СТРУКТУРНЫЙ АНАЛИЗ СИМВОЛЬНЫХ ПОСЛЕДОВАТЕЛЬНОСТЕЙ
(Вычислительные системы)
1984 год Выпуск 101

УДК 519.217.2:621.391.19

ОЦЕНКА НАДЕЖНОСТИ РАСПОЗНАВАНИЯ ДИСКРЕТНОЙ РЕЧИ ПРИ ЛИНГВИСТИЧЕСКИХ ОГРАНИЧЕНИЯХ

В.М.Величко, Э.Г.Крылов, Л.Я.Савельев

§1. Постановка задачи

Рассматривается задача оценки надежности распознавания устных фраз, составленных из изолированно произнесенных слов, с четкими паузами между словами (так называемая "дискретная речь"). Фразы составляются из слов заранее заданного проблемно-ориентированного словаря V. На фразы накладываются синтаксические ограничения, заключающиеся в запрете грамматически неправильных парных словосочетаний, и семантические ограничения, заключающиеся в запрете бессмысленных в данной проблемно-ориентированной области парных словосочетаний. Указанные лингвистические ограничения учитываются в алгоритме распознавания фраз следующим образом:

1. Каждая допустимая фраза может начинаться со слова из подсловаря $V_{нач} \subseteq V$.

2. За каждым распознанным словом фразы v_r может следовать слово v_{r+1}, принадлежащее подсловарю V_r, определяемому словом v_r, т.е. $v_{r+1} \in V_r \subseteq V$.

3. На $(r+1)$-м этапе распознавания принимается в расчет только одно распознанное слово (правило "первого лучшего"), давшее наилучший результат при распознавании подсловаря V_r.

Перечисленные синтаксические и семантические ограничения позволяют существенно повысить надежность распознавания дискретной речи в реальных системах [1] за счет значительного сокращения распознаваемого словаря V_r на каждом шаге распознавания по сравнению с полным словарем V. Целью данной работы является количественная оценка получаемого выигрыша в надежности и методика получения этой оценки на ЭВМ при указанных ограничениях. Оценка получается с помощью марковской модели, хорошо описывающей особенности исследуемой дискретной речи и реальных алгоритмов ее распознавания.

§2. Математическая модель системы связи

I. Предположения о языке и способе распознавания.

Я з ы к. Предположим, что слишком короткие и слишком длинные сообщения маловероятны и что в сообщении слова́, следующие за данным, не зависят от предшествующих ему.

Есть общий, начальный и переходный словари. Начальный словарь состоит из слов, которые могут быть в начале сообщения. В переходном словаре для каждого слова указаны слова, которые могут следовать за ним.

Математически предположения о языке сводятся к тому, что:

а) длина сообщения имеет пуассоновское распределение;

б) содержание сообщения имеет марковское распределение.

Это соответствует предполагаемым особенностям исследуемой речи: I) очень короткие и очень длиные сообщения встречаются редко; 2) распределение слов, следующих за данным, не зависит от слов, ему предшествующих.

С п о с о б р а с п о з н а в а н и я. При распознавании сообщения начальный сигнал сравнивают с каждым словом начального словаря и выбирают самое похожее. Сигнал, принятый первым после начального, сравнивают с каждым словом переходного словаря, которое может следовать после распознанного начального, и выбирают самое похожее. Вообще, распознав n-е слово, сравнивают $(n+1)$-й принятый сигнал с каждым словом переходного словаря, которое может следовать за распознанным n-м, и выбирают самое похожее. Принятым считается сообщение, составленное из выбранных самых похожих слов.

2. Описание системы связи. Система связи состоит из источника информации и канала передачи информации [2].

И с т о ч н и к и н ф о р м а ц и и. Словарь A составляют номера $1,\dots,m$. Он определяет язык $X = A\cup(A\times A)\cup(A\times A\times A)\cup\dots$. Сообщениями на этом языке являются конечные последовательности $x = i_0 i_1 \dots i_n$ номеров $1,\dots,m$. Возьмем число $\lambda > 0$, распределение вероятностей $p = (p_i)$ и стохастическую матрицу $Q = (q_{ij})$,

$i, j = 1, \dots, m$. Вероятность того, что сообщение x имеет длину $l(x) = n$ и состоит из номеров $i_0, i_1, \dots, i_n$, равна

$$P(x=i_0 i_1 \dots i_n) = p_{i_0} q_{i_0 i_1} \dots q_{i_{n-1} i_n} \cdot (\lambda^n/n!) e^{-\lambda} .$$

При таком определении вероятность того, что сообщение x имеет длину n, равна

$$P(l(x)=n) = \sum_{1 \le i_0, i_1, \dots, i_n \le m} P(x=i_0 i_1 \dots i_n) = (\lambda^n/n!) e^{-\lambda} .$$

А вероятность того, что сообщение длины n состоит из номеров $i_0, i_1, \dots, i_n$, равна

$$P(x=i_0 i_1 \dots i_n \mid l(x)=n) = p_{i_0} q_{i_0 i_1} \dots q_{i_{n-1}, i_n} .$$

Это соответствует сделанным предположениям.

К а н а л п е р е д а ч и и н ф о р м а ц и и. Входной код X канала состоит из конечных последовательностей $x = i_0 i_1 \dots i_n$ номеров $1, \dots, m$. Выходной код Y равен входному.

Условные вероятности определим так. Возьмем последовательность стохастических матриц $R^{(0)} = (r_{ij}^{(0)})$, $R^{(1)} = (r_{ij}^{(1)}), \dots, R^{(m)} = (r_{ij}^{(m)})$, $i, j = 1, \dots, m$. Вероятность того, что принято сообщение $y = j_0 j_1 \dots j_n$, когда передано сообщение $x = i_0 i_1 \dots i_n$, равна

$$P(y=j_0 j_1 \dots j_n \mid x=i_0 i_1 \dots i_n) = r_{i_0 j_0}^{(0)} \, r_{i_1 j_1}^{(j_0)} \dots r_{i_n j_n}^{(j_{n-1})} .$$

3. Частные случаи.

М а р к о в с к а я м о д е л ь. Для каждого $k = 0, 1, \dots, m$ выберем множество $A(k)$ номеров из $A = \{1, \dots, m\}$. Составим матрицы $R^{(k)}$ из элементов 0 и I так, чтобы

$$r_{ii}^{(k)} = 1 \quad (i \in A(k)), \quad r_{ii}^{(k)} = 0 \quad (i \notin A(k)).$$

В матрице $R^{(k)}$ каждая строка состоит из одного элемента I и $m-1$ элементов 0. В строках с номерами $i \in A(k)$, и только в них, единицы стоят на диагонали матрицы. (Расположение единиц в остальных строках, не на диагонали, несущественно.)

При передаче по такому каналу в начале сообщения каждый номер $i \in A(0)$ принимается правильно, а каждый номер $i \notin A(0)$ при -

нимается неправильно. Если $(n-1)$-м принят номер $k = 1, \ldots, m$, то n-м каждый номер $i \in A(k)$ принимается правильно, а каждый номер $i \notin A(k)$ принимается неправильно.

Возьмем еще тоже упрощенные начальное распределение $p = (p_i)$ и переходную матрицу $Q = (q_{ij})$. Для каждого $k=0,1,\ldots,m$ выберем множество $B(k)$ номеров из $A = \{1,\ldots,m\}$. Обозначим $b(k)$ число элементов множества $B(k)$. Положим:

$$p_i = \beta(0) = 1/b(0) \quad (i \in B(0)), \quad p_i = 0 \quad (i \notin B(0));$$

$$q_{kj} = \beta(k) = 1/b(k) \quad (j \in B(k)), \quad q_{kj} = 0 \quad (j \notin B(k)).$$

Лапласовская модель. Эта модель является специальным случаем марковской. Ее используют, когда в сообщении на каждом месте с равной вероятностью может появиться любое слово из общего словаря независимо от того, какие слова появились на других местах.

В лапласовской модели

$$A(0) = A(1) = \ldots = A(m) = C,$$

$$R^{(0)} = R^{(1)} = \ldots = R^{(m)} = R.$$

В матрице R каждая строка состоит из одного элемента I и $m-1$ элементов 0. В строках с номерами $i \in C$, и только в них, единицы стоят на диагонали матрицы.

При передаче по такому каналу каждый номер $i \in C$ всегда принимается правильно, а каждый номер $i \notin C$ всегда принимается неправильно. Предполагается еще, что

$$B(0) = B(1) = \ldots = B(m) = A,$$

$$\beta(0) = \beta(1) = \ldots = \beta(m) = 1/m.$$

§3. Вероятность правильного приема

Вычислим для рассматриваемых моделей вероятность того, что принятое и переданное сообщения совпадают.

I. Общая формула. Для каждых номеров $i_0, i_1, \ldots, i_n \in A$ вероятность того, что принятое и переданное сообщения совпадают с $i_0 i_1 \ldots i_n$, равна

$$P(y=x=i_0 i_1 \ldots i_n) = P(y=i_0 i_1 \ldots i_n \mid x=i_0 i_1 \ldots i_n) \cdot P(x=i_0 i_1 \ldots i_n) =$$

$$= r^{(0)}_{i_0 i_1} r^{(i_0)}_{i_1 i_1} \ldots r^{(i_{n-1})}_{i_n i_n} \cdot p_{i_0} q_{i_0 i_1} \ldots q_{i_{n-1} i_n} \cdot (\lambda^n / n!) e^{-\lambda}.$$

Вероятность того, что принятое и переданное сообщения совпадают, равна сумме этих вероятностей:

$$\varkappa = P(y=x) = \sum_{n\ge 0}\ \sum_{1\le i_0, i_1, \dots, i_n \le m} P(y=x=i_0 i_1,\dots,i_n).$$

Для каждого $n=0,1,2,\dots$ вероятность того, что принятое сообщение совпадает с переданным сообщением длины n, равна

$$\varkappa(n) = P(y=x\,|\,l(x)=n) =$$

$$= \sum_{1\le i_0,i_1,\dots,i_n\le m} r^{(0)}_{i_0 i_0}\, r^{(i_0)}_{i_1 i_1}\dots r^{(i_{n-1})}_{i_n i_n}\cdot p_{i_0}\, q_{i_0 i_1}\dots q_{i_{n-1} i_n}.$$

Поэтому

$$\varkappa = \sum_{n\ge 0}\varkappa(n)\cdot(\lambda^n/n!)e^{-\lambda}. \qquad (1)$$

2. Формулы для частных случаев. Выведем формулы вероятности правильного приема для марковской и лапласовской моделей.

М а р к о в с к а я м о д е л ь. В этой модели

$$r^{(0)}_{i_0 i_0}\ r^{(i_0)}_{i_1 i_1}\dots r^{(i_{n-1})}_{i_n i_n} = 1$$

тогда и только тогда, когда $i_0\in A(0)$, $i_1\in A(i_0),\dots,i_n\in A(i_{n-1})$. Во всех остальных случаях это произведение равно 0. Поэтому в марковской модели

$$\varkappa(n) = \sum_{i_0\in A(0)}\ \sum_{i_1\in A(i_0)}\dots\sum_{i_n\in A(i_{n-1})} p_{i_0}\, q_{i_0 i_1}\dots q_{i_{n-1} i_n}. \qquad (2)$$

Пусть p,Q упрощенные и $c(k)$ – число элементов пересечения $A(k)\cap B(k) = C(k)$. Тогда

$$\varkappa(n) = \sum_{i_0\in C(0)}\ \sum_{i_1\in C(i_0)}\dots\sum_{i_n\in C(i_{n-1})} p_{i_0}\, q_{i_0 i_1}\dots q_{i_{n-1} i_n} =$$

$$=\beta(0)\cdot\sum_{i_0\in C(0)}\ \sum_{i_1\in C(i_0)}\dots\sum_{i_{n-1}\in C(i_{n-2})}\beta(i_0)\beta(i_1)\dots\beta(i_{n-1})\cdot c(i_{n-1}).$$

В частности, если $C(0) = C(1)=\dots=C(m) = C$ и c – число элементов множества C, то $\varkappa(n) = \varkappa(0)\cdot\bar{\varkappa}$, где $\varkappa(0) = \beta(0)\,c = c/b(0)$, $\bar{\varkappa} = \sum_{i\in C}\beta(i)$. И, следовательно,

$$\varkappa = \varkappa(0)\cdot e^{-\lambda(1-\bar{\varkappa})}. \qquad (3)$$

Л а п л а с о в с к а я м о д е л ь. В этой модели:

$$A(k) = C(k) = C;$$

$$\varkappa(0) = c/m, \ \bar{\varkappa} = c/m, \ \varkappa(n) = (c/m)^{n+1}; \qquad (4)$$

$$\varkappa = (c/m)\cdot e^{-\lambda(1-c/m)}. \qquad (5)$$

§4. Практическое вычисление вероятности правильного приема по марковской модели

Пусть $\lambda > 0$, начальное распределение вероятностей $p = (p_i)$ и стохастическая матрица $Q = (q_{ij})$, $i,j = 1,\dots,m$,задающая однородную марковскую цепь.

I. Особенности вычислений на ЭВМ. Вычисление $\varkappa$ (вероятности правильного приема сообщения) по формуле (I) весьма затруднено без использования ЭВМ. Однако вычисление $\varkappa(n)$ по формуле (2) становится проблематичным даже с помощью ЭВМ,если m и n велики: произведения вида $q_{i_0 i_1} q_{i_1 i_2} \quad q_{i_{n-1} i_n}$ являются элементами n-й степени матрицы Q размерности $(m \times m)$.

2. Методология эффективных вычислений. Для эффективности вычислений предлагается:

а) понизить размерность матрицы Q, укрупнив состояния исходной марковской цепи, т.е. сведя исходную марковскую цепь к марковской цепи с меньшим числом состояний;

б) использовать специальное представление матрицы, учитывая наличие большого числа нулей в стохастической матрице;

в) вычислять $\varkappa(n)$ по рекуррентным формулам.

У к р у п н е н и е с о с т о я н и й и с х о д н о й м а р к о в с к о й ц е п и. Словарь $A = \{1,\dots,m\}$ будем отождествлять с множеством состояний исходной марковской цепи. Пусть $\hat{A} = \{\hat{A}_1, \hat{A}_2, \dots, \hat{A}_s\}$ - такое разбиение множества A, что для любых $i,j \in \{1,\dots,s\}$, $\hat{A}_i \cap \hat{A}_j = \emptyset$ при $i \neq j$ $\bigcup_{i=1}^{s} \hat{A}_i = A$ и $s < m$.

Укрупнение состояний исходной марковской цепи возможно, если выполняются условия теоремы об укрупнении [3]:

Для того чтобы состояния марковской цепи можно было укрупнить посредством разбиения $\hat{A}$, необходимо и достаточно, чтобы для любых двух множеств $\hat{A}_i$ и $\hat{A}_j$ вероятности $q_{k\hat{A}_j}$ имели одно и то же значение для всех $k \in \hat{A}_i$.

Эти общие значения $\hat{q}_{ij}$ и образуют переходную матрицу $\hat{Q}$ укрупненной цепи, причем $\hat{q}_{ij} = \sum_{l \in \hat{A}_j} q_{kl}$ для любого $k \in \hat{A}_i$.

В дальнейшем будем считать, что укрупнение состояний исходной цепи возможно, и будем опускать значок $\wedge$, подразумевая, что Q - матрица переходов, q_{ij} - вероятности переходов укрупненной марковской цепи. Заметим, что укрупнение состояний исходной цепи не изменяет результата вычислений.

Специальное представление матрицы Q позволяет значительно сэкономить память при реализации алгоритма на ЭВМ. Предположим, что произвольная i-я строка матрицы Q имеет вид:

$$\left(0, \ldots, t_1/b(i), \ldots, t_2/b(i), \ldots, t_j/b(i), \ldots, t_{\hat{b}(i)}/b(i), \ldots, 0\right),$$

где $\hat{b}(i)$ - число ненулевых элементов в строке укрупненной матрицы; $b(i)$ - число ненулевых элементов в соответствующей строке исходной матрицы; $t_j \geq 1$, $\sum_{1 \leq j \leq \hat{b}(i)} t_j = b(i)$; кружком выделены элементы, стоящие в столбцах j_1 и j_2, так как слова с номерами j_1 и j_2 с указанной вероятностью принимаются неправильно после слова с номером i.

Предлагается матрицу Q размерности $(m \times m)$ представить двумя целочисленными массивами: M1 длиной L, в котором строки расположены последовательно и $L = \sum_{1 \leq i \leq m} l(i)$, $l(i)$ - число слов, занимаемых i-й строкой, и M2, который содержит индексы первых элементов строк в M1.

Произвольная i-я строка представляется в массиве M1 набором чисел: $b(i)$ $z(i)$ $a(i)$, $u_1, u_2, \ldots, u_{a(i)}$, где $b(i)$ - число расположенных в i-й строке ненулевых элементов в соответствующей строке исходной матрицы; $z(i)$ - число неправильно принимаемых слов после i-го слова; $a(i)$ - число правильно принимаемых слов после i-го слова; u_j содержат номера правильно принимаемых слов после i-го слова, причем

$$u_j = \begin{cases} k_j, \text{ если } t_j = 1 \ (k_j \in A(i) - \text{множество номеров правильно принимаемых слов после } i\text{-го слова}); \\ k_j + t_j \cdot 10^h, \text{ если } t_j > 1, \end{cases}$$

число h подбирается таким, чтобы выполнялось условие $10^h \ge m_0$, где m_0 - размерность строки исходной стохастической матрицы до укрупнения. Отсюда следует, что $l(i) = a(i) + 3$ и $L = 3m + \sum_{1 \le i \le m} a(i)$, а полные затраты памяти ЭВМ на специальное представление матрицы Q равны $(L+m)$ слов памяти.

Массив М2 позволяет осуществлять доступ к произвольным элементам строки, так как содержит индексы (номера) слов, содержащих числа $b(i)$ в массиве М1. Отметим, что суть приведенного ниже алгоритма - в последовательном извлечении элементов строки из массива данных, а не в произвольном поиске этих элементов.

Рекуррентные формулы для вычисления $\varkappa(n)$. Алгоритм вычисления вероятности правильного приема n слов основан на применении рекуррентных формул для расчета $\varkappa(n)$ и позволяет значительно сэкономить память ЭВМ, так как в нем нет непосредственных возведений в степень матрицы Q.

По формуле (2) имеем:

$$\varkappa(n) = \sum_{i_0 \in A(0)} \sum_{i_1 \in A(i_0)} \dots \sum_{i_n \in A(i_{n-1})} p_{i_0} q_{i_0 i_1} q_{i_1 i_2} \dots q_{i_{n-1} i_n}.$$

Введем следующие обозначения:

$$\varkappa^i(0) = \sum_{j \in A(i)} q_{ij},\ \varkappa^i(1) = \sum_{j \in A(i)} q_{ij} \varkappa^j(0), \dots, \varkappa^i(n) = \sum_{j \in A(i)} q_{ij} \varkappa^j(n-1).$$

Проследим, как строятся вычисления $\varkappa(n)$ по шагам. На нулевом шаге вычислим:

$$\varkappa(0) = \sum_{i_0 \in A(0)} p_{i_0} \quad \text{и} \quad \varkappa^i(0) = \sum_{j \in A(i)} q_{ij},\ \forall i \in \{1,\dots,m\}.$$

Заметим, что в силу стохастичности матрицы Q, $\sum_{j \in B(i)} q_{ij} = 1$, $\forall i \in \{1,\dots,m\}$ (суммирование ведется по множеству индексов $B(i) \subseteq A$, на котором $q_{ij} > 0$). Однако существуют такие $i \in \{1,\dots,m\}$, для которых $\varkappa^i(0) < 1$, ввиду соотношений $A(i) \subseteq B(i)$.

Убедимся, что вычисления на текущем шаге зависят от выражений, полученных на предыдущем шаге:

$$\varkappa(1) = \sum_{i_0 \in A(0)} \sum_{i_1 \in A(i_0)} p_{i_0} q_{i_0 i_1} = \sum_{i_0} p_{i_0} \sum_{i_1} q_{i_0 i_1} = \sum_{i_0} p_{i_0} \cdot \varkappa^{i_0}(0).$$

Вместе с $\varkappa(1)$ вычислим на первом шаге

$$\varkappa^i(1) = \sum_{j \in A(i)} q_{ij}\, \varkappa^j(0), \quad \forall i \in \{1,\ldots,m\}.$$

Далее, на втором шаге вычислим:

$$\varkappa(2) = \sum_{i_0 \in A(0)} \sum_{i_1 \in A(i_0)} \sum_{i_2 \in A(i_1)} p_{i_0} q_{i_0 i_1} q_{i_1 i_2} =$$

$$= \sum_{i_0} p_{i_0} \sum_{i_1} q_{i_0 i_1} \sum_{i_2} q_{i_1 i_2} = \sum_{i_0} p_{i_0} \sum_{i_1} q_{i_0 i_1} \varkappa^{i_1}(0) = \sum_{i_0} p_{i_0} \varkappa^{i_0}(1)$$

и все $\varkappa^i(2)$; на k-м шаге $\varkappa(k)$ и $\varkappa^i(k)$, а на последнем n-м шаге достаточно вычислить $\varkappa(n)$:

$$\varkappa(n) = \sum_{i_0 \in A(0)} \sum_{i_1 \in A(i_0)} \cdots \sum_{i_{n-1} \in A(i_{n-2})} \sum_{i_n \in A(i_{n-1})} p_{i_0} q_{i_0 i_1} \cdots$$

$$\cdots q_{i_{n-2} i_{n-1}} q_{i_{n-1} i_n} = \sum_{i_0} p_{i_0} \sum_{i_1} q_{i_0 i_1} \cdots \sum_{i_{n-1}} q_{i_{n-2} i_{n-1}} \sum_{i_n} q_{i_{n-1} i_n} =$$

$$= \sum_{i_0} p_{i_0} \sum_{i_1} q_{i_0 i_1} \cdots \sum_{i_{n-1}} q_{i_{n-2} i_{n-1}} \varkappa^{i_{n-1}}(0) = \ldots = \sum_{i_0} p_{i_0} \varkappa^{i_0}(n-1).$$

Таким образом, $\varkappa(0), \varkappa(1), \ldots, \varkappa(n)$ вычисляются за $(n+1)$ шаг, а общая рекуррентная формула для вычислений $\varkappa(i)$ для $i \in \{1,2,\ldots$ $\ldots, N, \ldots\}$, $j \in \{1,2,\ldots,m\}$ имеет вид:

$$\varkappa(i) = p\, K(i-1), \tag{6}$$

где $p = (p_{i_0})$ - вектор начального распределения; $K(0) = (\varkappa^j(0))$, $K(i) = (\varkappa^j(i))$, $i \geq 1$,

$$\varkappa^j(0) = \sum_{i \in A(j)} q_{ji},$$
$$\varkappa^j(i) = \sum_{l \in A(j)} q_{jl}\, \varkappa^l(i-1). \tag{7}$$

При реализации алгоритма вычислений $\varkappa(i)$ по рекуррентным соотношениям через $\varkappa^j(i)$ необходимы дополнительные затраты памяти ЭВМ для хранения: массива $\varkappa(i)$ (длиной n); массива значений $\varkappa^j(i)$, вычисленных на предыдущем шаге (длиной m); массива значений $\varkappa^j(i)$, вычисляемых на текущем шаге (длиной m).

3. Алгоритм вычисления $\varkappa(i)$. Предположим, что начальное распределение вероятностей совпадает с первой строкой матрицы Q, на нулевом шаге вычислены $\varkappa^j(0)$ и записаны в массив D. Опишем шаг алгоритма вычисления $\varkappa^j$ на текущем l-м шаге. Используемые обозначения:

D – массив предыдущих значений $\varkappa^j$;

D1 – массив текущих значений $\varkappa^j$;

M1 – массив специального представления строк матрицы Q;

M2 – массив номеров первых элементов строк в массиве M1;

$H = 10^h$;

DKN – массив $\varkappa(l)$, $l \in \{0,1,\ldots,n\}$;

mod(I,J) – стандартная функция получения остатка от деления целых чисел I на J;

entier(X) – функция получения целой части действительного X;

K, K1, M3, N1, NT, DEL, A, C, F – рабочие ячейки.

Описание переменных и фрагмент программы на псевдоалголе

```
begin integer i,j,m,n,l,H,L,K,K1,M3,N1,NT,DEL;
input (m,n,H,L);
begin real A,C,F; integer array M1 [1:L], M2 [1:m];
real array D,D1 [1:m], DKN [0:n];
                         . . .
for j:= 1 until m do   начало цикла по j;
NT:= M2[j];            засылка индекса I-го элемента j-й строки;
DEL:= M1[NT];          засылка b(j);
NT:= NT+2;             вычисление индекса a(j);
N1:= M1[NT];           засылка a(j);
F:= 0;                 обнуление сумматора;
for i:= 1 until N1 do  начало цикла по i;
NT:= NT+1;             вычисление индекса u_i;
K:= M1[NT];            засылка u_i;
C:= K/H;               вычисление числителя
K1:= entier (C);       вероятности перехода t_i;
if K1=0 then A:= 1 else A:= K1;  засылка t_i в A;
M3:= mod (K,H);        вычисление номера k_i;
F:= F+A*D[M3]          накопление сумматора;
od;                    конец цикла по i;
```

`D1[j]:= F/DEL`	вычисление текущего $\varkappa^j(l) = 1/b(j)\cdot\sum_i t_i \cdot \varkappa^i(l-1)$ – реализация формулы (7);
`od;`	конец цикла по j;
`DKN[l]:= D1[1];`	засылка $\varkappa(l)$;
`for i:= 1 until m do`	перезапись массива текущих $\varkappa^j$ в
`D i := D1[i] od;`	массив предыдущих $\varkappa^j$.

. . .

Таким образом, основная тяжесть вычислений реализуется двумя вложенными циклами приведенного алгоритма.

4. <u>Резюме</u>. Вычисление $\varkappa = e^{-\lambda}\sum_{n\geq 0}(\lambda^n/n!)\cdot\varkappa(n)$, если вычислены $\varkappa(n)$, не представляет затруднений.

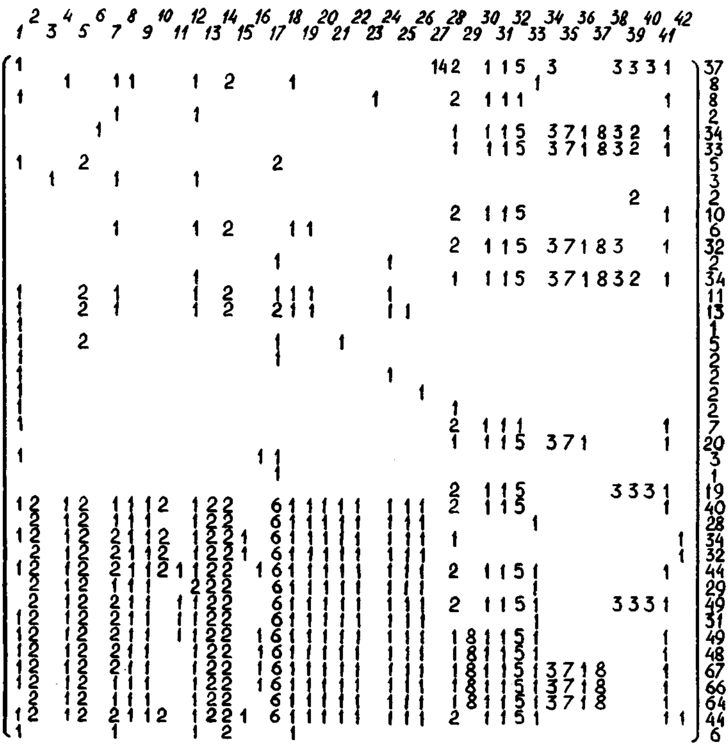

Рис. I

5. Численные эксперименты

Методы расчета вероятности правильности приема сообщения по марковской и лапласовской моделям были использованы для оценки работы алгоритма распознавания принимаемых слов в реальном эксперименте. Исходный словарь, интерпретируемый состояниями марковской цепи,содержал 106 слов. По экспериментальным данным были составлены матрица переходов и начальный вектор вероятностей, задающие марковскую цепь. При проведенных экспериментах в работе алгоритма распознавания варьировался критерий сравнения принятого слова с эталоном, в результате изменялось число неправильно при-

Таблица

n	МОДЕЛЬ ЛАПЛАСА	МОДЕЛЬ МАРКОВА	МАРКОВСК. ПРИБЛИЖ.	МОДЕЛЬ ЛАПЛАСА	МОДЕЛЬ МАРКОВА	МАРКОВСК. ПРИБЛИЖ.
	$k = 1$			$k = 4$		
0	.99056604	1.0	1.0	.96226415	1.0	1.0
1	.98122107	1.0	.99943978	.92595229	.98810455	.98881451
2	.97196427	.99929196	.99832027	.89101070	.97314840	.97775414
3	.96279479	.99819630	.99832027	.85738765	.96162434	.96681748
4	.95371183	.99738867	.99776099	.82503340	.95036845	.95600316
5	.94471455	.99672954	.99720202	.79390007	.93891663	.94530979
6	.93580215	.99599819	.99664336	.76394157	.92743831	.93473604
7	.92697382	.99522343	.99608502	.73511359	.91614680	.92428056
8	.91822879	.99446137	.99552698	.70737345	.90505456	.91394203
9	.90956625	.99371287	.99496927	.68068012	.89409179	.90371915
10	.90098544	.99296396	.99441186	.65499407	.88324483	.89361061
20	.81950595	.98547462	.98885495	.44584144	.78177847	.79853992
30	.74539495	.97804299	.98332909	.30347541	.69197093	.71358374
40	.67798609	.97066739	.97783411	.20656967	.61348011	.63766600
50	.61667327	.96334743	.97236984	.14060787	.54212087	.56982510
	$k = 8$			$k = 13$		
0	.92452830	.94594595	.94594595	.87735849	.89189189	.89189189
1	.85475258	.89398683	.90279381	.76975792	.80012826	.81550164
2	.79024295	.85034985	.86161019	.67535365	.73064290	.74565419
3	.73060197	.81186651	.82230528	.59252726	.67245131	.68178915
4	.67546220	.77344650	.78479339	.51985882	.61551733	.62339413
5	.62448392	.73572976	.74899271	.45610255	.56144537	.57000063
6	.57735306	.70018687	.71482518	.40016544	.51270438	.52118026
7	.53377924	.66666269	.68221631	.35108855	.46870629	.47654134
8	.49349402	.63472249	.65109498	.30803052	.42846006	.43572573
9	.45624218	.60423457	.62139335	.27025319	.39154446	.39840596
10	.42181529	.57519988	.59304665	.23710893	.35778876	.36428262
20	.19245288	.35164698	.37180172	.06407945	.14537135	.14878690
30	.08780647	.21497710	.23309552	.01731767	.05906427	.06077024
40	.04006163	.13142485	.14513575	.00468016	.02399777	.02482088
50	.01827809	.08034573	.09161762	.00126483	.00975027	.01013779

нимаемых слов от I до I3. Например, при самом слабом критерии сравнения неправильно принималось слово с номером 26, при уточнении критерия сравнения стали неправильно приниматься слова с номерами 26 и I06, далее, при следующем уточнении - 26, I06, 22 и т.д., т.е. при новом критерии сравнения появилось новое неправильно принимаемое слово.

Укрупнив исходные состояния, удалось исходную цепь свести к марковской цепи с 42 состояниями. На рис.I изображена укрупненная матрица, в которой ненулевые элементы представлены числителем t_j , а знаменатели $b(i)$ указаны справа от строки; нулевые элементы пропущены. Специальное представление матрицы переходов потребовало 637 слов памяти на массив строк и 42 слова на массив индексов первых элементов строк. Таким образом, затраты памяти на хранение специального представления матрицы переходов составили 679 слов вместо (42x42)x2 = 5528 слов, если бы вещественная матрица переходов хранилась в памяти полностью (на одно вещественное число отводится два слова памяти).

Алгоритмы вычисления вероятностей правильного приема сообщения при k неправильно принимаемых словах (k = I,2,...,I3) по марковской, лапласовской и приближенной марковской моделям были запрограммированы на языке Фортран и реализованы на ЭВМ СМ-I. Отметим, что шаг вычислений $\varkappa(i)$ и $\varkappa^j(i)$ по марковской модели при фиксированном i длится ≈5 сек.

В приближенной марковской модели вероятность правильного приема сообщения вычисляется по формуле

$$\tilde{\varkappa} = \tilde{\varkappa}(0)\cdot e^{-\lambda(1-\bar{\tilde{\varkappa}})} = \tilde{\varkappa}(0)\cdot e^{-\lambda}\cdot\sum_{n\geq 0}(\lambda^n/n!)\cdot\tilde{\varkappa}(n), \qquad (8)$$

где $\tilde{\varkappa}(0) = \dfrac{a(0)}{b(0)}$ - вероятность правильного приема начального слова, $a(0)$ - число правильно принимаемых слов после начального, $b(0)$ - число всех принимаемых слов после начального;

$$\bar{\tilde{\varkappa}} = \frac{1}{m-k}\sum_{1\leq i\leq m} a(i)/b(i)$$

- средняя вероятность правильного приема слова, m - число слов (состояний), k - число неправильно принимаемых слов, i пробегает значения по номерам строк, $a(i)$ - число правильно принимаемых слов после i-го слова, $b(i)$ - число всех принимаемых слов после i-го слова; а

$$\tilde{\varkappa}(n) = \tilde{\varkappa}(0)\cdot\bar{\tilde{\varkappa}}^n. \qquad (9)$$

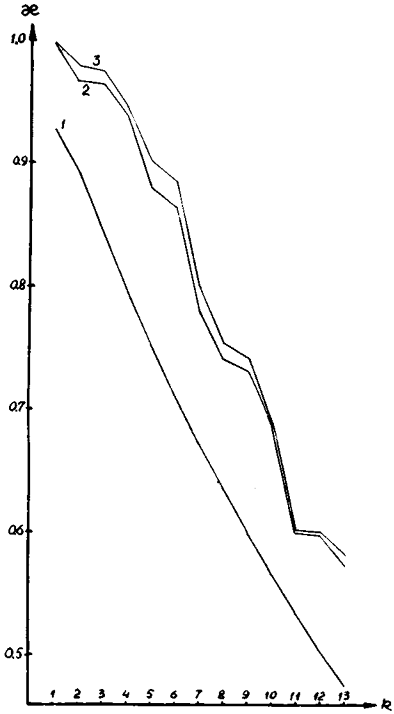

Рис. 2

В таблице представлены величины $\varkappa(n)$ – вероятности правильного приема сообщения длины n, если неправильно принимаются k слов, вычисленные по трем моделям: лапласовской (4), марковской (2) и приближенной марковской (9). На рис.2 отображено изменение $\varkappa$ при $\lambda = 5$, если варьируется k – число неправильно принимаемых слов (I – лапласовская модель (5), 2 – марковская (I), 3 – марковская приближенная (8)). Из таблицы и рис.2 видно, что марковское приближение дает результаты, близкие к точным, но несколько бо́льшие по величине.

Таким образом, учет лингвистических ограничений существенно повышает надежность распознавания фраз дискретной речи. Например, для типичного случая средней длины фразы 5 слов и надежности распознавания изолированных слов ≈96% при словаре 106 слов надежность распознавания дискретной речи повышается с 80% для лапласовской модели до 94% для марковской.

Литература

I. РУДНЫЙ Б.Н., ТРУНИН-ДОНСКОЙ В.Н. Современное состояние работ по речевому управлению. –В кн.: Речевое управление, 1972, с. 3–16.

2. ФАЙНСТЕЙН А. Основы теории информации.–М.:ИЛ,1960.–140 с.

3. КЕМЕНИ Дж., СНЕЛЛ Дж. Конечные цепи Маркова. –М.: Наука, 1970. – 272 с.

Поступила в ред.-изд.отд.
8 декабря 1983 года