МЕТОДЫ ОБРАБОТКИ ИНФОРМАЦИИ
(Вычислительные системы)
1978 год Выпуск 74

УДК 681.3.06

ОСНОВЫ ПОСТРОЕНИЯ СИСТЕМЫ АВТОМАТИЧЕСКОГО КОДИРОВАНИЯ, КОНТРОЛЯ И КОРРЕКЦИИ ТЕКСТОВОЙ ИНФОРМАЦИИ

В.В. Хабаров

Сложность процесса подготовки данных для ЭВМ служит одним из серьезных препятствий на пути внедрения многих автоматизированных систем. Как правило, на пользователя ЭВМ возлагается представление информации в виде, удобном для машины. Пользователь должен привычные для него термины естественного языка (точнее,его модификации, учитывающие специфику данной области) заменять специальными (обычно цифровыми) кодами.

Процесс такого кодирования заметно затрудняется как необходимостью пользоваться перекодировочными словарями, так и тем, что от пользователя требуется непривычно большая для него скрупулезность. В то же время информация после подобного кодирования теряет свою наглядность, что затрудняет обнаружение ошибок путем беглого просмотра, как это обычно делается с текстами на естественном языке.Поэтому кодирование приходится контролировать с помощью дублирования либо рекодирования. Возможности автоматического обнаружения ошибок, а тем более исправления, в данном случае ограничены из-за сложности, а нередко и невозможности построения формальных методов для обнаружения некоторых типичных видов ошибок (например, при замене одного кода другим, равным или близким ему по смыслу).

Важно также отметить, что в этом трудоемком и рутинном процессе подготовки данных (исключая перфорацию) участвуют, как правило, специалисты высокой квалификации. Естественной является попытка облегчить труд этих специалистов, переложив его хотя бы ча-

стично на ЭВМ, чтобы предоставить человеку возможность описывать данные в привычных для него терминах естественного языка, а остальную часть работы (кодирование, обнаружение и исправление ошибок) поручить машине.

Вполне понятно, что вывод о практической целесообразности перехода к словесному кодированию можно сделать лишь на основе количественных оценок и при непременном условии – создании простых и эффективных программных средств для реализации автоматического кодирования, обнаружения и исправления ошибок.

Для этого нами был проведен эксперимент, позволяющий получить требуемые количественные оценки для одной из областей применения – автоматизации проектирования маршрутной технологии изготовления деталей машин [I]. Для проведения эксперимента был создан входной язык технолога, который без употребления перекодировочных таблиц отражал семантику описываемых исходных данных и обладал достаточной избыточностью для исправления ошибок.

Исследуемый язык словесного кодирования сопоставлялся с языком цифрового кодирования.

Исследования показали, что языки описания данных необходимо строить как формальную модель естественных языков. Такие языки позволяют существенно облегчить взаимодействие человека с машиной, избежать трудоемких процедур при кодировании, значительно сократить число ошибок при подготовке данных.

Однако применение языков описания требует создания сложных трансляторов, программы которых характеризуются большими размерами, сложной логикой и большой трудоемкостью изготовления.

Для решения данной проблемы нами были проведены исследования с целью создания простых и эффективных средств обработки текстовой информации, подготовленной на различных языках входных сообщений.

I. Методика автоматического кодирования и исправления ошибок. Решалась следующая задача: задан словарь, каждому слову которого поставлен в соответствие код; требуется найти для каждого слова входного текста соответствующее слово из словаря и заменить его кодом.

При отсутствии искажений задача не вызывает затруднений. При искажениях нужно найти слово, наиболее близкое к данному, т.е. возникает задача распознавания образов [2]. Методы решения подобных задач связаны с большими затратами времени и памяти машины [3–5].

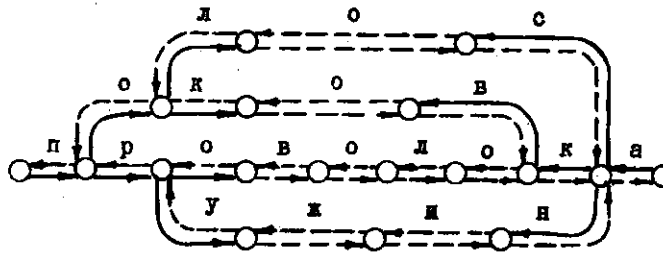
Мы применили алгоритм распознавания, который по затратам времени и памяти хорошо вписывается в технические возможности существующих ЭВМ и, как показал эксперимент, обладает достаточно высокой эффективностью обнаружения и исправления ошибок.

Введем следующие понятия. Б а з о й с л о в а назовем цепочку минимальной длины, которая отличает данное слово от всех других слов словаря; оставшуюся часть слова назовем о к о н ч а н и е м.

В зависимости от назначения порядка чтения слов может быть получено несколько таких цепочек (баз). Так, слово может анализироваться с начала, с конца либо по какому-то другому закону. Если в слове имеется ошибка, то для его распознавания нужно выбрать такой порядок чтения, чтобы получить неискаженную базу. Например, если известно, что первые символы входных слов искажены, то анализ можно провести со второго символа. При неизвестной позиции ошибки в слове, для распознавания слова можно использовать несколько баз. В этом случае слово будет опознано, если среди выбранных баз имеется неискаженная база.

Критерием выбора баз является наименьшая повторяемость символов слова в базах, т.е. лучшим будет такой случай, когда каждый символ слова принадлежит только одной базе. Такому условию для ограниченного объема словарей отвечают базы при чтении с начала (I) и конца (П) слова. Базой I назовем цепочки минимальной длины, которые отличают данное слово при анализе с начала слова, базой П – цепочки, отличающие слово при анализе с конца.

I. Поковка
2. Подоса
3. Проволка
4. Пружина



——— дерево баз I и П
– – – дерево окончаний

Рис. I

Таблица I

Распределение ошибок в словах различных длин

Кол-во букв в слове	Число ошибок в слове	Позиция в слове														
		I	2	3	4	5	6	7	8	9	I0	II	I2	I3	I4	I5
2	27	I4	I3													
3	28	I5	6	7												
4	39	I3	9	9	8											
5	9I	29	I8	2I	I2	II										
6	I42	29	30	36	I6	20	II									
7	I89	33	27	26	26	30	29	I8								
8	I65	20	22	29	I8	I9	25	I7	I5							
9	I88	I4	I5	2I	28	28	30	I7	2I	I4						
I0	I86	23	22	24	24	I3	2I	20	I3	I4	I2					
II	I5I	I4	I5	I0	I0	I5	I8	I5	I2	I3	I8	II				
I2	I24	6	8	II	I6	I2	I7	I0	I9	I0	6	4	5			
I3	99	4	9	I4	7	9	9	8	8	7	8	3	8	5		
I4	5I	3	I	9	4	3	7	2	I	6	5	2	3	2	4	
I5	40	4	2	4	5	3	4	I	I	0	5	2	2	2	3	2

Используя представление словаря в виде дерева, получаем два поисковых дерева для кодирования, состоящих соответственно из дерева баз I и дерева баз II и принадлежащих им окончаний. Объединив оба дерева, получим полное поисковое дерево для кодирования слов с ошибками по базе I и II (рис.I).

Алгоритм автоматического кодирования и исправления ошибок по поисковому дереву баз состоит в следующем. Слово входного текста анализируется буква за буквой, пока не выделится цепочка, которая совпадает с одной из баз эталонного словаря (т.е. нет ошибки в базе). После этого слову присваивается код, а затем проверяется окончание, при совпадении осуществляется переход к следующему слову. Если окончания не совпали, то к выделившейся базе входного слова подсоединяется окончание из поискового дерева. При ошибке в базе I слово анализируется по базе II. Если ошибка при анализе по базе II не обнаружена, то слову присваивается код. Если и в базе II ошибка, то слово заносится в список неисправленных ошибок.

Таблица 2

Распределение ошибок по видам и кратности

Кратность	Вид	Количество	%
Одиночные	X	71	4,650
	Y	44	2,881
	Z	1340	87,75
	Итого	1455	95,286
Двойные смежные	W	30	1,965
	XZ	5	0,327
	YZ	2	0,131
	YY	5	0,327
	ZZ	6	0,393
	Итого	48	3,143
Двойные раздельные	XY	1	0,065
	XZ	3	0,196
	YZ	2	0,131
	ZZ	11	0,721
	Итого	17	1,113
Тройные	XW	3	0,196
	YW	1	0,065
	ZXX	2	0,131
	XZZ	1	0,065
	Итого	7	0,458
Всего ошибок		1527	100,00%

2. Анализ ошибок. Эксперимент показал достаточно высокую эффективность метода по обнаружению и исправлению ошибок и, следовательно, возможность его практического применения. Это предопределило необходимость в дальнейших исследованиях метода.

Прежде всего было выполнено на более полном фактическом материале исследование ошибок в словах. С этой целью были использованы данные о словах с ошибками, допущенными машинисткой при печати технических текстов (табл.1). Ошибки анализировались по видам и кратности (табл.2).

Рассматривались четыре вида ошибок: пропуск (X), дополнение (У), замена букв (Z) и перестановка соседних букв (W). Из анализа полученных результатов следует, что наибольший удельный вес составляют одиночные ошибки – примерно 95%, двойные смежные ошибки – немногим более 3%, на двойные раздельные и тройные ошибки приходится менее 2%.

В рассматриваемом методе коррекции для распознавания слов используются две попытки чтения – с начала и конца. Результат можно получить быстрее, если знать, в какой части слова вероятнее ошибка. С этой целью оценим распределение ошибок по длине слова на основе данных табл.1.

Пусть событие А заключается в появлении ошибки в начале слова. Это означает, что ошибка сделана в одном из первых $\left[\frac{n}{2}\right]$ сим-

волов, где [ ] – целая часть, n – длина слова. Событие B заключается в появлении ошибок в конце слова, т.е. в одном из $\left[\frac{n}{2}\right]$ последних символов. Пусть C_n означает появление слова длины n. A/B_n и B/C_n – условные события, заключающиеся в появлении ошибок в первой и второй половинах слова длиной n. Тогда оценки вероятностей ошибок в первой и второй половинах относительно всех длин слов P(A) и P(B) могут быть получены по следующим формулам:

$$\bar{P}(A) = \sum_{n=2}^{15} \bar{P}(A/C_n)\bar{P}(C_n), \quad P(B) = \sum_{n=2}^{15} \bar{P}(B/C_n)\bar{P}(C_n), \qquad (I)$$

где $\bar{P}$ – оценки для вероятностей P ; 15 – максимальная длина слова.

По формулам (I) и данным табл.2 получены оценки для вероятностей ошибок соответственно в первой и второй половинах слов: $\bar{P}(A) = 0{,}513$; $\bar{P}(B) = 0{,}406$. Отсюда следует, что оценка вероятности ошибки в начале слова превышает оценку вероятности ошибки в конце слова.

Построим доверительные интервалы для $P(A_1)$ и $P(B_1)$. Из того, что величины $n\cdot\bar{P}(A_1) \approx 775$, $n\cdot(1-\bar{P}(A_1)) \approx 745$ и $n\bar{P}(B_1) \approx 608$, $n(1-\bar{P}(B_1)) \approx 912$ значительно больше 10, следует возможность использования формул для определения доверительных границ [6]

$$P_{1,2} = \bar{P} \pm t_\beta \sqrt{\frac{\bar{P}(1-\bar{P})}{n}} , \qquad (2)$$

где $\bar{P}$ – значение оценки для P ; t_β определяется из таблиц нормального закона; n – объем выборки.

При доверительном уровне $\beta = 0{,}05$ имеем $t_\beta = 1{,}96$. Тогда доверительные границы для $P(A_1)$ равны $P_1(A_1) = 0{,}495$; $P_2(A_1) = 0{,}535$; для $P(B_1)$ равны $P_1(B_1) = 0{,}32$; $P_2(B_1) = 0{,}425$.

Полученные доверительные границы определяют пределы, в которых с вероятностью 0,95 находятся истинные значения вероятностей ошибок, характеризуемые оценками вероятностей $\bar{P}(A_1) = 0{,}513$, $\bar{P}(B_1) = 0{,}406$.

Для того чтобы более обоснованно говорить о неслучайном расхождении в вероятностях ошибок в начале и в конце слова, проверим статистическую гипотезу H_0 о совпадении вероятностей $P(A_1)$ и $P(B_1)$, используя критерий χ^2 [7].

Пусть сравниваются вероятности двух событий C_1 и C_2 . В выборке объема n_1 событие C_1 имело успех K_1 раз, а в выборке объема n_2 событие C_2 имело успех K_2 раз. При выполнении равенства

$P(C_1) = P(C_2)$ величина

$$\chi^2 = \frac{\left(\frac{K_1}{n_1} - \frac{K_2}{n_2}\right)^2 n_1 n_2 (n_1 + n_2 - 1)}{(K_1 + K_2)(n_1 + n_2 - K_1 - K_2)} \qquad (3)$$

обладает следующим свойством: вероятность события $\{\chi^2 \le q^2\}$ приближенно равна

$$\frac{2}{\sqrt{2\pi}} \int_0^q e^{-\frac{1}{2}t^2} dt .$$

Задаваясь уровнями ошибки $2\beta = 0{,}05$ и $0{,}01$, по таблицам нормального распределения для $2\beta = 0{,}05$ находим $q_1 = 1{,}96$; для $2\beta = 0{,}01$, $q_2 = 2{,}58$.

Гипотеза $P(A_1) = P(B_1)$ принимается, если $\chi^2 \le q^2$, и отвергается – в противном случае. Подставляя в выражение (3) значения $n_1 = n_2 = 1520$, $K_1 = 780$, $K_2 = 618$, находим $\chi^2 = 34{,}9$, что больше величин $q_1^2 = 3{,}84$ и $q_2^2 = 6{,}66$. Таким образом, при уровнях ошибки первого рода (отвергнуть правильную гипотезу) $2\beta = 0{,}05$ и $2\beta = 0{,}01$ гипотеза о равенстве уверенно отвергается.

Сравним вероятность ошибок в первой и последней трети слова. Под событием "ошибка в первой трети слова" будем понимать ошибку в первых $\left[\frac{n}{3}\right]$ символах, под событием "ошибка в последней трети слова" будем понимать ошибку в $\left[\frac{n}{3}\right]$ последних символах.

Обозначим эти события через A_2 и B_2 соответственно. Из 1520 испытаний (табл.1) событие A_2 имело место 509 раз и B_2 имело место 394 раза. По формуле (1) получим соответствующие оценки вероятностей: $\bar{P}(A_2) = 0{,}335$; $\bar{P}(B_2) = 0{,}260$.

Построим доверительные интервалы для $P(A_2)$, $P(B_2)$. Задаваясь доверительным уровнем $2\beta = 0{,}05$ и используя формулу (2), получаем доверительные границы $P_1(A_2) = 0{,}311$; $P_2(A_2) = 0{,}359$; $P_1(B_2) = 0{,}238$; $P_2(B_2) = 0{,}282$.

Проверим гипотезу H_0 о равенстве $P(A_2)$ и $P(B_2)$. Подставляя значения $n_1 = n_2 = 1520$, $K_1 = 509$, $K_2 = 394$, вычислим по формуле (3) значение статистики $\chi^2 = 20{,}5$. Из сравнения полученного значения χ^2 с ранее найденными значениями q_1 и q_2 следует, что

при ошибках первого рода с уровнями $2\beta = 0{,}05$ и $0{,}01$ гипотеза о равенстве $P(A_2) = P(B_2)$ отвергается.

Согласно приведенным выше оценкам, ошибки по длине слова распределены неравномерно: ошибок в начале слова существенно больше, чем в конце слова.

3. Анализ корректирующей способности метода. Как следует из анализа ошибок, на долю одиночных и двойных смежных ошибок приходится подавляющее большинство (свыше 98%), поэтому корректирующая способность метода определится возможностью исправления этих ошибок.

Введем некоторые обозначения: n – число слов в словаре; i – номер слова в словаре; B_1, B_2 – число символов в 1-й и 2-й базах; D_j – ошибка в j-м символе слова.

Определим вероятность исправления одиночной ошибки в слове $P_1(C)$. Предположим, что если в слове имеется ошибка, то вероятность ее появления в каждом символе одна и та же, т.е. $P(D_j) = \frac{1}{m}$.

Тогда

$$P_1(C) = \begin{cases} \dfrac{2m - (B_1 + B_2)}{m}, & \text{если } m < B_1 + B_2, \\ 1, & \text{если } m \geq B_1 + B_2. \end{cases} \qquad (4)$$

Если слова из словаря появляются с равной вероятностью, то вероятность исправления ошибок по словарю $P_1(S)$ определится выражением

$$P_1(S) = \frac{1}{n} \sum P_1(C_i). \qquad (5)$$

Вероятность исправления двойных смежных ошибок в слове определится как

$$P_2(C) = \begin{cases} \dfrac{2m - (B_1 + B_2 + 2)}{m-1}, & \text{если } m \leq B_1 + B_2, \\ 1, & \text{если } m > B_1 + B_2. \end{cases} \qquad (6)$$

Вероятность исправления ошибок по словарю $P_2(S)$ будет равна

$$P_2(S) = \frac{1}{n} \sum_{i=1}^{n} P_2(C_i), \qquad (7)$$

В соответствии с выражениями (4)-(7) оценивалась вероятность автоматического исправления ошибок для словарей различной длины, из которых исключались одно- и двухсимвольные слова. На рис.2 приведено распределение вероятностей исправления одиночных (P_1) и двойных ошибок (P_2) в зависимости от объема словаря и используе -

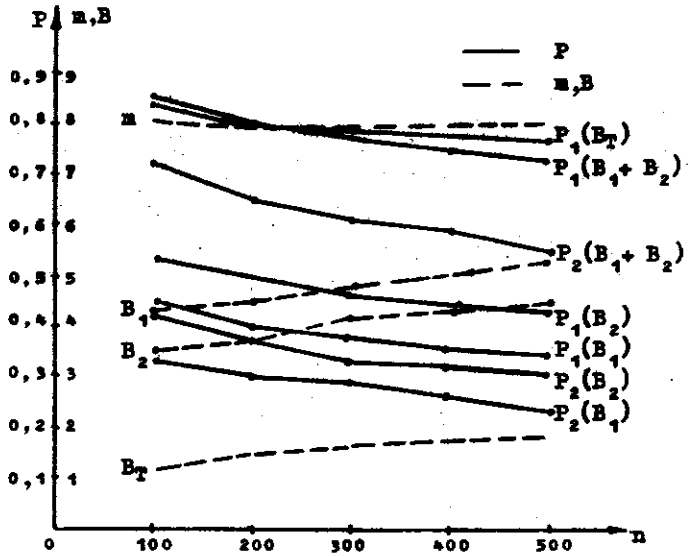


Рис. 2

мых баз слова; первой (B_1), второй (B_2) ,обеих (B_1+B_2), теоретической (B_T) - соответствующей словарю того же объема, но с минимальной длиной базы (случай равномерного употребления символов алфавита).

Вероятность исправления ошибок уменьшается с увеличением объема словаря, однако для словарей, применимых для специализированных языков, вероятность исправления ошибок остается в пределах от 0,85 до 0,75. Возможность аналитической оценки корректирующей способности словаря в соответствии с формулами (4)-(7) позволяет путем подбора слов довести уровень исправления одиночных ошибок для словарей в 200-400 слов до 90-95%. Это подтверждает и тот факт, что исправление ошибок по одной лишь теоретической базе (B_T) для рассмотренных словарей составляет 75-85% (рис.2).

Из рис.2 следует, что вторые базы слов (B_2) короче первых (B_1) на 0,8 символа и, если учесть, что число ошибок в конце слов существенно меньше, чем в начале, то можно сделать следующий важный вывод: при обработке текста лучше читать слова не с начала, а с конца. Это позволит сократить затраты на кодирование слова и исправление ошибок.

4. Требования к системе. Высокая эффективность словесного кодирования и наличие практического метода исправления ошибок обуславливает создание системы автоматического кодирования, контроля и коррекции текста. Состав и содержание требований к системе определяются из задач, решаемых при автоматизированном проектировании предприятий, перспектив ее развития и эксплуатации. Основные из них следующие:

– универсальность (система должна обеспечивать трансляцию текстов с различных языков входных сообщений);

– эффективность по обнаружению и исправлению ошибок (все синтаксические ошибки должны быть обнаружены и большинство их исправлено, исправления должны допускать последующую проверку человеком);

– достаточное быстродействие системы (система должна быть технологичной в эксплуатации; затраты времени и памяти ЭВМ на кодирование, контроль и исправление ошибок должны быть реалистичными для существующих ЭВМ);

– простота подготовки данных о словаре (при обучении системы новому языку или внесении изменений от пользователя должно требоваться только описание словаря либо изменения; всю остальную часть по созданию программных средств кодирования, контроля и коррекции текста должна выполнять система);

– удобство пользования системой (язык общения с системой должен быть доступным для массового пользователя-непрограммиста).

5. Применение R-языка при реализации системы. В основе метода автоматического кодирования лежит представление эталонного словаря в виде двух словарей. Прямого – для распознавания слов при чтении с начала слова и инверсного – для распознавания при чтении слов с конца слова. Каждый словарь представляется в виде двух ориентированных граф-схем (рис.I): дерева баз слов, в котором дуги направлены от начальной

вершины r_0 к вершинам $b_1,\ldots,b_n$, соответствующим концам без каждого из слов; дерева окончаний слов, в котором дуги направлены от вершин $b_1,\ldots,b_n$ к вершине r_ϕ, соответствующей концам слов.

Дерево без удобно предоставить с помощью R-языка Вельбицкого [8], модифицированного применительно к условиям задачи кодирования. Удобство применения R-языка заключается в естественности и простоте описания в нем данного алгоритма, в существовании простых и эффективных средств реализации алгоритмов, записанных на этом языке. Кроме того, что весьма важно, входная информация кодируется с помощью той же технологической базы, которая развита для производства трансляторов.

Дерево словаря описывается R-грамматикой следующим образом. Каждая дуга дерева есть правило R-грамматики, указывающее, что надо делать для кодирования слова. Узлы дерева объединяют правила в комплексы правил. Правила, дуги которых имеют общую вершину, образуют один комплекс правил.

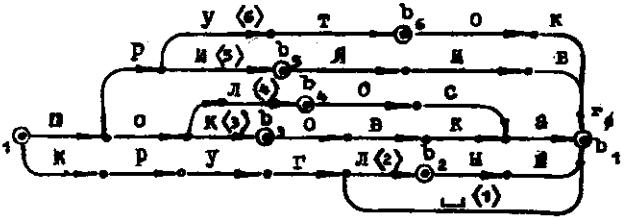


Рис. 3

Выделено три типа правил:

1. Правило $a \xrightarrow[T=1]{} r$. Первый тип правил (T = 1) соответствует линейным цепочкам в дереве (участки монарной R-грамматики по работе [2]). Текущий символ входного текста сравнивается с термом (a) правила. При совпадении символ считается синтаксически верным и осуществляется переход к комплексу правил, следующему за данным.

2. Правило $a \xrightarrow[T=2]{} r$. Ко второму типу (T = 2) относятся правила одного комплекса, которые соответствуют непоследним символам базы. При выполнении этого правила текущий символ входного текста сравнивается с термом первого правила в комплексе правил. При несовпадении - переход к следующему правилу в комплексе, при совпадении - переход к комплексу правил по адресу r.

3. Правило $a \xrightarrow[T=3]{k} A_{мок}$. К третьему типу (T = 3) относятся правила одного комплекса, соответствующие последним символам базы. При выполнении этого правила текущий символ входного текста сравнивается с термом (a) правила, при совпадении слову присваивается код (k) и указывается адрес для входления в массив окончаний ($A_{мок}$).

Дерево без слов (рис.3) с помощью данных правил запишется в следующем виде:

$$r_1 \sim \{к \to r_2,\ п \to r_3\}$$

$$r_2 \sim \{р \to r_4\}$$

$$r_4 \sim \{у \to r_5\}$$

$$r_5 \sim \{т \to r_6\}$$

$$r_6 \sim \{\sqcup \xrightarrow{\langle 1 \rangle} A_{мок},\ л \xrightarrow{\langle 2 \rangle} A_{мок}\}$$

$$r_3 \sim \{о \to r_7,\ р \to r_8\}$$

$$r_7 \sim \{к \xrightarrow{\langle 3 \rangle} A_{мок},\ л \xrightarrow{\langle 4 \rangle} A_{мок}\}$$

$$r_8 \sim \{и \xrightarrow{\langle 5 \rangle} A_{мок},\ у \xrightarrow{\langle 6 \rangle} A_{мок}\}$$

Дерево окончаний представляется в виде списковой структуры с использованием для снятия частичной и полной вкладываемости окончаний.

Применение аппарата R-грамматик позволяет одни и те же средства использовать как для кодирования, так и для раскодирования. При этом используется известная способность R-грамматик применяться (при одной и той же форме представления) в двух режимах: распознающем и порождающем. В нашем случае для целей раскодирования, когда порождается не все множество допустимых конструкций языка, а только одна единственная, введен управляемый режим порождения R-грамматик. Управление осуществляется с помощью дуте - водителя, который для каждого комплекса с числом правил более одного указывает,какое по порядку правило должно быть выбрано.

В памяти машины R-грамматики реализуются в виде R-таблиц, каждому словарю соответствуют две R-таблицы: R-таблица слов прямая и R-таблица слов инверсная. Деревья окончаний реализуют-

ся в виде прямого и инверсного массивов окончаний. Путеводитель - в виде раскодировочной таблицы слов.

6. О п и с а н и е с и с т е м ы. Программный комплекс системы автоматического кодирования, контроля и коррекции текста реализован на ЭВМ "Минск-32" и ЕС-1020 и представляет собой универсальное средство для обработки текстовой информации, подготовленной на различных языках входных сообщений. С помощью данной системы пользователь может получить систему программ для трансляции текстов с конкретного языка, задав в качестве исходных данных эталонный словарь и структуру входных данных. Процесс генерирования программных средств иллюстрируется схемой на рис. 4.

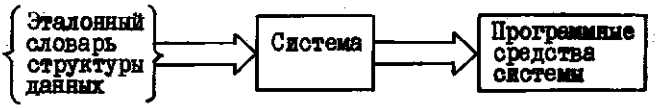


Рис. 4

Сформированный комплекс программ предоставляет пользователю следующие возможности:

- выполнять автоматическое кодирование текстовой информации;
- обнаруживать грамматические ошибки в тексте;
- исправлять обнаруженные ошибки;
- раскодировать тексты ранее закодированные в системе;
- накапливать статистику ошибок при решении задач пользователя.

Работа сформированного комплекса программ иллюстрируется схемой на рис. 5.

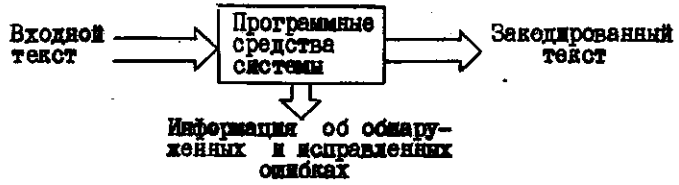


Рис. 5

В систему программ входят обслуживающий и процедурный сегменты, объединенные управляющей программой.

Процедурные сегменты сформированы по их функциональному назначению: "Обучение", "Кодирование" и "Раскодирование". Каждый сегмент является самостоятельным модулем и может использоваться в других пакетах прикладных программ.

Обслуживающий сегмент системы выполняет следующие функции:

- настраивает систему на заданный режим работы;
- вводит с перфокарт или перфолент кодируемый текст (эталонный словарь);
- записывает текст (словарь) на МД;
- выполняет синтаксический контроль текста (словаря);
- распечатывает кодируемый текст (эталонный словарь) с указанием синтаксических ошибок;
- корректирует текст (словарь) на МД согласно массиву изменений.

Система функционирует в трех режимах: обучение, кодирование и раскодирование, каждый режим обеспечивается соответствующим процедурным сегментом.

В режиме "Обучение" на основе эталонного словаря осуществляется автоматическое построение программных средств. Эталонный словарь представляется в виде словаря слов или словаря предложений. Соответствующие коды могут быть определены пользователем или сформированы системой. Система может формировать средства для кодирования на уровне слов (лексики) и на уровне предложений (синтаксиса).

Уровень обучения определяется директивами пользователя:

- обнаруживать ошибки лексики (ООЛ)*);
- исправлять ошибки лексики (ИОЛ);
- обнаруживать ошибки синтаксиса (ООС);
- исправлять ошибки синтаксиса (ИОС);

В общем случае могут быть построены следующие программные средства:

1. R-таблица слов (RTC).
2. R-таблица слов инверсная (RTCИ).
3. R-таблица предложений (RTП).

---

*) Здесь и ниже сокращения соответствуют табл. 3.

4. R-таблица предложений инверсная (RТПИ).
5. Массив окончаний слов (МОС).
6. Массив окончаний слов инверсный (МОСИ).
7. Массив окончаний предложений (МОП).
8. Массив окончаний предложений инверсный (МОПИ).
9. Раскодировочная таблица слов (РТС).
10. Раскодировочная таблица предложений (РТП).

Конкретный состав средств, формируемых в режиме "Обучение", определяется директивой пользователя (табл. 3).

Таблица 3

Программные средства	Обучение				Кодирование				Раскодирование	
	ООД	ИОЛ	ООС	ИОС	КООЛ	КИОЛ	КООС	КИОС	РЛ	РС
RТС	+	+	+	+	+	+	+	+	+	+
RТСИ		+	+	+		+	+	+		
RТП			+	+			+	+		+
RТПИ				+				+		
МОС	+	+	+	+	+	+	+	+	+	+
МОСИ		+	+	+		+	+	+		
МОП			+	+			+	+		+
МОПИ				+				+		
РТС	+	+	+	+					+	+
РТП			+	+						+

Кодирование является основным рабочим режимом системы. Конкретный режим кодирования определяется следующими директивами пользователя:

- кодировать, обнаруживать ошибки лексики (КООЛ);
- кодировать, исправлять ошибки лексики (КИОЛ);
- кодировать, обнаруживать ошибки синтаксиса (КООС);
- кодировать, исправлять ошибки синтаксиса (КИОС).

При кодировании используются программные средства, полученные в режиме обучения. Связь директив кодирования и программных средств для их выполнения определяется табл. 3.

При обращении к программе кодирования посредством одной из директив управляющая программа представляет необходимые программные средства для выполнения данной директивы (табл.3). Работа системы в режиме кодирования осуществляется следующим образом.

Текст для кодирования вводится с МЛ в виде предложений с соответствующими разделителями. Затем из предложений выделяются слова, подлежащие кодированию. Слова, не требующие кодирования, переносятся в специальное поле. Затем слово распознается по прямой В-таблице. Если база выделялась, слову присваивается код. При наличии ошибки в базе и директиве КООЛ слово считается с ошибкой и выдается на печать. Если задается директива КИОЛ, слово инвертируется и распознается инверсной таблицей. Если слово опознано прямой или инверсной В-таблицей и код для него определен, то анализируется правильность окончания. Затем производится контрольная печать слов с присвоенными кодами и слов с обнаруженными и исправленными ошибками.

Если кодирование выполняется на уровне синтаксиса, то пословно закодированное предложение на уровне лексики распознается по прямой В-таолице предложений. При выделении базы предложению присваивается код. Если в базе ошибка, то при директиве КООС предложение с ошибкой выдается на печать и осуществляется переход на кодирование следующего предложения, при директиве КИОС предложение проверяется по инверсной таблице. После распознавания предложения по одной из таблиц ему присваивается код и производится проверка окончания. Затем осуществляется печать исправленных ошибок синтаксиса.

В режиме "Раскодирование" осуществляется раскодирование текстов, ранее закодированных системой. Для раскодирования используются программные средства, полученные на стадии обучения системы (табл.3). Управление раскодированием осуществляется с помощью директив пользователя: раскодировать лексику (РЛ), раскодировать синтаксис (РС).

Опыт реализации показал, что система хорошо вписывается в возможности существующих ЭВМ, так, для машины "Минск-32" получены следующие параметры. Объем программ на языке ЯСК составляет 23,6 тыс. команд, в том числе:

- обслуживающий сегмент – 5,8 тыс. команд;
- сегмент "обучение" – 9,0 тыс. команд;
- сегмент "кодирование" – 5,0 тыс. команд;
- сегмент "раскодирование" – 3,8 тыс. команд.

Максимальный объем эталонного словаря для машины с памятью 32 тыс. команд – 4000 слов.

Время построения средств кодирования, раскодирования и исправления ошибок линейно зависит от объема словаря n и определяется по формуле t(сек.) ≈ 0,6n. Время кодирования t(сек.) ≈ 0,05n.

Коэффициент сжатия текста на уровне слов K_c = 3,0 – 4,0.

Коэффициент сжатия текста на уровне предложений $K_п$ = 6 – 10.

Литература

1. ХАБАРОВ В.В., КОСАРЕВ Ю.Г. Об эффективности автоматического кодирования и исправления ошибок при подготовке данных.–В кн.: Вычислительные системы. Вып.62. Ассоциативное кодирование. Новосибирск, 1975, с.106–118.

2. ЗАГОРУЙКО Н.Г. Методы распознавания и их применение. М., "Сов.радио", 1972, 206 с.

3. MORGAN H.L. Spelling correction in systems program.– "Commun ACM",1970,v.13,N 2,p.90–94.

4. WAGNER P.A. Spelling correction for regular languages . – "Communs ACM",1974,v.17,N 5,p.265–268.

5. RISEMAN E.M.,HANSON A.R. A contextual postprocessing system for error correction using binary n-grams. – "IEEE Trans.Computers",1974,v.23,N 5,p.480–493.

6. ВЕНТЦЕЛЬ Е.С. Теория вероятностей. М., "Наука", 1975.

7. Ван дер ВАРДЕН. Математическая статистика. М.,ИЛ, 1960.

8. ВЕЛЬБИЦКИЙ И.В. Метаязык n-грамматик. –"Кибернетика",1975, № 3, с.47–63.

Поступила в ред.-изд.отд.
12 апреля 1978 года