Тематический раздел: КИБЕРНЕТИКА

УДК 004.032.26+004.422.632.2+004.424.4

Автор: Соколов А.М.

Векторные представления для эффективного сравнения и поиска сходных строк

Ключевые слова

Аппроксимация расстояния редактирования, метрические вложения, приближенный поиск ближайшего соседа, нейросетевые информационные технологии.

Keywords

Edit distance approximation, metric embeddings, approximate nearest neighbor search, neural information technologies

Аннотация

Предлагается способ аппроксимации классического расстояния редактирования между символьными строками, основанный на преобразовании строк в векторы, принадлежащие пространству с легковычислимой метрикой, которое сохраняет близость в исходном пространстве и вычисляется быстрее, чем время, за которое вычисляется исходное расстояние.

Разработаный #-граммный метод аппроксимации расстояния редактирования улучшает качество аппроксимации, по сравнению с описанными в литературе методами. На основе предложенного метода разработаны две его рандомизированные версии, которые могут использоваться для эффективного решения задачи приближенного поиска ближайших строк.

Анотація

Пропонується спосіб апроксимації класичної відстані редагування між символьними рядками, що базується на перетворенні рядків в вектори, які належать простору з метрикою, яка обчислюється легко, і яке зберігає близкість у вихідному просторі й обчислюється швидше, ніж час, за який обчислюється вихідна відстань.

Розроблений #-граммний метод апроксимації відстані редагування покращує якість апроксимації, порівняно з описаними в літературі методами. На основі запропонованого методу розроблено дві його рандомізовані версії, які можуть використовуватися для ефективного вирішення задачі приблизного пошуку найближчих рядків.

Abstract

We propose a method of the classic string edit distance approximation, which is based on a mapping of strings to vector space endowed with an easy-calculable metrics. The mapping preserves similarity in the sense of the input metrics, and new distances can be calculated faster the time, needed to calculated distance between input strings.

The presented #-gram method of approximating edit distance improves the approximation quality, comparing to known results. Based on the method we present its two randomized versions, which can be used for an effective approximate nearest neighbor search.

Введение

Классическое расстояние редактирования $ed(x, y)$ между символьными строками x, y (расстояние Левенштайна [1] – минимальное количество операций вставки, замены и удаления символов для преобразования x в y) применяется во многих областях – от генетики и веб-поиска до распознавания речи. Современные объемы строковых данных и их продолжающийся экспоненциальный рост [2] приводят к тому, что классический $O(n^2)$-алгоритм [3, 4] вычисления этого расстояния между строками длины n часто практически не применим. Это обстоятельство породило новые области исследований, связанные с ускорением вычисления или аппроксимации расстояния редактирования [5], и использованием методов теории метрических вложений [6] для преобразования объектов в пространства, где упрощается вычисление сложных исходных метрик.

Одной из актуальных задач аппроксимации расстояния редактирования является поиск вложения метрики Левенштайна в векторное пространство [7]. Существующие методы аппроксимации расстояния Левенштайна и поиска сходных строк в больших масивах требуют усовершенствования для повышения вычислительной эффективности, снижения требования к ресурсам, повышения точности аппроксимации.

Для этого в статье развивается метод вложения строк в векторное пространство, основанный на применении нейросетевых распределенных представлений [8, 9]. Мы предлагаем детерминированный (раздел 3) метод вложения расстояния редактирования в манхетенново пространство, а также его рандомизированные версии для поиска ближайшего соседа (раздел 4).

1 Определения и вспомогательные утверждения

Будем обозначать алфавит символов Σ, а множество строк длины $n \in \mathbb{N}$, заданных над Σ, как Σ^n. Символ, находящийся в позиции i строки x, будем обозначать $x[i]$. *Подстроку* $x[i]x[i+1]\dots x[j]$ строки x будем обозначать $x[i, j]$ и писать $x[i, j] \subseteq x$, а диапазоны позиций вида $[i, j]$

назовем *интервалами*. Подстроки вида $x[i, i+q-1], q \in \mathbb{N}$ называются *q-граммами*, а множество всех q-грамм строки – ее *q-спектром*. Длину строки x обозначим $|x|$. Для $x \in \Sigma^n$ и $q \in \mathbb{N}$ *q-граммным вектором* строки будем называть вектор $v_{n,q}(x) \in (\mathbb{N} \cup \{0\})^{|\Sigma|}$, где каждой q-грамме $\sigma \in \Sigma^q$ соответствует элемент вектора $(v_{n,q}(x))_\sigma \in \mathbb{N} \cup \{0\}$, равный числу раз, которое σ встретилась в x: $(v_{n,q}(x))_\sigma = \sum_{i=1}^{n-q+1} [x[i, i+q-1] = \sigma]$. Когда это не вызывает неоднозначностей, будем вместо $v_{n,q}(x)$ писать $v_q(x)$ или просто $v(x)$. Для пары строк $x, y \in \Sigma^n$ *q-граммным расстоянием* $d_q(x, y)$, называется манхеттеново расстояние (l_1-расстояние) между соответствующими q-граммными векторами $||v_q(x) - v_q(y)||_{l_1} = \sum_{\sigma \in \Sigma^q} |(v_q(x))_\sigma - (v_q(y))_\sigma|$. Последовательность операций редактирования, которые трансформируют строку x в y, будем называть *восстановлением* y по x.

Графом де Брейна [10, 11] $B(\Sigma; q)$ для алфавита Σ и параметра $q \geq 3$ называется направленный граф $G(V, E)$, множеству вершин V которого соответствуют все $(q-1)$-граммы, а множеству дуг $E \subset V \times V$ – все q-граммы в данном алфавите. Соответствующие дугам и вершинам q- и $(q-1)$-граммы назовем *метками*. При этом для символов $l_i \in \Sigma$ дуга с меткой $l_1 l_2 \dots l_q$ соединяет вершины с метками $l_1 l_2 \dots l_{q-1}$ и $l_2 l_3 \dots l_q$. Каждой строке $x, |x| \geq q$ соответствует определенный *путь* π_x на графе де Брейна, состоящий из дуг, последовательно соединяющих вершины, метки которых есть последовательно входящие в строку $(q-1)$-граммы. *Подпутем* π'_x пути π_x будем называть связную последовательность дуг π_x, соответствующую подстроке $x' \subseteq x$ и обозначать как $\pi'_x \subseteq \pi_x$. Путь π_x, состоящий из двух подпутей π'_x и π''_x будем обозначать как $\pi_x = \pi'_x \pi''_x$.

Для $x, |x| \geq q$ обозначим $B[x; q]$ подграф графа де Брейна, вершинами которого являются все $(q-1)$-граммы строки x (элементы $(q-1)$-спектра x), а дуги соответствуют ее q-граммам. Обозначим граф, построенный на объединении $(q-1)$-спектров двух строк $x, y \in \Sigma^w$ одинаковой длины w, как $B[x, y; q]$.

Строка $x, |x| \geq w$ называется (q, w)-*неповторяющейся*, если в любом интервале из w символов все $w - q + 1$ штук q-грамм различны. Строку $x \in \Sigma^w$, являющуюся (q, w)-неповторяющейся,

будем называть просто *q-неповторяющейся*.

Рассмотрим два случая:

Случай I. обе строки $x, y \in \Sigma^w$ являются q-неповторяющимися и

Случай II. они ими не являются.

Сначала рассмотрим случай I. Если x и y содержат общую q-грамму $l_1 \dots l_q$, то соответствующие им пути π_x и π_y на $B[x, y; q]$ будут проходить через одну и ту же дугу, соединяющую вершины с метками $l_1 \dots l_{q-1}$ и $l_2 \dots l_q$. *Левой* (вида $- <$) и *правой* (вида $> -$) *точками ветвления* назовем вершины, где пути π_x и π_y, соответственно, расходятся после общей дуги или сходятся перед общей дугой. В этом случае в $B[x, y; q], x \neq y$ можно выделить смежные участки вида $> - <, - <, > -$, т.е. участки, содержащие как минимум одну общую дугу обоих путей π_x, π_y, и ограниченные с одной или двух строн точкой ветвления.

Введем для случая I понятия (полу)петли, (полу)вилки и сдвига. Для каждого из двух путей совокупность левой точки ветвления, ближайшей к ней в порядке прохождения дуг правой точки ветвления, и дуг этого пути между ними (вида $< \_ >$) назовем *полупетлей*. Каждой полупетле соответствует также полупетля (возможно, нулевой длины), состоящая из дуг второго пути и соединяющая те же точки ветвления (тогда вместе эти полупетли имеют вид $<=>$) или же, для конфигураций вида $> - <$, если левая и правая точки двух таких смежных конфигураций соединены полупетлей вида $> - < \_ > - <$, то соединяющая оставшиеся последнюю левую и следующую за ней, первую, правую точки ветвления. Такую пару соответствующих друг другу полупетель назовем *петлей*. По данному определению, как минимум одна из полупетель петли имеет ненулевою длину.

Граф $B[x, y; q]$ назовем *вилкой*, если существует такой общий для π_x, π_y подпуть $\pi_c, w > |\pi_c| > 0$, что $\pi_x = \pi'_x \pi_c \pi''_x$, а $\pi_y = \pi'_y \pi_c \pi''_y$, где для любых дуг $e \in \pi'_x \cup \pi''_x, e' \in \pi'_y \cup \pi''_y$, выполняется $e \neq e'$. Поскольку $|x| = |y|$, то $|\pi'_x| + |\pi''_x| = |\pi'_y| + |\pi''_y|$. Подпути $\pi'_x, \pi''_x, \pi'_y \pi''_y$ левой или правой вилок, составленые из дуг одной строки, будем называть *полувилками*. *Сдвигом* назовем граф, являющийся частным случаем графа-вилки, где существует такой общий для π_x и π_y подпуть

$\pi_c, w \geq |\pi_c| > 0$, что $\pi_x = \pi'_x\pi_c$, а $\pi_y = \pi_c\pi'_y$, где $\forall e \in \pi'_x, e' \in \pi'_y,\ e \neq e'$. Подпути π'_x и π'_y в этом случае будем также называть полувилками. Если $B[x, y; q]$ является сдвигом, то будем говорить, что x является сдвигом y и наоборот.

Справедливы следующие утверждения о зависимости числа общих дуг при наличии описанных конфигураций.

Утверждение 1. *Если $B[x, y; q]$ является вилкой (сдвигом), то $B[x, y; q+1]$ также является вилкой (сдвигом) с тем же количеством дуг в полувилках.*

Утверждение 2. *Если $B[x, y; q], x, y \in \Sigma^n$ является вилкой с длиной общей части $|\pi_c|$, причем правая вилка состоит из 2 полувилок длиной $|\pi'_x| = s_1$ и $|\pi'_y| = s_2$, а левая – длиной $|\pi''_x| = s_3$ и $|\pi''_y| = s_4$, то для преобразования x в y достаточно выполнить не более $\min(s_1, s_2) + \min(s_3, s_4)$ операций замены и $|s_2 - s_1| + |s_3 - s_4|$ операций вставки или удаления. Так как $s_3 = n - |\pi_c| - s_1, s_4 = n - |\pi_c| - s_2$, то суммарно $ed(x, y) \leq \max(s_1, s_2) + \max(s_3, s_4)$. Заметим, что верхняя оценка также справедлива для x, y, граф которых не является вилкой, но π_x и π_y имеют некоторую общую часть π_c, а π'_x и π'_y (или π''_x и π''_y) могут частично совпадать.*

Случай II. Когда строки не являются (q, w)-неповторяющимися, у одной или обеих подстрок существует хотя бы один подпуть $\pi'_x \subseteq \pi_x$ ($\pi'_y \subseteq \pi_y$) на графе $B[x; q]$ ($B[y; q]$), соответствующий подстроке $x' \subseteq x$ ($y' \subseteq y$), имеющий самопересечение и потому являющийся циклом. Такой цикл имеет хотя бы одну вершину, встречающуюся более одного раза и, поэтому, как минимум одну повторяющуюся $(q-1)$-грамму. Заметим, что один и тот же цикл может соответствовать разным подстрокам x' (y') – для этого достаточно начать прохождение цикла с другой вершины.

Следующая лемма утверждает, что при достаточно большом q на графе $B[x; q]$, где x – (q, w)-неповторяющаяся строка, имеется не более, чем один цикл.

Лемма 1. *Для $x \in \Sigma^w$ при $q > 2w/3$, если существует подстрока $x' \subseteq x$, такая, что $\pi_{x'}$ образует цикл C на $B[x; q]$, то одинаковые дуги вне цикла C отсутствуют.*

Доказательство. Пусть $f = |x'|$ и подстроки $x'' = x[i, i + q - 2]$, $x''' = x[j, j + q - 2], i < j$ являются парой максимально удаленных среди совпадающих $(q-1)$-грамм в $\pi_{x'}$. Поскольку

$q > 2w/3$, то эти $(q-1)$-граммы пересекаются в как минимум $2\lceil 2w/3 \rceil - w$ символах и символы подстроки $x[i, j-1]$ будут периодически повторяться подряд в x': если обозначить символы подстроки $x[i, j-1]$ как $b_1 b_2 \dots b_B$, то $x'[k] = b_{((k-1) \mod B)+1}, k = 1, \dots, f$.

Допустим, совпали две $(q-1)$-граммы $x[i', i'+q-2] = x[j', j'+q-2]$, $j' > i' > j$, т.е. $(q-1)$-граммы, выходящие своим правым концом за правую границу x'''. Поскольку $q > 2w/3$, то эти q-граммы полностью покрывают общий с x'', x''' участок, содержащий как минимум одно вхождение всей последовательности символов $b_1, \dots, b_B$. А поскольку для подстроки $x[i', j'+q-2]$ справедливы аналогичные рассуждения про периодический характер (со своей повторяющейся последовательностью), то $\pi_{x[i,j+q-2]}$ и $\pi_{x[i',j'+q-2]}$ проходят те же дуги, что и цикл C. Аналогично рассматривается случай совпадения $(q-1)$-грамм до x''. □

Лемма 2. *Для $x, y \in \Sigma^w$ при $q > 2w/3$, если первая (последняя) вершина подпутей $\pi_{x'} \subseteq \pi_x, \pi_{y'} \subseteq \pi_y$, образующих общий цикл C, не совпадают, то общих дуг у π_x и π_y перед (после) этими вершинами нет.*

Доказательство. Рассмотрим случай, когда подпути $\pi_{x'}, \pi_{y'}$, состоящие из дуг цикла C, оканчиваются в разных вершинах цикла. Пусть $x[i, i+q-1]$ и $y[j, j+q-1]$ – q-граммы, соответствующие последним дугам указанных подпутей в цикле C. Допустим для $i' > i, j' > j$ совпали две q-граммы $\tilde{x} = x[i', i'+q-1] = y[j', j'+q-1] = \tilde{y}$. Так как $q > 2w/3$, то они содержат хотя бы одну последовательность символов $b_1, \dots, b_B$, описаную в доказательстве Леммы 1.

Пусть позиции правых концов $x[i, i+q-1]$ и $y[j, j+q-1]$ внутри, соответственно, $\tilde{x}$ и $\tilde{y}$ равны i'', j''. Поскольку подпути заканчиваются в разных вершинах, а $\tilde{x} = \tilde{y}$, то $i'' \neq j''$.

Допустим, $i'' < j''$. Из $\tilde{x} = \tilde{y}$ следует, что $\tilde{x}[i''+1, j''] = \tilde{y}[i''+1, j'']$, что эквивалентно тому, что $x[i+q, i+q-1+(j''-i'')] = y[j+q-(j''-i''), j+q-1]$. Поскольку и y', и x' составлены из периодически повторяющихся последовательностей символов $b_1, \dots, b_B$, то получаем, что, во-первых, x' заканчивается не q-граммой $x[i, i+q-1]$, а $x[i+(j''-i''), i+q-1+(j''-i'')]$, и, во-вторых, x' и y' заканчиваются в одной и той же вершине, что противоречит условию леммы. □

Следующее утверждение говорит о характере поведения путей на графах с циклами при изменении q.

Утверждение 3. *Если для $x \in \Sigma^w$ существует цикл на $B[x;q]$, $q > 2w/3$ такой, что в нем присутствуют повторяющиеся дуги, то с каждым увеличением q на единицу длина участка пути между этими вершинами, проходящего более одного раза по одним и тем же дугам, сокращается. При этом общее количество уникальных дуг в цикле не изменяется. Если повторяющихся дуг в цикле нет (т.е. каждая дуга цикла присутствует в π_x ровно один раз), то при следующем увеличении q на единицу цикл пропадает.*

Доказательство. По Лемме 1 цикл единственный. Утверждение можно проверить, проследив на графе $B[x;q]$ с одним циклом изменение количества дуг в цикле и вне его при последовательном увеличении q. □

2 Нижняя и верхняя границы на расстояние редактирования

Будем искать такое определение расстояния между строками x и y и порог на его значение, чтобы для расстояний меньше этого порога (и при выполнении не зависящих от строк условий на определенные величины) граф $B[x,y;q]$ для случая I содержал бы сдвиг или вилку, и не содержал бы полупетель. Это позволило бы восстановить строки за небольшое число операций (чем меньше, тем точнее будет аппроксимация расстояния редактирования), используя Утверждение 2. Не всякое расстояние удовлетворяет этим требованиям: например, при использовании обычного q-граммного расстояния (l_1-расстояние между q-грамными векторами и при фиксированном q для любого допустимого значения q существуют строки x, y, где на графе $B[x,y;q]$ имеется петля, и найти искомый порог не представляется возможным.

В случае I (q,w)-неповторяющихся строк для полупетель и вилок отличаются зависимостями числа различных q-грамм при увеличении q. Чтобы различать полупетли и вилки на этом осно-

вании, мы предлагаем использовать следующее расстояние, основанное на обычном q-граммном расстоянии

$$d^{\Sigma}_{w,q_1,q_2}(x,y) = \sum_{q=q_1}^{q_2} d_q(x,y), \tag{1}$$

определенное для строк одинаковой длины w и фиксированных значений длины q-грамм q_1, q_2.

Покажем, что с его помощью можно определить наличие вилки или сдвига, т.е. возможность восстановления пары строк за небольшое число операций редактирования. Случай II наличия повторов q-грамм потребует отдельного рассмотрения (Лемма 4).

Обозначим $\Delta q = q_2 - q_1$, $Q = (\Delta q + 1)(\Delta q + 2)$. Будем называть пару строк x, y *плохой*, если неравенство $d^{\Sigma}_{w,q_1,q_2}(x,y) < Q$ не выполняется, и *хорошей* в противном случае. Покажем, что для хорошей пары строк x, y граф $B[x,y;q_2]$ является вилкой (или сдвигом).

В следующей лемме мы рассмотрим ситуацию, когда для определенных значений q_1, q_2 ($q_2 > q_1$) и w строки x, y являются (q_1, w)-неповторяющимися и найдем необходимые условия существования петли на графе $B[x,y;q_2]$ (при этом пара x, y будет плохой). Тогда невыполнение этих условий будет достаточным условием наличия сдвига или вилки при отсутствии общих циклов.

Лемма 3. *Пусть $x, y \in \Sigma^w$ (q_1, w)-неповторяющиеся, и $w \geq q_2 > q_1 \geq 3$, $\Delta q \leq \lfloor \frac{w-q_1+1}{2} \rfloor$,*

$$Q \leq 4(w - q_2 + 1), \tag{2}$$

$$d^{\Sigma}_{w,q_1,q_2}(x,y) < Q, \tag{3}$$

тогда $ed(x,y) < 2(\Delta q + 1)$.

Доказательство. Для любой полупетли в графе $B[x,y;q]$, имеющей как минимум по одной общей дуге левее и правее, соответственно, левой и правой точек ветвления, увеличение q на единицу приводит к увеличению на единицу количества дуг в каждой из полупетель, участвующих в формировании петли от каждой строки, по сравнению с $B[x,y;q]$. Поскольку, по определению полупетель, в $B[x,y;q]$ дуги одной полупетли не совпадают с дугами из соответствующей ей полупетли другой строки, то совпадений дуг петли в $B[x,y;q+1]$ тем более не

будет, т.к. метки дуг в $B[x,y;q+1]$ содержат метки дуг $B[x,y;q]$ как подстроки: $x[i,i+q-1] \to x[i,i+q] = x[i,i+q-1]x[i+q]$. Поэтому для строк x,y, содержащих хотя бы одну петлю, окруженную общими дугами, выполняется $d_{q+1}(x,y) \geq d_q(x,y)+2$. Если же $B[x,y;q]$ является вилкой, то по Утверждению 1 при увеличении q на единицу они сохранятся и $d_q(x,y)$ не изменится. Общее число дуг при этом уменьшается на единицу.

Поскольку при значении Δq большем, чем половина количества имеющихся в графе $B[x,y;q_1]$ дуг, ни одна полупетля не сохраняется, для сохранения полупетель наложим дополнительное условие $\Delta q \leq \lfloor (w-q_1+1)/2 \rfloor$.

Учитывая, что минимальное q-граммное расстояние между различными строками равно 2, и при условии, что правая и левая точки ветвления петли не становятся, соответственно, первой или последней вершиной путей в $B[x,y;q]$, при $q < q_2$ (что обеспечивается условием (2)), получаем

$$d^{\Sigma}_{w,q_1,q_2}(x,y) \geq \sum_{q=q_1}^{q_2}(d_{w,q_1}(x,y)+2(q-q_1)) \geq 2(q_2-q_1+1)+(q_2-q_1)(q_2-q_1+1)$$

$$=(\Delta q+1)(\Delta q+2) = Q. \tag{4}$$

Для того, чтобы можно было утверждать, что при $d^{\Sigma}_{w,q_1,q_2}(x,y) < Q$ не возникает конфигурация вида $=$ (отсутствие совпадающих вершин в путях π_y и π_x) вместо конфигураций сдвига или вилки, должно выполняться условие $|\pi_c| \geq 1$, содержащееся в определении вилки и сдвига, т.е. наличие как минимум одной общей дуги для $q = q_2 - 1$: $d_{q_2-1}(x,y) \leq 2(w-(q_2-1)+1)-2 = 2(w-q_2+1)$. Если же $d_{q_2-1}(x,y) > 2(w-q_2+1)$ (т.е. нет ни одной общей дуги в $B[x,y;q_2-1]$), то $d^{\Sigma}_{w,q_1,q_2}(x,y) = \sum_{q=q_1}^{q_2-1} d_q(x,y) + d_{w,q_2}(x,y) > 4(w-q_2+1)$. Следовательно, если $d^{\Sigma}_{w,q_1,q_2}(x,y) \leq 4(w-q_2+1)$, то $d_{q_2-1}(x,y) \leq 2(w-q_2+1)$ и существует хотя бы одна общая вершина при $q = q_2$.

Таким образом, из наличия петель на $B[x,y;q]$ для $q = q_1,\ldots,q_2$ с необходимостью следует $d^{\Sigma}_{w,q_1,q_2}(x,y) \geq Q$. Но так как по условию (3) $d^{\Sigma}_{w,q_1,q_2}(x,y) < Q$, то следовательно, петель на $B[x,y;q_2]$ быть не может, т.е. $B[x,y;q_2]$ является вилкой (сдвигом) либо пути на $B[x,y;q_2]$ имеют конфигурацию вида $=$. Последнее устраняется условием (2), как показано выше. Остается

единственная возможность – $B[x,y;q_2]$ является вилкой (сдвигом).

По Утверждению 2 для восстановления вилки при фиксированном q необходимо затратить не более $\max(s_1,s_2)+\max(s_3,s_4) \le s_1+s_2+s_3+s_4 = d_q(x,y)$ операций редактирования. Поскольку заранее не известно, при каком значении $q = q_1,\ldots,q_2$ образовалась вилка, то можно лишь утверждать, что $ed(x,y) \le d_{q_2}(x,y)$. Далее, при q_2 петли не может быть, т.к. $d^{\Sigma}_{w,q_1,q_2}(x,y) < Q$. Поэтому обозначим как $q^*, q_1 \le q^* < q_2$ последнее значение q, при котором еще имеется петля. Тогда $d^{\Sigma}_{w,q_1,q^*}(x,y) \ge (q^*-q_1+1)(q^*-q_1+2)$ и $Q > d^{\Sigma}_{w,q_1,q_2}(x,y) = d^{\Sigma}_{w,q_1,q^*}(x,y)+(q_2-q^*)d_{q^*}(x,y) \ge (q^*-q_1+1)(q^*-q_1+2)+(q_2-q^*)d_{q_2}(x,y)$, т.е.

$$ed(x,y) \le d_{q_2}(x,y) < \frac{(Q-(q^*-q_1+1)(q^*-q_1+2))}{q_2-q^*} = q_2+q^*-2q_1+3 \le 2(\Delta q+1).$$

□

Рассмотрим теперь случай II наличия повторов q-грамм. Поскольку нас интересует зависимость количества общих дуг на $B[x,y;q]$ у двух путей π_x и π_y от q, то отметим, что от случая I будут отличаться только те ситуации наличия циклов, где совпадения дуг одного пути с дугами другого имеются для дуг, которые в хотя бы одном цикле встречаются больше одного раза и где специфицированное в Утвержении 3 уменьшение количества дуг не компенсирует увеличение $d_q(x,y)$.

Ограничимся поэтому рассмотрением случаев, когда цикл, содержащийся в пути π_x и цикл, содержащийся в π_y, одинаковы при $q = q_1$, но могут отличаться как длины подпутей $\pi_{x'} \subseteq \pi_x$ и $\pi_{y'} \subseteq \pi_y$, принадлежащих циклу, так и вершины входа и выхода из него.

Лемма 4. *Пусть при $q_1 > 2w/3$ для строк $x,y \in \Sigma^w$ существуют такие подстроки $x' \subseteq x, y' \subseteq y, x' \ne y'$, что на графе $B[x,y;q]$ пути $\pi_{x'}$ и $\pi_{y'}$ состоят из дуг, принадлежащих общему циклу C. Пусть также $d^{\Sigma}_{w,q_1,q_2}(x,y) \le Q$. Тогда $ed(x,y) < 2(\Delta q+1)$.*

Доказательство. Допустим от противного, что $ed(x,y) \ge 2(\Delta q+1)$. Покажем, что в этом случае $d^{\Sigma}_{w,q_1,q_2}(x,y) > Q$. Для этого найдем минимально возможное значение $d^{\Sigma}_{w,q_1,q_2}(x,y)$ при заданных условиях леммы. Поскольку характер изменения $d_q(x,y)$ при наличии циклов может быть весьма сложным и разнообразным, а получение этой зависимости в замкнутом виде

громоздко, для выяснения множества параметров, при которых может реализовываться минимум $d^{\Sigma}_{w,q_1,q_2}(x,y)$, воспользуемся следующими качественными соображениями, основанными на сравнении зависимостей значений $d_q(x,y)$ от характера прохождения цикла обоими путями π_x, π_y.

Пусть меньшее число дуг цикла C между вершинами, соответствующими начальным вершинам (первым (q_1-1)-граммам) путей $\pi_{x'} \subseteq \pi_x$ и $\pi_{y'} \subseteq \pi_y$, равно s. Пусть длина подпути $\pi_{x'}$, принадлежащего циклу C, есть L_x, и, аналогично, L_y – длина подпути $\pi_{y'}$, принадлежащего циклу C.

Поведение $d_q(x',y')$ в случае наличия одного цикла полностью определяется указанными параметрами s, L, L_x, L_y, пока $L_x > L$, $L_y > L$, поэтому можно рассматривать $d^{\Sigma}_{w,q_1,q_2}(x',y')$ как функцию от s, L, L_x, L_y. При увеличении длины q-граммы $d_q(x',y')$ изменяется вдоль линий постоянного значения величины $|L_x - L_y|$ с уменьшением L_x и L_y на единицу. Если $L_x < L$ и $L_y < L$, то циклы в обоих графах $B[x;q]$ и $B[y;q]$ пропадают, и дальнейшее поведение $d_q(x',y')$ при увеличении q описывается Леммой 3.

По Лемме 2 совпадения дуг разных путей вне цикла возможны, если совпадают первые или последние вершины путей $\pi_{x'}$ и $\pi_{y'}$. Вклад в $d_q(x,y)$ дуг путей вне цикла будет не меньше $|L_x - L_y|$ при $L_x \geq L, L_y \geq L$ (что реализуется при $s = 0$ или совпадении последних вершин $\pi_{x'}$ и $\pi_{y'}$). Поэтому $d_q(x,y) \geq d_q(x',y') + |L_x - L_y|$.

Можно показать что сумма вдоль такой линии ряда идущих подряд значений $d_q(x',y')$ для $s > 0$ будет не меньше, чем для $s = 0$ для этого же значения $|L_x - L_y|$ в том же диапазоне суммирования $q_1, \ldots, q_2$. Тогда, если $(s', L'_x, L'_y) = arg\min_{s,L_x,L_y} d^{\Sigma}_{w,q_1,q_2}(s, L_x, L_y)$, то $d^{\Sigma}_{w,q_1,q_2}(s', L'_x, L'_y) \geq d^{\Sigma}_{w,q_1,q_2}(0, L'_x, L'_y)$.

Рассмотрим поэтому только ситуацию $s = 0$, когда первые вершины подпутей $\pi_{x'}$ и $\pi_{y'}$ совпадают. Тривиальный случай, когда при этом также $|L_x - L_y| = 0$, рассматривать не будем, т.к. он сводится к отсутствию цикла.

Учитывая, что согласно Леммы 2 для рассматриваемых конфигураций можно применить

Утверждение 2, получаем, что для $|L_x - L_y| = \max(s_1, s_2) = \max(s_3, s_4)$ должно выполняться $|L_x - L_y| \geq \Delta q{+}1$. Иначе получим $ed(x, y) \leq \max(s_1, s_2){+}\max(s_3, s_4) < 2(\Delta q{+}1)$, что противоречит предположению в начале данного доказательства. Отсюда $d^{\Sigma}_{q_1,q_2,w}(x, y) \geq \sum_{q=q_1}^{q_2} 2|L_x - L_y| > 2(\Delta q + 1)^2 > (\Delta q + 1)(\Delta q + 2) = Q$. □

Объединим Лемму 4 с Леммой 3 и сформулируем Лемму 5.

Лемма 5. *Пусть* $x, y \in \Sigma^w$, $w \geq 6$, $w \geq q_2 > q_1 \geq 2w/3$ *и* $\Delta q \leq \lfloor (w{-}q_1{+}1)/2 \rfloor$,$Q \leq 4(w{-}q_2{+}1)$, $Q > d^{\Sigma}_{w,q_1,q_2}(x, y)$,*тогда* $ed(x, y) < 2(\Delta q + 1)$.

Доказательство. Из $\Delta q \leq \lfloor (w - q_1 + 1)/2 \rfloor$ и $q_1 \geq 2w/3$, следует $\Delta q \leq w/6$, поэтому значения Δq могут быть больше или равными единице при $w \geq 6$. Применяем в случае (q_1, w)-неповторяющихся строк x, y Лемму 3, а в противном случае – Лемму 4. □

Введем понятие *хорошего* и *плохого* интервала аналогично понятию хорошей или плохой пары строк. Хорошим интервалом назовем такой интервал вида $[i, j]$, что для пары ограничиваемых им в x и y строк $x[i, j]$ и $y[i, j]$ выполняется $d^{\Sigma}_{w,q_1,q_2}(x[i, j], y[i, j]) < Q$ (выражение (3) при условиях Леммы 5).

Далее рассмотрим интервал вида $[it' + 1, it' + w]$, где i – натуральное число из определенного диапазона, т.е. интервалы взятые с шагом t'. Следующая лемма говорит, что мы можем «пакетно» восстановить строки, содержащие последовательно идущие хорошие интервалы, для $t' < t$, где $t = w - \Delta q + 1$.

Лемма 6. *Пусть* $x, y \in \Sigma^m$, $t' \in \mathbb{N}$ *делит* $(m - w)$ *и* $t' < t$. *При выполнении условий Леммы 5, если для всех* $i = 0, \ldots, \frac{m-w}{t'}$ *выполняется* $d^{\Sigma}_{w,q_1,q_2}(x[it' + 1, it' + w], y[it' + 1, it' + w]) < Q$, *то* $ed(x, y) < 2(\Delta q + 1)$.

Доказательство. Из Леммы 3 следует, что при отсутствии совпадающих q_1-грамм для каждой пары удовлетворяющих условию Леммы 5 строк $x' = x[it' + 1, it' + w]$ и $y' = y[it' + 1, it' + w]$ пути $\pi_{x'}$ и $\pi_{y'}$ на $B[x, y; q_2]$ образуют вилку или сдвиг с общим участком пути, состоящим как минимум из $w - q_2 + 1 - 2\Delta q$ дуг и $ed(x', y') < 2(\Delta q + 1)$. Если на интервале $[it' + 1, it' + w - 1]$

имеются повторяющиеся q_1-граммы в $\pi_{x'}$ и/или в $\pi_{y'}$, то из Леммы 4 следует, что $x[it'+1, it'+w], y[it'+1, it'+w]$ можно интерпретировать также как сдвиг или вилку (у соответствующих путей есть общий центральный участок дуг длиной как минимум $w - q_2 + 1 - 2\Delta q$, различные дуги могут быть только по краям интервала) и что также $ed(x', y') < 2(\Delta q + 1)$.

Рассмотрим два соседних интервала $[it'+1, it'+w]$ и $[(i+1)t', (i+1)t'+w]$. Поскольку $t' \leq t - 1 = w - \Delta q$, то они пересекаются не менее, чем Δq символами, что равно максимально возможному размеру полувилки у подпутей, ограниченных хорошими интервалами шириной w.

Правая вилка (сдвиг) на графе $B[x[it'+1, it'+w], y[it'+1, it'+w]; q_2]$ и левая вилка (сдвиг) на $B[x[(i+1)t', (i+1)t'+w], y[(i+1)t', (i+1)t'+w]; q_2]$, принадлежащие пересечению интервалов, состоят из одних и тех же дуг. Поэтому вариант, когда указанные графы являются вилками с непустыми полувилками, соответственно, правой и левой вилок, или сдвигами на разную величину или в разную сторону, исключается. Следовательно, интервал $[it'+1, (i+1)t'+w]$ весь является сдвигом или вилкой, причем размер сдвига или полувилок также ограничен сверху Δq, как и у исходных интервалов. Следовательно, $ed(x[it'+1, (i+1)t'+w], y[it'+1, (i+1)t'+w]) \leq 2(\Delta q + 1)$.

Распространяя такие же рассуждения на остальные интервалы, получим, что $ed(x[1, m], y[1, m]) < 2(\Delta q + 1)$. □

Следствие 1. *Последовательность (при $t' = 1$) из смежных плохих интервалов, окруженная не менее, чем одним хорошим интервалом с каждой стороны, состоит из, как минимум, t интервалов.*

3 Детерминированный метод вложения расстояния редактирования в l_1

В предлагаемом детерминированном методе вложения расстояния редактирования, входная строка $x \in \Sigma^n$ преобразовывается в вектор $v(x)$ конкатенацией всех q-граммных векторов $v_{i,w,q} = v_{w,q}(x[i, i+w-1])$, для $q = q_1, \ldots, q_2$ и $i = 1, \ldots, n-w+1$. В качестве расстояния между векторами принимается манхетенново (l_1) расстояние. В этом разделе на основании утверждений, полученных в разделе 1, мы получим временные, ресурсные и точностные характеристики такого детерминированного вложения.

Для строк $x, y \in \Sigma^n$ определим, с использованием введенного в (1) расстояния d^{Σ}_{w,q_1,q_2}, расстояние

$$D(x,y) = \frac{\sum_{i=1}^{n-w+1} d^{\Sigma}_{w,q_1,q_2}(x[i, i+w-1], y[i, i+w-1])}{(n-w+1)(\Delta q+1)}, \tag{5}$$

и с использованием доказанных в предыдущем разделе утверждений покажем, насколько хорошо аппроксимирует расстояние редактирования предлагаемый метод вложения.

Будем искать ограничение снизу на количество плохих интервалов при $ed(x,y) > k_2$, где k_2 – некоторый параметр метода. Пусть количество плохих интервалов фиксировано, обозначим его N. Найдем сначала ограничение сверху на стоимость редактирования по всем возможным расположениям N интервалов, используя следующий алгоритм редактирования (восстановления) строк.

Будем последовательно переходить от интервала к интервалу, смещаясь на один символ. Допустим, мы восстановили строку до позиции $j-1$. Если последовательно все интервалы, начиная с j-го и до $(j+r)$-го включительно, хорошие, то мы восстанавливаем их за не более, чем $2(\Delta q+1)$ операций, используя результат Леммы 6 для $t' = 1$. Все интервалы в позициях, затронутых этим восстановлением (т.е. начинающиеся между j и $j+r$), назовем *накрытыми*. Если следующий, j-й интервал плохой, мы тратим одну операцию редактирования, замещая $x[j]$ на $y[j]$ и восстанавливая таким образом один символ, и далее переходим к следующему,

$(j+1)$-у интервалу.

Лемма 7. *Обозначим* $S = \lfloor \frac{n-1}{w} \rfloor$. *Восстановление строки* y *из* x *с помощью описанного алгоритма потребует не более*

$$2(\Delta q+1)\max(N, 2(\Delta q+1)\cdot\min(S,N) + \min[N, n-1-(w-1)\cdot\min(S,N)]) \quad < 2(\Delta q+1)(N+1)$$

операций редактирования.

Доказательство. Найдем расположение плохих интервалов, которое максимизирует стоимость редактирования по описанному алгоритму. По определению расстояния редактирования, это будет верхней оценкой на $ed(x,y)$.

Максимальная стоимость редактирования будет достигаться при максимальном количестве возникновений ситуации, описанной в Лемме 6. Поэтому расположение плохих интервалов, на котором достигается максимум стоимости редактирования, будет иметь вид повторяющихся серий вида +-===, где плюсом (+) обозначен хороший интервал, минусом (-) – плохой, а знаком равенства (=) – накрытый хороший (достаточно только одного минуса, следующего за плюсом, для добавления $2(\Delta q+1)$ операций в общую стоимость). Кроме того, в искомой конфигурации последний интервал будет хорошим, т.к. это дает дополнительные $2(\Delta q+1)$ операций редактирования к суммарной стоимости.

Итак, общая стоимость редактирования всех конфигураций вида +− === есть $2(\Delta q+1)\min(S,N)$ (стоимость восстановления таких конфигурации, умноженная на максимальное их число). Остается или $n - w\cdot\min(S,N) - 1$ плохих интервалов (минус единица здесь из-за последнего, всегда хорошего, интервала), или $N - \min(S,N)$ – в зависимости от того, какое из этих выражений имеет меньшую величину. □

Следствие 2. *Пусть* $ed(x,y) > k_2$, $q_1 > 2w/3$, *а также выполняются условия Леммы 5, тогда* $N > \frac{k_2}{2(\Delta q+1)} - 1$.

Учитывая Следствие 1, можно доказать следующую лемму, которая аналогична Лемме 7.

Лемма 8. *Пусть число плохих интервалов для строк* x, y *длиной* n *фиксировано и равно* N. *Тогда* $ed(x,y) \le 2(\Delta q+1)(\lceil \frac{N}{t} \rceil + 1)$.

Следствие 3. *Пусть* $ed(x,y) > k_2$, *тогда* $N > t(\frac{k_2}{2(\Delta q+1)} - 2)$.

Для $x, y \in \Sigma^w$ назовем q-грамму $x[i, i+q-1]$ *k-хорошей*, если найдется такая q-грамма $y[i, i+q-1]$, что $|j-i| \leq k$, т.е. q-грамма в строке y, сдвинутая на не более, чем k символов. Если такой q-граммы в y нет, то назовем $x[i, i+q-1]$ *k-плохой*.

Интервал $[i, i+q-1]$ назовем *k-хорошим*, если $B[x[i, i+q-1], y[i, i+q-1]; q]$ является сдвигом и число k-хороших q-грамм в обеих строках больше или равно $w-q+1-k$, откуда $d_q(x[i, i+q-1], y[i, i+q-1]) \leq 2k$. И *$k$-плохим* в противном случае.

Пусть $N[R]$ – количество интервалов $[i, i+w-1]$, где выполняется некоторое, налагаемое на эти интервалы, условие R.

Лемма 9. *Для* $x, y \in \Sigma^n$ *и* $q_1 < q_2 \leq w \leq \frac{n+1}{k_1+1}$, *если* $ed(x,y) \leq k_1$, *то*

$$N[d^{\Sigma}_{w,q_1,q_2}(x[i, i+w-1], y[i, i+w-1]) \leq 2k_1(\Delta q+1)] > n - w(k_1+1) + 1.$$

Доказательство. Подсчитаем минимальное количество k_1-хороших интервалов при $ed(x,y) \leq k_1$ аналогично доказательству леммы Укконена [12]. Всего имеется $n-w+1$ интервалов. Поскольку каждая операция редактирования может изменить максимум w интервалов, то общее количество k_1-плохих интервалов не превышает $k_1 w$. Оставшиеся $n-w+1-k_1 w$ интервалов будут k_1-хорошими, поскольку могут сдвинуться максимум на k_1 символов. Для k_1-хорошего интервала $[i, i+w-1]$ $d^{\Sigma}_{w,q_1,q_2}(x[i, i+w-1], y[i, i+w-1]) = \sum_{q=q_1}^{q_2} d_q(x[i, i+w-1]), y[i, i+w-1]) < 2k_1(\Delta q+1)$. □

Теперь с использованием Следствия 3 и Леммы 9 можно определить верхнюю и нижнюю границу на $D(x,y)$ при, соответственно, $ed(x,y) \leq k_1$ и $ed(x,y) > k_2$. Положим для этого $w = n^{\gamma}, \gamma \leq 1 - \frac{\ln(k_1+1)}{\ln n}$.

Обозначим множество позиций, где начинаются плохие интервалы, как $I_B = \{i = 1, \ldots, n-w+1 \mid d^{\Sigma}_{w,q_1,q_2}(x[i, i+w-1], y[i, i+w-1]) \geq Q\}$ и множество позиций, где начинаются хорошие интервалы, как $I_G = \{i = 1, \ldots, n-w+1 \mid d^{\Sigma}_{w,q_1,q_2}(x[i, i+w-1], y[i, i+w-1]) < Q\}$. Тогда,

учитывая Следствие 2, получаем

$$(n-w+1)(\Delta q+1)D(x,y) = \left(\sum_{i\in I_B} + \sum_{i\in I_G}\right)\sum_{q=q_1}^{q_2} d_q(x[i,i+w-1], y[i,i+w-1])$$

$$\text{(по Следствию 2, } N > t(\frac{k_2}{2(\Delta q+1)} - 2)\text{, поэтому)}$$

$$\geq NQ + 0 \geq Q(\frac{k_2}{2(\Delta q+1)} - 1).$$

Учитывая группировку плохих интервалов (Следствие 3), получаем

$$D(x,y) \geq \frac{Qt(\frac{k_2}{2(\Delta q+1)} - 2)}{(n-w+1)(\Delta q+1)} = d_2. \tag{6}$$

Обозначим множество позиций, где начинаются k_1-плохие интервалы, как $I_B^{k_1} = \{i = 1,\ldots,n-w+1 \mid \exists\ j \in [i-k_1,\ldots,k_1+i],\ y[j,j+q-1] = x[i,i+q-1]\}$ и где начинаются k_1-хорошие – $I_G^{k_1} = \{i = 1,\ldots,n-w+1 \mid \nexists j \in [i-k_1,\ldots,k_1+i],\ y[j,j+q-1] = x[i,i+q-1]\}$.

$$(n-w+1)(\Delta q+1)D(x,y) = \left(\sum_{i\in I_B^{k_1}} + \sum_{i\in I_G^{k_1}}\right)\sum_{q=q_1}^{q_2} d_q(x[i,i+w-1], y[i,i+w-1])$$

$$\leq (\Delta q+1)[|I_B^{k_1}|(2w+2-q_2-q_1) + ((n-w+1) - |I_B^{k_1}|)2k_1]$$

$$\text{(по Лемме 9, } |I_B^{k_1}| \leq k_1 w\text{, поэтому)}$$

$$\leq 2k_1[w(w+1-\frac{q_1+q_2}{2} - k_1) + (n-w+1)] < 2k_1[w^2+(n+1)](\Delta q+1)\text{, откуда}$$

$$D(x,y) \leq \frac{2k_1[w^2+(n+1)]}{n-w+1} = d_1. \tag{7}$$

Построение векторов $v(\cdot)$ с помощью префиксного дерева потребует порядка $O(n(q_2+w))$ операций, соответственно, общая оценки времени построения равна $O(n^{1+\gamma})$. Размерность вектора $v(\cdot)$ при условии фиксированого алфавита определяется количеством уровней префиксного дерева Δq и количеством узлов на каждом уровне – $O(n)$, итого размерность имеет порядок всего $O(n^{1+\gamma/2})$. Так же ограничено и время вычисления расстояния между векторами.

Из (6) и (7) следует, что для обеспечения $d_2 > d_1$ необходимо, чтобы $k_2 = \Omega(k_1(n^\gamma + n^{1-\gamma}))$. Отсюда, оптимальным значением будет $\gamma = 1/2$, при этом $k_2 = \Omega(k_1 n^{1/2})$ и время построения векторов $v(\cdot)$ – $O(n^{3/2})$, а их размерность – $O(n^{5/4})$.

4 Вероятностные методы формирования векторных представлений для поиска ближайших строк

В этом разделе мы приводим две рандомизированные версии детерминированного метода вложения, описанного в разделе 3, которые могут использоваться для практических приложений в задаче поиска ближашего соседа, поскольку применение детерминированной версии затруднено по причине больших размерностей получаемых векторов и больших размеров базы строк.

Рассмотрим задачу поиска ближайшего соседа NN. Имеется набор строк $P = \{p_1, \ldots, p_P | p_i \in \Sigma^n\}$ и входная строка-проб $p_0 \in \Sigma^n$, тогда задача NN состоит в поиске хотя бы одной строки p^*, такой что $\forall p \in P,\ ed(p, p_0) \geq ed(p^*, p_0)$.

Для больших размерностей входного пространства существующие алгоритмы точного поиска ближайшего соседа сводятся к линейному поиску по P [13], что часто слишком «дорого» для применения на практике. Кроме того, размерность $O(n^{5/4})$ векторов, получаемых в детерминированной схеме вложения расстояния редактирования в векторное пространство, может сама по себе оказаться слишком большой для эффективной обработки. С другой стороны, поиск приближенного ближайшего соседа бывает достаточным для приложений, что вызвало большое количество работ, связанных с разработкой таких алгоритмов. Задача поиска приближенного ближайшего соседа ε-NN, состоит в поиске такого $p^* \in P$, что $\forall p' \in P,\ ed(p^*, p_0) \leq (1 + \varepsilon) ed(p', p_0)$.

Для поиска приближенного ближайшего соседа мы предлагаем рандомизированную версию детерминированного алгоритма, (подраздел 4.1), а также процедуру, основанную на локально-чувствительном хешировании (подраздел 4.2) на 1-стабильных распределениях, релевантную нейроподобным распределенным представлениям [9]. Обе эти версии могут применятся для решения ε-NN задачи.

Приведем вероятностные варианты Леммы 9 и Следствия 3 при случайном выборе значения i из диапазона $1, \ldots, n - w + 1$:

Следствие 4. *Если* $ed(x,y) \le k_1$, $q_1 < q_2 \le w \le \frac{n+1}{k_1+1}$, *то*

$$Prob[d^{\Sigma}_{w,q_1,q_2}(x[i,i+w-1],y[i,i+w-1]) \le 2k_1(\Delta q+1)] > 1 - \frac{k_1 w}{n-w+1}.$$

Следствие 5. *Если* $ed(x,y) > k_2$, *то*

$$Prob[d^{\Sigma}_{w,q_1,q_2}(x[i,i+w-1],y[i,i+w-1]) \ge Q] > \frac{t(\frac{k_2}{2(\Delta q+1)}-2)}{n-w+1}.$$

4.1 Решение задачи ε-NN с помощью рандомизации детерминированного метода

Используем разновидность схемы, описанной в [14]. Пусть зафиксирована позиция i из диапазона $1,\ldots,n-w+1$ и соответствующий ему интервал $[i,i+w-1]$. Если позиция i выбрана случайно и равновероятно (с возвращением) среди всех $n-w+1$ возможных значений, то выражение (5) является матожиданием величины $\frac{d^{\Sigma}_{w,q_1,q_2}(x[i,i+w-1],y[i,i+w-1])}{\Delta q+1}$. Определим вектор $v_i(x)$ как конкатенацию q-граммных векторов $v_q(x)$ этого интервала для значений $q = q_1,\ldots,q_2$.

Выберем случайно, независимо и равновероятно (с возвращением) U значений i_f, $f = 1,\ldots,U$, обозначив их множество как I_U и конкатенируем соответствующие им $v_{i_f}(x)$ в один вектор $\tilde{v}(x)$. Конкатенацию всех $x[i_f, i_f+w-1]$ обозначим $x(I_U)$. Обозначим $\tilde{D}(x,y) = \frac{||\tilde{v}(x)-\tilde{v}(y)||_{l_1}}{(\Delta q+1)U}$.

Следующая лемма доказывается аналогично Лемме 2 из [14].

Лемма 10. *Пусть строки* $p_0, p_a, p_b \in P$ *такие, что* $ed(p_0,p_a) \le k_1$, $ed(p_0,p_b) > k_2$. *Тогда*

$$Prob[\tilde{D}(p_0,p_a) > d_1+\varepsilon] < e^{-2U\varepsilon^2}, \quad Prob[\tilde{D}(p_0,p_b) < d_2-\varepsilon] < e^{-2U\varepsilon^2}.$$

Создадим структуру S, состоящую из n структур S_k, $k = 1,\ldots,n$, по одной на каждое из возможных значений расстояний редактирования между строками длины n. Каждая из S_k состоит из M структур $F_1,\ldots,F_M$. Каждая структура F_m, $m = 1,\ldots,M$ содержит U значений позиций $i_{m1},\ldots,i_{mU}$ (указывающих на начало соответствующих интервалов $I_{i_{mu}} = [i_{mu}, i_{mu}+w-1]$), выбранных случайно и равновероятно, с возвращением, из диапазона $[1,\ldots,n-w+1]$, и таблицу Λ_m, содержащую $|\Sigma|^n$ ячеек (по числу возможных строк длиной n). Содержимое интервала $I_{i_{mu}}$ в структуре F_m для строки x обозначим как $x[I_{i_{mu}}]$. Конкатенацию всех $v_{w,q_1,q_2}(x[I_{i_{mu}}]), u =$

$1, \ldots, U$ обозначим $v_m(x)$. Ячейка таблицы Λ_m, соответствующая $z \in |\Sigma|^{wU}$, содержит строку $p \in P$, если $\tilde{D}(p, z) \leq d_1$, если такая p существует, иначе ячейка пуста. Структура S при простейшей реализации будет занимать объем памяти порядка $O(nM(U + \log P|\Sigma|^{wU}) + nP)$ и может быть построена за время $O(nMP(wU + |\Sigma|^{wU}))$.

Объем таблиц Λ_m можно уменьшить до $|\Sigma|^{wU}$, индексируя ее строками, получающимися конкатенацией U интервалов. Далее, для принятия решения о внесении строки в ячейку нам необходимы q-спектры подстрок $z[1 + (u-1)w, uw]$ при $q = q_1, \ldots, q_2$. Поэтому можно сэкономить место под хранение таблиц Λ_m, объединив ячейки с совпадающими q-спектрами интервалов $z[1 + (u-1)w, uw]$ для всех $q = q_1, \ldots, q_2$. В этом случае можно обозначить

$$\tilde{D}_m(p; z) = \sum_{u=1}^{U} d^{\Sigma}_{w,q_1,q_2}(p[I_{i_{mu}}], z[1 + (u-1)w, uw]) = \sum_{u=1}^{U} ||\tilde{v}(p[I_{i_{mu}}]) - \tilde{v}(z[1 + (u-1)w, uw])||_{l_1}.$$

И ячейка таблицы Λ_m, соответствующая $z \in |\Sigma|^{wU}$, будет содержать строку $p \in P$, если $\tilde{D}_m(p; z) \leq d_1$.

Для проба $p_0 \in \Sigma^n$ будем говорить, что *структура F_m ошибается на P*, если существует $p_1 \in P$, такая, что $ed(p_0, p_1) \leq k_1$ (или строка p_2 с $ed(p_0, p_2) \geq k_2$) и $\tilde{D}_m(p_0, p_1) > d_1$ ($\tilde{D}_m(p_0, p_2) < d_2$). Если существует k такое, что более, чем $\mu M / \log n$ структур F_m в S_k ошибаются на P, то будем говорить, что *структура S ошибается на p_0*. Будем говорить, что *структура S ошибается*, если существует $p_0 \in \Sigma^n$, такое, что S ошибается на p_0.

По теореме 4 из [14] для любого $\delta > 0$ и запроса p_0, если положить $M = (n \log_2 \Sigma + \log n - \log \delta) \ln n / \mu$ и $F = 0.5\varepsilon^{-2} \ln(2eP \ln n / \mu)$, то вероятность, что S ошибается на p_0 – не больше $\delta 2^{-n}$. Таким образом S на всей P ошибается с вероятностью δ.

Процедура поиска: Допустим S не ошибается (это происходит с вероятностью $1 - \delta$). Используем двоичный поиска для нахождения ближайшего соседа. Выберем произвольно начальное k и выберем случайно и равновероятно одну из F_m в S_k. Подсчитаем значение $p_0(I_{F_m})$ и проверим соответствующую ячейку. Если ячейка не пуста, переходим и проверяем аналогично S_{k-1}, иначе переходим к S_{k+1}. По Лемме 6 из [14] вероятность, что выбранное F_i ошибается на p_0, меньше или равна μ.

Приведем аналог Леммы 7 из [14]

Лемма 11. *Если все S_k не ошибаются на p_0, тогда расстояние от p', возвращенной алгоритмом поиска, отличается в не более, чем $1+\varepsilon$ раз от расстояния от p_0 до настоящего ближашего соседа.*

И аналогично теореме 8 [14], получаем, что приведенная процедура поиска находит с высокой вероятностью $(1+\varepsilon)$-ближайшего соседа, для любого $\varepsilon > 0$: *Если S не ошибается, то для любого запроса p_0 алгоритм находит $(1+\varepsilon)$-приближенного соседа с вероятностью $1-\mu$ за время $O(\varepsilon^{-2}n(\log|P| + \log\log n - \log\mu)\log n)$.*

4.2 Решение задачи ε-NN с помощью LSH и 1-стабильного распределения

Сначала опишем оригинальную схему локально-чувствительного хеширования (LSH) [15] в применении к строкам с классической метрикой редактирования.

Определим шар радиуса k, состоящий из точек, расстояния которых от центра шара t не превышает k: $S(t,k) = \{q : ed(q,s) \leq k\}$. Задача (ε,k)-PLEB (Point Location in Equal Balls) определяется как поиск алгоритма, который для любого проба $p_0 \in P$ делает следующее: (1) если существует $p \in P$, такое что $p_0 \in S(p,k)$, возвращает YES и любую из точек $p' \in P$, таких что $p_0 \in S(p',(1+\varepsilon)k)$, (2) если $p_0 \notin S(p,(1+\varepsilon)k)$ для всех $p \in P$, возвращает NO, (3) если ближайшая точка $p \in P$, такая что $r \leq ed(p_0,p) \leq (1+\varepsilon)r$, то возвращает или YES или NO. В работе [15] показана сводимость задачи ε-NN к задаче ε-PLEB. Задача (k_1,k_2)-PLEB является вариантом ε-PLEB задачи и формулируется как поиск алгоритма, который: (1) если существует $p \in P$, такое что $p_0 \in S(p,k_1)$, возвращает YES и любую из точек $p' \in P$, таких что $p_0 \in S(p',k_2)$, (2) в остальных случая возвращает NO.

Идея схемы LSH состоит в том, чтобы, используя определенное количество некоторых хеш-функций, добиться высокой вероятности коллизии (совпадения значений хеш-функций) – между близко находящимися объектами, и низкой – для далеких объектов. Тогда, применяя такое

же хеш-преобразование к пробу, мы проверяем, не равен ли хэш какому-нибудь из ранее посчитанных хешей векторов из P. При совпадении найденный вектор $p \in P$ будет являться приближенным соседом к пробу. Насколько он будет отличаться от точного ближайшего соседа, зависит от свойств использованных хеш-функций.

Семейство $H = \{h : \Sigma^n \to X\}$ (где X – некоторое конечное или счетное множество значений) называется (k_1, k_2, p_I, p_{II})-чувствительным или просто локально-чувствительным, если для любых $x, y \in \Sigma^n$ и любой независимо и равновероятно выбранной хеш-функции $h \in H$ выполняется:

$$\begin{aligned} &\text{если } x \in S(y, k_1), \text{ то } Prob[h(x) = h(y)] > p_I, \\ &\text{если } x \notin S(y, k_2), \text{ то } Prob[h(x) = h(y)] < p_{II}. \end{aligned} \tag{8}$$

Для того, чтобы семейство H было «полезным», необходимо, чтобы выполнялось условие $k_1 < k_2$, т.е. «близкая» точка x должны находятся ближе к y, чем точки, считающиеся «дальними», и $p_{II} < p_I$, («близкие» точки должны вызывать коллизию хеш-функций с большей вероятностью, чем «дальние»).

На предварительном этапе подготовки структуры для приближенного поиска ближайшего соседа определяется семейство функций $\{g : \Sigma^n \to X^K\}$, где $g(p) = (h_1(p), h_2(p), \ldots, h_K(p))$. Всего из семейства G выбирается случайно и равновероятно L штук функций $g_1, \ldots, g_L$. Создается также таблица, в ячейку которой заносятся строки p из базы P на основании значения хеш-вектора $g(p)$: каждая строка p попадает в ячейку с идентификатором, равным значению хеш-вектора, посчитанном на ней, а именно $(h_1(p), h_2(p), \ldots, h_K(p))$. Размер таблицы ограничен $O(|P|L)$, кроме того, еще необходимо хранить исходную базу – $O(n|P|)$. Построение таблицы завершено, когда каждая $p \in P$ занесена в соответствующую ячейку.

Процедура поиска: Для строки-проба p_0 вычисляются все хеш-векторы $g_i(p_0), i = 1, \ldots, L$ и проверяются соответствующие ячейки в таблице. Если проверяемая ячейка содержит строку $p^* \in S(p, k_2)$, мы возвращаем YES и p^*. Если после проверки $2L$ строк не было найдено ни одной строки из P в $S(p, k_2)$, возвращаем NO.

В [15] доказывается, что можно добиться выполнения следующих условий с константной вероятностью (большей 1/2), при которых описанная процедура поиска приближенного ближайшего соседа корректна: (1) если существует $p^* \in S(p, k_1)$, то должно выполняться $g_j(p) = g_j(p^*)$, для некоторого $j = 1, \ldots, L$; (2) количество коллизий хеш-функций со строками вне $S(p, k_2)$ в L проверенных ячейках должно быть меньше $2L$: $\sum_{j=1}^{L} |(P - S(q; k_2)) \cap g_j^{-1}(g_j(p_0))| < 2L$. Это достигается это определенным выбором значений K, L [15]: *Если H является (k_1, k_2, p_I, p_{II})-чувствительным семейством функций, а $K = \log_{\frac{1}{p_{II}}} |P|$ и $L = |P|^{\rho}$, где $\rho = \ln(\frac{p_I}{p_{II}})$, тогда алгоритм решения (k_1, k_2)-PLEB задачи занимает $O(n|P| + |P|^{1+\rho})$ памяти, использует $O(|P|^{\rho})$ вычислений расстояния и $O(|P|^{\rho} \log_{\frac{1}{p_{II}}} |P|)$ вычислений хеш-функций.* Заметим, что из $p_{II} < p_I$ и выражения для ρ следует, что время поиска сублинейно от $|P|$.

В работе [16] для построения локально-чувствительного к l_p-норме семейства хеш-функций предлагается использовать то свойство p-стабильных распределений, что линейные комбинации случайных величин ϕ_i из этого распределения распределены так же, как и одна случайная величина, умноженная на норму коэффициентов данной линейной комбинации [17]. Поскольку скалярное произведение линейно, то $(v_1, \phi) - (v_2, \phi)$ распределено так же, как и $||v_1 - v_2||_{l_1}\phi$. Поэтому, если поделить действительную ось на равные промежутки, то, интуитивно понятно, что скалярные произведения векторов с близкой нормой на случайный вектор из соответствующего стабильного распределения, будут попадать в одни и те же или близкие промежутки и на этом основании можно построить локально-чувствительное семейство.

В случае $p = 1$ (1-стабильное распределение), для использования этого свойства хеш-функции задаются в виде

$$h(v) = \lfloor \frac{(v, \bar{\phi}) + b}{r} \rfloor, \tag{9}$$

где $r \in \mathbb{R}$ – некоторое число, b – равномерно распределенная случайная величина из $[0, r]$, а $\bar{\phi}$ – вектор из элементов из 1-стабильного распределения Коши с плотностью $\frac{1}{\pi(1+x^2)}$. Можно показать [16], что для двух фиксированных векторов v_1, v_2, в зависимости от значения l_1-нормы

разности векторов $c = ||v_1 - v_2||_{l_1}$, вероятность коллизии хеш-фукции (9) равна

$$p(c) = \int_0^r \frac{1}{c} f(c)(1 - \frac{t}{c})dt = \frac{1}{\pi}(2\tan^{-1}(\frac{r}{c}) - \frac{c}{r}\ln(1 + (\frac{r}{c})^2)), \tag{10}$$

где $f(\cdot)$ – функция плотности модуля случайной величины, распределенной по Коши. Функция $p(c)$ есть монотонно убывающая функция, поэтому, по определению (8) семейство функций (9) будет локально-чувствительным и их можно использовать в схеме LSH.

Нам понадобятся асимптотические значения (10) при $r >> c$ и при $r << c$:

1. При $r << c$, $\tan^{-1}(x) \simeq x - \frac{x^3}{3} + O(x^3)$, поэтому

$$p(c) = \frac{1}{\pi}(2\tan^{-1}(\frac{r}{c}) - \frac{c}{r}\ln(1 + (\frac{r}{c})^2)) < \frac{1}{\pi}(2\frac{r}{c} - \ln e^{\frac{r}{c}}) = \frac{1}{\pi}\frac{r}{c}. \tag{11}$$

2. При $r >> c$, поскольку $\tan(x) \to \frac{\pi}{2}$ при $x \to \infty$, то

$$p(c) = \frac{1}{\pi}(2\tan^{-1}\frac{r}{c} - \frac{c}{r}\ln(1 + (\frac{r}{c})^2)) < 1 - \frac{2}{\pi}\frac{c}{r}\ln\frac{r}{c}. \tag{12}$$

Предложим новую хэш-функцию, основанную на определении (9). В работе [16] для формирования вектора $g_j = (h_1(v), h_2(v), \ldots, h_K(v))$ фиксированный вектор v умножается скалярно на ряд случайных векторов $\phi_i, i = 1, \ldots, K$. Вместо фиксированного вектора, возьмем K случайных векторов $v^i_{w,q_1,q_2}(x)$, получаемых случайным и независимым выбором окна шириной w в строке x (см. Следствия 5 и 4). Для каждого из них сгенерируем свой вектор ϕ_i, элементы которого взяты из распределения Коши. Модифицированные хеш-функции $h'_i(x)$, – элементы вектора $h'(x) = (h'_1(x), h'_2(x), \ldots, h'_1(x))$, – определим как

$$h'_i(x) = \lfloor \frac{(v^i_{w,q_1,q_2}(x), \phi_i) + b}{r} \rfloor, \tag{13}$$

где b, r – параметры из определения (9).

Для определения значений p_I, p_{II} необходимо выяснить распределение величин $h'_i(x)$ и $h'_i(y)$. Пусть $Pprob[h'(x) = h'(y)|c] = p(c)$, где c – целочисленное значение $d^{\Sigma}_{w,q_1,q_2}(x, y)$.

Обозначим $d_1 = 2k_1(\Delta q + 1), \quad d_2 = Q, \quad p_1 = 1 - \frac{k_1 w}{n-w+1}, \quad p_2 = \frac{t(\frac{k_2}{2(\Delta q+1)} - 2)}{n-w+1}$.

$$Prob[h'_i(x) = h'_i(y)|ed(x, y) \le k_1] \ge \sum_{c \le d_1} p(c|ed(x, y) \le k_1)p(c) \ge p(d_1) \sum_{c \le d_1} p(c|ed(x, y) \le k_1)$$

$$= p(d_1)p(c \le d_1|ed(x, y) \le k_1) \ge p(d_1)p_1,$$

$$Prob[h'_i(x) = h'_i(y)|ed(x,y) > k_2] \le p(d_2) \sum_{c \ge d_2} p(c|ed(x,y) > k_2) + \sum_{c < d_2} p(c|ed(x,y) > k_2)p(c)$$

$$\le p(d_2)p(c \ge d_2|ed(x,y) > k_2) + \sum_{c < d_2} p(c|ed(x,y) > k_2)p(0)$$

$$= p(d_2)p_2 + (1-p_2)p(0) \le 1 - p_2(1-p(d_2)).$$

Итак, для семейства новых хеш-функций (13) получаем

$$Prob[h'_i(x) = h'_i(y)|ed(x,y) \le k_1] \ge p(d_1)p_1 = p_I, \tag{14}$$

$$Prob[h'_i(x) = h'_i(y)|ed(x,y) > k_2] \le 1 - p_2(1-p(d_2)) = p_{II}. \tag{15}$$

Чтобы такое семейство было локально-чувствительным, должно выполняться $1 - p_2(1 - p(d_2)) < p(d_1)p_1$. В свою очередь, для его выполнения необходимо, чтобы

$$k_2 > 2(\Delta q + 1)(2 + \frac{n-w+1}{t(1-p(d_2))}(1 - p(d_1) + \frac{wk_1p(d_1)}{n-w+1}). \tag{16}$$

Для определения асимптотического поведения k_2 положим $w = n^\gamma,\ 0 < \gamma < 1$, и $r = n^\mu,\ \mu > 0$. Из условия 2 Леммы 3 $\Delta q = \Theta(\sqrt{w})$ то

$$d_1 = 2k_1(\Delta q + 1) = \Theta(k_1 n^{\gamma/2}), \quad d_2 = Q = \Theta(n^\gamma), \quad t = w - q_2 - \Delta q = \Theta(n^\gamma).$$

При $n \to \infty$, используя (10), (11), (12), найдем оценки $1 - p(d_1)$ и $1 - p(d_2)$ в зависимости от соотношения параметров $\mu,\ \gamma$:

1. если $\gamma = \mu$, то $r = \omega(d_1),\ r = \Theta(d_2)$, то

$$1 - p(d_2) = const, \quad 1 - p(d_1) > \frac{2}{\pi}\frac{d_1}{r}\ln(\frac{r}{d_1}).$$

2. если $\mu > \gamma$, то $r = \omega(d_1)$ и $r = \omega(d_2)$, то

$$1 - p(d_1) > \frac{2}{\pi}\frac{d_1}{r}\ln(\frac{r}{d_1}), \quad 1 - p(d_2) > \frac{2}{\pi}\frac{d_2}{r}\ln(\frac{r}{d_2}).$$

3. если $\gamma/2 < \mu < \gamma$, то $r = \omega(d_1)$ и $r = o(d_2)$, то

$$1 - p(d_2) > 1 - \frac{1}{\pi}(\frac{r}{d_2}), \quad 1 - p(d_1) > \frac{2}{\pi}(\frac{d_1}{r}\ln(\frac{r}{d_1})).$$

4. если $\mu = \gamma/2$, то $r = \omega(d_1)$ и $r = o(d_2)$, то

$$1 - p(d_2) > 1 - \frac{1}{\pi}(\frac{r}{d_2}), \quad 1 - p(d_1) = const;$$

Подставляя в (16), найдем выражение для k_2 для каждого из случаев

1. если $\gamma = \mu$, то $k_2 = \Omega(k_1(n^{1-\gamma}\ln n + n^{\gamma/2})$. При $\gamma > 2/3$, $n^{\gamma/2}$ доминирует $n^{1-\gamma}\ln n$ и k_2 будет иметь асимптотику $\Omega(n^{\gamma/2}), \gamma > 2/3$. При $\gamma \leq 2/3$, $n^{1-\gamma}\ln n$ доминирует $n^{\gamma/2}$ и k_2 имеет асимптотику $\Omega(n^{1-\gamma}\ln n), \gamma \leq 2/3$. Отсюда, оптимальным значением будет $\gamma = 2/3$, и $k_2 = \Omega(k_1 n^{1/3}\ln n)$.

2. если $\mu > \gamma$, то $k_2 = \Omega(n^{\gamma/2} + k_1[n^{1-\gamma} + n^{\mu}/\ln n]))$. Поскольку $\mu > \gamma$, то $\frac{n^{\mu}}{\ln n} = \Omega(n^{\gamma})$ и k_2 в этом случае растет быстрее, чем в варианте с $\mu = \gamma$.

3. если $\gamma/2 < \mu < \gamma$, то $k_2 = \Omega(k_1(n^{1-\mu}(\mu - \gamma/2)\ln n + n^{\gamma/2})$. Поскольку $\mu < \gamma$, то $n^{1-\mu}\ln n = \Omega(n^{1-\gamma}\ln n)$, что является более плохой оценкой, чем оценка в варианте с $\mu = \gamma$.

4. если $\mu = \gamma/2$, то $k_2 = \Omega(n)$, что также хуже, чем вариант $\mu = \gamma$.

Итак, оптимальным значением γ есть $\gamma = 2/3$, при которой $k_2^* = \Omega(k_1 n^{1/3}\ln n)$.

Допустим, мы увеличим k_2, положив $k_2 = 2(\Delta q+1)(2+z\frac{n-w+1}{t(1-p(d_2))}(1-p(d_1)+\frac{wk_1p(d_1)}{n-w+1})$, для $z > 1$. Это повлияет на значение и асимптотику p_2, поскольку его значение зависит от k_2. Оценив в таком случае величину $\rho = \frac{\ln(p_1p(d_1))}{\ln(1-p_2(k_2)(1-p(d_2)))} = \frac{\ln(1-A)}{\ln(1-zA)}$, где $A = 1 - p_1p(d_1)$, аналогично оценке ρ для случая хеммингова расстояния [15, 18] получаем $\rho = O(\frac{1}{1+z})$ при условии $\ln|P| > p_1p(d_1)$.

Таким образом, метод приближенного поиска ближайшей строки, построенный на описанном модифицированном варианте LSH на 1-стабильном распределении, возвратит с вероятностью большей 1/2 строку из P, которая принадлежит шару $S(q, O(zk_1n^{1/3}\ln n))$, затратив на это порядка $O(|P|^{1/(1+z)})$ операций. Поскольку схема имеет константную вероятность ошибки, повторив описанную процедуру LSH паралельно и независимо порядка $O(\ln\frac{1}{\alpha})$ раз, мы можем добиться вероятности успеха хотя бы в одном запуске процедуры не менее $1 - \alpha$ для любого заданого уровня ошибки $\alpha < 1$.

5 Заключение

С точки зрения теории метрических вложений проанализировано, развиваемое в рамках нейросетевого распределенного представления информации в ассоциативно-проективных нейронных сетях [19], представление символьных строк [8, 9].

Разработан новый q-граммный метод аппроксимации расстояния редактирования и приведены значения параметров метода, при которых достигается уточнение оценок качества аппроксимации, предложенных в [9]. Также достигнуто улучшение асимптотики роста k_2 по сравнению с методами вложения расстояния редактирования в пространство l_1, описанными в [20]. Разработана рандомизированная версия метода аппроксимации и показана возможность построения семейства рандомизированных локально-чувствительных функций, которые могут использоваться для эффективного решения задачи приближенного поиска ближайших строк.

Дальнейшим направлением исследований является уточнение полученных оценок, исследование возможности модификации предложенного или разработки нового алгоритма для более общих версий расстояния редактирования (с перестановками, с переменными стоимостями операций), а также исследование эффективности разработанного метода в практических задах, требующих сравнения длинных символьных строк.

Список литературы

[1] *Levenshtein V. I.* Binary codes capable of correcting deletions,insertions, and reversals // *Soviet Physics - Doklady.* — 1966. — February. — Vol. 10, no. 8. — Pp. 707–710.

[2] *Burks C., Cinkosky M. J., Gilna P.* Decades of nonlinearity: the growth of DNA sequence data // Los Alamos Science / Ed. by N. G. Cooper. — 1992. — No. 20. — Pp. 254–255.

[3] *Vintsyuk T. K.* Speech discrimination by dynamic programming // *Kibernetika (Cybernetics).* — 1968. — January-February. — Vol. 4. — Pp. 81–88.

[4] *Wagner R. A., Fischer M. J.* The string-to-string correction problem // *Journal of the ACM.* — 1974. — January. — Vol. 21, no. 1. — Pp. 168–173.

[5] *Navarro G.* A guided tour to approximate string matching // *ACM Computing Surveys.* — 2001. — Vol. 33, no. 1. — Pp. 31–88.

[6] *Indyk P.* Embedded Stringology. — Talk at Fifteenth Annual Combinatorial Pattern Matching Symposium.

[7] *Indyk P.* Open problems // Workshop on Discrete Metric Spaces and their Algorithmic Applications / Ed. by J. rí Matouˇsek. — Haifa: 2002. — March.

[8] *Sokolov A., Rachkovskij D.* Some approaches to distributed encoding of sequences // Proc. of XI-th International Conference Knowledge-Dialogue-Solution. — Vol. 2. — Varna, Bulgaria: 2005. — June. — Pp. 522–528.

[9] *Sokolov A.* Nearest string by neural-like encoding // Knowledge-Dialogue-Solution, KDS-2006. — Varna, Sofia, Bulgaria: FOI BG, 2006. — June.

[10] *de Bruijn N. G.* A combinatorial problem // Koninklijke Nederlandsche Akademie van Wetenschappen. — Vol. 49. — 1946.

[11] *Knuth D. E.* Seminumerical Algorithms. — Second edition. — Reading, Massachusetts: Addison-Wesley, 1981. — 10 January. — Vol. 2 of *The Art of Computer Programming*.

[12] *Ukkonen E.* Approximate string-matching with q-grams and maximal matches // *Theor. Comput. Sci.* — 1992. — Vol. 92, no. 1. — Pp. 191–211.

[13] *Borodin A., Ostrovsky R., Rabani Y.* Lower bounds for high dimensional nearest neighbor search and related problems // Proceedings of 31-st annual ACM STOC. — 1999. — Pp. 312–321.

[14] *Kushilevitz E., Ostrovsky R., Rabani Y.* Efficient search for approximate nearest neighbor in high dimensional spaces // Proc. of 30th STOC. — 1998. — Pp. 614–623.

[15] *Indyk P., Motwani R.* Approximate nearest neighbors: towards removing the curse of dimensionality // STOC '98: Proceedings of the thirtieth annual ACM symposium on Theory of computing. — New York, NY, USA: ACM Press, 1998. — Pp. 604–613.

[16] Locality-sensitive hashing scheme based on p-stable distributions / M. Datar, N. Immorlica, P. Indyk, V. Mirrokni // Twentieth annual symposium on Computational geometry. — Brooklyn, New York, USA: 2004. — Pp. 253–262.

[17] *Nolan J.* An introduction to stable distributions. — `http://academic2.american.edu/~jpnolan/stable/chap1.pdf`.

[18] *Gionis A., Indyk P., Motwani R.* Similarity search in high dimensions via hashing // VLDB '99: Proceedings of the 25th International Conference on Very Large Data Bases. — San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1999. — Pp. 518–529.

[19] *D.A. R.* Representation and processing of structures with binary sparse distributed codes // *IEEE Transactions on Knowledge and Data Engineering.* — 2001. — Vol. 13, no. 2. — Pp. 261–276.

[20] Approximating edit distance efficiently / Z. Bar-Yossef, T. S. Jayram, R. Krauthgamer, R. Kumar // 45th Annual IEEE Symposium on Foundations of Computer Science. — IEEE, 2004. — October. — Pp. 550–559.

В печать

На перевод на английский язык согласен