УДК 51:621.396.801

О ЗАДАЧЕ ПОИСКА ПОВТОРЯЮЩИХСЯ ОТРЕЗКОВ ТЕКСТА

В.Д.Гусев, Ю.Г.Косарев, Т.Н.Титкова

В задачах, связанных с распознаванием образов [1], с машинным переводом [2], с кодированием и поиском информации [3,4], а также во многих других зачастую приходится иметь дело с анализом статистических свойств текстовых или символьных последовательностей значительной длины. Такой анализ, в частности, включает в себя построение распределений по частоте встречаемости ℓ-грамм-связных подпоследовательностей длины ℓ ($\ell = 1,2,3,\ldots$).

Известные в настоящее время статистики ℓ-грамм [3,5,6] получены, как правило, для малых значений ℓ (биграммы, триграммы, слова) на текстах сравнительно небольшой длины N (10^4, редко $10^5 - 10^6$ символов). Это объясняется главным образом тем, что при больших значениях ℓ и N объем вычислений T у применяемых алгоритмов начинает нелинейно расти с увеличением длины текста ($T \sim N \log N$ или даже $\sim N^2$) [7].

В связи с этим предпринята попытка 1) проанализировать, какие параметры текста и используемых вычислительных средств существенны для выбора алгоритма получения распределений ℓ-грамм по частоте встречаемости, и 2) разработать алгоритмы получения вышеуказанных распределений с объемом вычислений, л и н е й н о (или хотя бы к в а з и л и н е й н о) зависящим от длины N анализируемого текста в достаточно ш и р о к о м диапазоне возможных значений ℓ и N. Ответ на первый и частично второй вопросы содержится в данной статье. Дополнением к ней являются работы [8,9].

Введем следующие обозначения:

N – объем (длина) обрабатываемого текста (в символах исходного алфавита);

ℓ – грамма – подпоследовательность текста, содержащая ℓ расположенных подряд (без пропуска) символов ($\ell = 1,2,3,\ldots$);

A_0 – исходный алфавит, т.е. полный набор различных символов, из которых составлен текст;

$n = |A_0|$ – мощность исходного алфавита A_0;

A_ℓ – алфавит ℓ-грамм, т.е. набор всевозможных комбинаций из ℓ символов исходного алфавита, отличающихся друг от друга по составу или по порядку следования элементов ($\ell = 1,2,3,\ldots$; A_1 – упорядоченный в соответствии с частотой встречаемости исходный алфавит);

$|A_\ell|$ – мощность алфавита A_ℓ ($\ell = 1,2,3,\ldots$);

a_i^ℓ – элементы алфавита A_ℓ ($\ell = 0,1,2,3,\ldots$; $i = 1,2,\ldots, |A_\ell|$);

M_ℓ – число различных ℓ-грамм в тексте (M_ℓ может не совпадать с $|A_\ell|$ ввиду наличия запрещенных комбинаций, либо комбинаций разрешенных, но отсутствующих в данном конкретном тексте);

E_ℓ^k – количество различных ℓ-грамм, каждая из которых встретилась в тексте ровно k раз ($k = 0,1,2,\ldots$);

$F_\ell(m)$ – закон распределения частоты встречаемости в тексте различных ℓ-грамм; m – порядковый номер ℓ-граммы, соответствующий её месту в ранжированном по абсолютной величине ряду частот; $m = 1,2,3,\ldots, M_\ell$; $\ell = 1,2,3,\ldots$;

S – объем оперативной памяти (ОП) в битах, отводимой под рабочее поле программы;

T_ℓ – трудоемкость алгоритма (число операций).

1. П р е д в а р и т е л ь н ы е з а м е ч а н и я. Можно указать несколько подходов к задаче построения распределений ℓ-грамм по частоте встречаемости. Перечислим их, снабдив соответствующими названиями*), отражающими, на наш взгляд, идею метода.

1.1. М е т о д п р о т я ж к и (корреляционный метод). Выделяем первую ℓ-грамму и, последовательно сдвигая её на одну позицию от начала текста к концу, осуществляем каждый раз сравнение данной ℓ-граммы с соответствующей ℓ-граммой текста, фиксируя число совпадений. Аналогичную операцию проделываем с каждой из оставшихся ($N-\ell$) ℓ-грамм, которые не подвергались анализу на одном из предыдущих шагов. В общем случае число сравнений в этом методе (T_ℓ) пропорционально квадрату длины N текста, т.е. при значениях $N \sim 10^6$ и выше метод уже практически не реализуем на современных ЭВМ.

*) Установившейся терминологии в этой области пока не существует.

1.2. М е т о д с о р т и р о в к и. Все ℓ-граммы текста выписываются посимвольно и упорядочиваются (скажем, лексикографически) с использованием процедуры внутренней сортировки [10], линейно зависящей от N, если N относительно невелико, либо н е л и н е й н о й по N процедуры внешней сортировки [11], когда N в е л и к о. Искомые частотные распределения легко получаются на основе упорядоченного массива.

1.3. Н у м е р а ц и о н н ы й м е т о д. Это название отражает традиционно используемый для набора статистики метод, заключающийся в том, что каждой из n^ℓ возможных ℓ-грамм отводится свой счетчик в ОП. Задача набора статистики сводится при этом к установлению взаимно-однозначного соответствия (нумерации) между значением соответствующей ℓ-граммы и номером счетчика, в который заносится информация об этой ℓ-грамме.

Эта задача решается тривиально в случае малых ℓ с линейными по N затратами. Однако с увеличением ℓ мощность алфавита ℓ-грамм нарастает экспоненциально. При этом ввиду ограниченности ОП уже невозможно снабдить каждую ℓ-грамму алфавита "персональным" счетчиком. К тому же такая стратегия оказалась бы явно нерациональной по расходу памяти, ввиду того, что, как правило, не все возможные ℓ-граммы реально присутствуют в тексте (их всего $N-\ell+1$, включая повторяющиеся).

1.4. М е т о д а с с о ц и а т и в н о г о к о д и р о в а н и я является естественным развитием нумерационного метода. Сохраняя достоинства последнего (быстрый поиск нужного счетчика), он позволяет в то же время более рационально использовать ОП за счет выделения счетчиков не под каждую из возможных ℓ-грамм, а лишь под те из них, которые реально присутствуют в тексте. Общая идея метода и оценки его трудоемкости содержатся в [12]. Анализ возможностей метода применительно к за-

даче поиска повторяющихся отрезков текста содержится в настоящей работе и в [8].

2. Анализ существенных параметров. Трудоемкость алгоритма отыскания частот встречаемости в тексте ℓ -грамм произвольной длины зависит от значений параметров N, n, ℓ, M_ℓ, априорной информации о характере распределений ℓ - грамм по частоте встречаемости $F_\ell(m)$, от параметров технических средств и, в первую очередь, от объема S ОП, а также типов и числа вспомогательных запоминающих устройств. Ниже влияние этих факторов рассматривается более подробно. При этом в первую очередь нас будет интересовать, при каком соотношении между значениями перечисленных параметров начинается нелинейный рост объема вычислений. Забегая вперед, отметим, что каждому из параметров соответствует некоторая область значений, для которой при широком диапазоне изменений других параметров может быть указан эффективный метод решения рассматриваемой задачи, и существуют содержательные примеры практических задач для каждой области.

2.1. Мощность исходного алфавита n. Величина n непосредственно определяет мощность алфавитов ℓ -грамм: $|A_\ell| = n^\ell$. Чем меньше n, тем для больших ℓ выполняется условие

$$n^\ell \le c_S , \qquad (2.1)$$

где c_S - количество счетчиков, которое может быть размещено в ОП объемом S бит. Для любого ℓ, удовлетворяющего (2.1), используя нумерационный метод, можно получить искомое распределение за один "прогон" текста. Под "прогоном" здесь понимается процедура просмотра каждой ℓ -граммы текста, включающая в себя считывание ℓ -граммы в ОП и её анализ. Трудоемкость прогона является линейной функцией N.

Требуемое число и емкость счетчиков c_S определяются не только мощностью алфавита n^ℓ, но и длиной текста N, а также априорными сведениями о характере распределения $F_\ell(m)$. По аналогии с эффективным кодированием [13] легко видеть, что наибольшая ОП потребуется для размещения информации о распределениях $F_\ell(m)$, близких к равномерному (наименее избыточному) закону распределения, при котором каждая из n^ℓ возможных комбинаций встречается в тексте одинаковое (N/n^ℓ) число раз.

Очевидно, что объем вычислений T будет линейной функцией длины текста N, если размер ОП удовлетворяет соотношению

$$S \ge k_\ell \cdot n^\ell \cdot \left] \log_2 \frac{N}{n^\ell} \right[, \qquad (2.2)$$

где $]x[$ означает наименьшее целое, большее или равное x, а k_ℓ - коэффициент, учитывающий неравномерность распределения ($0 < k_\ell \le 1$).

Величина k_ℓ для равномерного распределения равна 1, а в случае неравномерного распределения может быть существенно уменьшена путем адаптивной подстройки под имеющуюся неравномерность. Алгоритм построения счетчиков переменной длины, реализующий такую подстройку, описан в [17].

Неравенство (2.2) выделяет в пространстве интересующих нас параметров область, для которой объем вычислений линейно зависит от N. Эта область характеризуется либо малой мощностью исходного алфавита, либо малыми значениями ℓ. На длину текста и характер распределения ℓ -грамм принципиальных ограничений при этом не накладывается. Примером задач для данной области является набор статистики биграмм, триграмм для текстов с $n \approx 30$–50, а также получение распределений по частоте встречаемости блоков длины ℓ в двоичных последовательностях ($n = 2$; $\ell \simeq 2$–15), что представляет интерес при оценке их избыточности [14].

2.2. Длина текста N. В тексте длиной N символов содержится $N-\ell+1$ ℓ -грамм, что уже при сравнительно небольших ℓ может быть значительно меньше их возможного многообразия n^ℓ. Это позволяет при относительно небольших значениях N (сравнимых с объемом ОП) получать статистики ℓ -грамм для достаточно больших значений n, ℓ и произвольных $F_\ell(m)$.

Линейная зависимость объема вычислений от N в этом случае обеспечивается при выполнении неравенства

$$S \ge (N-\ell+1)\cdot\ell \cdot \left] \log_2 n \right[, \qquad (2.3)$$

когда все ℓ -граммы могут быть размещены в ОП, лексикографически упорядочены с использованием линейной по N процедуры внутренней сортировки [10] и подсчитаны. Применение такого подхода для текстов большой длины потребовало бы привлечения про-

цедуры внешней сортировки [II], что означало бы нелинейный рост трудоемкости в зависимости от длины текста ($T \sim N \log_{\alpha} N/S$, где 2α – число лент, используемых при внешней сортировке).

Неравенство (2.3) выделяет в пространстве рассматриваемых параметров область, частично пересекающуюся (при малых N, n и ℓ) с областью, определяемой неравенством (2.2). Однако при больших значениях N, n и ℓ эти области существенно расходятся. Область, определяемая неравенством (2.2), вытянута вдоль оси N, а область, определяемая неравенством (2.3), вытянута по осям n и ℓ. Содержательными примерами задач, удовлетворяющих неравенству (2.3), могут служить многочисленные задачи упорядочивания и статистики информационных массивов, составляющие один из существенных аспектов функционирования АСУ.

2.3. Число M_ℓ различных ℓ-грамм в тексте. Этот параметр определяется видом распределения $F_\ell(m)$, которое, в свою очередь, изменяется с величиной ℓ. Для естественных языковых систем, как правило, $M_\ell << N$ при небольших значениях ℓ и достаточно больших N. С ростом ℓ величина M_ℓ возрастает и становится близкой к $N - \ell$.

Для небольших значений M_ℓ, когда объем ОП позволяет разместить в ней в явном виде (посимвольно) все различные ℓ-граммы, каждую со своим счетчиком, можно предложить эффективный линейный по N метод решения рассматриваемой задачи. В основе его лежит процедура ассоциативного кодирования [I2], получившая в американской литературе наименование "hash coding".

Эта процедура позволяет указать адрес соответствующей ℓ-граммы (номер счетчика) по её значению. Процедура состоит из двух шагов, первый из которых включает в себя собственно отображение ℓ-граммы в соответствующий адрес, а второй – устранение неоднозначности адресации, возникающей на первом шаге и характеризующейся тем, что две и более различных ℓ-граммы могут быть снабжены одним и тем же адресом. Выписывание в явном виде всех различных ℓ-грамм и обусловлено необходимостью разделения налагающихся ℓ-грамм.

Область линейности будет определяться при данном подходе соотношением $S \geq S_1 + S_2$, где

$$S_1 = M_\ell \cdot \ell \cdot \;] \log_2 n \; [\qquad (2.4)$$

– объем ОП, требующийся для размещения ℓ-грамм, а

$$S_2 \approx k_\ell \; M_\ell \;] \log_2 \frac{N}{M_\ell} \; [\qquad (2.5)$$

– объем ОП, отводимой под счетчики. Последнее соотношение получается из соображений, аналогичных тем, что и для (2.2). В развернутой форме для зоны линейности имеем неравенство

$$S \geq M_\ell \; (\ell \cdot] \log_2 n \; [+ k_\ell \;] \log_2 \frac{N}{M_\ell} \; [ ), \qquad (2.6)$$

которое выделяет в пространстве параметров область, целиком покрывающую подобласть, определяемую неравенством (2.3). Действительно, при больших ℓ, как уже отмечалось, $M_\ell \approx N$, $\log \frac{N}{M_\ell} \to 0$ и (2.6) переходит в (2.3). При малых ℓ область, определяемая (2.6), охватывает существенно бо́льшие значения N, нежели область, определяемая (2.3) (параметр n при этом может варьироваться весьма широко). Таким образом, в области, определяемой неравенством (2.3), могут работать две различные процедуры набора статистики с линейной зависимостью T от N.

Примером задачи с параметрами, удовлетворяющими неравенству (2.6), является задача получения частот встречаемости слов по достаточно длинным текстам на машине среднего класса ($n \approx 30$; $\ell \approx$ I–I0; $M_\ell \approx I0^4$; $N \approx I0^6$; $S \approx (I0^5 \div I0^6)$ бит).

2.4. Характер распределения ℓ-грамм по частоте встречаемости $F_\ell(m)$. Рассмотрим на примере некоторых е с т е с т в е н н ы х языковых систем, какой вид имеют распределения ℓ-грамм по частоте встречаемости и какова динамика изменения характера распределений с увеличением ℓ.

Как правило, для малых значений ℓ функции $F_\ell(m)$ монотонно и достаточно быстро убывают с увеличением порядкового номера m ℓ-граммы. Количественно закономерность такого типа была сформулирована Ципфом применительно к распределению частоты встречаемости слов в английских текстах (закон Ципфа) [I5]:

$$F(m) = \frac{C}{m^{\alpha}} \; , \qquad (2.7)$$

где C и α – некоторые положительные константы (α близко к I).

С увеличением ℓ мощность алфавита ℓ-грамм растет как n^ℓ. Однако, начиная с некоторого $\ell = \ell^*$, подавляющая часть их будет представлять уже запрещенные комбинации (можно считать, что доля ℓ-грамм, разрешенных, но отсутствующих в данном конкретном тексте, пренебрежимо мала при условии, что N достаточно велико и выборка представительна). Приближенную оценку для ℓ^* можно получить из соотношения $n^{\ell^*} \approx N$, откуда

$$\ell^* \approx \frac{\log N}{\log n} \quad . \tag{2.8}$$

Для рассмотренного выше примера ($n = 50$, $N = 1{,}5 \cdot 10^6$) $\ell^* \approx 4$, что хорошо согласуется с имеющейся информацией по триграммам ($E_3^0 \approx 0{,}82\ |A_3|$, т.е. запрещенных комбинаций уже существенно больше, чем разрешенных).

Очевидно, что с ростом ℓ величина M_ℓ может также лишь увеличиться и при $\ell > \ell^*$ становится уже сравнимой с N. В то же время объем текста, покрываемый ℓ-граммами (площадь под графиком $F_\ell(m)$), остается неизменным. Отсюда следует, что с ростом ℓ число повторяющихся ℓ-грамм ($M_\ell - E'_\ell$) уменьшается и бóльшая часть их вырождается в единичные (однократно встречающиеся). Иными словами, с ростом ℓ распределения $F_\ell(m)$ все в бóльшей степени отклоняются от закона Ципфа и становятся все более равномерными. Динамика изменения распределений частот встречаемости ℓ-грамм с ростом ℓ ($\ell_1 < \ell_2 < \ell_3$) схематически представлена на рис. I.

Начиная с некоторых значений $\ell = \ell^{**}$, величина ($M_\ell - E'_\ell$) становится столь малой, что все повторяющиеся ℓ-граммы могут быть посимвольно выписаны в ОП, упорядочены и подсчитаны. Фактически это означает, что при $\ell > \ell^{**}$ $M_\ell - E'_\ell \to 0$, $E'_\ell \to N$, и задача выписывания повторяющихся ℓ-грамм сводится к устранению из текста единичных ℓ-грамм, относительно которых известно, что они при данном ℓ составляют уже п о д а в л я ю щ у ю часть текста.

Такой подход к построению распределений ℓ-грамм при $\ell > \ell^{**}$ развит авторами в [8]. Алгоритм основан на некоторой модификации традиционной процедуры ассоциативного кодирования, позволяющей как минимум на порядок уменьшить трудоемкость

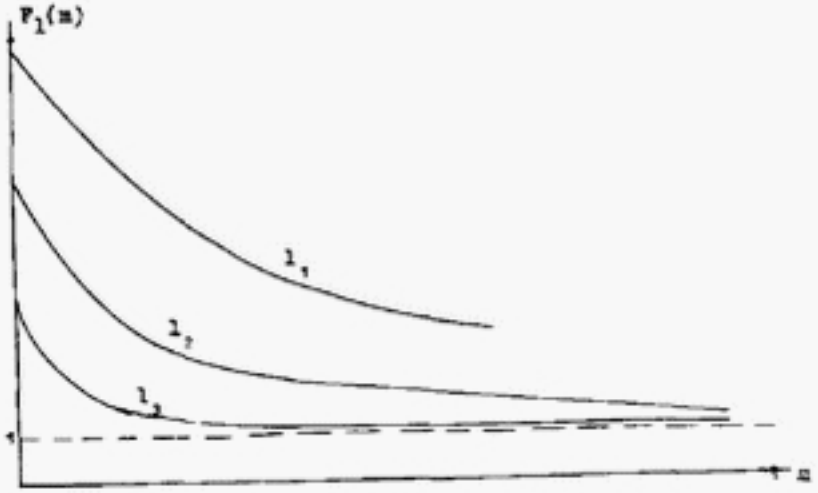


Рис. I

этого метода по сравнению с известными. Полученные в [8] оценки показывают, что в области значений N, сравнимых с объемом ОП S, трудоемкость алгоритма квазилинейно зависит от N.

Приближенную оценку величины ℓ^{**} для значений $N \sim 10^6$ можно получить, исходя из второго закона Ципфа [16], согласно которому

$$\frac{\mathcal{J}_k}{\mathcal{J}_1} \approx \frac{2}{k(k+1)} \quad , \tag{2.9}$$

где $\mathcal{J}_k$ ($k = 1,2,\ldots,k_{max}$) – количество разновидностей слов, каждое из которых встречалось в тексте ровно k раз. Этот закон позволяет оценить число слов, однократно встретившихся в тексте. Действительно, общее количество слов в тексте

$$V = \sum_{k=1}^{k_{max}} k \cdot \mathcal{J}_k \approx 2 \cdot \mathcal{J}_1 \cdot \sum_{k=1}^{k_{max}} \frac{1}{k+1} \approx 2\mathcal{J}_1 \cdot \ln k_{max} \quad . \tag{2.10}$$

Полагая $\mathcal{J}_{k_{max}} = 1$, для k_{max} из (2.9) имеем оценку $k_{max} \approx \sqrt{2\mathcal{J}_1}$, откуда $V \approx \mathcal{J}_1 \cdot \ln 2\mathcal{J}_1$. В интересующем нас диапазоне $N \sim 10^6$ символов, что соответствует $V \sim 2 \cdot 10^5$ слов, последнее соотношение

дает $\gamma_1 \approx 0{,}1\,V$, т.е. в текстах указанного объема примерно десятая часть слов – однократно встречающиеся. Принимая среднюю длину слова ℓ_0 примерно равной 5 – 6 символам и экстраполируя полученную оценку на ℓ-граммы соответствующей длины, имеем $E^1_{\ell_0} \approx 0{,}1\,N$. Можно считать, что единичные ℓ_0-граммы равномерно распределены по тексту, так что средняя длина серий неединичных ℓ_0-грамм между каждой парой единичных примерно равна 9. При переходе от ℓ_0 к ℓ_0+1 эта длина уменьшится по крайней мере на единицу, поскольку каждая единичная ℓ_0-грамма порождает две единичные рядом расположенные (ℓ_0+1)-граммы. На следующем шаге длина каждой серии единичных ℓ-грамм вновь увеличится на единицу за счет соответствующего сокращения длины серий неединичных ℓ-грамм и т.д. В итоге приходим к линейной схеме роста числа единичных ℓ-грамм:

$$E^1_\ell \approx E^1_{\ell_0}(\ell-\ell_0+1), \quad \ell=\ell_0+1, \ell_0+2,\ldots,\ell^{**}. \qquad (2.11)$$

Поскольку $E^1_{\ell^{**}} \approx N$, для ℓ^{**} отсюда получаем

$$\ell^{**} \approx \frac{N}{E^1_{\ell_0}} + \ell_0 - 1, \qquad (2.12)$$

что в рассматриваемом нами случае ($N \sim 10^6$) дает $\ell^{**} \sim 15$.

Более точный анализ, учитывающий различие длин серий неединичных элементов, показывает, что в действительности количество единичных ℓ-грамм в рамках рассматриваемой модели должно нарастать несколько медленнее, чем в соответствии с (2.11). Следует, однако, отметить наличие компенсирующего фактора, не учитываемого в данной модели, а именно: единичные ℓ-граммы возникают не только на стыке серий из единичных и неединичных элементов, но и внутри последних, становясь своего рода новыми "центрами кристаллизации" единичных ℓ-грамм. Указанный фактор особенно существен при малых ℓ. Вследствие этого оценка (2.11) при малых начальных ℓ_0, когда $E^1_{\ell_0}/N$ пренебрежимо мало, может оказаться существенно заниженной.

2.5. Промежуточные выводы. Анализ методов, изложенных в п.1, показывает, что с их помощью относительно легко (с линейными в широком диапазоне изменения N затратами) можно выделить

короткие ($\ell < \ell^*$), но частые, а также длинные ($\ell > \ell^{**}$), но редкие ℓ-граммы [8]. Для значений же $\ell^* \le \ell \le \ell^{**}$ при условии, что не выполняется ни одно из облегчающих предположений, рассмотренных выше, метод, изложенный в [8], становится неэффективен, поскольку единичные ℓ-граммы еще не составляют подавляющую часть текста. Потенциально возможное количество ℓ-грамм уже столь велико, что прямое использование нумерационного метода также потребует большого числа прогонов. Поскольку N велико, операция сортировки может быть только внешней и, следовательно, нелинейной по затратам. И, наконец, процедура ассоциативного кодирования в отличие от модификации, изложенной в [8], требует большой ОП, поскольку необходимо выписывать посимвольно в ОП каждую из равновидностей ℓ-грамм.

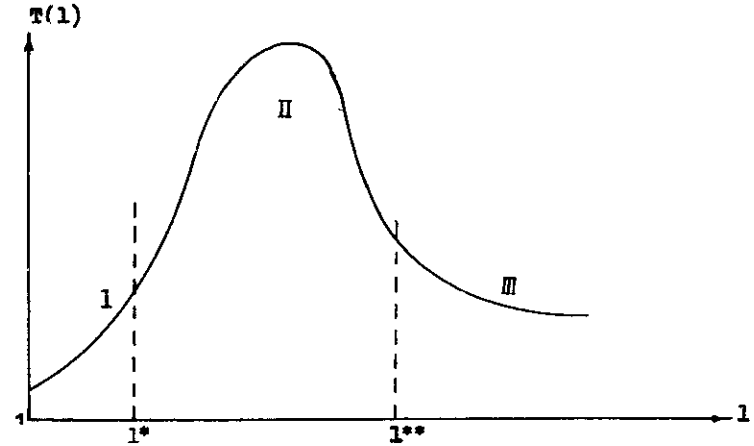


Рис. 2

Таким образом, при получении полного спектра распределений для разных ℓ вычислительные затраты не являются м о н о т о н н о й функцией параметра ℓ и максимум их приходится на диапазон п р о м е ж у т о ч н ы х значений $\ell^* < \ell < \ell^{**}$ (рис.2). Приведенные рассуждения не носят окончательного характера и, быть может, отражают лишь положение, сложившееся на сегодняшний день. Вместе с тем ситуация, когда задача относительно легко решается при малых и при больших значениях некоторого

характеризующего её параметра, но плохо поддается решению в области промежуточных значений этого параметра, является, вообще говоря, довольно типичной (центральная предельная теорема, задача коммивояжера, дифракция в ближней и дальней зонах и т.д.). Это позволяет предполагать здесь существование некоторого информационного принципа, точная формулировка которого, равно как и указание области его применимости, представляет, по всей видимости, нетривиальную задачу.

В заключение отметим, что ограничение на оперативную память S в принципе не позволяет избавиться от нелинейности вычислительных затрат для произвольных значений N ни для одной из областей I,II,III (рис.2). Однако если в случае областей I и III граница нелинейности отстоит весьма далеко ($N \approx 10^6 - 10^7$), что позволяет говорить о линейной (или квазилинейной) зависимости T от N в этих областях, то для II граница нелинейности существенно ниже, чем для I и III.

3. Алгоритм Поиска Повторяющихся Отрезков Текста-I ("АПОРТ-I"). Рассмотрим алгоритм, ориентированный на получение полного спектра распределений, $\ell = 1, \ldots, \hat{\ell}$, где $\hat{\ell}$-любое наперед заданное значение ℓ. В отличие от алгоритма [8], который предназначен для получения распределений лишь для достаточно больших значений $\ell (\ell > \ell^{**})$ и не требует для своего использования информации о распределениях с меньшими значениями ℓ (что позволяет обходить пик вычислительных затрат (область II) в случае, когда соответствующие распределения нас не интересуют), данный алгоритм на каждом последующем этапе использует информацию о распределении ℓ-грамм меньшей длины.

Алгоритм представляет итерационную по ℓ процедуру, на каждом шаге которой получается статистика ℓ-грамм длины, на единицу большей, чем на предыдущем ($\ell = 1,2,3,\ldots$). В основу алгоритма положены следующие (в порядке их использования) процедуры:

1) последовательное перекодирование текста из алфавита $A_{\ell-1}$ в алфавит A_ℓ;

2) набор статистики ($\ell+1$)-грамм с помощью счетчиков переменной длины;

3) статистическая нумерация ($\ell+1$)-грамм;

4) ограничение мощности алфавита на каждом шаге путем введения групповых кодов;

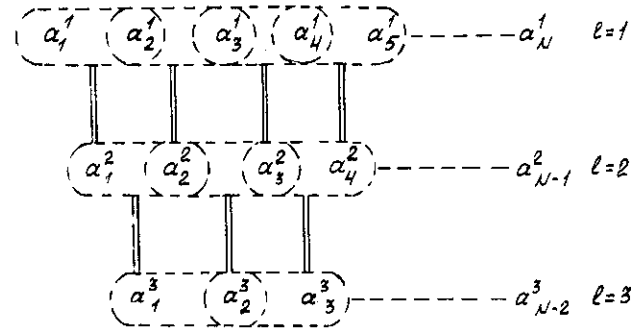
5) введение единичных кодов для исключения из рассмотрения на последующих этапах ℓ-грамм с частотой встречаемости ниже пороговой;

6) классификация ℓ-грамм по последнему (или первому) символу первоначального алфавита;

7) обработка групповых кодов для выявления повторяющихся и единичных ℓ-грамм.

Рассмотрим эти процедуры подробнее:

I. Схема последовательной перекодировки выглядит следующим образом:



и т.д.

Последовательная перекодировка позволяет на каждой итерации алгоритма ($\ell = 1,2,3,\ldots$) представлять анализируемый текст непосредственно в символах алфавита A_ℓ. При этом частично удается избежать соответствующего увеличения объема текста в раз, перекодируя взаимно-однозначно лишь фиксированную часть элементов алфавита A_ℓ и объединяя оставшиеся элементы в группы с единым кодом для всех элементов группы (см. процедуру 4).

Фактически процедура перекодировки представляет модификацию нумерационного метода, приспособленного на случай ограничений по ОП (введение групповых кодов). Она позволяет с помощью перекодировочной таблицы присваивать одинаковым ℓ-граммам один и тот же код, определяющий номер счетчика для соответствующей ℓ-граммы.

К достоинствам последовательной перекодировки следует отнести существенное уменьшение перекодировочной таблицы, поскольку не требуется хранить сами значения ℓ-грамм (см.процедуру 3), и ускорение процессов перекодирования и набора статистики, которые осуществляются по адресу.

2. Благодаря неравномерности частного распределения ℓ-грамм, вместо счетчиков фиксированной длины можно использовать счетчики переменной длины [I7], размещая в ОП гораздо большее их количество. Аналогом этого в теории кодирования является использование кодов Хаффмена [I3] вместо обычных блочных кодов при известной статистике источника. Достоинством данного метода по сравнению с методикой Хаффмена является возможность подстройки под неизвестную статистику источника и быстрый поиск нужного счетчика (естественно, за счет некоторого отхода от оптимальности).

3. Под статистической нумерацией ℓ-грамм понимается нумерация их в соответствии с частотой встречаемости в тексте, причем меньшему значению номера (кода) соответствует бо́льшая частота. Это позволяет легко выделять ℓ-граммы с нужной частотой встречаемости, эффективно (по адресу) набирать статистику и минимизировать затраты на получение статистики ℓ-грамм путем использования статистики $(\ell-1)$-грамм, полученной на предыдущем этапе. Конкретно это выражается в минимизации числа переходов из основного поля в дополнительное при работе со счетчиком переменной длины [I7].

4. Как уже упоминалось выше, экспоненциальный характер роста $|A_\ell|$ с увеличением ℓ не позволяет отводить под каждую из возможных ℓ-грамм "персональный" код, а следовательно, и "персональный" счетчик (размер перекодировочной таблицы, а также рабочее поле счетчиков лимитированы объемом ОП). Поэтому на каждой итерации фиксируется какое-то количество наиболее частых кодов (ℓ-грамм), которым разрешается при следующей итерации иметь для каждого из своих возможных расширений ($(\ell+1)$-грамм) отдельный счетчик. Все оставшиеся коды объединяются в группы, и каждой такой группе присваивается свой групповой код.

Таким образом, введение групповых кодов позволяет набирать на каждой итерации статистику наиболее частых ℓ-грамм, в принципе, без потерь ℓ-грамм с меньшей частотой встречаемости. Поскольку, однако, нас интересует полная статистика всех ℓ-

грамм, то введение групповых кодов, облегчая набор статистики частных ℓ-грамм, приводит и к известным трудностям, связанным с решением обратной задачи – с разделением групповых ℓ-грамм на "персональные" или "истинные".

На каждой последующей итерации эта задача усложняется из-за роста доли текста, покрываемого групповыми кодами. Однако одновременно с этим часть ℓ-грамм, описываемых групповыми кодами, начинает вырождаться в единичные ℓ-граммы, не представляющие интереса для дальнейшего рассмотрения. Этим и обусловлено включение в алгоритм процедуры, описываемой в пункте 5.

5. Введение единичных кодов для ℓ-грамм, частота встречаемости которых ниже фиксированного порога, задаваемого пользователем, является основным средством уменьшения трудоемкости алгоритма при последующих итерациях. Действительно, выявление единичной ℓ-граммы равносильно уменьшению на единицу длины N анализируемого текста на всех последующих итерациях (ℓ-грамма считывается в ОП, но не анализируется), поскольку все возможные расширения единичной ℓ-граммы могут дать не более чем единичную $(\ell+\ell')$-грамму ($\ell' = 1,2,3,\ldots$).

Единичные ℓ-граммы растут в основном за счет групповых кодов. Таким образом, выявление единичных ℓ-грамм существенно сокращает число групповых кодов и облегчает задачу их последующей обработки.

6. Наряду с кодом, присваиваемым каждой ℓ-грамме в процессе перекодировки, сохраняется информация о последнем (или первом) символе ℓ-граммы. Это позволяет на любом шаге итерации восстановить значение ℓ-граммы в символах исходного алфавита. Для однозначности такого восстановления необходимо, чтобы число групп, на которые разбиты все групповые (а также и единичные) коды, было не меньше чем n (по символу исходного алфавита на группу).

Нетрудно показать, что процедуры I–6 могут быть выполнены за один прогон текста и трудоемкость их является линейной функцией от N. Потенциальная нелинейность заложена в процедуре разделения групповых кодов на "истинные" и возникает, как уже говорилось, при значениях $\ell^* < \ell < \ell^{**}$, когда групповые коды составляют значительную долю текста. Для разделения может быть использована либо процедура внешней сортировки, либо

алгоритм, описанный в [8] (в последнем случае предполагается, что ℓ-граммы, представленные групповыми кодами, в большинстве своем единичные).

Имея в виду, что каждый раз мы фиксируем L самых частых ℓ-грамм и набираем полную статистику для всевозможных их расширений (на I элемент), можно оценить количество групповых кодов, подлежащих анализу на ℓ-й итерации:

$$N_\ell^{2P} = N - E_{\ell-1}^1 - \sum_{m=1}^{L} F_{\ell-1}(m). \qquad (3.1)$$

Воспользовавшись выражениями (24) и (I5) из [8], затраты на обработку групповых кодов можно представить либо в виде

$$T_\ell^{2P} \approx C_\cdot \rceil \log_2 n \lceil \cdot \ell \cdot N_\ell^{2P} \cdot \log_2 \frac{N_\ell^{2P} \cdot \ell_\cdot \rceil \log_2 n \lceil}{S} \qquad (3.2)$$

– внешняя сортировка, либо в виде

$$T_\ell^{2P} \approx C_\cdot \rceil \log_2 n \lceil \cdot \ell \cdot e \cdot \frac{(N_\ell^{2P})^2}{S} \qquad (3.3)$$

– алгоритм [8]. Отсюда полные затраты на ℓ-й итерации составят величину

$$T_\ell \approx c_0 \cdot N + T_\ell^{2P}, \qquad (3.4)$$

где c_0 – константа, определяющая затраты на обработку одной ℓ-граммы в соответствии с процедурами I–6.

К недостаткам описанного алгоритма следует отнести: а) нерациональное использование поля счетчиков, обусловленное тем, что не все из возможных расширений ℓ-грамм, фиксируемых на данной итерации, реально присутствуют в тексте (часть счетчиков оказывается пустой); б) дополнительные затраты, связанные с необходимостью слияния статистики наиболее частых ℓ-грамм со статистикой групповых кодов и целесообразностью выявления при перекодировке по результирующей статистике тех ℓ-грамм, которые описывались групповыми кодами, но фактически оказались единичными. Путем некоторого усложнения процедуры обработки указанные недостатки могут быть частично устранены.

4. Алгоритм Поиска Повторяющихся Отрезков Текста-2 ("АПОРТ-2"). Рассмотрим другой алгоритм поиска полного спектра распределений, основанный на идее ассоциативного кодирования [I2] и позволяющий ориентироваться не на все возможные расширения ℓ-грамм (см. п.4, стр.62), а лишь на те из них, которые реально присутствуют в тексте.

Каждая итерация соответствует получению распределения $F_\ell(m)$ для одного из значений ℓ и состоит из последовательных прогонов еще необработанной (или частично обработанной) доли текста. На каждом прогоне заполняется таблица (расстановочное поле), содержащая $m_\ell \approx \frac{S}{\ell_\cdot \rceil \log_2 n \lceil}$ позиций. Под позицией понимается участок ОП длиной в $(\ell_\cdot \rceil \log_2 n \cdot \lceil + \gamma_\ell)$ двоичных разрядов, куда посимвольно записывается ℓ-грамма ($\ell \rceil \log_2 n \lceil$ разрядов) вместе со своей частотой встречаемости в тексте (γ_ℓ разрядов, где $\gamma_\ell < \ell_\cdot \rceil \log_2 n \lceil$ для $\ell > \ell^*$).

Заполнение таблицы осуществляется в соответствии с процедурой ассоциативного кодирования [I2] (используется лишь первая часть этой процедуры – нумерация, не обладающая свойством взаимной однозначности). Из текста последовательно выбираются еще не обработанные на предыдущих прогонах из-за ограниченности ОП ℓ-граммы, и для каждой из них определяется номер позиции, в которой должна содержаться информация об этой ℓ-грамме:

$$\tau(x_i) = x_i \,(mod\, m_\ell), \qquad (4.1)$$

где x_i – числовой код ℓ-граммы ($0 \le \tau(x_i) \le m_\ell - 1$). Если соответствующая позиция пуста, то ℓ-грамма записывается в неё, а в счетчик заносится единица. В противном случае обрабатываемая ℓ-грамма сравнивается с ℓ-граммой, уже записанной в позиции, и при их совпадении в счетчик добавляется единица, а при несовпадении анализ ℓ-граммы откладывается на последующие прогоны. Поскольку $N >> m_\ell$, расстановочное поле практически не будет содержать пустых позиций, несмотря на случайных характер процедуры заполнения. Очевидно также, что все позиции содержат отличные друг от друга ℓ-граммы, причем в результате прогона

оказывается определенной частота встречаемости каждой из них вне зависимости от их расположения в тексте.

Для реализации описанной процедуры требуется помечать обработанные ℓ-граммы. Кроме того, необходимо отмечать и единичные ℓ-граммы, выявляемые в ходе итерации. Единичные ℓ-граммы на следующей $(\ell+1)$-й итерации не должны обрабатываться. С этой целью каждая ℓ-грамма снабжена двумя информационными двоичными разрядами. Совокупность $(N-\ell+1)$ таких пар назовем информационной лентой (ИЛ1), сопутствующей тексту. Четыре возможных состояния каждой пары разрядов имеют следующую интерпретацию: 00 – ℓ-грамма подлежит обработке; 11 – ℓ-грамма обработана (не единичная); 01 – единичная ℓ-грамма; 10 – ℓ-грамма обработана на предыдущем прогоне, а информация о её типе (единичная – неединичная) должна быть занесена в ИЛ1 на данном прогоне.

Кодом 10 помечаются ℓ-граммы, впервые заносимые в таблицу, относительно которых лишь по окончании прогона можно сделать однозначное заключение (единичная – неединичная). С целью сохранения этой информации для следующего прогона заполненная таблица отображается по окончании прогона в таблицу ИЛ2 длиной в m_ℓ разрядов, хранимую в отличие от ИЛ1 в ОП. Единица в k-м разряде ИЛ2 означает, что в k-й позиции таблицы находилась единичная ℓ-грамма.

Использование ИЛ1 и ИЛ2 позволяет избежать удвоения числа прогонов, обусловленного необходимостью фиксации единичных ℓ-грамм. Обработке на каждом прогоне подвергаются лишь ℓ-граммы с кодами 00 и 10, причем для первых эта обработка заключается в занесении их в таблицу и изменении состояния счетчика, а для вторых – в изменении состояния соответствующей пары разрядов в ИЛ1 на 01 или 11 в зависимости от значения соответствующего разряда (1 или 0) в ИЛ2 (нужный разряд выделяется с помощью отображения (4.1)). Единичные ℓ-граммы, выявленные на всех прогонах ℓ-й итерации, автоматически переводятся в единичные $(\ell+1)$-граммы на первом прогоне $(\ell+1)$-й итерации (и только на нем!) с учетом возможности их естественного "размножения" (см. п.2.4, стр. 55).

После окончания прогона и формирования ИЛ2 осуществляется внутренняя сортировка ℓ-грамм, выписанных в таблице; в соответствии с их частотой встречаемости таблица переписывается на ленту, и осуществляется очередной прогон текста. Фиксируя на каждом прогоне в таблице ℓ-грамму с максимальной частотой, можно определить к концу итерации величину $F_\ell(1)$. Распределение $F_\ell(m)$ может быть после этого получено за один дополнительный прогон применением процедуры ассоциативного кодирования к значениям счетчиков ℓ-грамм, выписанных на ленту. Расстановочное поле будет содержать при этом $F_\ell(1)$ позиций, в каждой из которых будет накапливаться одна из величин E^k $(k = 2,\ldots, F_\ell(1))$.

Отметим три особенности алгоритма, учет которых существенно снижает его трудоемкость.

1. В отличие от алгоритма "АППОРТ-1" и алгоритма [8] процедура обработки построена так, что на каждой итерации длина прогона уменьшается с ростом номера прогона. Действительно, после первого прогона почти все из m_ℓ первых подряд расположенных ℓ-грамм текста, включая и их повторения, распределенные по всему тексту, оказываются обработанными. После второго прогона оказываются обработанными около $2m_\ell$ первых ℓ-грамм текста и т.д. Каждый последующий прогон устраняет пробелы, оставшиеся от предыдущих прогонов и увеличивает длину участка с полностью обработанным текстом, не подлежащим в дальнейшем считыванию с ленты и анализу. Учитывая наличие единичных ℓ-грамм, выявленных на предыдущей итерации, а также распределенных по всему оставшемуся тексту и зафиксированных кодом 11 повторений уже обработанных ℓ-грамм, можно ожидать даже более быстрой, нежели линейная, скорости уменьшения длины прогона, что соответствует средней по итерации длине прогона, не превышающей половины длины текста.

2. Предполагая, что одинаковые ℓ-граммы распределены по тексту равномерно, можно показать, что с увеличением номера прогона на каждой итерации средняя частота встречаемости ℓ-грамм, попадающих в таблицу, будет уменьшаться. При $f_\ell \approx 1$ остаток текста целесообразно обработать алгоритмом [8].

3. В описанной модификации алгоритма от предыдущей итерации сохраняется и используется лишь информация о единичных ℓ-граммах. В принципе алгоритм позволяет сохранять и использовать информацию о k-ичных ℓ-граммах $(k = 2,3,\ldots, \ell-1)$ предыду-

щей итерации, для которых может быть предложена эффективная схема обработки за счет экономии ОП примерно в ℓ/k раз. Возможность такой экономии становится очевидной, если учесть, что $(\ell-1)$-грамма, встретившаяся в тексте k раз, может породить не более чем k различных ℓ-грамм, для идентификации которых достаточно выписать лишь k $(\ell+1)$-х символов по адресу, определяемому ассоциативным кодом $(\ell-1)$-граммы.

Оценим трудоемкость алгоритма, исходя из приведенной в п.2.4. схемы роста числа единичных ℓ-грамм. Для числа прогонов на ℓ-й итерации имеем

$$\rho_\ell \approx \frac{M_\ell - E'_\ell}{m_\ell} = \frac{(M_\ell - E'_\ell)\cdot \ell \cdot]\ \log_2 n\ [}{S} . \qquad (4.2)$$

Оценим параметр M_ℓ, связав его с известными параметрами N и E'_ℓ (выражение 2.II). Исходя из определения E^k_ℓ, можно записать:

$$M_\ell = \sum_{k=1}^{F_\ell(1)} E^k_\ell , \qquad (4.3)$$

$$N = \sum_{k=1}^{F_\ell(1)} k \cdot E^k_\ell , \qquad (4.4)$$

откуда получаем очевидное неравенство

$$2(M_\ell - E'_\ell) = 2\sum_{k=2}^{F_\ell(1)} E^k_\ell \le \sum_{k=2}^{F_\ell(1)} k \cdot E^k_\ell = N - E'_\ell . \qquad (4.5)$$

Из (4.5) и (4.2) для M_ℓ и ρ_ℓ имеем:

$$M_\ell \le \frac{N + E'_\ell}{2} , \qquad (4.6)$$

$$\rho_\ell \le \frac{(N - E'_\ell)\cdot \ell \cdot]\ \log_2 n\ [}{2 \cdot S} . \qquad (4.7)$$

Учитывая, что средняя длина прогона в данном алгоритме не превышает половины длины текста, для трудоемкости T_ℓ алгоритма получим следующее выражение:

$$T_\ell \le \frac{1}{4} c \cdot N \frac{(N - E'_\ell)\cdot \ell \cdot]\ \log_2 n\ [}{S} , \qquad (4.8)$$

где c – константа, определяющая время считывания в ОП и обработки одной ℓ-граммы текста (операции считывания и обработки совмещены).

Возвращаясь к примеру, рассмотренному в п.2.4 ($N = 1.5\cdot10^6$ символов, $n = 50$, $S = 0{,}75\cdot10^6$ бит, $\ell_0 = 5$, $N/E'_{\ell_0} \approx 10$), оценим номер итерации (ℓ_{max}), соответствующей максимальным вычислительным затратам, и число прогонов на этой итерации. Подставляя (2.II) в (4.7) и приравнивая производную $\frac{dP_\ell}{d\ell}$ нулю, получаем $\ell_{max} = 7$, $P_{\ell_{max}} \le 30$, что, с учетом укороченности прогонов, не превышает 15 прогонов полного текста. Заметим, что полученная оценка существенно завышена (см. выражение (4.5)); кроме того, не принималось в расчет уменьшение вычислительных затрат, обусловленное учетом замечаний 2 и 3.

Итак, представлены два алгоритма получения спектра распределений ℓ-грамм ($\ell = 1,2,3,\ldots,\hat{\ell}$) по частоте встречаемости в символьных последовательностях большой длины. В диапазоне значений N, сравнимых с объемом ОП S, трудоемкость алгоритмов квазилинейно зависит от длины последовательности.

"АППОРТ-2" алгоритмически проще, чем "АППОРТ-I", легче для программирования и не требует увеличения объема хранящегося на л е н т е текста, возникающего при перекодировках ("АППОРТ-I") или при расщеплении текста на отдельные ℓ-граммы (внешняя сортировка). Однако в тех случаях, когда коэффициент заполнения поля счетчиков в "АППОРТ-I" достаточно высок, среднее число прогонов на итерацию в этом алгоритме будет меньше, чем в "АППОРТ-2".

Ограничивающим фактором для второго алгоритма при малом объеме ОП является необходимость посимвольного представления в ОП (но не во внешней памяти!) каждой из разновидностей ℓ-грамм. Это обусловлено тем обстоятельством, что для диапазона значений $\ell < \ell^{**}$ предпосылка $E'_\ell \approx M_\ell \approx N$, положенная в основу [8], не имеет места. С другой стороны, выписывание в ОП ℓ-грамм в явном виде позволяет в отличие от [8] не осуществлять предва –

рительного разбиения текста на попарно непересекающиеся подмножества ℓ-грамм.

В заключение отметим, что в описанных алгоритмах на каждой итерации анализируются лишь ℓ-граммы одинаковой длины. В трехмерном пространстве $\ell, m, F_\ell(m)$ это соответствует получению двумерных срезов, перпендикулярных оси ℓ. Дальнейшим развитием описанных подходов явилось бы создание методики, позволяющей оптимизировать затраты на получение спектра распределений не по отдельным двумерным срезам, а сразу во всем трехмерном пространстве.

Литература

I. Распознавание слуховых образов. Под ред. Н.Г.Загоруйко и Г.Я.Волошина. Новосибирск, изд-во "Наука", 1966.

2. Автоматический перевод,1949-1963.Критико-библиографический справочник под редакцией Г.С.Цвейга и Э.К.Кузнецовой.М., 1967.

3. БЕЛОНОГОВ Г.Г., НОВОСЕЛОВ А.П. Некоторые количественные закономерности в автоматизированных информационных системах. -В кн.: Цифровая вычислительная техника и программирование под ред. А.И.Китова. Вып.6. Изд-во "Сов.радио", 1971.

4. STEINACKER Ivo. Aspects of Computer text processing. - "Data Processing",1973,vol.15, N 3,p.148-153.

5. Статистика речи и автоматический анализ текста.Под ред. Р.Г.Пиотровского, Л., изд-во "Наука", 1971.

6. 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.

7. ГЛУШКОВ В.М., ГЛАДУН В.П., ЛОЗИНСКИЙ Л.С., ПОГРЕБИНСКИЙ С.Б. Обработка информационных массивов в автоматизированных системах управления. Киев, "Наукова думка", 1970.

8. ГУСЕВ В.Д., КОСАРЕВ Ю.Г., ТИТКОВА Т.Н. Отыскание статистических закономерностей текстов методом ассоциативного кодирования. -Настоящий сборник, с. 72-89.

9. КОСАРЕВ Ю.Г. Автоматные поля. -Настоящий сборник, с. 90-96.

10. КОРМИЛИЦЫН Н.С., КОСАРЕВ Ю.Г.Программа внутренней сортировки для ЭВМ "Минск-32". -В кн.: Вычислительные системы. Вып. 59. Новосибирск, 1974, с. 78-83.

11. ЛАВРОВ С.С., ГОНЧАРОВ Л.И. Автоматическая обработка данных. Хранение информации в памяти ЭВМ. М.,Изд."Наука",1974.

12. 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.

13. ХАФФМЕН Д.А. Метод построения кодов с минимальной избыточностью. -В кн.: Кибернетический сборник. Вып.3. Изд. ИЛ., 1961, с. 79-87.

14. JERMANN W.H. Redundancy in Deterministic Sequences. - "IEEE Trans on Systems Science and Cybernetics",1970,vol.SSC-6, N 4,p.358-360.

15. ZIPF I.K. Human Behavior and the principle of least effort. Cambridge (Mass), Addison-wesley,1949.

16. ANDREW D.Booth. A "Law" of Occurences for words of Low Frequency. - "Information and Control.",1967,N 10,p.386-393.

17. ЛОСЕВА В.Л., КОСАРЕВ Ю.Г. Программа определения больших гистограмм (БОГ). -В кн.: Вычислительные системы. Вып. 59. Новосибирск, 1974, с. 101-107.

Поступила в ред.-изд.отд.
29 мая 1974 года