

З Чеботарева, З. В. Система управления дебиторской задолженностью на предприятии железнодорожного транспорта / З. В. Чеботарева, О. А. Яковлева // Актуальные проблемы гуманитарных и естественных наук. – 2017. – № 4-2. – URL: <https://cyberleninka.ru/article/n/sistema-upravleniya-debitorskoy-zadolzhennostyu-na-predpriyatii-zheleznodorozhnogo-transporta/viewer> (дата обращения: 01.02.2026).

Получено 31.05.2026

ISSN 2227-1155. Сборник студенческих научных работ.

Вып. 31. Гомель, 2026

УДК 004.94

С. И. РОДИОНОВ (СИ-21)

Научный руководитель – ст. преп. В. В. РОМАНЕНКО

ПОСТРОЕНИЕ СИСТЕМЫ ПОИСКА ДУБЛИКАТОВ И ПОХОЖИХ ИЗОБРАЖЕНИЙ НА ОСНОВЕ MINHASH И LSH

Описан практический метод построения системы быстрого поиска похожих изображений с помощью MinHash и LSH. Рассмотрены извлечение признаков, построение сигнатур и формирование ключей для хэш-таблиц, что позволяет находить кандидатов без полного перебора. Приведены пояснения поиска дубликатов и похожих изображений.

В задачах анализа больших коллекций геоснимков часто требуется находить похожие снимки, но прямое попарное сравнение имеет квадратичную сложность. Эффективное решение дает сочетание MinHash и LSH.

Выделение признаков документа

Документ представляется в виде множества признаков. Признаками могут быть идентификаторы слов, n -граммы символов или любые другие характеристики, которые можно сравнить между собой

Пример. Рассмотрим два коротких текста:

- документ А: «мама мыла раму»;
- документ В: «мама мыла».

Если в качестве признаков взять целочисленные идентификаторы слов (например, «мама» $\rightarrow 1$, «мыла» $\rightarrow 2$, «раму» $\rightarrow 3$), то получим:

- множество $A = \{1, 2, 3\}$;
- множество $B = \{1, 2\}$.

Для оценки похожести двух множеств используется **коэффициент Жаккара** (Jaccard similarity):

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|}.$$

В данном случае пересечение $\{1, 2\}$ имеет мощность 2, объединение $\{1, 2, 3\}$ – мощность 3, поэтому $J = 2 / 3 \approx 0,667$.

Чем больше коэффициент, тем более похожи документы. На практике вместо слов часто используют n -граммы (последовательности из n символов), чтобы учитывать частичные совпадения и орфографические вариации.

На рисунке 1 показано: слева – исходная фраза, справа – соответствующее множество целых чисел.

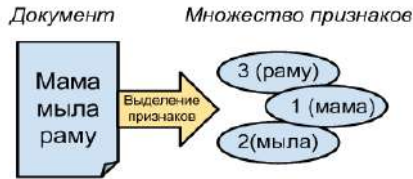


Рисунок 1 – Пример преобразования документа в множество признаков (идентификаторы слов)

Создание сигнатуры с помощью MinHash

Прямое вычисление коэффициента Жаккара для всех пар множеств требует перебора. MinHash оценивает сходство через компактную сигнатуру фиксированной длины.

Определение MinHash. Для множества S и семейства хэш-функций $h_1, \dots, h_k$ определим:

$$\text{minhash}_i(S) = \min_{x \in S} h_i(x).$$

Вектор из таких минимальных значений длины k называется MinHash-сигнатурой множества S .

Важное свойство – вероятность того, что минхэши двух множеств совпадут для случайной выбранной хэш-функции, в точности равна коэффициенту Жаккара этих множеств:

$$P(\text{MinHash}(S_1) = \text{MinHash}(S_2)) = J(S_1, S_2).$$

Данное свойство значительно упрощает оценку сходства.

Для повышения точности берут не одну, а множество различных хэш-функций. Практически их легко получить, генерируя случайные числа a_i, b_i и используя линейное преобразование.

Пример с тремя хэш-функциями. Пусть для чисел $\{1, 2, 3\}$ и $\{1, 2\}$ вычислены значения:

- 1 минхэш для обоих множеств = 1078;
- 2 минхэша = 1674 (совпадает);
- 3 минхэша для первого = 145, для второго = 4437 (разные).

Из трех минхэшей совпали два $\rightarrow$ оценка сходства = $2/3$, что совпадает с истинным коэффициентом Жаккара. Чем больше хэш-функций (длиннее сигнатура), тем точнее оценка.

На рисунке 2 показана сигнатура из четырех случайных чисел для множества $\{1, 2, 3\}$.

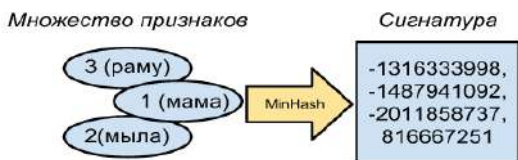


Рисунок 2 – Сигнатура документа как набор минхэшей

Иллюстрируется вычисление трех минхэшей для двух документов и их попарное сравнение.

Создание ключей с помощью Locality Sensitive Hashing (LSH)

Наличие MinHash-сигнатур для всех документов не устраняет необходимость попарного сравнения, так как для поиска документов, похожих на запрос, требуется сравнить его сигнатуру с сигнатурами всех остальных. LSH решает эту проблему, группируя похожие сигнатуры в корзины (buckets) хэш-таблицы.

Если использовать полную сигнатуру (например, из 100 чисел) в качестве ключа, то в одну корзину попадут только документы с абсолютно идентичными сигнатурами, что соответствует практически полному совпадению (вероятность такого события равна s^k , где s – сходство). Это слишком жесткое условие.

Сигнатура длины N разбивается на b полос по r чисел ($N = b \cdot r$). Для каждой полосы вычисляется хэш-ключ, который добавляется в хэш-таблицу. При поиске сигнатура запроса разбивается аналогично, и кандидатами считаются документы, у которых совпала хотя бы одна полоса [4, 5]. Вероятность такого совпадения для документов со сходством s выражается формулой

$$P_{cand}(s) = 1 - (1 - s^r)^b,$$

где P – коэффициент вероятности совпадения; s – коэффициент Жаккара (истинная похожесть); r – длина полосы (количество минхэшей в одной полосе); b – число полос.

Пример. Пусть сигнатура состоит из 100 минхэшей, разбитых на 10 полос по 10 элементов в каждой. Тогда $b = 10, r = 10$. Вероятность выборки в зависимости от s приведена в таблице 1.

На рисунке 3 показано, как сигнатура из четырех чисел (на примере) разбивается на две полосы, и для каждой полосы вычисляется свой ключ для хэш-таблицы.

Таблица 1 – Вероятность того, что документ будет выбран как кандидат

Истинное сходство (s)	0,0	0,25	0,50	0,75	1,0
Вероятность выборки	0,0	~0	~0	~0,94	1,0

Точные значения $P = 1 - (1 - s^{10})^{10}$. Например, при $s = 0,75$: $0,75^{10} \approx 0,056$; $1 - 0,056 = 0,944$; $0,944^{10} \approx 0,56$, тогда $P \approx 0,44$. Но в исходной статье указана аппроксимация, дающая близкие результаты. Для практики важно, что порог можно регулировать, выбирая b и r .

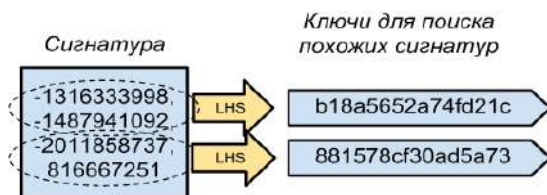


Рисунок 3 – Преобразование сигнатуры в ключи LSH (разбиение на полосы)

График на рисунке 4 демонстрирует резкий переход в районе $s \approx 0,8$.



Рисунок 4 – График вероятности выборки кандидата от истинной похожести s (для $b = 10, r = 10$)

Практический алгоритм поиска

Индексация базы документов:

- для каждого документа выделить множество признаков (например, идентификаторы слов или n -грамм).
- вычислить MinHash-сигнатуру длины N (например, 100) с помощью N различных хэш-функций.

Поиск похожих на запрос:

- для запросного документа вычислить его сигнатуру и разбить на полосы (те же параметры);
- для каждой полосы взять соответствующий ключ и извлечь все идентификаторы документов из соответствующей корзины – это **кандидаты**;
- удалить дубликаты идентификаторов.

Достоинства метода:

- линейная сложность индексации $O(N \cdot |S|)$ для каждого документа;
- поиск выполняется за время, пропорциональное числу полос и среднему размеру корзины;
- возможность контроля компромисса между скоростью и точностью через параметры b и r .

На простых примерах объяснены MinHash и LSH для поиска похожих документов: от признаков до сигнатуры и индексации без попарного перебора. Метод применяется в системах обнаружения дубликатов, плагиата и рекомендациях. Для углубленного изучения рекомендованы классические работы [1–3] и библиотеки типа datasketch.

СПИСОК ЛИТЕРАТУРЫ

1 **Бродер, А. З.** О сходстве и содержании документов / А. З. Бродер // Сжатие и сложность последовательностей : материалы конф. VLDB. – Эдинбург, 1997. – С. 21–29.

2 **Гионис, А.** Поиск сходства в высоких размерностях с помощью хеширования / А. Гионис, П. Индик, Р. Мотвани // 25-я конф. по очень большим базам данных (VLDB) : материалы конф. VLDB. – Эдинбург, 1999. – С. 518–529.

3 Поиск похожих документов с помощью MinHash и LSH // Habr. – URL: <https://habr.com/ru/articles/250673/> (дата обращения: 11.05.2026).

4 **Хан, А.** LSHBloom: эффективная по памяти дедупликация документов экстремального масштаба / А. Хан, Р. Андервуд, К. Сибеншу // arXiv. – URL: <https://arxiv.org/abs/2411.04257> (дата обращения: 11.05.2026).

5 **Индик, П.** Приближённый поиск ближайших соседей: преодоление «проклятия размерности» / П. Индик, Р. Мотвани // 30-й ежегод. симп. ACM по теории вычислений (STOC). – Даллас, 1998. – С. 604–613.

Получено 30.07.2026