1975 год ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ Выпуск 62

УДК 51:621.396.801

ОТЫСКАНИЕ СТАТИСТИЧЕСКИХ ЗАКОНОМЕРНОСТЕЙ ТЕКСТОВ МЕТОДОМ АССОЦИАТИВНОГО КОДИРОВАНИЯ

В.Д.Гусев, Ю.Г.Косарев, Т.Н.Титкова

Для произвольной символьной последовательности значительной длины рассматривается задача получения распределений по частоте встречаемости ℓ -грамм - связных подпоследовательностей из ℓ символов.

Предполагается, что ограничения сверху на величину ℓ (как в [2] и [3]), на мощность алфавита n и объем словаря n^{ℓ} (как в [2] и [4]) отсутствуют, а допустимая длина самого анализируемого текста N достигает 10^6 - 10^7 символов (что на несколько порядков больше, чем в [4]).

В отличие от [1] предлагается алгоритм для случая $\ell > \ell_0$, когда подавляющая часть текста представлена единичными ℓ -граммами, а доля повторяющихся ℓ -грамм такова, что их список может разместиться в оперативной памяти объемом S бит.

Величина порога ℓ_0 в общем случае зависит от длины N анализируемого текста, мощности исходного алфавита n и ряда побочных факторов. Например, приближенная оценка величины порога ℓ_0, полученная в [1] на основе анализа второго закона Ципфа на материале английского языка ($N=10^6$, $n=50$), составляет величину порядка 10-15.

В качестве возможных сфер применения алгоритма можно указать задачи выявления структуры связей между конструкциями языка и оценки его избыточности [5], задачи дешифровки [6], сортировки массивов с большим числом единичных элементов и ряд других.

I. И д е я а л г о р и т м а . Задача решается в два этапа. На первом - в тексте выявляются и устраняются из дальнейшего анализа единичные ℓ -граммы. На втором - строится распределение для повторяющихся ℓ -грамм.

Число ℓ -грамм, остающихся после первого этапа, мало, поэтому второй этап не вызывает больших затруднений и может выполняться, например, с помощью алгоритма внутренней сортировки.

Алгоритм реализации первого этапа основывается на общей идее ассоциативного кодирования [8] (" hash coding " [II], функция расстановки [7]), но использует ее в более простом и в то же время более эффективном виде.

Традиционная процедура обычно подразумевает: I) ф о р м и р о в а н и е адреса элемента информационного массива по его з н а ч е н и ю ; 2) з а н е с е н и е с а м о г о э л е м е н т а в участок памяти, соответствующий сформированному адресу; и 3) о р г а н и з а ц и ю с п и с к о в о й с т р у к т у р ы для элементов, попадающих в уже занятые участки памяти.

Занесение в память самого элемента информационного массива обусловлено тем, что процедура адресации не взаимно-однозначна, то есть по полученному адресу не удается единственным образом восстановить значение элемента информационного массива. Возникающая неоднозначность обусловлена, во-первых, спецификой самой процедуры адресации, которая может двум или более различным элементам поставить в соответствие один и тот же адрес, а, во-вторых, тем, что в информационном массиве могут быть повторяющиеся элементы, которые в принципе будут давать один и тот же адрес.

Предлагаемая модификация традиционной процедуры касается двух пунктов:

- организации процедуры адресации в случае, когда размер адресного поля существенно меньше λ (требование, связанное с ослаблением ограничений на длину массива);

- упрощения процедуры и более эффективного использования объема оперативной памяти благодаря отказу от требования однозначного отождествления элементов, связанного обычно с занесением самих элементов в оперативную память.

Разбиение анализируемого массива на группы, каждую из которых можно было бы обработать с учетом реальных ограничений на размер адресного поля, не может быть выполнено простым расчленением массива на части, поскольку при этом одинаковые элементы попадут в разные группы. Поэтому для разбиения массива на группы нужно, в свою очередь, воспользоваться процедурой ассоциативного кодирования, то есть разбиение совокупности ℓ-грамм на части индуцируется некоторым фиксированным разбиением на множестве соответствующих им адресов.

В нашем случае, в отличие от традиционных применений процедуры ассоциативного кодирования, важно не столько само по себе выявление всех одиночных ℓ-грамм, сколько сокращение числа ℓ-грамм, поступающих на второй этап, до некоторой заданной величины. Это позволяет использовать процедуры, которые не гарантируют обнаружения всех одиночных ℓ-грамм, например, последовательно использовать набор различных функций адресации $f_1, f_2, \ldots, f_\rho$, подобранный таким образом, что для достаточно большого числа элементов информационного массива выполняется соотношение

$$f_i(\lambda) \neq f_j(\lambda); \quad i, j = 1, \ldots, \rho; \quad i \neq j. \qquad (I)$$

В этом случае ℓ-граммы, не классифицировавшиеся однозначно (по признаку - одиночные или нет) при использовании функции адресации f_1, подвергаются обработке последующими функциями набора до тех пор, пока в отношении их не будет вынесено однозначное решение либо пока число их не сократится до некоторого порогового значения. Такой способ не предполагает выписывания в явном виде самих ℓ-грамм в оперативной памяти с последующей организацией списковой структуры. Это позволяет существенно сэкономить объем оперативной памяти и, как показано дальше, не менее чем на порядок увеличить (по сравнению с традиционной процедурой " hash coding ") количество ℓ-грамм, однозначно классифицируемых за один "прогон" исходного текста. Под "прогоном" здесь понимается процедура просмотра всей последо -

вательности, включающая в себя применительно к каждой ℓ -грамме анализ того, подлежит ли ℓ -грамма обработке на данном этапе, и когда это так, саму обработку.

Фактически в участке памяти, определяемом применением соответствующей функции адресации к какой-либо конкретной ℓ -грамме, может храниться информация всего лишь о трех состояниях (0,I,2), соответствующих числу обращений по данному адресу: 0, I, 2 и более, для чего требуется два бита. На практике к этим двум битам имеет смысл добавить третий под метку для единичных ℓ -грамм, выявленных на предыдущем просмотре, что позволяет избежать повторного прогона текста.

2. Вероятностная модель. В качестве вероятностной модели, имитирующей процедуру формирования адреса в методе ассоциативного кодирования, можно воспользоваться классической схемой случайного размещения q шаров по Q ящикам (так называемая задача о дробинках [9]). При этом количество ящиков Q соответствует размеру адресного поля S/u, где u - число битов, отведенное для хранения информации по каждому адресу, а число шаров q соответствует величине группы ℓ -грамм, обрабатываемых за один прогон текста. Ввиду малости доли повторяющихся ℓ -грамм будем исходить из модели с равновероятным попаданием каждого шара в любой из ящиков. В этом случае [10] математическое ожидание M_{μ_r} числа ящиков, в каждом из которых содержится ровно по r ($r = 0,1,2,\dots,q$) шаров, равно:

$$M_{\mu_r} = Q C_q^r \frac{1}{Q^r} \left(1 - \frac{1}{Q}\right)^{q-r}. \qquad (2)$$

При $q, Q \to \infty$ и ограниченном $q/Q = k$

$$M_{\mu_r} \approx Q \frac{k^r}{r!} e^{-k}. \qquad (3)$$

а дисперсия

$$\mathcal{D}\mu_r \approx M_{\mu_r}\left[1 - \frac{M_{\mu_r}}{Q}\left(1 + \frac{(k-r)^2}{k}\right)\right]. \qquad (4)$$

Отсюда для математического ожидания числа ящиков, содержащих ровно по одному шару, имеем

$$\mathcal{J}_1 = M_{\mu_1} = q\left(1 - \frac{1}{Q}\right)^{q-1} . \qquad (5)$$

При больших q и Q

$$\mathcal{J}_1 \approx q e^{-q/Q} . \qquad (6)$$

Можно видеть, что при фиксированном Q максимум $\mathcal{J}_1$ достигается при $q = Q$. При этом

$$\mathcal{J}_1^* = \mathcal{J}_1(q = Q) \approx \frac{Q}{e} , \qquad (7)$$

$$\sigma(\mathcal{J}_1^*) = \sqrt{\mathcal{D}_{\mu_1}(q = Q)} = \left[\frac{Q}{e}\left(1 - \frac{1}{e}\right)\right]^{1/2} \approx 0.79 \cdot \sqrt{\mathcal{J}_1^*} . \qquad (7а)$$

3. Схема алгоритма и оценка его трудоемкости. Трудоемкость предлагаемого подхода можно оценить уже на данном этапе, отвлекаясь от детальных подробностей алгоритма, изложенных в следующем разделе. Кратко опишем лишь принципиальную схему алгоритма.

Алгоритм представляет собой итеративную процедуру, каждому шагу которой соответствует один прогон текста, связанный с выделением и обработкой ℓ -грамм лишь одной группы. Формирование группы осуществляется таким образом, что в нее входят лишь ℓ -граммы, в отношении которых еще не выяснено, являются они одиночными или нет. Ядро итерации составляет описанная выше процедура формирования адреса каждой ℓ -граммы выделенной группы методом ассоциативного кодирования. Оценка количества одиночных ℓ -грамм, выявленных при обработке одной группы, осуществляется на основании соотношения (7). Поскольку выделяемые группы примерно равномощны, то в результате каждой итерации выявляется и устраняется из дальнейшего анализа примерно одинаковое количество одиночных ℓ -грамм. По выходе из последней итерации число ℓ -грамм, подлежащих дальнейшему анализу, не должно превышать определенной пороговой величины, гарантирующей нам пренебрежимо малые затраты на заключительной стадии обработки.

Итак, применительно к нашему случаю, заменяя в (7) Q на S/u , получим

$$\mathcal{J}_1^* \approx \frac{S}{ue} \quad , \tag{8}$$

В простейшем варианте, когда $u = 3$,

$$\mathcal{J}_1^* \approx 0,125\ S, \tag{9}$$

то есть за один прогон текста число ℓ -грамм, подлежащих дальнейшему анализу, сокращается на величину порядка 0,125 S .

Полученная оценка справедлива до тех пор, пока длина анализируемого текста в результате ρ^* прогонов не уменьшится до размеров одной группы: $N(\rho^*) \approx \frac{S}{3}$. Начиная с этого момента, число обращений к адресному полю размером $\frac{S}{3}$ уже не будет постоянной величиной, равной $\frac{S}{3}$, а полностью определяется уменьшающейся длиной анализируемого текста $N(\rho)$ (ρ -номер прогона). Количество прогонов текста ρ^* ,которое потребуется для того, чтобы сократить длину обрабатываемого текста до размеров $\frac{S}{3}$, определяется из соотношения

$$\rho^* \approx \frac{N - N(\rho^*)}{\mathcal{J}_1^*} = e\left(\frac{3N}{S} - 1\right). \tag{10}$$

На последующих прогонах ($\rho = \rho^*+1,\ \rho^*+2, \ldots$) изменение величин $\mathcal{J}_1(\rho)$ и $N(\rho)$ описывается системой рекуррентных соотношений:

$$\mathcal{J}_1(\rho+1) \approx N(\rho)\, e^{-\frac{3N(\rho)}{S}},$$

$$N(\rho+1) = N(\rho) - \mathcal{J}_1(\rho+1). \tag{11}$$

Окончанию процесса соответствует уменьшение длины текста, подлежащего обработке, до размера

$$N(\rho^{**}) = \frac{S}{\ell\,]\log_2 n[} \quad , \tag{12}$$

где $]x[$ означает наименьшее целое, большее или равное x . При выполнении этого условия все оставшиеся ℓ -граммы могут быть

выписаны посимвольно в оперативной памяти объемом S, упорядочены с использованием линейной (в функции от N) по затратам процедуры внутренней сортировки и подсчитаны. Этому моменту соответствует номер прогона ρ^{**}, после которого будет справедливо приближенное соотношение

$$N(\rho^{**}) \approx N(\rho^{*}) - \sum_{\rho=\rho^{*}+1}^{\rho^{**}} \mathcal{J}(\rho). \qquad (13)$$

Интересующая нас добавка в числе прогонов, равная $\rho^{**} - \rho^{*}$, составляет незначительную долю от ρ^{*} и практически не зависит от N и S (параметр S автоматически исключается из обеих частей соотношения (13)). Имеющейся слабой зависимостью $(\rho^{**} - \rho^{*})$ от ℓ и n можно пренебречь. Представление о характере изменения величин $\mathcal{J}(\rho)$ и $N(\rho)$ дает таблица I, полученная на основании рекуррентных соотношений (II) при начальном условии $N(\rho^{*}) \approx S/3$.

Таблица I

Изменение параметров $\mathcal{J}$ и N на последних прогонах текста

ρ	$\mathcal{J}(\rho)$	$N(\rho)$
$\rho^{*}+1$	$\frac{S}{3e} \approx 0{,}125S$	$\frac{S}{3}\frac{e-1}{e} \approx 0{,}2S$
$\rho^{*}+2$	$\frac{S}{3}\frac{e-1}{e}\exp\left(-\frac{e-1}{e}\right) \approx 0{,}11S$	$\frac{S}{3}\frac{e-1}{e}\left(1-\exp\left(-\frac{e-1}{e}\right)\right) \approx 0{,}09S$
$\rho^{*}+3$	$0{,}07S$	$0{,}02S$
$\rho^{*}+4$	$0{,}019S$	$0{,}001S$

Для оценки величин ρ^{*} и ρ^{**} рассмотрим в качестве примера задачу получения распределения IO-грамм на тексте длиной $N = 1{,}5 \cdot 10^{6}$ символов с использованием ЭВМ типа "Минск-32" ($S \approx 0{,}74 \cdot 10^{6}$ бит – 20000 ячеек по 37 разрядов). Мощность исходного алфавита $n = 50$. Имеем: $]\log_2 n[= 6$; $N(\rho^{*}) \approx 0{,}25 \cdot 10^{6}$; $\rho^{*} \approx 14$: $N(\rho^{**}) \approx 0{,}017\,S$; $N(\rho^{*}+3) \approx 0{,}02\,S \approx N(\rho^{**})$, то есть $\rho^{**} - \rho^{*} \approx 3$. Таким образом, на всю обработку потребуется порядка 17 прогонов текста.

С учетом вышеизложенного трудоемкость T предлагаемого метода определяется соотношением

$$T \approx c \cdot N \cdot \rho^{**} \approx c \cdot N \cdot (\rho^{*}+3) \approx c\,\frac{3e\,N^{2}}{S}\,, \qquad (14)$$

где c -константа, определяющая среднее время считывания ℓ -граммы (фактически одного символа) в оперативную память, а также время формирования адреса ℓ -граммы и занесения информации по адресу в соответствии с процедурой ассоциативного кодирования.

Предлагаемый метод не предусматривает записи информации на ленту за исключением сопутствующей информации о типе ℓ -граммы (см. п.4), объем которой составляет N бит, то есть существенно меньше, чем объем самого текста. Эту информацию можно записать в режиме совмещения. Перемотка лент на начало совмещается со считыванием с других лент.

И наконец, операции считывания ℓ -граммы и формирования адреса также могут проводиться в режиме совмещения. Кроме того, операция считывания на каждом прогоне применяется ко всем N ℓ -граммам, в то время как дальнейшей обработке с увеличением номера прогона подвергается все меньшее и меньшее число ℓ -грамм. В связи с этим можно полагать, что константа c в данном методе в основном определяется временем считывания. Учитывая это, для трудоемкости метода получим окончательное выражение

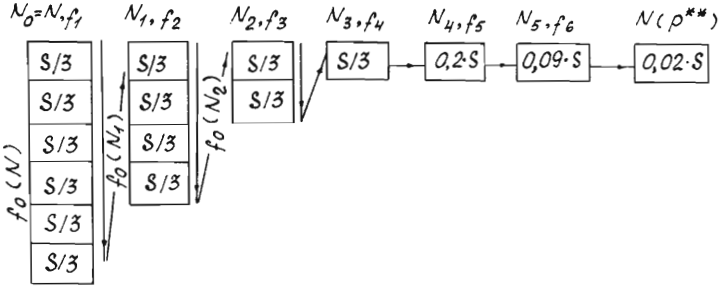
$$T \approx 3 C_{c} \rceil \log_{2} \sqcap \left[e\,\frac{N^{2}}{S} \right. , \qquad (15)$$

где C_{c} - время считывания одного бита информации в оперативную память.

4. Реализация алгоритма.

4.1. Разбиение на группы. Процедура обработки текста (см. схему) основана на том, что исходный текст длины N делится на примерно одинаковые группы. Размер адресного поля, отводимого для процедуры ассоциативного кодирования, равен, как уже упоминалось выше, $S/3$, следовательно, число обращений (размер группы) к этому полю за один прогон текста при оптимальной стратегии выявления единичных ℓ -грамм также должно примерно равняться $S/3$.

Схема разбиения текста на группы, соответствующая последовательному сокращению числа анализируемых ℓ -грамм в процессе обработки



Пусть $\{f_j\}$ ($j=1,\dots,n_f$; n_f – количество различных функций адресации, необходимых для полной обработки текста) – набор последовательно используемых функций адресации, а N_j – количество неисключенных ℓ -грамм после использования функций набора с первой по j -ю включительно.

Примем за основу следующую схему обработки текста. Выявим вначале в тексте все те единичные ℓ -граммы, которые позволяет обнаружить функция адресации f_1 , сократив тем самым длину текста, подлежащего дальнейшему анализу, до N_1 . Оставшиеся N_1 ℓ -грамм вновь разобьем на группы и применим для их обработки функцию f_2 и т.д., до тех пор, пока после использования n_f функций не получим текст, все ℓ -граммы которого могут быть в явном виде выписаны в оперативной памяти $\left(N_{n_f} \approx \frac{S}{\ell\ ]\log_2 n[}\right)$.

Для разбиения на группы используем какую-либо простейшую функцию адресации f_0 и решающее правило типа: ℓ -грамма λi ($i=1,\dots, N_j-1$) принадлежит k -й группе $\left(k = 1,2,\dots, N_{j-1}/\frac{S}{3}\right)$, если

$$\frac{S}{3}(k-1) < f_0(\lambda i) \le \frac{S}{3}k. \qquad (16)$$

Очевидно, что процедура разбиения на группы обусловлена лишь ограниченностью оперативной памяти. Как только размер текста сократится до величины $N_j \approx \frac{S}{3}$, надобность в таком разбиении отпадает, и на каждом последующем прогоне к остатку тек-

ста должна применяться новая функция адресации (одной из них может быть функция f_0). Количество прогонов, которые необходимы для того, чтобы сократить длину текста от N_{j^*} до N_{n_f}, уже оценивалось выше (порядка трех). Следовательно, количество необходимых для анализа функций адресации составит величину

$$n_f \approx j^* + 3. \qquad (17)$$

Оценка величины j^* может быть получена из следующих соображений. За один прогон текста мы обрабатываем ℓ-граммы лишь одной группы (на схеме каждая группа обозначена клеточкой). При этом каждая используемая функция адресации f_j выделяет на одной группе порядка $\frac{S}{3e}$ единичных ℓ-грамм (см. формулу (6)), а на всех своих $\frac{3N_{j-1}}{S}$ группах, то есть на всем оставшемся тексте, примерно N_{j-1}/e единичных ℓ-грамм. Таким образом, после использования j функций адресации длина текста сократится до величины

$$N_j = N_{j-1} - \frac{N_{j-1}}{e} = \frac{e-1}{e} N_{j-1}. \qquad (18)$$

Используя полученное рекуррентное соотношение и начальное условие $N_0 = N$, можно выписать выражение для N_j в явном виде:

$$N_j = \left(\frac{e-1}{e}\right)^j N. \qquad (19)$$

Оценку j^* теперь можно получить из приближенного соотношения

$$\left(\frac{e-1}{e}\right)^{j^*} N \approx \frac{S}{3},$$

откуда

$$j^* \approx \left\lceil \ln \frac{S}{3N} \Big/ \ln\left(\frac{e-1}{e}\right) \right\rceil. \qquad (20)$$

Полученная оценка зависит лишь от соотношения между S и N. Для рассматривавшегося выше примера ($S = 0{,}74 \cdot 10^6$; $N = 1{,}5 \cdot 10^6$) $j^* \approx 4$. С учетом (17) получим $n_f = 7$, то есть для анализа текста с указанными выше значениями параметров S и N понадобится 7 различных функций адресации.

Вместо набора различных функций $\{f_j\}$ можно использовать всего одну функцию адресации, которая зависит от некоторого параметра. Более того, можно использовать одну и ту же функцию и для разбиения текста на группы, и для разбрасывания ℓ -грамм выделенной группы в адресном поле размером $\frac{S}{3}$.

Скажем, если в качестве параметра использовать величину N_j , то, определив функцию f_0 в виде

$$f_0(\lambda_i) = x_i \pmod{N_j}, \quad i = 1, \ldots, N_{j-1}, \qquad (21)$$

где x_i – числовой код, соответствующий ℓ -грамме λ_i , мы получаем разбиение текста на группы в соответствии с решающим правилом (16). Одновременно с этим величина $f_0(\lambda_i) \pmod{\frac{S}{3}}$ определяет адрес пары разрядов, отводимых для данной ℓ -граммы в адресном поле.

Поскольку при каждом новом разбиении на группы величина N_j меняется, то ℓ -граммы, попавшие в зону неоднозначности при предыдущем разбиении, получат при следующем разбиении новые случайные адреса, не говоря уже о том, что и попасть они могут в разные группы. Это эквивалентно использованию для разбрасывания новой функции адресации.

Когда размер текста сократится до величины $\frac{S}{3}$, адресацию можно проводить в соответствии с выражением (21), подставив вместо N_j величину $\frac{S}{3}$ и незначительно варьируя ее от прогона к прогону. Это ведет к некоторому отклонению от оптимального режима выявления единичных кодов, но позволяет не привлекать новых функций адресации.

4.2. О б р а б о т к а ℓ – г р а м м к а ж д о й г р у п п ы . Предположим, что на данном прогоне обрабатывается k -я группа ℓ -грамм посредством использования функции адресации f_j (или $f_0(N_j)$ с учетом того, что было сказано выше).

Будем считать, что информация об ℓ -граммах, которые подлежат дальнейшей обработке, содержится в сопутствующей тексту условной ленте, состоящей из N двоичных символов, где единицы стоят на месте уже выявленных единичных ℓ -грамм, а нули на месте ℓ -грамм, подлежащих дальнейшему анализу.

Использование сопутствующей двоичной информационной ленты обусловлено тем фактором, что, если бы мы отказались от нее, нам

пришлось бы вместо слитного исходного текста выписывать в па - мять в явном виде все ℓ -граммы, о которых еще не выяснено, являются они единичными или нет. Это привело бы к увеличению памяти, а следовательно, и времени считывания, определяющего в основном трудоемкость алгоритма, примерно в ℓ -раз. Ясно, что отказ от сопутствующей информационной ленты был бы целесообразен к моменту, когда объем текста, подлежащего дальнейшей об - работке, был бы сокращен примерно в ℓ -раз, но фактически это происходит где-то на самых последних шагах обработки.

Оперативная память S используется следующим образом. $\frac{S}{3}$ разрядов хранят информацию о единичных ℓ -граммах, выявленных при обработке предыдущей $(k-1)$-й группы. При обработке k -й группы эта информация переносится на двоичную ленту, сопутствующую тексту. $\frac{2S}{3}$ разрядов служат адресным полем для процедуры адресации f_j ℓ -грамм k -й группы (по 2 разряда на ℓ -грам - му). Об информации, заносимой в каждую пару разрядов, упомина - лось выше. По окончании обработки k -й группы эта информация сжимается в 2 раза посредством отображения $00 \to 0, 10 \to 0, 01 \to 1$ (легко видеть, что такое отображение можно осуществить лишь по - сле того, как вся k -я группа будет обработана) и переписыва - ется в первые $\frac{S}{3}$ разрядов.

Анализу на каждом прогоне подвергаются лишь те ℓ -граммы, которым соответствует ноль в сопутствующей информационной лен - те. К соответствующей ℓ -грамме применяется функция адресации f_0 и определяется, принадлежит ли ℓ -грамма к группам $(k-1)$ или k . Если она не принадлежит к этим группам, то ее обра - ботка на данном прогоне заканчивается и осуществляется переход на следующую помеченную нулем ℓ -грамму.

Если ℓ -грамма принадлежит группе $k-1$, то в соответст - вии с тем, как это было сказано в предыдущем разделе этого па - раграфа, формируется её адрес, лежащий в диапазоне от I до $\frac{S}{3}$, и проверяется содержимое соответствующего разряда в первых $S/3$ разрядах памяти. Наличие единицы в этом разряде свидетельству - ет о том, что ℓ -грамма единичная (это было выяснено на преды - дущем прогоне), и эта информация заносится в сопутствующую ин - формационную ленту, после чего начинается обработка следующей ℓ -граммы.

Если ℓ -грамма принадлежит группе k, то формируется ее адрес, лежащий в диапазоне от I до $\frac{S}{3}$, и в соответствующую пару разрядов, расположенную во второй части оперативной памяти ($\frac{2S}{3}$), заносится информация о типе ℓ -граммы (OI, если она встретилась впервые, и IO - в противном случае). Затем осуществляется переход на следующую ℓ -грамму.

5. Сравнение по трудоемкости предложенного метода с альтернативными. Реальный интерес представляет лишь сравнение с методом, основанным на использовании традиционной процедуры " hash coding ", и методом, использующим внешнюю сортировку. Корреляционный метод (или метод протяжек) по своей трудоемкости представляется практически нереализуемым. Что же касается нумерационного метода (речь идет об однозначной формульной нумерации), то для интересующего нас случая ($n^{\ell} >> S$) соответствующий подход, насколько нам известно, пока не разработан.

Для сравнения по трудоемкости предложенного метода ассоциативного кодирования с методом, использующим традиционную процедуру " hash coding ", достаточно сравнить число ℓ -грамм, однозначно классифицируемых за один прогон текста в каждом из методов. Описанный выше метод позволяет выявить за один прогон $\mathcal{J} \approx 0{,}125\ S$ единичных ℓ -грамм. Используя традиционную процедуру " hash coding ", за один прогон текста можно классифицировать не более чем $\mathcal{J} = \frac{S}{\ell\,] \log_2 n [}$ ℓ-грамм, то есть столько, сколько их поместится в явном виде в оперативной памяти. На самом деле эта оценка сильно завышена, поскольку не учитывается, что часть памяти (зачастую близкая к половине) уходит на организацию списковой структуры (адреса пересылок и счетчики для каждой ℓ -граммы).

Подставляя в интересующем нас случае для ℓ среднее значение оценки порога ℓ_0 ($\bar{\ell}_0 \approx$ I3) и считая $] \log_2 n [= 6$, получаем значение $\mathcal{J} \approx 0{,}0I3\ S$, то есть величину, на порядок меньшую, чем $\mathcal{J}_1^*$.

Более перспективным, нежели традиционный метод " hash coding", выглядит метод, использующий внешнюю сортировку. Рассмотрим в качестве примера один из наиболее известных методов

упорядочивания, реализующий слияние частей массива, расположенных на разных лентах, так называемый метод фон Неймана [7].

Предполагается, что исходный массив записан неупорядоченными порциями на входной ленте. В начале процесса выполняется подготовительный этап, состоящий в последовательном считывании этих порций в оперативную память, упорядочивании элементов внутри каждой порции одним из методов внутренней сортировки и поочередном записывании порций на две выходные ленты (рассматривается модификация метода фон Неймана, использующая четыре ленты). Затраты на этом этапе пропорциональны длине текста: $T_0 \approx c_0 \cdot N$. Каждая из исходных упорядоченных порций содержит при этом $\frac{S}{\ell \rceil \log_2 n \lceil}$ ℓ-грамм. Число порций в массиве длиной N составляет величину $L \approx N\ell \rceil \log_2 n \lceil / S$.

В дальнейшем на каждом этапе обработки используются две входные и две выходные ленты, роли которых меняются после очередного прогона текста. В оперативной памяти выделяются три кармана. Вся последующая обработка заключается в последовательном слиянии порций, полученных в результате первого подготовительного этапа. В итоге каждого прогона удваиваются размеры порций, состоящих из упорядоченных элементов.

При этой процедуре элементы двух объединяемых порций последовательно считываются с каждой из входных лент в закрепленный за этой лентой карман. Затем данные из этих карманов сливаются, заполняя третий карман. Как только он заполнится, его содержимое п е р е п и с ы в а е т с я на одну из выходных лент, и затем слияние данных из первых двух карманов продолжается. Когда входные ленты оказываются исчерпанными и завершится формирование выходных лент, очередной этап упорядочивания заканчивается. Входные и выходные ленты обмениваются ролями и п е р е м а т ы в а ю т с я на начало.

Число прогонов текста при обработке данным методом с помощью четырех лент примерно равно $\log_2 L$. Общие затраты на всех этапах составят величину

$$T_{сорт} \approx (c_1 + c_2 + c_3) \log_2 L + c_0 N, \qquad (22)$$

где c_1 – константа, определяющая время считывания одной ℓ-граммы в оперативную память и время ее обработки в процессе упоря-

дочивания; c_2 – константа, определяющая время записи ℓ –граммы на ленту; и c_3 – константа, характеризующая скорость перемотки.

Обычно [7] "обработка содержимого кармана в оперативной памяти (например, упорядочивание считанной в него порции данных или слияние с другим массивом) происходит быстрее, чем обмен между этим карманом и зоной ленты. Поэтому скорость решения задачи, использующей ленты, в основном или даже полностью (если возможно совмещение) определяется суммарной продолжительностью операций обмена". Применительно к (22) это означает, что под c_1 будем понимать просто константу, характеризующую скорость считывания.

Константа записи c_2, как правило, больше или равна константе считывания c_1, нередко превышая ее в 2-3 раза (запись с контролем). Константа c_3, характеризующая скорость перемотки, обычно близка к c_1. Учитывая вышеизложенное, а также тот факт, что операции считывания, записи и перемотки не могут быть совмещены в описываемом алгоритме, получим следующую оценку трудоемкости метода внешней сортировки (добавкой $c_0 N$ в (22) пренебрегаем):

$$T_{copm} \geq 3c_1 N \log_2 \frac{N\ell\,]\log_2 n[}{S} \quad . \qquad (23)$$

Учитывая, что в данном методе каждая ℓ –грамма должна в явном виде выписываться в оперативной памяти, то есть связность текста нарушается в отличие от метода ассоциативного кодирования, выражение (23) можно переписать в виде:

$$T_{copm} \geq 3c_c\,]\log_2 n\,[\cdot \ell \cdot N \cdot \log_2 \frac{N\ell\,]\log_2 n[}{S} \quad . \qquad (24)$$

Отсюда для отношения трудоемкости алгоритмов, использующих метод ассоциативного кодирования и метод внешней сортировки, получаем

$$\frac{T}{T_{copm}} \approx \frac{\ell N}{\ell \cdot S \cdot \log_2 \frac{N\ell\,]\log_2 n[}{S}} \quad . \qquad (25)$$

Сравнивая выражения (I5) и (24), видно, что при фиксированном S и $N \to \infty$ оба метода являются нелинейными, причем пред –

почтительнее оказывается метод, использующий внешнюю сортировку. Однако следует учитывать, что в практических приложениях мы имеем дело хотя и с большими, но с конечными значениями N, в частую сравнимыми с величиной S, стоящей в знаменателе, как в выражении (I5), так и в (24). Применительно к этой ситуации нелинейность, содержащаяся в (I5), будет проявлена весьма слабо, и при фиксированном объеме оперативной памяти S может быть указана весьма широкая область значений $N < N_S$, для которой метод ассоциативного кодирования оказывается существенно экономичнее, чем метод внешней сортировки.

Можно считать, что в указанной области ($N < N_S$) трудоемкость предложенного метода квазилинейно зависит от длины текста N. Граница области N_S может быть определена из приближенного соотношения $T(N_S) \approx T_{copm}(N_S)$. Получающееся при этом нелинейное (относительно N_S) уравнение легко может быть решено методом подстановок (см. разобранный ниже пример).

Для иллюстрации в табл. 2 приведены результаты численного расчета отношения $\frac{T}{T_{copm}}$ в функции от длины текста N при двух фиксированных значениях S. Параметры ℓ, n и S взяты из уже рассматривавшегося выше примера ($n = 50$; $\ell = 10$; $S_1 = 0{,}74 \cdot 10^6$ бит; $S_2 = 2 \cdot S_1 = 1{,}48 \cdot 10^6$ бит).

Таблица 2

N		$0{,}75 \cdot 10^6$	$1{,}5 \cdot 10^6$	$3 \cdot 10^6$	$6 \cdot 10^6$	$9 \cdot 10^6$	$12 \cdot 10^6$
$\frac{T}{T_{copm}}$	$S = S_1$	0,05	0,08	0,14	0,25	0,35	0,44
	$S = S_2$	–	0,05	–	0,14	–	0,25
N		$18 \cdot 10^6$	$24 \cdot 10^6$	$30 \cdot 10^6$	$42 \cdot 10^6$	$54 \cdot 10^6$	$62 \cdot 10^6$
$\frac{T}{T_{copm}}$	$S = S_1$	0,62	0,8	0,98	–	–	
	$S = S_2$	0,35	0,44	0,54	0,72	0,9	≈ 1

Как видно из анализа этой таблицы, диапазон значений N, для которого метод ассоциативного кодирования оказывается экономичнее, чем метод внешней сортировки, весьма широк ($N_{S_1} \approx 3 \cdot 10^7$, $N_{S_2} \approx 6{,}2 \cdot 10^7$). Эффективность метода ассоциативного кодирования

растет практически линейно с увеличением объема оперативной памяти S. В то же время увеличение S в методе, использующем внешнюю сортировку, не приводит к существенному росту эффективности метода. Это является причиной расширения области предпочтительности первого метода по сравнению со вторым при увеличении объема оперативной памяти.

Итак, предложен алгоритм получения распределений по частоте встречаемости в тексте ℓ-грамм значительной длины, в основу которого положен метод ассоциативного кодирования ℓ-грамм текста. Алгоритм предназначен для обработки текстов большого объема ($N \sim 10^6 \div 10^7$) и в указанном диапазоне изменения N существенно экономичнее (в среднем на порядок) альтернативных методов.

Подводя итог вышеизложенному, уместно сделать несколько замечаний общего характера.

1. Получив распределение по частоте встречаемости для ℓ-грамм, мы фактически решили аналогичную задачу и для $(\ell + \ell')$-грамм ($\ell' = 1,2,\ldots$). Это вытекает из того очевидного факта, что единичные ℓ-граммы переходят в единичные $(\ell + \ell')$-граммы. Следовательно, этап вычеркивания единичных $(\ell + \ell')$-грамм может быть опущен, и искомое распределение можно получить, дополнив выписанные на последнем этапе в явном виде ℓ-граммы до $(\ell + \ell')$-грамм и упорядочив их с помощью линейной процедуры внутренней сортировки.

2. Незначительно усложнив процедуру обработки, можно предложить модификацию алгоритма, выявляющую на каждом шаге обработки 3 типа ℓ-грамм: единичные, двойные наложившиеся и все остальные. Оптимизация процедуры обработки сводится в этом случае к максимизации числа ($\mathcal{T}_1 + \mathcal{T}_2$) единичных и дважды наложившихся ℓ-грамм. Тогда при оптимальном отношении $\frac{q}{Q} = \sqrt{2}$ эффективность алгоритма возрастает примерно на 12%.

3. Алгоритм с успехом может быть использован для упорядочивания по частоте встречаемости элементов произвольного информационного массива со значительным числом единичных элементов. Вследствие того, что мы имеем дело уже не со связным текстом, его эффективность падает в ℓ раз (добавляется множитель ℓ в выражении (15)), но диапазон предпочтительности по сравнению с

методом внешней сортировки продолжает оставаться достаточно большим.

4. Легко получить обобщение алгоритма для случая неединичного (по частоте встречаемости ℓ -граммы в тексте) порога. Это достигается благодаря увеличению числа разрядов, отводимых для кодирования информации об одной ℓ -грамме. Такая процедура эквивалентна уменьшению S, за счет чего эффективность алгоритма несколько падает, но зато расширяется сфера применимости алгоритма за счет ослабления ограничения, сформулированного во введении.

Литература

I. ГУСЕВ В.Д., КОСАРЕВ Ю.Г., ТИТКОВА Т.Н. О задаче поиска повторяющихся отрезков текста. -Настоящий сборник, с.49-71.

2. ЁЛКИНА В.Н., ЮДИНА Л.С. Статистика слогов русской речи. -В кн.: Вычислительные системы. Вып. 10, Новосибирск, 1964. с. 58-78.

3. ЁЛКИНА В.Н., ЮДИНА Л.С., ХАЙРЕТДИНОВА А.Г. Статистика двух-и трехфонемных сочетаний русской речи. -В кн.: Вычислительные системы. Вып. 37. Новосибирск, 1969, с. 48-74.

4. Сб."Статистика речи и автоматический анализ текста",под редакцией Р.Г.Пиотровского. Изд-во "Наука", Ленинград, 1971.

5. ШЕННОН К. Предсказание и энтропия печатного английского текста. -"Работы по теории информации и кибернетики", М., ИЛ., 1963.

6. ЕВРЕИНОВ Э.В.. КОСАРЕВ Ю.Г.. УСТИНОВ В.А.Применение ЭВМ в исследовании письменности древних майя, т.[1-4] Новосибирск, Изд-во СО АН СССР, 1961-1969.

7. ЛАВРОВ С.С., ГОНЧАРОВА Л.И. Автоматическая обработка данных. Хранение информации в памяти ЭВМ. М., Изд-во "Наука", 1971.

8. ВЕЛИЧКО В.М., ГУСЕВ В.Д., КОСАРЕВ Ю.Г., ЛОЗОВСКИЙ В.С. ТИТКОВА Т.Н. Ассоциативное кодирование: реализация и применение. -Настоящий сборник, с. 3-37.

9. БЕРНШТЕЙН С.Н. Теория вероятностей. М., Гостехиздат, 1946.

10. СЕВАСТЬЯНОВ Б.А., ЧИСТЯКОВ В.П. Асимптотическая нормальность в классической задаче о дробинках. -"Теория вероятностей и ее применения". 1964, т. IX, вып. 2.

11. LUM V.Y.,YUEN P.S.T., DODD M. Key to Address Transform Techniques: A Fundamental Performance Study on Large Existing Formatted Files. - "Communications of the ACM",1971,vol.14,N 4.

Поступила в ред.-изд.отд.
I июня 1973 года