`, `rapidfuzz` и `jellyfish`.

В качестве исходного набора данных был взят публичный датасет пассажиров Титаника. Из него были извлечены 1000 уникальных имен. Для симуляции реальных условий ввода информации была применена функция-генератор, которая создала 300 искусственных дубликатов, внеся в них случайные искажения. Итоговый размер датасета составил 1300 записей.

Для проведения попарного сравнения методами нечеткого поиска были сформированы все возможные уникальные комбинации записей (8385 пар). Порог отсечения для всех алгоритмов был установлен на уровне 80 из 100. Выбор такого порога обусловлен эмпирическим балансом между отсевом явного шума и сохранением потенциальных дубликатов

III. Результаты и сравнение

В ходе тестирования фиксировались время выполнения каждого алгоритма, а также количество найденных и пропущенных дубликатов. Итоговые результаты представлены в таблице 1.

Таблица 1. Сравнение алгоритмов

Метрика	Левенштейн	Джаро-Винклер	Token Sort
Время (с)	0.8806	3.8534	**1.2214**
Найдено (TP)	260	278	**292**
Ошибок (FP)	250	4015	**163**
Пропущено (FN)	40	22	**8**
Precision	0.510	0.065	**0.642**
Recall	0.867	0.927	**0.973**
F1-Score	0.642	0.121	**0.774**

Анализ полученных данных показывает, что классическое расстояние Левенштейна отличается самой высокой скоростью выполнения (0.8806 с), однако обеспечивает весьма посредственный баланс метрик ($F_1 = 0.642$) и пропускает значительное количество дублирующих записей ($FN = 40$).

Аномально высокое количество ложных срабатываний ($FP = 4015$) у алгоритма Джаро-Винклера объясняется спецификой метрики: она отдает приоритет совпадению префиксов. В контексте имен собственных алгоритм завышает коэффициент сходства из-за идентичных начальных букв, что при установленном пороге в 80 приводит к ошибочному признанию разных людей дубликатами. Для задач очистки ФИО данная метрика в чистом виде оказалась неприменимой.

Наилучшие показатели в ходе тестирования продемонстрировал алгоритм Token Sort ($F_1 = 0.774$). За счет этапов предварительной токенизации и алфавитной сортировки слов данный подход успешно выявил подавляющее большинство скрытых дубликатов (пропущено всего 8 записей), существенно снизив количество ошибок по сравнению с другими методами. При этом скорость обработки (1.2214 с) осталась на приемлемом уровне, сопоставимом с базовой метрикой Левенштейна.

IV. Выводы

В ходе работы было доказано, что выбор алгоритма нечеткого поиска напрямую влияет на качество очистки данных. Использование метрики Джаро-Винклера для дедупликации записей, содержащих перестановки слов (например, ФИО), приводит к неприемлемому росту ложных срабатываний и падению производительности. Классическое расстояние Левенштейна работает быстро, но обладает низкой точностью на зашумленных данных. Комбинированные подходы, такие как Token Sort, являются наиболее оптимальным выбором: они эффективно справляются с перестановками и опечатками, обеспечивая лучший баланс между полнотой и точностью при адекватных затратах вычислительного времени.

В качестве перспективного направления развития данной работы рассматривается переход от чисто метрических методов к семантическим. Использование векторных представлений слов (Word Embeddings) и предобученных трансформеров (BERT) позволит выявлять дубликаты даже при использовании синонимов или разных языковых раскладок, что недоступно классическим алгоритмам редактирования строк.

Список литературы

1. Макаров А. В., Намиот Д. Е. Обзор методов очистки данных для машинного обучения // International Journal of Open Information Technologies. – 2023. – № 10.
2. Remove duplicate rows from a SQL Server table by using a script // Microsoft Learn : [сайт]. – 2025. – URL: https://learn.microsoft.com (дата обращения: 27.03.2026).
3. Winkler W. E. String Comparator Metrics and Enhanced Decision Rules in the Fellegi-Sunter Model of Record Linkage // Proceedings of the Section on Survey Research Methods. – American Statistical Association, 1990. – P. 354–359.
4. Рубцов, Д. Н., Барахнин, В. Б. (2009). Выявление дубликатов в разнородных библиографических источниках. Вестник Новосибирского государственного университета. Серия : Информационные технологии, 7 (3), 86-93.
5. Пыхтеев С. Д., Глухих И. Н. Программный комплекс для поиска неявных дубликатов в массивах текстовых данных // Вестник кибернетики. – 2021. – № 2 (42). – С. 45–52.

Арсенович Павел Александрович, студент кафедры информационных технологий автоматизированных систем БГУИР, arsenpashamail@gmail.com.

Малайчук Валерий Викторович, студент кафедры информационных технологий автоматизированных систем БГУИР, mvvbor@gmail.com.

Научный руководитель: Трофимович Алексей Фёдорович, заместитель декана по идеологической и воспитательной работе ФИТиУ, старший преподаватель кафедры информационных технологий автоматихированных систем, trofimaf@bsuir.by.