

ОБ ИСПОЛЬЗОВАНИИ КОМПЬЮТЕРОВ ДЛЯ ОЦЕНКИ СЛОЖНОСТИ УЧЕБНЫХ ПОНЯТИЙ

Майер Роберт Валерьевич,

доктор педагогических наук, профессор кафедры физики и дидактики физики ГОУ ВПО «Глазовский
государственный педагогический институт»

robert_maier@mail.ru

Аннотация

В статье обсуждается проблема автоматизации процедуры оценки семантической сложности «обычных» слов и научных понятий, что необходимо для определения сложности учебных текстов. Под сложностью понятия P относительно тезауруса Z понимают величину, численно равную наименьшему количеству «простых» слов из тезауруса Z , требующихся для объяснения сущности понятия P . При этом все научные понятия могут быть разделены на 7 категорий, которым соответствуют следующие значения сложности: 1 – 2 – 4 – 8 – 16 – 32 – 64. Для оценки сложности понятий применяются метод парных сравнений, метод сопоставления понятий со шкалой сложности и метод «вычисления» сложности. Рассмотрены три компьютерных программы на языке Pascal: 1) программа, которая в случайном порядке предъявляет пары сравниваемых понятий (объектов), воспринимает ответы эксперта и записывает результаты в файл; 2) программа, обрабатывающая получающийся файл и вычисляющая сложность понятия; 3) программа, выводящая на экран оцениваемые объекты (научные понятия) и шкалу сложности, содержащую опорные понятия, воспринимающая оценки эксперта и вычисляющая результирующую оценку. Предложенные программы могут быть использованы для оценки других качеств научных понятий (объектов).

Ключевые слова — парные сравнения, оценивание, понятие, программирование, сложность, тезаурус.

I. ВВЕДЕНИЕ

Одна из актуальных проблем дидактики состоит в разработке методов оценки дидактической сложности учебных текстов [1]. Ее решение связано с оценкой семантической сложности «обычных» слов и научных понятий, составляющих текст. При этом под сложностью понятия относительно некоторого тезауруса Z (например, тезауруса пятиклассника) будем понимать величину s , равную наименьшему количеству «простых» слов из тезауруса Z , которое необходимо произнести, чтобы объяснить сущность данного понятия. Считается, что все «обычные» слова и научные понятия, встречающиеся в школе, могут быть разделены на 7 категорий, которым соответствуют следующие значения сложности: 1 – 2 – 4 – 8 – 16 – 32 – 64 [2].

Для оценки семантической сложности понятий могут использоваться следующие методы: 1) метод подсчета слов в максимально кратком определении или объяснении понятия; 2) метод карточек, заключающийся в разложении карточек по шкале сложности; 3) метод парных сравнений, состоящий в сравнении пар понятий друг с другом и выявлении более сложного.

Обсуждаемая проблема носит более широкий характер, чем кажется, так как сравниваться и оцениваться могут не только понятия, но и другие объекты. Для ее решения удобно использовать компьютер, позволяющий предъявлять эксперту различные пары понятий (объектов), воспринимать вводимую информацию, вычислять среднее, записывать результаты оценок или парных сравнений в текстовый файл.

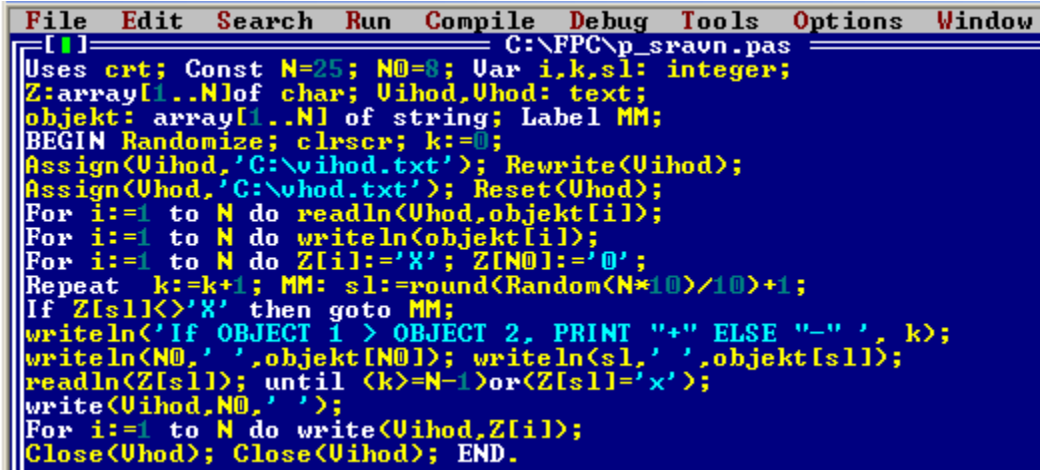
Цель статьи состоит в создании компьютерных программ, автоматизирующих и рандомизирующих процесс парного сравнения и оценки сложности понятий, сохраняющих результаты оценки в текстовый файл. В качестве **методологической основы** использовались работы Ю.А. Алябышевой, А.Ю. Антонова и А.А. Веряева [3], В.В. Гучука [4], Н.В. Лукашевича [5], Вал.А. Лукова и Вл.А. Лукова [6, 7], И.С. Тулохоновой [8] (тезаурусный подход), Г. Дэвида [9], В.А. Дюка [10], Е.К. Корноушенко [11], А.И. Орлова [12], И.В. Садовникова [13], В.С. Черепанова [14], (методы оценок), Р.В. Майера [1, 2], В. Davis и D. Sumara [15] (сложность текстов), В.Б. Попова [16] (программирование на Pascal).

II. АВТОМАТИЗАЦИЯ ПАРНЫХ СРАВНЕНИЙ ПОНЯТИЙ С ЦЕЛЮ ОЦЕНКИ ИХ СЛОЖНОСТИ

Оценка и ранжирование нескольких десятков понятий (объектов) вручную является непростой задачей. Например, процедура оценивания 30 объектов методом парных сравнений требует осуществления около 450 различных сопоставлений. Во время каждого из них эксперт должен сравнить степень наличия оцениваемого качества (то есть сложности) в двух сопоставляемых понятиях. Для парного сравнения понятий и получения матрицы оценок удобно применять электронные таблицы Excel. Оцениваемые понятия располагают в левом столбце и верхней строке; оценки -1, 0 и 1 ставят в ячейках пересечения строк и столбцов. Недостаток этого метода состоит в невозможности сопоставления понятий в случайном порядке и влиянии ранее поставленных оценок на текущие оценки эксперта.

Для повышения объективности результатов следует автоматизировать и рандомизировать этот процесс, используя специальную компьютерную программу. Она должна в случайном порядке выводить на экран сравниваемые понятия, воспринимать ответы эксперта и сохранять результаты в текстовом файле [1]. Так как оценивание может длиться 5–8 часов, то эксперт должен иметь возможность прервать работу с программой, сохранить результаты сравнений, а на следующий день вернуться и продолжить сопоставление понятий.

Применяемая нами программа ПР–1 написана в среде Free Pascal и представлена на рис. 1. Рассмотрим процесс оценивания сложности $N=25$ понятий из школьного учебника физики. Используя `File Manager`, создают файл `vhod.txt`, в который помещают список оцениваемых понятий (рис. 2.1). Случайно выбирают понятие с номером $N_0=8$, после чего запускают программу ПР–1. На экране появляются понятие $N_0=8$ и другое случайным образом выбранное понятие из списка, например, понятие 11. При этом эксперт должен сравнить сложности этих двух понятий и ввести с клавиатуры символы «+», «0» или «-». Плюс означает, что оцениваемого качества (то есть сложности) в понятии 8 заметно больше, чем в понятии 11, ноль — примерно одинаково, а минус — заметно меньше, чем в понятии 11.



```

File Edit Search Run Compile Debug Tools Options Window
C:\FPC\p_sraavn.pas
Uses crt; Const N=25; N0=8; Var i,k,sl: integer;
Z:array[1..N]of char; Uihod,Uhod: text;
objekt: array[1..N] of string; Label MM;
BEGIN Randomize; clrscr; k:=0;
Assign(Uihod,'C:\vhod.txt'); Rewrite(Uihod);
Assign(Uhod,'C:\uhod.txt'); Reset(Uhod);
For i:=1 to N do readln(Uhod,objekt[i]);
For i:=1 to N do writeln(objekt[i]);
For i:=1 to N do Z[i]:='X'; Z[N0]:='0';
Repeat k:=k+1; MM: sl:=round(Random(N*10)/10)+1;
If Z[sl]<>'X' then goto MM;
writeln('If OBJECT 1 > OBJECT 2, PRINT "+" ELSE "-" ', k);
writeln(N0,' ',objekt[N0]); writeln(sl,' ',objekt[sl]);
readln(Z[sl]); until (k>=N-1)or(Z[sl]='x');
write(Uihod,N0,' ');
For i:=1 to N do write(Uihod,Z[i]);
Close(Uhod); Close(Uihod); END.

```

Рис. 1. Программа ПР–1 (Free Pascal), автоматизирующая парные сравнения понятий.



Рис. 2. Список оцениваемых понятий (файл vnod.txt), результаты сравнений понятия 8 с другими понятиями (файл vnod.txt) и получающаяся матрица оценок.

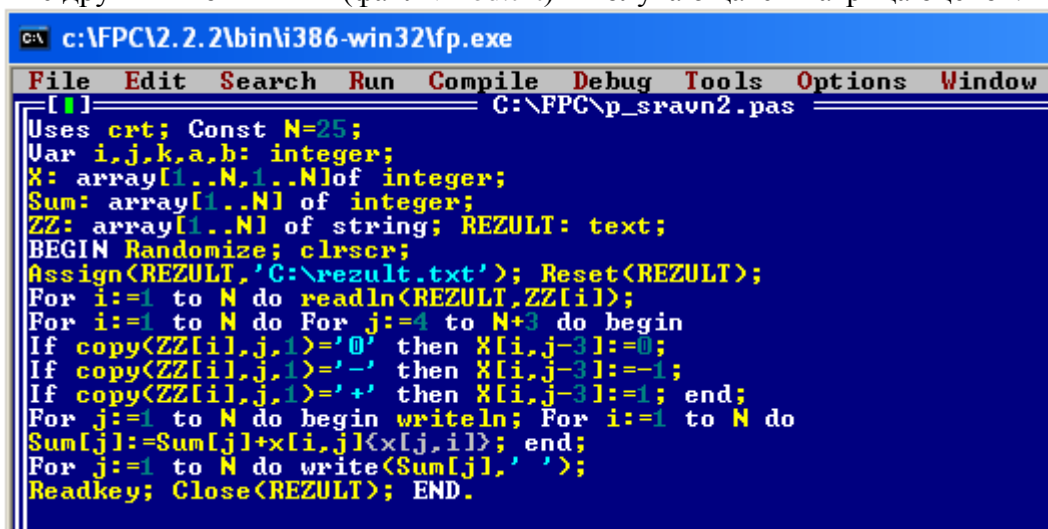


Рис. 3. Программа ПР-2 для обработки матрицы оценок, получающихся после использования программы ПР-1 .

Эксперт нажимает на «Enter», на экране монитора снова появляется понятие 8 и случайно выбранное понятие 19. Эксперт опять осуществляет сравнение их сложностей и с клавиатуры вводит “+”, “-“ или “0”. При этом понятие N_0 не сопоставляется повторно ни с каким другим понятием. Оно также не сравнивается с самим собой, так как результат известен, – оценка “0”. Когда процесс сравнения понятия 8 со всеми остальными понятиями из входного файла (рис. 2.1) заканчивается, программа ПР-1 создает текстовый файл vnod.txt. В нем присутствует одна строка, содержащая номер $N_0=8$ и оценки, поставленные экспертом в результате сопоставления понятия 8 с понятием 1, понятием 2, ..., понятием N . Например: «+ + + ... + 0 0 - + ... - 0 + - -» (рис. 2.2). Ее следует сохранить в файле rezult.txt. После этого все повторяется с другим объектом, например, с понятием $N_0=10$. Если необходимо, эксперт может сделать перерыв и выключить компьютер, предварительно сохранив результаты оценивания.

После перебора всех понятий, используя текстовый редактор, из получающихся строчек (рис. 2.2) формируют матрицу оценок, как это показано на рис. 2.3. Это двумерная матрица, с

левого верхнего угла к правому нижнему идет диагональ из нулей, а результаты сравнения двух понятий оказываются записанными на пересечении соответствующих строки и столбца. Полный перебор всех понятий приводит к тому, что каждая пара понятий сравнивается два раза (сначала i -ое понятие с j -тым, а потом наоборот). Это позволяет повысить надежность оценок.

Эта двумерная матрица оценок из плюсов, нулей и единиц (рис. 2.3), находящаяся в файле rezult.txt, анализируется компьютерной программой ПР-2 (рис. 3). Она для каждой строки определяет сумму всех плюсов, из нее вычитает сумму всех минусов и результат x_i для каждой i -той строки выводит на экран. Чтобы программа ПР-2 аналогичным образом обрабатывала столбцы матрицы, необходимо активизировать $x[j,i]$ и закоментировать $x[i,j]$. Получающиеся результаты y_i ($i = 1, 2, \dots, N$) также выводятся на экран. Сложность s_i оцениваемого понятия пропорциональна разности $x_i - y_i$. К этой разности можно прибавить некоторую постоянную, а результат умножить на коэффициент так, чтобы максимальное значение s_i из всего списка понятий равнялось единице, а минимальное – нулю.

III. АВТОМАТИЗАЦИЯ ОЦЕНИВАНИЯ ПОНЯТИЙ С ПОМОЩЬЮ ШКАЛЫ СЛОЖНОСТИ

Возможен другой подход к оценке сложности понятий. Компьютерная программа выводит на экран оцениваемое понятие и шкалу сложности 1 – 2 – 4 – 8 – 16 – 32 – 64 с указанием соответствующих опорных понятий. Эксперт, сопоставляя оцениваемое понятие с опорными, приблизительно определяет сложность s понятия и вводит ее значение с клавиатуры (например, 14). Эта процедура повторяется 3 – 5 раз; оцениваемое понятие каждый раз сравнивается с опорными терминами, которые выбираются случайным образом и не повторяются.

Недостаток этого метода состоит в том, что эксперт запоминает значения сложности, присвоенные оцениваемому понятию на предыдущем шаге, и это влияет на последующие оценки. Для того, чтобы оценки были независимыми, надо оценивать не одно, а 3-5 понятий, которые при этом чередуются. Так работает программа ПР-3, написанная в ABCPascal. Оцениваемые понятия присваиваются переменным $W[1] - W[5]$; на каждом шаге одно из них вместе со шкалой сложности выводится на экран. Эксперт может вводить любое целое значение сложности от 1 до 120 (не обязательно 1 – 2 – 4 – 8 – 16 – 32 – 64). Программа ПР-3 запоминает 3 оценки, полученные каждым понятием, вычисляет средние и выводит результаты на экран.

Программа ПР-3.

Uses crt;

```
Const S1:array[1..8]of string=('ВОЗДУХ, ОБЛАКО, ОГОНЬ','СТОЛ, СВЕТ, СОЛНЦЕ',
'ЧЕЛОВЕК, ВРЕМЯ, РЕКА', 'КАРАНДАШ, ГРИБ, ДОМ', 'ГОРОД, КАРТОШКА, КНИГА',
'РУКА, ВОДА, СОБАКА','ТАРЕЛКА, КОШКА, ГОЛОВА', 'СТУЛ, ПТИЦА, ЛОЖКА');
S2:array[1..8]of string=('ВЕС, ЖИДКОСТЬ, ДОЛЯ', 'ВЫЧЕСТЬ, МЕТАЛЛ, ПРИБОР',
'ГОРИЗОНТАЛЬ, УРАВНЕНИЕ, ИСПАРЕНИЕ', 'СЛОЖИТЬ, ЗАПАД, МАССА', 'ТЕЛО,
ЧИСЛО, ПРУЖИНА', 'ОСЬ, ПРИТЯЖЕНИЕ, ЧАСТИЦА', 'ВЕСЫ, ПЕРЕМЕННАЯ, ГАЗ',
'РАКЕТА, ОПЫТ, ОТРЕЗОК');
S4:array[1..8]of string=('ПЛОТНОСТЬ, ГРАВИТАЦИЯ, КОМПРЕССОР', 'КОМПЬЮТЕР,
СИЛА ТЯЖЕСТИ, ИМПУЛЬС', 'КИСЛОТА, ПРЯМАЯ ПРОПОРЦИОНАЛЬНОСТЬ,
ЮПИТЕР', 'КРИСТАЛЛ, МОЛНИЯ, КАТЕТ, ЭЛ_ЗАРЯД', 'КООРДИНАТА, УМНОЖИТЬ,
ПРЕЛОМЛЕНИЕ СВЕТА', 'ПРИНЦИП, КОНДЕНСАТОР, ЧАСТОТА', 'ЖЕСТКОСТЬ,
ГИПОТЕНУЗА, СКОРОСТЬ', 'КИСЛОТА, ДЕЛИТЬ, КОЛИЧЕСТВО ТЕПЛА');
S8:array[1..8]of string=('АТОМ, МАГНИТНОЕ_ПОЛЕ, УСКОРЕНИЕ', 'МОЛЕКУЛА,
ЭЛЕМЕНТАРНЫЙ ЗАРЯД, СИЛА АМПЕРА',
'ДЛИНА ВОЛНЫ, УГЛОВАЯ СКОРОСТЬ, ЭЛЕКТР_ПОЛЕ', 'ЭЛ_ЕМКОСТЬ,
ОПТИЧ_СИЛА, НАПРЯЖЕННОСТЬ', 'ИЗВЛЕЧЕНИЕ КОРНЯ, МОЛЬ, ЭЛЕКТРОН, СИЛА
```

ТОКА', 'ВОЗВЕДЕНИЕ В КВАДРАТ, УФ-ЛУЧИ, НАПРЯЖЕНИЕ, ГАЛЬВАН_ЭЛЕМЕНТ',
 'ГЕНЕРАТОР ЭЛ_КОЛЕБАН, СПЕКТР, ВОЛЬТМЕТР, УД_ТЕПЛОЕМКОСТЬ',
 'КИНЕТИЧЕСКАЯ ЭНЕРГИЯ, ДИФфуЗИЯ, ВНУТР_СОПРОТИВЛЕНИЕ');

S16:array[1..8]of string=('ВАЛЕНТНОСТЬ, ИЗОТОП, АРКТАНГЕНС,
 РЕЛИКТ_ИЗЛУЧЕНИЕ',

'ИНДУКЦИЯ_МП, КОВАЛЕНТНАЯ СВЯЗЬ, РАДИОАКТ_РАСПАД, АЛЬФА-ЧАСТИЦА,
 ПОЛУПРОВОДНИК', 'МОЛЯРНАЯ МАССА, МОЩНОСТЬ, ЦЕПНАЯ РЕАКЦИЯ, ФОТОН,
 ДИФРАКЦИЯ', 'ПРОТОН, ЭЛ_МАГН_ПОЛЕ, ЭЛ_МАГН_ИНДУКЦИЯ, ЯДЕРНАЯ
 РЕАКЦИЯ',

'ПОЛЯРИЗАЦИЯ, ПОТЕНЦИАЛ, ПРЕДЕЛ, ЯДЕРНЫЙ РЕАКТОР, ДЫРКА,
 ИНТЕНСИВНОСТЬ',

'ЭДС_ИНДУКЦИИ, ЭЛ_МОЩНОСТЬ, НЕЙТРОН, ИНДУКТИВНОСТЬ,
 СИНУС', 'ТАНГЕНС, ДИФРАКЦИОННАЯ РЕШЕТКА, ФУНКЦИЯ,
 ФЛУКТУАЦИЯ', 'ЭНТРОПИЯ, РАБОТА ВЫХОДА, АРКСИНОС, ЕМКОСТЬ,
 МАГНИТНЫЙ_ПОТОК');

S32:array[1..8]of string=('ДНК, АДРОНЫ, ВЕКТ_ПРОИЗВЕДЕНИЕ, РЕНТГЕН_ТРУБКА',
 'СПИН, ВОЛНОВАЯ_ФУНКЦИЯ, ОРБИТАЛЬ, ИМПЕДАНС', 'КОВАРИАНТНОСТЬ,
 ВЕКТОРНОЕ ПРОИЗВЕДЕНИЕ, ТОК НАСЫЩЕНИЯ, КВАРК',

'ИМПЕДАНС, ПРОСТРАНСТВО-ВРЕМЯ, ПАРСЕК, ГАММА-ЛУЧИ', 'ГЛАВНОЕ
 КВАНТОВОЕ ЧИСЛО, ЛОГАРИФМ, ДЕЙТРОН, АНТИГЕЛИЙ', 'ЭНЕРГИЯ СВЯЗИ,
 АНТИЧАСТИЦА, ДИФФЕРЕНЦИАЛ ФУНКЦИИ, РН-ПЕРЕХОД', 'МАСС-СПЕКТРОГРАФ,
 ИНТЕРФЕРЕНЦИЯ, МЕЗОН, МОМЕНТ ИНЕРЦИИ, МГД-ГЕНЕРАТОР', 'ГЛЮОН
 ОРБИТАЛЬ, ГРАВИТОН, СИНГУЛЯРНОСТЬ, ФОТОЭФФЕКТ');

S64:array[1..5]of string=('ЭФФЕКТ МЕССБАУЭРА, ЯДЕРНЫЙ МАГНИТНЫЙ РЕЗОНАНС,
 ВАРИАЦИЯ', 'ТЕНЗОР, ПРОИЗВОДНАЯ, СПЕКТР_ПЛОТНОСТЬ ИНТЕНСИВНОСТИ
 ИЗЛУЧЕНИЯ', 'УРАВНЕНИЕ ШРЕДИНГЕРА, ФУНКЦИОНАЛ, ГРАДИЕНТ', 'ТЕНЗОР,
 ФУНКЦИОНАЛ, ВАРИАЦИЯ', 'ФУНКЦИОНАЛ, ГРАДИЕНТ, ЯДЕРНЫЙ МАГНИТНЫЙ
 РЕЗОНАНС');

S128:array[1..3]of string=('НАБЛА, ДИВЕРГЕНЦИЯ, РОТОР', 'ЦИРКУЛЯЦИЯ, РОТОР,
 ДИВЕРГЕНЦИЯ', 'НАБЛА, ДИВЕРГЕНЦИЯ, ДВОЙНОЙ ИНТЕГРАЛ');

var i,j,k: integer; x:array[1..5,1..8] of integer; SL:array[1..5] of single;

W: array[1..5] of string; c1,c2,c4,c8, c16,c32,c64:array[1..5,1..8] of integer; Label
 m1,m2,m4,m8,m16,m32,m64,m128;

BEGIN W[1]:='АННИГИЛЯЦИЯ'; W[2]:='ИОН'; W[3]:='ФОКУС'; W[4]:='ДИСПЕРСИЯ';
 W[5]:='МОМЕНТ СИЛЫ'; Randomize;

for j:=1 to 3 do for k:=1 to 5 do

begin clrscr; writeln;

writeln('ОЦЕНИВАЕМОЕ ПОНЯТИЕ ', W[k]); writeln;

m1: i:=round(random(74)/10)+1; if c1[k,i]=1 then goto m1; c1[k,i]:=1; writeln('S=1: ',S1[i]);
 writeln;

m2: i:=round(random(74)/10)+1; if c2[k,i]=1 then goto m2; c2[k,i]:=1; writeln('S=2: ',S2[i]);
 writeln;

m4: i:=round(random(74)/10)+1; if c4[k,i]=1 then goto m4; c4[k,i]:=1; writeln('S=4: ',S4[i]);
 writeln;

m8: i:=round(random(74)/10)+1; if c8[k,i]=1 then goto m8; c8[k,i]:=1; writeln('S=8: ',S8[i]);
 writeln;

m16: i:=round(random(74)/10)+1; if c16[k,i]=1 then goto m16; c16[k,i]:=1; writeln('S=16:
 ',S16[i]); writeln;

m32: i:=round(random(74)/10)+1; if c32[k,i]=1 then goto m32; c32[k,i]:=1; writeln('S=32:
 ',S32[i]); writeln;

m64: i:=round(random(44)/10)+1; if c64[k,i]=1 then goto m64; c64[k,i]:=1; writeln('S=64:
 ',S64[i]); writeln;


```

m128: i:=round(random(24)/10)+1; writeln('S > 100 ',S128[i]); writeln;
write('СЛОЖНОСТЬ '); readln(x[k,j]); end; writeln;
for k:=1 to 5 do begin SL[k]:=((x[k,1]+x[k,2]+x[k,3])/3);
writeln(W[k], ' = ',SL[k]:2:2); end; readkey;
END.

```

Для оценки семантической сложности абстрактных понятий, обозначающих не измеряемые школьником физические величины, а также объекты (явления), которые школьник не может пронаблюдать в повседневной жизни или на уроке, может быть использован **метод вычисления сложностей**. При этом учитывается, что между понятиями существуют логические связи, а их система имеет иерархическую структуру, в которой из абстракций k -ого уровня формируются определения абстракций $(k+1)$ -ого уровня сложности. Поэтому появляются цепочки взаимосвязанных понятий, например: сила \Rightarrow работа \Rightarrow мощность \Rightarrow интенсивность, магнитный поток \Rightarrow электромагнитная индукция \Rightarrow ЭДС индукции и т.д.

В этом случае для нахождения сложности образуемого понятия к сложности образующего понятия прибавляется некоторое целое m из интервала $[1; 5]$, например: $s(\text{ускорение}) = s(\text{скорость}) + m$. При этом число m показывает, сколько простых слов с $s=1$ нужно добавить к понятию «скорость», чтобы дать определение ускорению. Переход на следующий уровень соответствует увеличению сложности на величину m , приблизительно равную 2 – 5 (например: «скорость – путь делить на время»). Это объясняется тем, что система научных понятий долго эволюционировала в соответствии с принципом экономии мышления, из которого следует, что появление нового понятия оправдывает себя в том случае, когда его определение сложнее самого сложного из образующих понятий на 2 – 5 слов. Например, $s(\text{ускорение}) \approx s(\text{скорость}) + 3$.

При оценке сложности текста получается список из N научных понятий, сложности которых s_i ($i = 1, 2, \dots, N$) необходимо определить. Это можно сделать так: 1) для каждого понятия P сформулировать самое простое определение и выявить образующие его понятия; 2) составить уравнения, связывающие сложность P со сложностями одного, двух или трех самых сложных образующих его понятий; 3) создать компьютерную программу, с помощью которой вычислить приблизительные значения сложностей s_i различных понятий; 4) проанализировать и подкорректировать полученные результаты так, чтобы они не противоречили здравому смыслу. При этом следует учитывать встречаемость обозначаемого объекта в повседневной жизни, его принадлежность к микро-, макро- и мегамиру, степень абстрактности понятия, а так же степень вхождения термина в тезаурус учащихся начальной, базовой и старшей школы.

Этот подход позволяет избежать логических противоречий возникающих при оценке большого количества (более 100) научных понятий из различных разделов физики, когда сложность понятия k -ого уровня, ошибочно оценивается выше сложности понятия, находящегося на $k+1$ или $k+2$ уровне. Для оценки сложности большой совокупности понятий нами использовались компьютерная программа на Pascal, содержащая формулы типа: $uskorenie:=skorost+3$, $moshnost:=rabota+3$, $diod:= 1,3* \text{fotoresistor}$.

В случае, когда количество понятий (с оценками сложности) достигает нескольких сотен, работать с получившимся списком, хранящемся в файле slovar.txt, становится непросто. Автором создана простая программа, которая автоматически просматривает этот файл, находит нужные термины, выводит их и значения s_i на экран, а также печатает в текстовый файл vihod.txt. Для определения сложности новых понятий, не входящих в уже составленный список (файл slovar.txt), эксперт составляет запрос, включая в него термины, максимально близкие по сложности к оцениваемому понятию P , и запускает программу. Например, при оценивании понятия «ион» осуществляется поиск терминов «атом», «электрон», «заряд», «ионизация». В результате получается список терминов с указанием их сложностей, что позволяет эксперту более точно определить сложность понятия «ион», не вступая в

противоречия с оценками других понятий. После этого новые термины и их сложности также добавляют в файл slovar.txt.

IV. ВЫВОДЫ

Рассмотренные компьютерные программы позволяют автоматизировать и рандомизировать процессы оценки сложности понятий (или иных объектов) и сопоставления оцениваемых понятий со шкалой сложности. Обсуждаются три компьютерных программы, написанные на языке Pascal: 1) программа, которая в случайном порядке предьявляет сравниваемые объекты, воспринимает ответы эксперта и записывает результаты в файл; 2) программа, обрабатывающая получающийся файл; 3) программа, выводящая на экран оцениваемый объект (научное понятие) и шкалу сложности, воспринимающая оценки эксперта и вычисляющая среднее. При парном сравнении эксперт может сделать перерыв, сохранить результаты, а затем через некоторое время продолжить работу. Предложенные методы позволяют получить более объективные оценки семантической сложности научных понятий и могут быть использованы для оценивания других качеств иных объектов.

БИБЛИОГРАФИЯ

- [1] Майер Р.В. Дидактическая сложность учебных текстов и ее оценка: монография. – Глазов: ГГПИ, 2020. 149 с. ISBN 978-5-93008-305-7
- [2] Mayer R.V. Results of estimating the didactic complexity of section of the general physics course // Proceedings of ICERI2021 Conference 8th-9th November 2021. pp. 179-187.
- [3] Алябышева Ю.А., Антонов А.Ю., Веряев А.А. Цифровизация тезаурусного подхода в образовании // Информатика и образование. 2020. № 1. С. 51–58.
- [4] Гучук В.В. Технология объективизации экспертной кластеризации слабоформализуемых объектов // Управление, обработка информации и принятие решений. 2014. Т. 18, № 5 (66). С. 153–158.
- [5] Лукашевич Н. В. Тезаурусы в задачах информационного поиска. – М., 2010. 396 с.
- [6] Луков Вал. А., Луков Вл. А. Методология тезаурусного подхода: стратегия понимания // Знание. Понимание. Умение. 2014. № 1. С. 18–35.
- [7] Луков Вал.А., Луков Вл.А. Тезаурусы: Субъектная организация гуманитарного знания: науч. монография. – М.: Изд-во Нац. ин-та бизнеса, 2008. 784 с.
- [8] Тулохонова И.С. Моделирование системы обучения на основе тезаурусного подхода // Интернет-журнал «Науковедение». Том 9, №4. 2017.
- [9] Дэвид Г. Метод парных сравнений. – М.: Статистика, 1978. 144 с.
- [10] Дюк В. А. Компьютерная психодиагностика. – СПб.: Братство, 1994. 364 с.
- [11] Корноушенко Е.К. Алгоритм классификации путем парного сравнения признаков // Автоматика и телемеханика, № 11, 2017. С. 151-166.
- [12] Орлов А.И. Организационно-экономическое моделирование: учебник: в 3 ч. Ч. 2: Экспертные оценки. – М.: Изд-во МГТУ им. Н.Э. Баумана. 2011. 486 с.
- [13] Садовников, И.В. Квалиметрия: учеб. пособие. – Чита: ЧитГУ, 2009. 150 с.
- [14] Черепанов В.С. Экспертные оценки в педагогических исследованиях. – М.: Педагогика, 1989. 150 с.
- [15] Davis B., Sumara D. Complexity and Education: Inquiries Into Learning, Teaching, and Research. – Mahwah, New Jersey, London, 2006. 201 p.
- [16] Попов В.Б. Turbo Pascal для школьников: учебно-методическое пособие. – Москва: РИОР: ИНФРА-М, 2012. 372 с.