Майер Роберт Валерьевич,
д-р пед. наук, профессор кафедры ФДФ,
ФГБОУ ВО «Глазовский государственный педагогический институт»,
г. Глазов, Удмуртская Республика, Россия

ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ ЧТЕНИЯ ТЕКСТА

Рассмотрена компьютерная программа, моделирующая чтение текста, которая учитывает вероятностный характер восприятия слогов, слов и предложений, закономерности понимания и забывания информации.

Ключевые слова: кибернетика, моделирование, программирование, чтение.

Robert V. Mayer,
Doctor of Pedagogical Sciences, Professor of Physics and Physics Didactics Chair,
FSBEI of HE «The Glazov Korolenko State Pedagogical Institute»,
Glazov, Udmurt Republic, Russia

IMITATING MODELING OF THE TEXT READING

The computer program simulating reading of the text is considered which takes into account the stochastic character of syllables, words and sentences recognition, the regularities of understanding and forgetting of the information.

Keywords: cybernetics, modeling, programming, reading.

Рассмотрим кибернетическую модель чтеца [2, 3]. Память чтеца можно промоделировать тремя массивами $Z_1(i)$, $Z_2(j,k)$, $Z_3(k)$, где i, j, k – номера слога, слова и предложения соответственно. Их элементами являются вероятности правильного воспроизведения чтецом слогов, слов и предложений, лежащие в интервале [0; 1]. Пусть человек читает j–ое слово в k–ом предложении. Распознавание слогов моделируется как случайный процесс с вероятностью $p=0{,}05$ – 1, при этом время t увеличивается на T_1. Если случайное x из интервала [0; 1] меньше вероятности p, то чтец сумел прочитать слог, элементу $Z_1(i)$ ($i=1,2,\ldots,L$) присваивается 1; если $x \geq p$ – все повторяется снова. Когда человек правильно прочитал все слоги, проверяется условие $Z_1(1)\cdot Z_1(2)\cdot\ldots\cdot Z_1(L) > 0{,}8 + x'$, где x' – случайная величина из интервала [0; 0,1]. Если оно выполняется, то чтец понял $j-$ое слово в $k-$ом предложении, $Z_2(j,k)$ присваивается 1, t увеличивается на T_2. Чтец понимает предложение, если помнит все входящие в него слова. Вероятность понимания $k-$ого предложения из N слов равна $P_k = Z_2(1,k)\cdot Z_2(2,k)\cdot\ldots\cdot Z_2(N,k)$. Если $P_k > 0{,}8 + x'$, то будем считать, что предложение понято. Степень понимания текста характеризуется суммой всех $Z_3(k)$: $Zn = Z_3(1) + Z_3(2) + \ldots + Z_3(M)$. Если $Zn > 0{,}8 + x'$, то считается, что чтец понял текст, а иначе он читает его еще раз.

Ребенок, читающий медленно, дойдя до конца длинного слова (предложения), часто не может вспомнить его начало, но при следующей попытке читает это слово (предложение) быстрее. Модель должна учитывать, что во время каждой операции частично забывается запомнившаяся ранее информация. До того, как чтец сложил слово, слоги относятся к логически несвязанной информации; они быстро забываются по экспоненциальному закону. Забывание логически связанной информации (слов, предложений) происходит по логистическому закону [1]. Получаем уравнения:

$$dZ_1(i)/dt = -\gamma\cdot Z_1(i)(0{,}3 + \exp\{-s_1(i,j)/5\}),\ \gamma = 0{,}015,\ a = 0{,}07,\ b = a/10,$$

$$dZ_2(j,k)/dt = -a(Z_2(i,k) - 0{,}1)\cdot(1{,}002 - Z_2(i,k))\cdot(0{,}3 + \exp\{-s_2(j)/5\}),$$

$$dZ_3(k)/dt = -b(Z_3(k) - 0{,}1)\cdot(1{,}002 - Z_3(k))\cdot(0{,}3 + \exp\{-s_3/5\}),$$

где $s_1(i,j)$, $s_2(j)$, s_3 позволяют учесть количество обращений к $(i,j)-$слову, $j-$предложению или всему тексту. Если чтец правильно понял слово или предложение, то $s_1(i,j)$, $s_2(j)$ увеличиваются на 2, а если неправильно – на 1; все это приводит к уменьшению скорости забывания информации и времени, требующемуся для выполнения этих операций. В течение одной попытки чтения текста одно и то же слово или предложение чтец прочитывает не более трех раз подряд. Используется программа 1.

Программа 1. Компьютерная модель чтения текста (Free Pascal).

```
{$N+}uses crt,graph; const L=6; M=20; N=10; dt=0.2; g1=0.009; a=0.07;
b=a/10; Mt=0.12;var DV,MV,Q,i_pr,i_sl,j_sl,i_b,j_pr,j_b,s3,ii,d1,d2,w:
integer; p,pp,sl,x,t,t1,iq: single; s1:array[0..11,0..M+1]of integer;
s2:array[0..M+1]of integer; z1:array[0..M+1]of single; z2:array[0..11,
0..M+1]of single; z3: array[0..M+1] of single; Label m1, m2, m3;
Procedure Time; begin For ii:=1 to w do begin t:=t+dt; For j_b:=1 to L
do z1[j_b]:=z1[j_b]-g1*(z1[j_b])*dt*(0.3+exp(-0.2*s1[i_sl,i_pr])); For
j_sl:=1 to N do For j_pr:=1 to M do begin If z2[j_sl,j_pr]>0.1 then
z2[j_sl,j_pr]:=z2[j_sl,j_pr]-a*(z2[j_sl,j_pr]-0.1)*(1.002-z2[j_sl,j_pr]
)*dt*(0.3+1*exp(-0.2*s2[j_pr])); end; For j_pr:=1 to M do If z3[j_pr]>
0.15 then z3[j_pr]:=z3[j_pr]-b*(z3[j_pr]-0.15)*(1.002-z3[j_pr])*dt*(0.3
+1*exp(-0.2*s3)); end; end;
BEGIN DV:=Detect; InitGraph(DV,MV,''); Randomize; line(0,600,1400,600);
p:=0.6; Repeat i_pr:=1; Repeat i_sl:=1; t1:=t; iq:=i_pr; circle(10+round
(t*Mt),600-round(i_pr*20),2); line(10+round(t*Mt),600-round(i_pr*20),10+
round(t1*Mt),600-round(iq*20)); Repeat i_b:=1; {==} Repeat m2: w:=round
(2+2*exp(-0.1*s1[i_sl,i_pr])); Time; sl:=random(100)/100; if sl<p+0.0*
s1[i_sl,i_pr] then begin z1[i_b]:=1; inc(i_b); end else goto m2; until
i_b>L; w:=3; Time; inc(s1[i_sl,i_pr]); pp:=1; sl:=random(10)/100; For
i_b:=1 to L do pp:=pp*z1[i_b];If 0.8<pp then begin z2[i_sl,i_pr]:=1; inc
(s1[i_sl,i_pr]); d1:=s1[i_sl,i_pr]; inc(i_sl); end else begin z2[i_sl,
i_pr]:=0.2; end;If (s1[i_sl,i_pr]-d1>2)and(i_sl<=N) then begin inc(i_sl);
d1:=s1[i_sl,i_pr]; end; until (KeyPressed)or(i_sl>N); {==} w:=3; Time;
inc(s2[i_pr]); pp:=1; sl:=random(10)/100; For j_sl:=1 to N do pp:=pp*
z2[j_sl,i_pr]; If 0.7+0*sl<pp then begin z3[i_pr]:=1; inc(s2[i_pr]);
inc(i_pr); d2:=s2[i_pr]; end else begin z3[i_pr]:=0.33;end; If (s2[i_pr]
-d2>2)and(i_pr<=M) then begin inc(i_pr); d2:=s2[i_pr]; end; pp:=0; For
j_pr:=1 to M do pp:=pp+z3[j_pr]/M; circle(10+round(Mt*t),600-round(pp*
400),2); circle(10+round(Mt*t),600-round(pp*400),3); circle(10+round(Mt
*t),600-round(400),1); sl:=random(10)/100; If (0.85<pp)and(s3>0) then
begin Q:=1; goto m1; end; until (KeyPressed)or(i_pr>M); circle(10,10,10);
pp:=0; For j_pr:=1 to M do pp:=pp+z3[j_pr]/M; sl:=random(10)/100; If 0.85
+0*sl>pp then begin inc(s3); i_pr:=1; i_sl:=1; end else Q:=1; m1: w:=2;
Time; until (KeyPressed)or(Q=1); Readkey; CloseGraph; END.
```

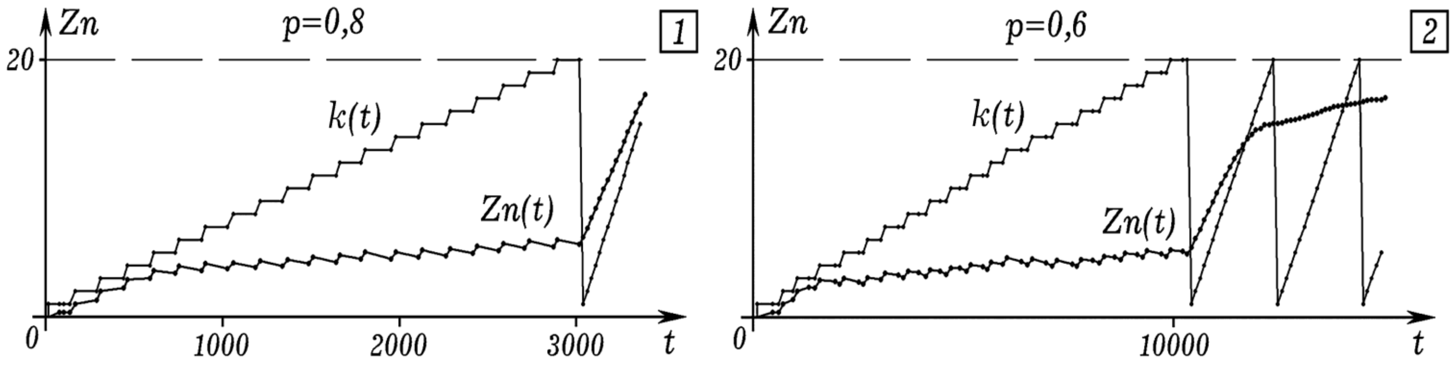

Рисунок 1 – Результаты моделирования чтения текста при $p = 0{,}8$ и 0,6.

На рис. 1 представлены графики $Zn(t)$ и $k(t)$ (k – номер читаемого предложения) для двух чтецов с различными вероятностями правильного чтения слогов p (0,8 и 0,6; $\gamma = 0{,}015$). Когда $p = 0{,}8$, чтец понимает текст со второй попытки за время 3300 УЕВ (усл. ед. времени). Если $p = 0{,}6$, то чтец понимает текст с четвертого раза за время 16000 УЕВ. Видно, что при чтении сложного текста (например, на иностранном языке) во время первого прочтения уровень понимания $Zn(t)$ растет медленно, чтец как бы подготавливает основу для понимания текста. Во время второй и/или третьей попытки наблюдается скачок Zn, – ученик схватывает суть текста, его знания резко увеличиваются. Затем Zn снова растет медленно. Так решается любая сложная проблема: сначала человек набирает информацию, создавая предпосылки для «скачка понимания», затем происходит «озарение», а потом решение постепенно совершенствуется. Параметры чтеца (p, T_1, T_2 и T_3) вместе с параметрами текста (L, N, M) определяют время чтения текста. Рассмотрим графики зависимостей понимания различных предложений $Z_3(k)$ в моменты времени t_1 и $t_2 > t_1$ (рис. 2). Когда человек дочитывает текст, он хорошо помнит последние предложения $(k > 15)$ и плохо первые $(k > 5)$. При повторном чтении уровень знаний первых предложений снова повышается, после чего они забываются медленнее.

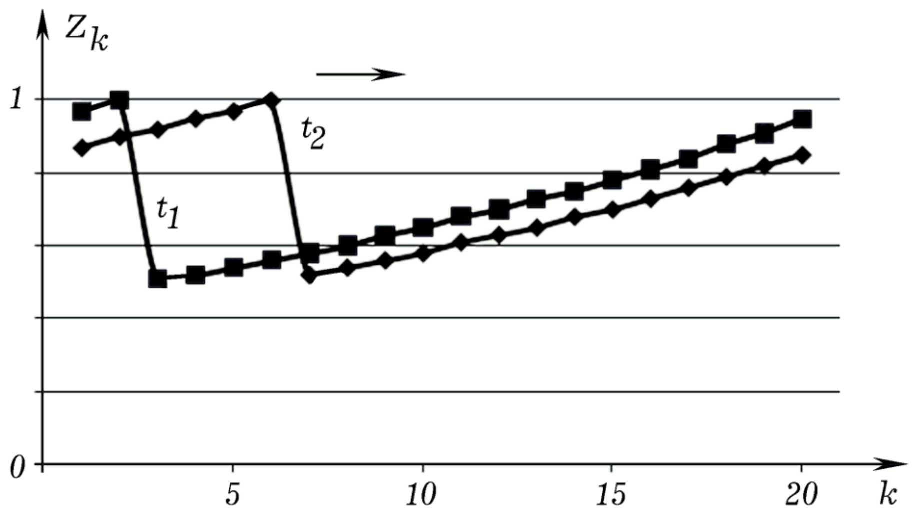

Рисунок 2 – Зависимость понимания предложений от k в моменты t_1 и $t_2 > t_1$.

СПИСОК ЛИТЕРАТУРЫ

1. Майер Р.В. Исследование математических моделей дидактических систем на компьютере: монография. – Глазов: Глазов. гос. пед. ин-т, 2018. – 160 с.

2. Теслер Г.С. Новая кибернетика. – Киев: Логос, 2004. – 401 с.

3. Hunt E. The Mathematics of Behavior. – New York: Cambridge University Press, 2007. – 346 p.