РОССИЙСКАЯ АКАДЕМИЯ НАУК

ВСЕРОССИЙСКИЙ ИНСТИТУТ НАУЧНОЙ И ТЕХНИЧЕСКОЙ ИНФОРМАЦИИ
(ВИНИТИ)

НАУЧНО • ТЕХНИЧЕСКАЯ
ИНФОРМАЦИЯ

Серия 2. ИНФОРМАЦИОННЫЕ ПРОЦЕССЫ И СИСТЕМЫ

ЕЖЕМЕСЯЧНЫЙ НАУЧНО-ТЕХНИЧЕСКИЙ СБОРНИК

Издается с 1961 г. № 11 Москва 2013

ИНФОРМАЦИОННЫЙ АНАЛИЗ

УДК [001.102:002] : 004

В.В. Нешитой

Методы вычисления границ ядра и зон рассеяния публикаций

Рассматриваются три метода вычисления границ ядра и зон рассеяния публикаций, а точнее – координат трех характерных точек A, C, B на кривой распределения: аналитический, графический и метод наименьших квадратов. Первые два метода могут быть применены к любому статистическому ранговому распределению в случае однородной выборки. Такие распределения описываются второй системой непрерывных распределений, которая является универсальным законом рассеяния публикаций. В частном случае, если справедлив закон Вейбулла, используется метод наименьших квадратов.

Приводятся практические примеры, подтверждающие высокую точность аппроксимации статистических ранговых распределений второй системой непрерывных распределений автора.

Ключевые слова: *три характерные точки кривых распределения, границы ядра и зон рассеяния, аналитический метод, графический метод, метод наименьших квадратов, вторая система непрерывных распределений, форма представления ранговых распределений, универсальный закон рассеяния публикаций*

ВВЕДЕНИЕ

В научных исследованиях широко используются статистические ранговые распределения различных объектов – журналов, книг и других изданий, представленных в списке по убыванию частоты встречаемости; лексические единицы частотного словаря; ключевые слова, упорядоченные по убыванию частоты их использования при индексировании документов, и многие другие объекты. Порядковый номер журнала, книги, слова в этом списке называется рангом. На базе статистических ранговых распределений

можно решать множество практических задач, в том числе вычислять границы ядра и зон рассеяния публикаций. К сожалению, анализ научных работ с использованием ранговых распределений показывает, что некоторые авторы не могут вычислить теоретический закон и, следовательно, границы ядра и зон рассеяния. Это можно осуществить лишь при использовании теории обобщенных распределений.

В 1948г. С. Бредфорд окончательно сформулировал закон рассеяния журнальных публикаций, который заключается в следующем: «Если научные журналы расположить в порядке убывания числа помещенных в них статей по какому-либо заданному предмету, то в полученном списке можно выделить ядро журналов, посвященных непосредственно этому предмету, и несколько групп или зон, каждая из которых содержит столько же статей, что и ядро. Тогда числа журналов в ядре и последующих зонах будут относиться как $1 : n : n^2$ » [1, с. 93, 94].

Однако из этой формулировки неясно, как по статистическому ранговому распределению вычислить границы ядра и зон рассеяния, поскольку нет никаких формул для их вычисления. Неизвестно также, какое теоретическое ранговое распределение принято за основу закона рассеяния, какие точки на графике рангового распределения приняты в качестве таких границ, сколько может быть зон рассеяния, как вычисляется величина n. В формулировке С.Бредфорда лишь указывается, что можно выделить ядро журналов и несколько зон, при этом предполагается, что число статей в каждой зоне такое же, как и в ядре. Однако это предположение не согласуется с фактическими данными и не обосновывается теоретически.

МЕТОД ПОДБОРА

Приняв формулировку С.Бредфорда за основу, исследователи вынуждены по своему усмотрению устанавливать объем ядра журналов, а количество зон рассеяния и их размер получаются на основании фактических данных из условия, что каждая зона содержит столько же статей, что и ядро. При таком подходе количество зон рассеяния для одного и того же статистического рангового распределения у разных исследователей может сильно колебаться.

Главное в законе С.Бредфорда из того, что не подлежит сомнению – это расположение журналов по убыванию количества опубликованных в них статей по заданному предмету, т.е. ранжирование. Ранговые распределения с наибольшей полнотой отражают суть рассеяния публикаций. Но многие исследователи обходят этот главный вопрос стороной. Вместо изучения свойств статистических ранговых распределений и теоретического закона распределения с такими же свойствами они пытаются выделить ядро и зоны рассеяния, используя только формулировку С. Бредфорда 1948 года!

К сожалению, в литературных источниках не приводятся статистические ранговые распределения наименований книг, упорядоченных по убыванию частоты встречаемости, недостаточно активно проводится работа по составлению частотных списков журналов, упорядоченных по различным признакам. Но для изучения свойств ранговых распределений можно обратиться к математической лингвистике. В этой области знания накоплено большое количество частотных словарей, в которых лексические единицы расположены в порядке убывания (точнее, невозрастания) частоты их употребления в текстах. Выработана удобная для анализа таблица частот слов, в которой приводятся необходимые сведения о частотном словаре: ранг или интервал рангов слов, их абсолютная частота, количество слов с данной частотой, накопленная абсолютная частота, относительная частота, накопленная относительная частота слов. Такая таблица приводится, как правило, в конце частотного словаря. Она позволяет представить в компактной форме частотную структуру словаря любого объема.

Для аппроксимации статистических распределений, в том числе ранговых, созданы универсальные математические модели – обобщенные распределения, разработаны методы вычисления законов распределения по статистическим данным и оценок их параметров. Предложена новая форма представления ранговых распределений, а на ее основе – критерии однородности и достаточности объема выборки. Имеются компьютерные программы для построения частотных словарей, в том числе свободно распространяемые.

Это значит, что многие **задачи по обработке статистических ранговых распределений в лингвистике, информатике, библиотечном деле давно решены**. Но, к сожалению, накопленные в этих областях знания результаты научных исследований практически не используются. Причина заключается в том, что исследователю необходимо затратить немалый труд и время на изучение и практическое применение этих результатов в своей научной работе. Значительно проще традиционно выдвигать гипотезы и проверять их по различным критериям согласия, но таким путем нельзя получить новое знание.

Упомянутый выше метод подбора размеров ядра и зон рассеяния – это тот же метод выдвижения гипотез, но без проверки результатов по критериям согласия. Такой метод в научных исследованиях неприемлем.

Аналитический метод

Итак, задано ранговое распределение журналов. Если найти теоретическое распределение, которое достаточно точно описывает статистическое ранговое распределение, то оно позволит дать математически точную формулировку закона рассеяния в смысле С.Бредфорда, т.е. этот закон должен следовать из рангового распределения как частный случай. Наиболее же общим, **универсальным законом рассеяния может быть только теоретическое ранговое распределение**, поскольку закон распределения является наиболее полной характеристикой любой случайной величины.

Многие авторы при определении границ ядра и зон рассеяния пытаются обрабатывать статистические ранговые распределения без предварительного решения общей задачи – нахождения закона распределения. Но в этом случае им приходится для аппроксимации каждого статистического распределе-

ния подбирать различные формулы вместо использования одного теоретического закона, но с разными значениями параметров. Кроме того, для подбора теоретической кривой используется неудачная форма представления статистических ранговых распределений в системе координат «логарифм ранга – логарифм частоты», которая несет слишком мало информации о ранговом распределении и более того – не имеет вероятностного смысла.

Таким образом, проблема рассеяния публикаций состоит в решении общей задачи – нахождении такой универсальной формулы или системы формул, которые способны с высокой точностью аппроксимировать все многообразие статистических ранговых распределений. Поскольку «…рассеяние научной информации является краеугольным камнем всей научно-информационной деятельности, а изучение этого свойства научной информации – важнейшей проблемой информатики» [1, с. 93], то эту проблему необходимо разрешать весьма серьезными средствами. Такими средствами являются обобщенные распределения, предложенные автором [2–6].

Рассмотрим первую и вторую системы непрерывных распределений, каждая из которых задана тремя обобщенными плотностями. Запишем первые плотности этих систем:

$$p(x) = Ne^{k\beta x}\left(1-\alpha u e^{\beta x}\right)^{\frac{1}{u}-1}, \qquad (1)$$

$$\mathrm{p(t)} = \mathrm{Nt}^{\mathrm{k}\beta-1}\left(1-\alpha u t^{\beta}\right)^{\frac{1}{u}-1}. \qquad (2)$$

Здесь α, β, k, u – параметры распределения. Они вычисляются по статистическому распределению. N – нормирующий множитель, который выражается через параметры распределения при условии, что площадь под кривой распределения равна единице.

Первая плотность обладает тем замечательным свойством, что при значениях параметра формы $u \le 1/2$ она **имеет моду** x_C, т.е. такое значение случайной величины X, при котором плотность $p(x)$ максимальна, и **две точки перегиба –** x_A, x_B, **расположенные на равных расстояниях по обе стороны от моды**. Это такие точки на графике плотности распределения, которые отделяют выпуклую часть кривой от вогнутой или вогнутую часть от выпуклой. При $1/2<u<1$ имеются лишь две характерные точки – x_A и x_C (для распределений с левосторонней асимметрией). При $u \ge 1$ характерных точек не существует. Распределение (1) может быть задано на всей числовой оси, т.е. $-\infty < x < \infty$.

Распределение (2) задано на положительной полуоси $t > 0$. График плотности $p(t)$ может принимать различные формы. При определенных значениях параметров формы k, β, u график плотности $p(t)$ может иметь вид убывающей кривой распределения, а это значит, что плотность $p(t)$ описывает не только одновершинные статистические распределения, но и ранговые (убывающие).

Для вычисления закона распределения случайной величины по статистическому распределению, в том числе ранговому, нами разработаны два метода – универсальный метод моментов и общий устойчивый метод [4–6]. С целью упрощения расчетов и извлечения новой информации из рангового распределения оно приводится к форме одновершинного распределения [7]. Это достигается путем приведения формы второй плотности к форме первой. Для этого умножим левую и правую части плотности $p(t)$ на t, а величину t^{β} представим в виде $e^{\beta \ln t}$. Тогда плотность (2) примет вид

$$tp(t) = Ne^{k\beta \ln t}\left(1-\alpha u e^{\beta \ln t}\right)^{\frac{1}{u}-1}. \qquad (3)$$

Сравнивая (1) и (3), можем записать:

$$tp(t) = p(x),\ lnt = x. \qquad (4)$$

Таким образом, ранговое распределение (2), приведенное к форме $tp(t) = f(lnt)$, с учетом равенств (4) представляет собой распределение (1) и обладает всеми его свойствами: оно имеет моду $\ln \mathrm{t_C}$ и две точки перегиба $\ln t_A$ и $\ln t_B$. В этом заключается новая информация о ранговом распределении. **Примем абсциссы точек *A, C, B* обобщенных распределений в качестве границ ядра и зон рассеяния** различных объектов.

Поскольку точки $\ln t_A$ и $\ln t_B$ расположены на равных расстояниях от моды $\ln t_C$, то можем записать равенство $\ln \mathrm{t_C} - \ln t_A = \ln t_B - \ln t_C$, откуда имеем $\ln(\mathrm{t_C} / t_A) = \ln(t_B / t_C)$, или

$$\frac{t_C}{t_A} = \frac{t_B}{t_C} = n. \qquad (5)$$

Формула (5) может служить уточненной формулировкой закона рассеяния публикаций в смысле С. Бредфорда, хотя она представляет собой новую формулировку закона рассеяния публикаций.

Из (5) можно записать еще две формулы:

$$\mathrm{t_A} : t_C : t_B = t_A(1 : n : n^2), \qquad (6)$$

$$t_A : t_I : t_{II} = t_A\left[1 : (n-1) : n(n-1)\right]. \qquad (7)$$

Здесь t_A, t_C, t_B – количество журналов от начала частотного списка до точек *A, C, B;* $\mathrm{t_1}, t_{II}$ – количество журналов в первой и второй зонах рассеяния. При этом $\mathrm{t_1} = t_C - t_A$, $t_{II} = t_B - t_C$. С учетом этих равенств получена формула (7) из формулы (6).

Все три формулировки, заданные формулами (5) – (7), отличаются от формулировки С. Бредфорда и уточняют ее. Однако **приведенные формулы** (как и закон Бредфорда) **не являются законом рассеяния**, так как не дают полной информации о нем. Они устанавливают лишь общие соотношения между абсциссами трех характерных точек – моды и двух точек перегиба. Но для вычисления координат этих точек требуется знание закона рангового распределения. Только при известных оценках параметров этого распределения можно получить нужную информацию – вычислить границы ядра журналов и зон рассеяния, долю статей в ядре и зонах рассеяния, а также долю статей для любого числа журналов от начала частотного списка. Она выражается через функцию распределения. В формулах же (5) – (7) функция распределения не задействована.

Таким образом, наиболее полную информацию дает закон рангового распределения. Это значит, что обобщенная плотность (2), а точнее, **вторая система непрерывных распределений, заданная тремя обобщенными плотностями, является универсальным законом рассеяния публикаций**. Здесь уместно отметить, что первая система непрерывных распределений, включающая плотность (1), является универсальным законом старения публикаций [2].

Некоторые исследователи утверждают, что закон С. Бредфорда – это другая форма представления закона Дж. Ципфа – $p_r = k/r$. Эта формула была предложена им для описания ранговых распределений слов частотного словаря. Здесь p_r – относительная частота слова с рангом r, k – параметр. Для закона Дж. Ципфа произведение относительной частоты слова на ранг равно постоянной величине k, что на графике в системе координат $rp_r = f(\ln r)$ изображается горизонтальной прямой, на которой нет никаких характерных точек. Это убедительно свидетельствует о том, что законом Дж. Ципфа нельзя аппроксимировать статистические ранговые распределения, поскольку в данной системе координат в случае однородной выборки они имеют вид одновершинной кривой с двумя точками перегиба. Такие кривые с высокой точностью описываются второй системой непрерывных распределений. Здесь следует отметить, что закон Дж.Ципфа, а в более общей формулировке, с двумя дополнительными параметрами – закон Эсту-Ципфа-Мандельброта, – является частным случаем обобщенного распределения (2) при u=1, $\beta < 0$. Но при этом условии кривая распределения $rp_r = f(\ln r)$ не имеет характерных точек. Поэтому никоим образом нельзя из них получить закон рассеяния, нельзя их использовать для аппроксимации статистических ранговых распределений, тем более в серьезных научных исследованиях. Наилучшее теоретическое распределение для аппроксимации статистического рангового распределения должно быть вычислено по второй системе непрерывных распределений [6].

Рассмотрим далее закон рассеяния С.Бредфорда (при r=t, $t_Я = t_A$)

$$t_Я : t_1 : t_{II} = t_Я(1:n:n^2). \qquad (8)$$

Сравнивая эту формулу с формулами (6) и (7), видим, что закон рассеяния С.Бредфорда является неправильной комбинацией двух правильных формул: левая часть соответствует формуле (7), а правая – формуле (6).

Формулу (8) можно представить в другом виде – с учетом трех характерных точек, о которых С.Бредфорд в своей формулировке закона рассеяния не упоминал. Из (8) имеем

$$t_Я : (t_C - t_Я) : (t_B - t_C) = t_Я(1:n:n^2).$$

Вынесем в левой части этого равенства величину $t_Я$ за скобки

$$t_Я[1:(t_C - t_Я)/t_Я : (t_B - t_C)/t_Я] = t_Я(1:n:n^2).$$

Теперь можем записать два равенства: $(t_C - t_Я)/t_Я = n$, $(t_B - t_C)/t_Я = n^2$, откуда находим, что отношение $t_C : t_Я = n+1$, а отношение $t_B : t_Я = n^2 + n + 1 = (n+1)^2 - n$. В результате имеем формулу

$$t_Я : t_C : t_B = t_Я[1:(n+1):(n^2+n+1)]. \qquad (9)$$

В нашей точной формуле (6) первое отношение равно n, второе n^2. Поскольку эти отношения в формуле (9) не соблюдаются, то по закону С.Бредфорда точки перегиба на кривой рангового распределения $tp(t) = f(\ln t)$ должны располагаться **на разных** расстояниях от моды $\ln t_C$, а это противоречит свойствам статистических и теоретических ранговых распределений. Другое противоречие закона С.Бредфорда состоит в утверждении, что каждая зона рассеяния содержит такое же число статей, как и в ядре. Естественно, что этим двум требованиям не может удовлетворить ни одно теоретическое распределение. Поэтому все попытки усовершенствовать закон рассеяния С.Бредфорда при сохранении присущих ему противоречий оказались неудачными. И это закономерно, потому что предпринимались попытки решить частную задачу без предварительного решения общей задачи – нахождения такого теоретического рангового распределения, которое позволило бы с высокой точностью аппроксимировать широкое разнообразие статистических ранговых распределений.

Оценим погрешность формулы (9). Пусть величина n=5. Вычислим отношения $t_C : t_Я$ и $t_B : t_Я$. В первом случае оно равно n+1=6 , а во втором n^2+n+1=31. По точной формуле (6) имеем соответственно 5 и 25. Погрешность вычисления абсциссы точки С составила 20%, а точки В – 24%. Размер ядра в обоих случаях принят одинаковым, так как по закону С.Бредфорда его вычислить нельзя. Следует отметить, что с ростом величины n эта погрешность уменьшается.

Из полученных результатов следует, что закон рассеяния С.Бредфорда, представленный в виде формул (8) и (9), относительно близок к точным формулам (6) и (7), хотя при его формулировке им не были использованы теоретические ранговые распределения. Утверждение же С.Бредфорда о том, что число статей в ядре и зонах рассеяния одинаково, не соответствует действительности и вводит в заблуждение некоторых исследователей.

Однако огромная заслуга С.Бредфорда заключается в том, что он первым обратил внимание на явление рассеяния публикаций и побудил многих исследователей к углубленному изучению этого явления.

Вопрос этот оказался очень сложным. Универсальный закон рассеяния был найден нами лишь после разработки теории обобщенных распределений. Выше отмечалось, что таким законом является вторая система непрерывных распределений. Из обобщенной плотности (2) выводятся формулы для вычисления границ ядра и зон рассеяния.

Мода t_C находится из условия $dtp(t)/d\ln t = 0$ и в общем случае для распределений I-V типов равна [4]

$$t_C = \left(\frac{k}{\alpha(1+ku-u)} \right)^{1/\beta}. \qquad (10)$$

Величина n задается формулой

$$n=\left[1+\frac{1-u+\sqrt{\left[4k\left(1+ku-u\right)+\left(1-u\right)\right]\left(1-u\right)}}{2k\left(1+ku-u\right)}\right]^{1/\beta}. \quad (11)$$

Абсциссы точек перегиба вычисляются по формулам:

$$\mathrm{t_A}=t_C/n;\ t_B=t_C\cdot n. \quad (12)$$

Доли статей в каждой зоне и для любого другого интервала рангов вычисляются с помощью функции распределения или по статистическому ранговому распределению.

Из формул (11), (12) следует, что при заданных значениях параметров формы k, u с уменьшением параметра β величина $\mathrm{n}=\mathrm{t_C}/t_A=t_B/t_C$ растет. Это значит, что кривая распределения $tp(t)=f(\ln t)$ становится более широкой и пологой.

Методы вычисления аппроксимирующих четырехпараметрических распределений изложены в ряде работ [2–6], но эти методы рассчитаны на подготовленного исследователя.

Графический метод

Для нахождения границ ядра и зон рассеяния по статистическому ранговому распределению без предварительного вычисления теоретического закона можно предложить простой графический метод, который следует из анализа свойств обобщенных распределений (1) и (2). Суть этого метода покажем на примере.

Рассмотрим для примера закон Вейбулла, функция и плотность распределения которого заданы формулами

$$\mathrm{F(t)}=1\text{-}e^{-\alpha t^{\beta}}, \quad (13)$$

$$\mathrm{p(t)}=\alpha\beta\mathrm{t}^{\beta\text{-}1}e^{-\alpha t^{\beta}}. \quad (14)$$

При значениях параметра $\beta \le 1$ плотность (14) может с высокой точностью описывать некоторые статистические ранговые распределения.

Чтобы получить из рангового распределения полезную информацию, преобразуем плотность (14) к форме *tp(t) = f(lnt)* [7]:

$$\mathrm{tp(t)}=\alpha\beta\mathrm{t}^{\beta}e^{-\alpha t^{\beta}}=\alpha\beta e^{\beta\ln t}e^{-\alpha e^{\beta\ln t}} \quad (15)$$

С учетом равенств *tp(t) = p(x); lnt = x* плотность (15) примет вид

$$\mathrm{p(x)}=\alpha\beta\mathrm{e}^{\beta\mathrm{x}}e^{-\alpha e^{\beta x}}. \quad (16)$$

Полученная формула представляет собой частный случай плотности (1) при *u→0, k=1*. Плотность (16) дает новую информацию о ранговом распределении. График этой плотности, т.е. кривая распределения содержит три характерные точки – моду *C* и две точки перегиба *A* и *B*, расположенные на равных расстояниях по обе стороны от моды. Найдем абсциссы этих точек. Продифференцируем плотность (16) по *x* и приравняем первую производную нулю. Из полученного уравнения найдем выражение для моды

$$\mathrm{x_C}=\frac{1}{\beta}\ln\frac{1}{\alpha}. \quad (17)$$

В точках перегиба вторая производная равна нулю. Из этого условия для плотности (16) найдем

$$\mathrm{x_A}=x_C-\frac{1}{\beta}\ln\frac{3+\sqrt{5}}{2},$$

$$\mathrm{x_B}=x_C+\frac{1}{\beta}\ln\frac{3+\sqrt{5}}{2}.$$

Введем обозначение

$$n=\left(\frac{3+\sqrt{5}}{2}\right)^{\frac{1}{\beta}}. \quad (18)$$

С учетом (18) последние две формулы примут более простой вид

$$\mathrm{x_A}=x_C-\ln n; \quad (19)$$

$$\mathrm{x_B}=x_C+\ln n. \quad (20)$$

Переходя к распределению Вейбулла, из формул (19), (20) с учетом равенства x=lnt найдем

$$\mathrm{lnt_C}-\ln t_A=\ln t_B-\ln t_C=\ln n,$$

откуда $\ln(t_C/t_A)=\ln(t_B/t_C)=\ln n$, или

$$\frac{t_C}{t_A}=\frac{t_B}{t_C}=n. \quad (21)$$

Здесь

$$\mathrm{t_C}=\left(\frac{1}{\alpha}\right)^{\frac{1}{\beta}};\quad t_A=\frac{t_C}{n};\quad t_B=t_C\cdot n. \quad (22)$$

Формула (21), полученная на базе закона Вейбулла, совпадает с аналогичной формулой (5), полученной на базе четырехпараметрического распределения (2). Она остается также справедливой для любого частного случая распределения (2) при значениях параметра формы *u ≤ 1/2*. Закон Вейбулла является частным случаем распределения (2) при *u→0, k=1*. В некоторых случаях он с высокой точностью описывает статистические ранговые распределения. Но универсальным законом остается обобщенная плотность (2), которая позволяет вычислять и координаты характерных точек, и функцию распределения [4, с. 155–160] практически для любого статистического рангового распределения. В некоторых случаях статистические ранговые распределения с высокой точностью описываются дополнительными плотностями второй системы непрерывных распределений.

Предположим для определенности, что ранговое распределение задано законом Вейбулла с параметрами *α=0,1; β=0,5*. Приведем его к плотности *p(x),* которая представлена формулой (16). Тогда для этой плотности по формулам (17)–(20) найдем: n=6,8541; $\mathrm{x_C}=4,6052;\ \mathrm{x_A}=2,6804;\ x_B=6,53$.

Рассчитаем по формуле (16) значения плотности *p(x)* с интервалом *Δx=0,5* и сведем результаты в таблицу 1.

Построим график плотности *p(x),* т.е. кривую распределения (рис. 1а).

На кривой распределения абсциссу точки *C* легко найти графически путем проведения горизонтальной касательной к кривой (см. рис. 1а).

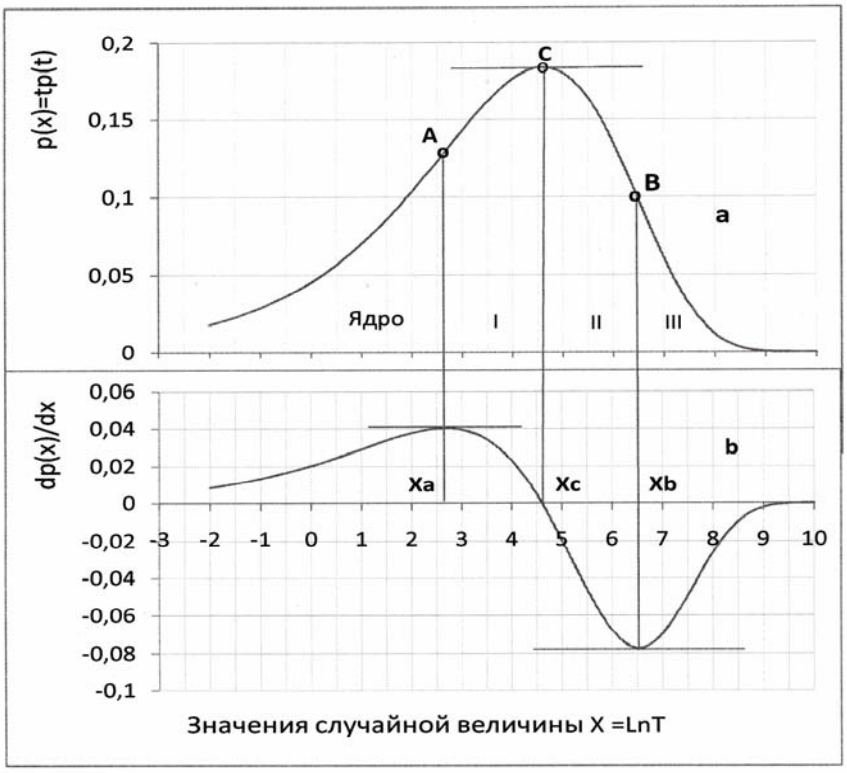

Рис. 1. Графики плотности распределения (a) и ее первой производной (b)

Таблица 1

Значения плотности и тангенса угла наклона касательной к кривой в серединах интервалов

x	p(x)	dp(x)/dx	x	p(x)	dp(x)/dx
-2	0,01773	0,008539	4	0,176464	0,023037
-1,5	0,022529	0,010732	4,5	0,18369	0,004705
-1	0,028542	0,013405	5	0,180147	-0,01966
-0,5	0,036022	0,016609	5,5	0,163655	-0,04617
0	0,045242	0,020359	6	0,134756	-0,06795
0,5	0,056465	0,024607	6,5	0,097806	-0,07722
1	0,069906	0,02919	7	0,060369	-0,06977
1,5	0,085655	0,033761	7,5	0,030263	-0,04921
2	0,103565	0,037706	8	0,011614	-0,0259
2,5	0,123099	0,040067	8,5	0,003163	-0,00951
3	0,143144	0,039496	9	0,000554	-0,00222
3,5	0,161833	0,034352	9,5	5,52E-05	-0,00029

Для нахождения абсцисс точек перегиба воспользуемся тем свойством кривой распределения, что в точках *A* и *B* первая производная принимает экстремальные значения: в точке *A* она имеет максимум, а в точке *B* – минимум. Вычислим тангенс угла наклона отрезков кривой к горизонтальной оси на всех интервалах как отношение разности между значениями плотности *p(x)* на границах интервала к ширине интервала *Δx*. Другими словами, найдем приближенные значения первой производной в серединах интервалов (в табл. 1 приведены расчетные значения производной dp(x)/dx) и построим график (см. рис. 1b).

Первая производная *dp(x)/dx* имеет максимум в точке x_A и минимум в точке x_B. Эти точки легко определить путем проведения горизонтальных касательных к кривой на рис. 1b.

Из построенного графика можно приближенно найти абсциссы трех характерных точек: $x_A = 2,7$; $x_C = 4,6$; $x_B = 6,5$.

Переходя к ранговому распределению, находим: $t_A = \exp(x_A) \approx 15$; $t_C = \exp(x_C) \approx 99$; $t_B = \exp(x_B) \approx 665$. Точные значения для распределения Вейбулла равны: $t_A = 14{,}59$; $t_C = 100$; $t_B = 685$.

Таким простым методом приближенно могут быть найдены абсциссы трех характерных точек любого статистического рангового распределения без предварительного вычисления теоретического закона распределения, но с использованием его свойств.

Значения функции распределения *F(t)* при любом заданном значении ранга *t,* в том числе в трех характерных точках, могут быть вычислены по статистическому ранговому распределению.

Используя графический метод, разные исследователи получат близкие результаты по определению границ ядра и зон рассеяния для одного и того же статистического рангового распределения.

Здесь необходимо отметить, что представленная на рис. 1a кривая теоретического распределения плавно возрастает до максимального значения и затем плавно убывает. Поэтому вычисление тангенса угла наклона отрезков кривой на всех интервалах не вызывает затруднений. Аналогичная кривая статистического распределения *tp(t) = f(lnt)* имеет многочисленные всплески и впадины, что затрудняет построение рис. 1b. Поэтому предварительно ее необходимо сгладить, например, с помощью лекала.

Отметим, что на рис. 1a горизонтальная касательная к кривой распределения *tp(t) = f(lnt)* в точке C представляет собой закон Дж.Ципфа. Отсюда следует, что этим законом невозможно описать никакое ранговое распределение.

Метод наименьших квадратов

В некоторых случаях статистическое ранговое распределение может с высокой точностью описываться законом Вейбулла, функция распределения и плотность вероятностей которого заданы формулами (13) и (14). Этот закон впервые использовал Г.Г. Белоногов для описания рангового распределения слов частотного словаря [8]. Поскольку этот закон весьма простой, его целесообразно проверять в первую очередь при отыскании подходящего рангового распределения. Для такой проверки функцию распределения необходимо преобразовать к линейному виду

$$\ln\ln(1/(1\text{-}F(t))) = \ln\alpha + \beta \ln t . \qquad (23)$$

Введем обозначения:

$$Y = \ln\ln(1/(1\text{-}F(t))), \; X = \ln t . \qquad (24)$$

Тогда последнее уравнение запишется в виде

$$Y = \ln\alpha + \beta X . \qquad (23')$$

Для проверки применимости закона Вейбулла необходимо по статистической функции распределения вычислить значения X, Y по формулам (24) и построить график зависимости Y=f(X). Если эмпирические точки расположатся вдоль прямой (23′), то далее по методу наименьших квадратов следует вычислить оценки параметров α и β этой прямой:

$$\beta = \frac{\overline{XY} - \overline{X}\,\overline{Y}}{\overline{X^2} - \left(\overline{X}\right)^2}, \quad \alpha = \exp\left(\overline{Y} - \beta \overline{X}\right). \qquad (25)$$

Для оценки тесноты линейной связи между переменными Y, X вычисляется выборочный коэффициент корреляции

$$R_{y/x} = \frac{\overline{XY} - \overline{X}\,\overline{Y}}{\sigma_x \sigma_y}, \qquad (26)$$

где средние квадратические отклонения σ_x, σ_y равны:

$$\sigma_x = \sqrt{\overline{X^2} - (\overline{X})^2}, \; \sigma_y = \sqrt{\overline{Y^2} - (\overline{Y})^2} .$$

При этом

$$\overline{X} = \frac{1}{N}\sum_{i=1}^{N} X_i , \quad \overline{Y} = \frac{1}{N}\sum_{i=1}^{N} Y_i ,$$

$$\overline{X^2} = \frac{1}{N}\sum_{i=1}^{N} X_i^2 , \quad \overline{Y^2} = \frac{1}{N}\sum_{i=1}^{N} Y_i^2 ,$$

где N – количество значений случайных величин X, Y.

Абсциссы точек A, C, B для закона Вейбулла вычисляются по формулам (18), (22). Значения функции распределения в этих точках при любых значениях параметров α и β соответственно равны:

$$F(t_A) = 0{,}31748; \; F(t_C) = 0{,}63212; \; F(t_B) = 0{,}92705. \qquad (27)$$

Отметим, что статистические данные расположатся вдоль прямой лишь в случае однородной выборки, для которой справедлив закон Вейбулла. Однако, если попытаться описать этим законом ранговое распределение слов частотного словаря, то окажется, что первые 50–100 наиболее частых слов не подчиняются закону Вейбулла. Это в основном служебные слова. Они составляют неоднородную часть выборки. Поэтому для более точного описания таких ранговых распределений можно предварительно удалить первые 50–100 слов с последующим пересчетом рангов и относительных частот слов, получив, таким образом, однородную выборку. Если же имеется необходимость аппроксимировать ранговое распределение всех слов частотного словаря, включая служебные, то следует ввести дополнительный параметр в теоретическое распределение. Наши исследования в свое время привели к выводу, что закон Вейбулла с учетом третьего параметра (обозначим его δ) можно представить в следующем виде [9]

$$F(t) = 1\text{-}e^{-\alpha[(t+1)^{\beta} - e^{-\delta t}]}, \qquad (28)$$

$$p(t) = \frac{\alpha\beta(t+1)^{\beta-1} + \alpha\delta e^{-\delta t}}{e^{\alpha[(t+1)^{\beta} - e^{-\delta t}]}} . \qquad (29)$$

Параметры α, β могут быть вычислены по методу наименьших квадратов по формулам (23)–(25) для рангов слов частотного словаря от 50–100 до рангов слов с частотой 2–3. Далее при известных оценках этих параметров вычисляется дополнительный параметр δ по формуле, которая следует из функции распределения (28):

$$\delta = -\frac{1}{t}\left[\ln\left((t+1)^{\beta} - \frac{1}{\alpha}\ln\frac{1}{1-F(t)}\right)\right]. \qquad (30)$$

Его можно вычислить один раз при заданной относительной частоте самого частого слова, которая равна функции распределения F(t=1).

Двухпараметрическим законом Вейбулла хорошо описываются некоторые ранговые распределения журналов, терминов, ключевых слов, образующих

статистически однородные выборки, а трехпараметрическим – некоторые неоднородные выборки.

Рассмотрим для примера *ранговое распределение слов «Частотного словаря современного русского языка»* [10]. Он построен на выборке огромного размера – 135 млн словоупотреблений. Количество разных лексем (в источнике – лемм) составило 739930 единиц. Из них количество лемм с частотой 2 и более раза составило 360755, с частотой 3 и более раза – 268106. В указанном источнике приводится частотный список первых 20000 лемм, что позволяет вычислить накопленные относительные частоты, т.е. функцию распределения для всех рангов от 1 до 20000. При известном количестве лемм с частотами употребления 1 и 2 раза (соответственно 379175 = 739930 – 360755 и 92649 = 360755 – 268106) можно дополнительно вычислить два значения функции распределения:

F(360755)=1–379175/135000000=0,9971913;

F(268106)=1–(379175+92649*2)/135000000=0,9958187.

Здесь из суммарной доли употреблений всех лемм, равной единице, вычитается доля употреблений лемм с частотой один раз – в первом случае и один и два раза – во втором.

Составим на базе Частотного словаря табл. 2, где приведем отдельные ранги слов (R≥80) и соответствующие этим рангам значения функции распределения (см. первые два столбца). Вычислим далее по методу наименьших квадратов оценки параметров закона Вейбулла, а также коэффициент корреляции. Параметр β оказался равным 0,309427, параметр α=0,111757. Коэффициент корреляции Ry/x=0,999789. Это значит, что эмпирическая зависимость оказалась близкой к теоретической прямой (23), которая представлена на рис. 2.

При известных оценках параметров α, β и функции распределения F(t=1)=0,035802 по формуле (30) найдем значение третьего параметра, который необходим для более точного описания наиболее частых слов: δ=0,091. Вычислим далее значения функции распределения по трехпараметрическому закону Вейбулла (см. табл. 3) и построим в полулогарифмическом масштабе график функции распределения (см. рис. 3) с учетом служебных слов, где отдельными точками показана эмпирическая функция распределения, сплошной линией – теоретическая.

Таблица 2

Расчет параметров закона Вейбулла по статистическому распределению

Ранги слов	Функция распределения	lnr	$\ln\ln\frac{1}{1-F(r)}$					
r	F(r)	X	Y	XY	X^2	Y^2	Yрасч.	Fрасч.
80	0,354324	4,382027	-0,82678	-3,62295	19,20216	0,683558	-0,83551	0,351864
100	0,374545	4,60517	-0,75656	-3,48411	21,20759	0,57239	-0,76646	0,371648
150	0,411675	5,010635	-0,63398	-3,17665	25,10647	0,401932	-0,641	0,409488
250	0,458992	5,521461	-0,48724	-2,69026	30,48653	0,2374	-0,48294	0,460423
400	0,506106	5,991465	-0,34894	-2,09067	35,89765	0,12176	-0,3375	0,510098
666	0,561671	6,50129	-0,19263	-1,25236	42,26677	0,037107	-0,17975	0,566333
1097	0,620558	7,000334	-0,03144	-0,22006	49,00468	0,000988	-0,02533	0,622802
1809	0,681615	7,500529	0,134963	1,012291	56,25794	0,018215	0,129442	0,679603
3000	0,740589	8,006368	0,299617	2,398842	64,10192	0,08977	0,285962	0,735798
4900	0,792717	8,49699	0,453411	3,852626	72,19885	0,205581	0,437774	0,787594
8100	0,838483	8,999619	0,600563	5,404838	80,99315	0,360676	0,593301	0,836338
13360	0,875102	9,50002	0,732492	6,958688	90,25039	0,536544	0,748139	0,879133
20000	0,898718	9,903488	0,828485	8,204889	98,07907	0,686387	0,872983	0,90874
268106	0,995819	12,49914	1,700595	21,25597	156,2284	2,892023	1,676148	0,995228
360755	0,997191	12,79595	1,770694	22,65771	163,7364	3,135356	1,767992	0,997146
Summa		116,7145	3,243251	55,2088	1005,018	9,979688		
Srednee		7,780966	0,216217	3,680587	67,0012	0,665313		
Beta=	0,309427	Sx=	2,541215					
Alfa=	0,111757	Sy=	0,786488					
Ry/x=	0,999789	δ=	0,091					

Из табл. 3 и графика функции распределения видно, что введение третьего параметра в закон Вейбулла позволило весьма точно аппроксимировать статистическое распределение первых 50–80 слов Частотного словаря. Ранговое распределение остальных слов хорошо описывается классическим законом Вейбулла с двумя параметрами.

Вычислим абсциссы трех характерных точек по формулам (18) и (22) при известных оценках параметров α и β: n=22,4287; $t_C = 1191$; $t_A = 53$; $t_B = 26704$. Отметим, что логарифмы рангов слов в точках A, C, B равны: 3,97187; 7,08221; 10,19256. Теоретическая функция распределения в этих точках равна: 0,31748; 0,63212; 0,92705.

Из этих расчетов следует, что ядро Частотного словаря составляют первые 53 слова. Они покрывают 31.7% текста. В первую зону рассеяния A–C входит 1191-53=1138 слов, которые покрывают 31.5% текста (63.2-31.7=31.5). Во вторую зону C–B входит 26704-1191=25513 слов, которые покрывают 29.5% текста (92.7-63.2=29.5). В третью зону рассеяния входит вся остальная лексика 739930-25513=714417 слов. Этот огромный словарь покрывает лишь 7.3% текста (100-92.7=7.3).

Рассмотрим еще один пример – *ранговое распределение периодических изданий по химии и химической технологии* [1]. Поскольку в данном случае выборка однородная, то для аппроксимации статистического рангового распределения может быть использован закон Вейбулла с двумя параметрами. Проведем необходимые расчеты по методу, изложенному выше. Результаты представлены в виде табл. 4 и рис. 4.

Эти результаты свидетельствуют о высокой точности аппроксимации законом Вейбулла статистического рангового распределения журналов, упорядоченных по убыванию опубликованных в них статей по химии и химической технологии. Коэффициент корреляции Ry/x=0,999705. Ядро образуют 88 журналов. Количество журналов до точки C, т. е. входящих в ядро и первую зону рассеяния, равно 552. В ядро и в первые две зоны рассеяния, т. е. до точки B входят 3469 журналов, в которых содержится 92,705% статей от их общего количества 187911. На третью зону рассеяния приходятся все остальные журналы 10850-3469=7381, и в этих журналах содержится 100-92,705=7,295 % статей.

Рис. 2. Прямая Вейбулла

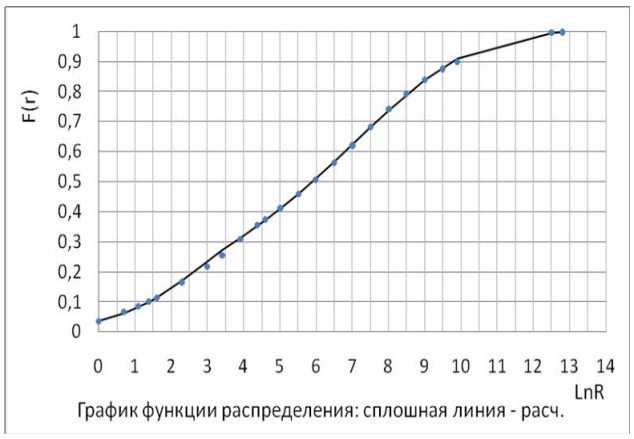

Рис. 3. Функция распределения Вейбулла

Таблица 3

Эмпирическая и теоретическая функции распределения наиболее частых слов

Ранги слов	Эмпирическая функция распределения	Уточненная теоретическая функция распределения
1	0,035802	0,035798
2	0,067176	0,061847
3	0,085204	0,082922
4	0,101071	0,100782
5	0,113756	0,116317
10	0,165571	0,172804
20	0,217115	0,235534
30	0,255761	0,271023
50	0,309294	0,313454

Таблица 4

Рассеяние журнальных публикаций по химии и химической технологии (10850 журналов, 187911 статей)

Кол.-во журналов	Доля статей	lnt	$\ln\ln\frac{1}{1-F(t)}$					
t	F(t)	X	Y	XY	X^2	Y^2	F(t)расч.	Yрасч.
18	0,15	2,890372	-1,817	-5,25181	8,3542489	3,301489	0,1536	-1,7912
50	0,25	3,912023	-1,2459	-4,87399	15,303924	1,552267	0,24772	-1,25649
100	0,34	4,60517	-0,8782	-4,04426	21,207592	0,771235	0,33577	-0,89371
500	0,62	6,214608	-0,033	-0,20508	38,621354	0,001089	0,61323	-0,05138
1000	0,75	6,907755	0,3266	2,25607	47,717083	0,106668	0,7447	0,3114
2000	0,85	7,600902	0,6403	4,86686	57,773718	0,409984	0,85948	0,67418
Summa		32,13083	-3,0072	-7,25221	188,97792	6,142732		
Srednee		5,355138	-0,5012	-1,2087	31,49632	1,023789		
Beta=	0,523374	Sx=	1,67893183	tc=	551,5708	F(tc)=	0,63212	
Alfa=	0,036738	Sy=	0,87896938	ta=	87,69708	F(ta)=	0,31748	
Ry/x=	0,999705	n=	6,28949982	tb=	3469,104	F(tb)=	0,92705	

Рис. 4. Прямая Вейбулла – рассеяние журнальных публикаций

Однако, несмотря на высокую точность аппроксимации некоторых ранговых распределений законом Вейбулла, в исследованиях по информатике и математической лингвистике он применяется весьма редко. Более часто используется закон Дж. Ципфа, который вовсе нельзя применять в таких исследованиях. Принимая во внимание то обстоятельство, что оба этих закона и множество других являются частными случаями обобщенного распределения (2), для описания различного рода ранговых распределений следует использовать вторую систему непрерывных распределений.

ЗАКЛЮЧЕНИЕ

При обработке статистических рядов распределения главной задачей является вычисление теоретического закона распределения. Она решается довольно просто по методам, изложенным в теории обобщенных распределений. Для аппроксимации статистических ранговых распределений используется вторая система непрерывных распределений.

На основании анализа свойств обобщенных распределений нами предлагаются математически точные формулировки закона рассеяния публикаций в

смысле Бредфорда. Но такие формулировки, как и закон С.Бредфорда, не могут быть приняты в качестве полноценного закона рассеяния публикаций. Универсальным законом рассеяния является вторая система непрерывных распределений, поскольку обобщенная четырехпараметрическая плотность, т. е. закон распределения, наиболее полно характеризует случайную величину.

Для вычисления закона распределения и оценок его параметров по статистическому ранговому распределению используется общий устойчивый метод. При известных оценках параметров по заранее выведенным формулам вычисляются абсциссы трех характерных точек A, C, B, которые приняты автором в качестве границ ядра и зон рассеяния. Абсциссы точек C и B, вычисленные по закону С.Бредфорда и универсальному закону, различаются на 20–25 % при условии, что величина $n = 5$, а размер ядра в обоих случаях одинаков. С ростом n эта погрешность уменьшается.

Для использования аналитического метода необходимо знать хотя бы некоторые сведения из теории обобщенных распределений. Этот метод рассчитан на подготовленного исследователя.

На базе свойств ранговых распределений нами предложен графический метод приближенного вычисления границ ядра и зон рассеяния. Он значительно проще аналитического метода, поскольку не требует вычисления закона распределения.

В случае однородных выборок некоторые статистические ранговые распределения могут быть описаны законом Вейбулла (13), (14). Оценки параметров этого закона наиболее просто вычисляются по методу наименьших квадратов. Если при этом ранговое распределение содержит неоднородную часть, например, служебные слова, то по формуле (30) следует дополнительно вычислить третий параметр δ при известных значениях параметров α, β и относительной частоты первого слова.

Проведенное исследование показало высокую точность аппроксимации некоторых статистических ранговых распределений законом Вейбулла. Однако для гарантированного вычисления наилучшего теоретического рангового распределения по статистическому ряду следует использовать вторую систему непрерывных распределений и общий устойчивый метод.

СПИСОК ЛИТЕРАТУРЫ

1. Михайлов А.И., Черный А.И., Гиляревский Р.С. Основы информатики. – М.: Наука, 1968. – 756 с.
2. Нешитой В.В. Универсальные законы рассеяния и старения публикаций // Веснік Беларус. дзярж. ун-та культуры і маст.– 2007.– № 8.– С. 128–133.
3. Нешитой В.В. Элементы теории обобщённых распределений: монография.– Минск: РИВШ, 2009. – 204 с.
4. Нешитой В.В. Математико-статистические методы анализа в библиотечно-информационной деятельности: учеб.-метод. пособие. – Минск: БГУ культуры и искусств, 2009. – 203 с.
5. Нешитой В.В. Методы статанализа в библиотечной деятельности: вычисление непрерывных распределений: учеб.-метод. пособие. – Минск: Бел. гос. ун-т культуры и искусств, 2010. – 61с.
6. Нешитой В.В. Законы Ципфа, Бредфорда и универсальные модели // Научно-техническая информация. Сер. 2. – 2010. – № 1. – С. 26–33; Neshitoi V.V. Zipf's and Bradford's laws and universal models // Automatic Documentation and Mathematical Linguistics. – 2010. – Vol. 44, № 1. – P. 30–37.
7. Нешитой В.В. Форма представления ранговых распределений // Ученые записки Тартуского гос. ун-та. – 1987. – Вып. 774. – С. 123–134.
8. Белоногов Г.Г. О некоторых статистических закономерностях в русской письменной речи // Вопросы языкознания. – 1962. – № 1. – С. 100–101.
9. Нешитой В.В. Законы распределения слов в тексте и его лексическая параметризация: дис... канд. филол. наук. – Минск, 1973. – 135 с.
10. Ляшевская О. Н., Шаров С. А. Частотный словарь современного русского языка (на материалах Национального корпуса русского языка). – М.: Азбуковник, 2009. – URL: http://dict.ruslang.ru/freq.php.

Материал поступил в редакцию 03.05.13.

Сведения об авторе

НЕШИТОЙ Василий Васильевич – доктор технических наук, профессор, заведующий кафедрой информационных ресурсов УО «Белорусский государственный университет культуры и искусств», Минск
e-mail**:** neshitoi_vv@tut.by

ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ

УДК [004.89:510.64]:001.103

А.Ю. Волкова

Опыт создания интеллектуальной ДСМ-системы для исследования данных различных предметных областей

Представлена ДСМ-система для интеллектуального анализа данных различных предметных областей: фармакологии, медицинской диагностики и социологии. Описаны особенности структуры и программная реализация основных модулей системы, а также результаты ее практического применения.

Ключевые слова: *ДСМ-метод, индукция, аналогия, абдукция, интеллектуальный анализ данных, качественный анализ данных, ДСМ-Решатель, фармакология, медицина, социология.*

ВВЕДЕНИЕ

ДСМ-рассуждения успешно применяются в различных предметных областях: фармакологии, технической диагностике, доказательной медицине, социологии, криминалистике, при создании интеллектуальных роботов [1-10].

ДСМ-метод автоматического порождения гипотез (АПГ) предполагает возможность применения различных стратегий для интеллектуального анализа данных (под интеллектуальным анализом понимается анализ с помощью интеллектуальной системы). Созданные ранее системы, по большей части, ограничивались одной предметной областью и предоставляли реализацию стандартных стратегий: простой метод сходства и запрет на контрпример [2, 3]. В настоящей статье речь пойдет о создании интеллектуальной системы, которая позволяет исследователю проводить анализ данных различных предметных областей (фармакология, медицинская диагностика, социология), применяя методы ДСМ-рассуждений, в том числе методы, отсутствующие в других системах: метод различия, метод сходства-различия, метод остатков [11]. Кроме того, в системе доступны прямой и обратный тип ДСМ-рассуждений, а для анализа социологических данных предусмотрено применение ситуационного расширения ДСМ-метода. Изучение этих данных с использованием разнообразных стратегий может не только упростить процесс оценки экспертом полученных результатов работы ДСМ-системы (так как позволяет сократить количество найденных причинно-следственных зависимостей, выделив наиболее существенные), но и выявить зависимости, которые не могут быть получены с помощью базовых стратегий (эту возможность предоставляет метод остатков).

ДСМ-метод АПГ является мощным логико-комбинаторным методом интеллектуального анализа данных [1-3], в его основе лежит синтез трех познавательных процедур: «эмпирическая индукция – структурная аналогия – абдукция». Эти процедуры реализуются в виде когнитивных правдоподобных рассуждений.

Данные предметной области должны быть хорошо структурированы, что предполагает представление их в виде отдельных объектов с четким описанием, допускающим определение операции сходства, вложения, объединения и разности на объектах-фактах. Среди признаков, описывающих факт, должно присутствовать некоторое свойство («эффект»), по которому мы можем разделить факты на положительные (+) и отрицательные (–)-примеры (обладающие и не обладающие «эффектом»), а также, возможно, противоречивые примеры (0-примеры, т. е. обладающие и не обладающие эффектом одновременно) и неопределенные (обозначаются через τ) примеры (факты, требующие доопределения) [2, с. 399 – 400].

На этапе индукции мы находим сходства фактов (гипотезы), применяя различные процедуры ДСМ-метода и их комбинации – стратегии [11-13]. На этапе аналогии происходит попытка доопределить (τ)-примеры полученными гипотезами, т. е. сделать прогноз, обладают ли эти факты исследуемым «эффектом». Наконец, процедура абдукции предполагает проверку обоснованности полученных гипотез путём объяснения начальной базы фактов найденными гипотезами.

Очевидно, что успешное применение ДСМ-метода возможно только в том случае, если в исходных данных действительно содержатся скрытые причинно-следственные зависимости.

В языке ДСМ-метода АПГ каждый факт представляется с помощью двухместного предиката $\Rightarrow_1$, означающего «объект обладает множеством свойств» [3]. Структурно это предполагает, что у каждого объекта есть «левая» (описание объекта) и «правая» (набор свойств) часть. Отталкиваясь от этого определения, назовем ДСМ-метод атомарным в том случае, если правая часть объекта – это одноэлементное множество, и неатомарным – иначе.

Например, в качестве объекта выступает пациент, описание объекта – это определенные характеристики такие, как пол, возраст, хронические болезни и прочее. Пусть в исследуемой задаче описания пациентов можно разделить на (+), (–) и (τ)-примеры, например, по наличию/отсутствию некоторой болезни. В этом случае применим атомарный ДСМ-метод. Если же исследуемая болезнь допускает представление в виде набора характеристик, то эти свойства можно отнести в правую часть. Тогда применим неатомарный ДСМ-метод.

На этапе индукции операцию сходства фактов можно применять сначала к левым частям объектов (описание пациента), а затем к правым (характеристики болезни), если правая часть присутствует. В этом случае ДСМ-метод мы назовем прямым. Если же операция сходства выполняется сначала на правых частях объекта, а затем на левых, то назовем ДСМ-метод обратным. Подробное описание решающих предикатов прямого ДСМ-метода можно найти, например, в [3], обратного – в [14].

Следует также отметить, что представление фактов может предполагать разделение как левой, так и правой части на группы. Например, описание респондента может иметь группу характеристик, включающих какие-либо биографические данные человека, и группу характеристик, описывающих некоторые ситуационные характеристики (место работы, зарплата и прочее), группу свойств, отражающих мнение респондента по некоторым вопросам. При таком представлении данных необходимо уже использовать ситуационный ДСМ-метод [15].

Создание «универсальной» программы, в которой были бы реализованы разные ДСМ-стратегии применительно к разным предметным областям, было необходимо не только для научного исследования, но и как практический инструмент для качественного анализа данных. Созданная программа лишена графического предметно-ориентированного интерфейса, но всё-таки включает минимальный набор меню для настройки параметров при проведении эксперимента. Таким образом, приложение можно считать «неполной» ДСМ-системой. Входные данные программы должны быть заполнены в файлах с заранее определенной структурой. Наличие же специфических визуальных стредств для подготовки данных неизбежно повлекло бы за собой зависимость программы от предметной области.

1. ОБЩАЯ СТРУКТУРА СИСТЕМЫ

ДСМ-система была создана на языке программирования C++ с использованием библиотеки классов MFC (Microsoft Foundation Classes) и библиотеки стандартных шаблонов STL (Standard Template Library), а также интерфейса для доступа и манипулирования внешними данными ADO и языка запросов SQL.

Структуру ДСМ-системы можно представить в виде трех блоков:

Представление данных	ДСМ-Решатель	Графический интерфейс пользователя

Модуль «Представление данных» необходим для внутреннего описания данных различных предметных областей. В текущей реализации определены классы для представления фармакологических, медицинских и социологических данных.

«ДСМ-Решатель» отвечает за реализацию различных стратегий и процедур ДСМ-метода АПГ. При этом допускается использование только обобщенных классов, никак не зависящих от конкретной предметной области.

«Графический интерфейс пользователя» предоставляет пользователю системы возможности настройки параметров и проведения экспериментов.

Язык C++ позволяет обеспечить следующие важные принципы системы:

Независимость модуля «ДСМ-Решатель» от модуля «Представление данных» (предметной области). Этот принцип реализуется благодаря возможностям языка C++: поддержка объектно-ориентированного программирования, поддержка виртуальных функций и возможность динамического приведения типов наследуемых классов к базовому [16]. Таким образом, все процедуры ДСМ-Решателя используют один и тот же класс для представления факта и тела гипотезы, который является базовым для всех классов предметных областей.

Оптимизация скорости выполнения процедур (временна́я сложность). Необходимо минимизировать время формирования гипотез. Технически это достигается благодаря использованию динамических массивов данных, передачи в качестве входных/ выходных параметров указателей (ссылок) на объекты [16], сохранение ссылок на зависимые данные. Например, все объекты исходной базы фактов подаются на вход ДСМ-Решателю в виде динамического массива (контейнера vector библиотеки STL) указателей базового класса, что позволяет иметь быстрый доступ к каждому объекту, значительно ускоряет разбиение массива на подмассивы (+), (–), (0) и (τ)-примеров и обеспечивает хорошую скорость при получении пересечений примеров (алгоритм Норриса), наиболее затратному по времени алгоритму в цикле ДСМ-рассуждений.

Оптимизация расхода памяти (ёмкостная сложность). Это свойство достигается благодаря использованию динамических массивов данных и указателей на объекты, динамическому выделению памяти и освобождению памяти от неиспользуемых объектов во время выполнения.

2. ДСМ-РЕШАТЕЛЬ

Характеристиками ДСМ-Решателя являются:

независимость от предметной области;

поддержка атомарной/неатомарной версии ДСМ-метода;

поддержка прямой/обратной версии ДСМ-метода;

поддержка простой / ситуационной версии ДСМ-метода;

реализация четырех индуктивных методов Д.С. Милля [11]: простой метод сходства, метод различия (миллевский метод различия и упрощение метода сходства-различия), метод соединенного сходства-различия, метод остатков;

реализация усеченного метода остатков;

реализация ограничений на базовые стратегии: единственность причины/следствия, запрет на контрпримеры, установка значения на порог родителей гипотез;

реализация алгоритма определения (не)протиоречивости гипотез (для двух и более массивов с полученнымии гипотезами);

реализация алгоритма восстановления связей между гипотезами (построение дерева гипотез);

поддержка конъюнктивных и дизъюнктивных (±) фильтров (выделение конъюнктивных признаков, обязательно присутствующих в гипотезе, и дизъюнктивных признаков, из которых в гипотезе должен присутствовать хотя бы один признак);

реализация процедуры аналогии (доопределение фактов, требующих прогнозирования);

реализация процедуры «доопределение по одному» (Последовательно каждому примеру исходной базы фактов (с оценками «+», «–» и «0») присваивается значение «τ», производится доопределение этого объекта средствами ДСМ-Решателя с выбранными параметрами, затем доопределенное значение сравнивается с существующим. Подсчитывается общее количество правильных и неправильных доопределений.);

реализация процедуры для определения абдуктивной сходимости [11] гипотез.

Модуль « ДСМ-Решатель» включает в себя следующие классы:

algorithmJSM (базовый класс реализации ДСМ-метода, содержит общие настройки и необходим для унифицированной работы с разновидностями версий ДСМ-метода),

algorithmJSM_atomicDirect (подкласс algorithmJSM, содержит процедуры атомарного прямого ДСМ-метода),

algorithmJSM_nonAtomicDirect (подкласс algorithmJSM, содержит процедуры неатомарного прямого ДСМ-метода),

algorithmJSM_reverse (подкласс algorithmJSM, содержит процедуры обратного ДСМ-метода),

truncResid (класс, который реализует усеченный метод остатков),

hypothesis (класс, представляющий гипотезу),

filter (класс, который содержащит информацию о фильтрах, применяемых в эксперименте),

ContradictionCalculation_Algorithm (алгоритм подсчета (не)противоречивости массивов гипотез),

algorithm_HypothesesBush (алгоритм для восстановления данных о «дереве» гипотез).

Основные детали реализации и соотнесение их с формальным описанием ДСМ-метода описаны в [17-19]. Реализация алгоритма определения непротиворечивости гипотез базируется на описании, представленном в [20].

Стоит отметить, что ситуационная версия ДСМ-метода основана на представлении данных в системе. Операция сходства для каждого объекта содержится в классе соответствующей ему предметной области. Таким образом, ДСМ-Решателю не требуется информация о структуре объекта. В процедурах вызывается просто операция сходства, возвращающая результат пересечения. Ситуационная версия автоматически поддерживается и для прямого, и для обратного ДСМ-метода.

Для обратного ДСМ-метода поддерживаются стратегии простого метода сходства, единственность следствия и запрет на контрпримеры.

Методы различия, метод соединенного сходства-различия и методы остатков (усеченный и обобщенный) были реализованы впервые. По сравнению с существующими ДСМ-системами, в которых обычно доступны прямой метод сходства и запрет на контрпримеры, ДСМ-Решатель предоставляет широкий набор возможностей при проведении экспериментов, так как позволяет исследовать данные при помощи различных стратегий.

3. ПРЕДСТАВЛЕНИЕ ДАННЫХ В СИСТЕМЕ

Блок «Представление данных» включает классы, описывающие объекты предметной области. С одной стороны, это абстрактные классы, которые затем используются в решателе, а с другой – это классы, представляющие объекты конкретных предметных областей.

В текущей реализации с помощью атомарного ДСМ-метода возможна обработка медицинских, социологических и фармакологических данных. Для неатомарного ДСМ-метода при выборе предметной области доступны медицина и социология.

Таким образом, для компьютерной обработки фактов необходимо было создать такую структуру данных, которая, с одной стороны, позволяла бы изменять представление объекта, а с другой – не влияла на процедуры ДСМ-Решателя.

Была разработана структура, которая позволяет сделать решатель универсальным, т. е. не зависящим от предметной области.

В основе представления лежит иерархия классов, представленная на рис. 1.

Опишем структуру каждого класса.

objectX – класс, который используется в модуле ДСМ-Решателя и представляет собой объект, не зависящий от предметной области.

В классе objectX определены виртуальные функции, которые вызываются в решателе: функции пересечения, вложения, равенства, разности, объединения и др. В классах-потомках эти функции переопределены в соответствии со спецификой рассматриваемой предметной области, кроме того, у каждого класса-потомка есть свои дополнительные функции (считывание/запись из файла и т.д.). Также класс objectX содержит наиболее общие поля-переменные: идентификатор объекта, имя, тип («+», «–», «0» или «?»

для (τ)-примеров) и шаг ДСМ-рассуждений. Для (τ)-примеров шаг – это номер такта, на котором последний раз была попытка доопределения, для остальных примеров шаг равен 0, что соответствует понятию факта в ДСМ-терминологии.

objectPharma – класс, который описывает объект фармакологических данных. В исходных базах фактов примеры – это химические соединения, представленные как набор дескрипторов ФКСП (Фрагментарный Код Суперпозиции Подструктур). В состав каждого соединения каждый дескриптор входит 0 или более раз.

В программе объект класса objectPharma – это динамический массив, хранящий количество вхождений дескрипторов. Существует вспомогательный класс descriptor, который применяется для представления дескрипторов. Данные на вход программы поступают в виде .xls файла особого вида. В файле присутствуют листы для описания дескрипторов и объектов, а также листы, куда записываются полученные гипотезы, результат доопределения (τ)-примеров и общая информация об эксперименте (какие стратегии ДСМ-метода АПГ были применены в эксперименте, использовались ли фильтры, количество примеров и полученных гипотез, количество необъясненных гипотезами примеров и др.). Структура такого файла описана подробно в [17].

objectMed_atomic – класс, который описывает объект медицинских данных, который использует атомарный ДСМ-метод.

Объект медицинских данных – это пациент, которому сопоставлен набор признаков. Каждый признак имеет свой тип [2, с. 420-424]. Все типы в программе сводимы к трем общим типам: кортеж заданной длины, признак иерархической структуры («дерево»), интервал («указывается норма признака или интервал отклонения от нормы с указанием направления отклонения» [2, с. 421]).

Для описания типов признаков в системе присутствуют вспомогательные классы: simple_descr для описания кортежа, norma_descr для описания интервала и tree_descr для описания «дерева».

Сам признак представлен классом element, в котором присутствуют функции для считывания признаков и их обработки в зависимости от типа.

Например, в случае типа дерева присутствие/отсутствие значения признака на дочернем элементе влечет присутствие/отсутствие значения родительского признака. В качестве признака пациента, мы рассмотрим «Проведенное ранее лечение». Если в иерархической структуре

1. Химиоиммунотерапия
   1.1 Араноза+Реаферон
   1.2 Дакарбазин + Интерферон

у пациента присутствуют значения 1.1 или 1.2, то это означает, что присутствует и значение 1.

Если же 1.1 и 1.2 отсутствуют, то отсутствует и 1. В исходных данных подобные зависимости не отмечены, поэтому требуется отметить эти связи на стадии считывания данных для эксперимента.

Подобная предобработка данных позволяет свести описание пациента к набору кортежей, где каждый элемент - это ‘+’ (обозначает «присутствие признака»), или ‘x’ («отсутствие признака»), или ‘t’ (в зависимости от типа признака трактуется, как «значение признака неизвестно» или «отсутствие признака»).

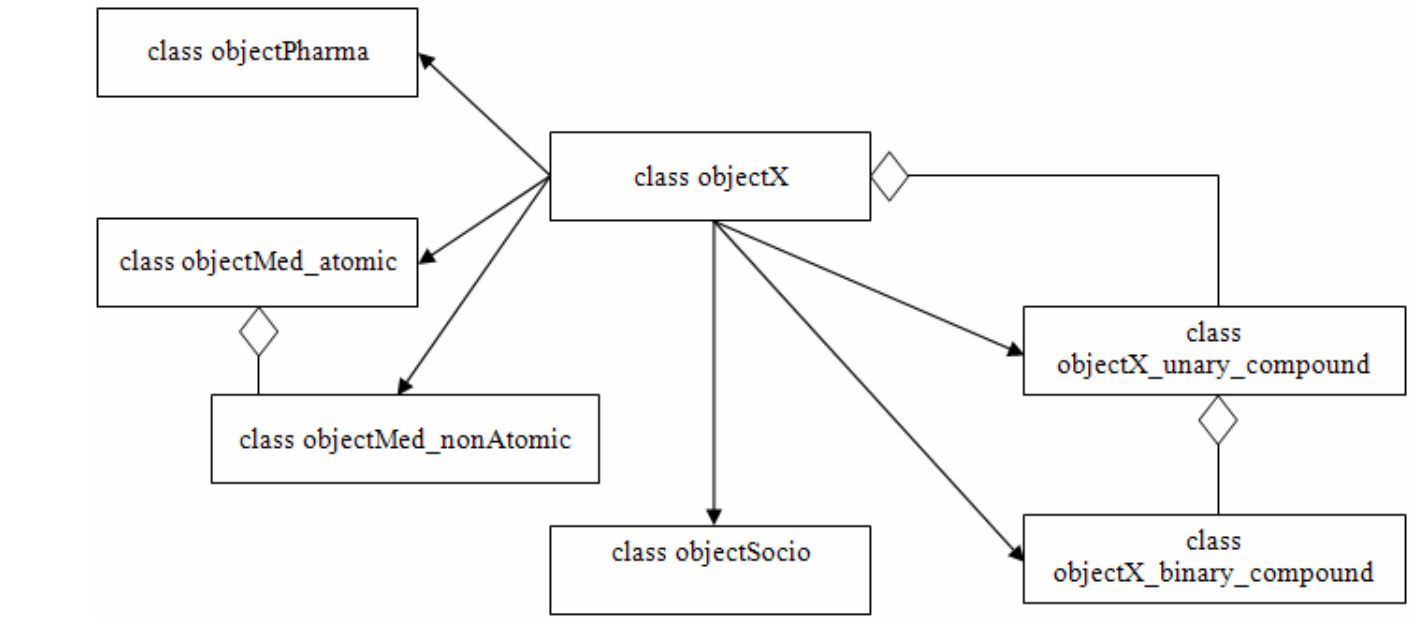

Рис. 1. Иерархия классов внутреннего представления данных

– отношение «родительский – производный классы»

– отношение «содержит».

Приведем табл. 1 соответствие между обозначениями типов признаков в программе и кодированием кортежей.

Таблица 1

Типы признаков медицинских данных

Обозначение типа признака	Краткое описание
single	Указывается одно из возможных значений, в строке помечается «+», невыбранные пункты помечаются «t».
korteg2	Допускается выбор нескольких значений признака: «+» – признак присутствует, «t» – значения нет.
korteg3	Допускается выбор нескольких значений признака: «+» – признак присутствует, «x» – признак отсутствует, «t» – значения нет.
norma	Указывается норма признака или интервал отклонения от нормы, в строке помечается «+», невыбранные пункты помечаются «t».
tree2	Признак иерархической структуры: «+» – признак присутствует, «t» – значения нет.
tree3	Признак иерархической структуры: «+» – признак присутствует, «x» – признак отсутствует, «t» – значения нет.

Набор всех значений признаков пациента кодируется с помощью 32-битовых машинных слов. Это обеспечивает минимизацию памяти и времени при работе решателя.

Данные на вход программы поступают либо в виде .mdb файла, либо в виде .xls и .txt файлов определенного вида.

Файл .mdb содержит базу данных (БД), а также интерфейс для просмотра результатов. На рис. 2 представлена схема БД, а в табл. 2 приводится ее краткое описание.

В отличие от Excel файла в интерфейсе для просмотра результатов доступно как схематичное описание, так и текстовое описание пациентов и гипотез, что удобно для пользователя при изучении данных (рис. 3).

Структура файла описана подробно в [17].

Файл .txt имеет определенную структуру и содержит описание исходных признаков. Создан специальный формальный язык, на котором необходимо записать признак и варианты его значений в зависимости от типа признака. Язык был разработан Д.А. Добрыниным (к.т.н., научный сотрудник ОТиППИ ВИНИТИ РАН, научный и технический руководитель Лаборатории робототехники и искусственного интеллекта) при проведении экспериментов с применением ДСМ-метода в области медицинской диагностики в ВИНИТИ РАН [1, 2].

В файле .xls присутствуют листы для краткого описания признаков (имя признака, число значений, поля для фильтров) и пациентов, а также листы, на которых записываются полученные гипотезы, результат доопределения (τ)-примеров и общая информация об эксперименте.

Структура файлов описана подробно в [17, 21].

objectMed_nonAtomic – класс, который описывает объект медицинских данных, используется в процедурах неатомарного ДСМ-метода.

Как следует из представленного выше рис. 1, этот класс основан на атомарном классе objectMed_atomic. В его состав входят два объекта класса objectMed_atomic: один для представления правой части, другой – левой.

Функции неатомарного объекта усложняются, так как требуется выполнять различные операции либо применительно к левой и правой частям, либо применительно только к одной из частей, при этом некоторые операции, например, пересечение или разность, могут создавать новую гипотезу или изменять уже полученную. Для этого все необходимые операции реализованы как для целого объекта, так и для его частей.

Файл .xls отличается только тем, что на листе со списком признаков появляются дополнительные поля (Left, Right), которые необходимо заполнить значениями 0 или 1 в зависимости от того, относится ли признак к левой части или входит в правую часть.

Что же касается обработки социологических данных, то структура представления объекта выглядит сложнее. Здесь каждый объект – это респондент, представленный набором признаков-ответов на вопросы анкеты. Признаки могут делиться на группы. В проводимых экспериментах было выделено три компоненты объекта: признаки, соответствующие описанию объекта, признаки, описывающие мнение, и признаки, описывающие ситуацию.

objectSocio – класс, который представляет собой одну из выделенных компонент. Вспомогательные классы необходимы для описания вопросов (в программе это класс question), а также компонент (класс component_info). При считывании вопросов в поля классов заносятся код и текст вопроса, а также хранится информация о его принадлежности к одной из компонент объекта. В классе, представляющем компоненту, мы храним такую информацию, как её тип (описание респондента, мнение, ситуация), положение в объекте (левая/ правая часть), обязательно ли или нет ее присутствие в гипотезе.

objectX_unary_compound – класс, который представляет собой левую или правую часть и содержит динамический массив элементов класса objectX. При заполнении данных в качестве элемента класса objectX создается объект класса objectSocio. Таким образом, представление объекта становится легко конфигурируемым: мы можем с легкостью выбирать, какую из компонент необходимо определить в правую или левую часть, и добавлять новые компоненты.

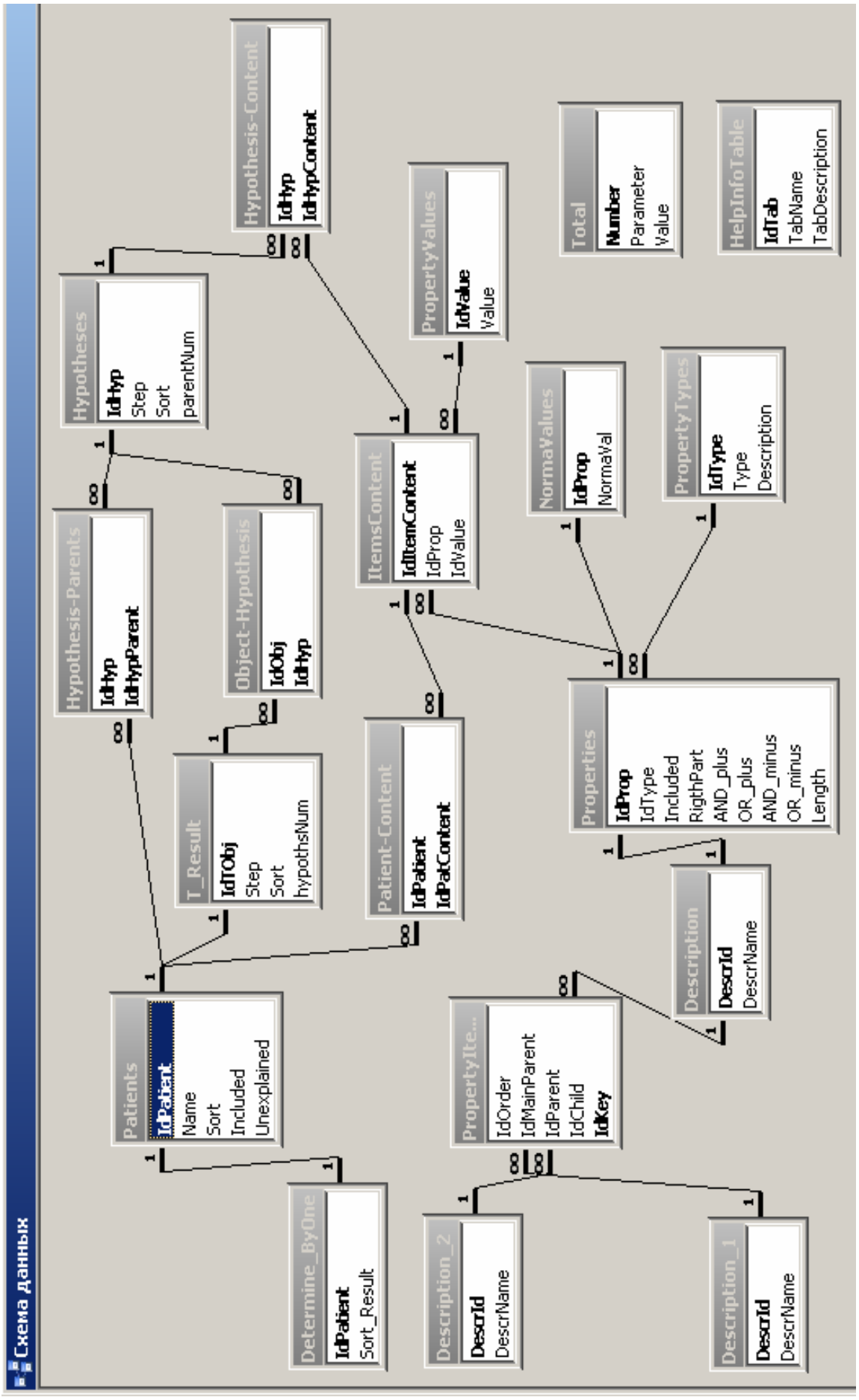

Рис. 2. Медицинские данные. Схема базы данных.

objectX_binary_compound – класс, который использует неатомарный ДСМ-метод. В качестве членов класса он включает два объекта типа objectX_unary_compound.

При применении атомарного ДСМ-метода в качестве класса, который описывает факт, можно использовать класс objectX_unary_compound.

Данные на вход программы поступают в виде .xls файла определенной структуры. В файле присутствуют листы для описания вопросов и объектов, а также лист с параметрами эксперимента. Полученные гипотезы записываются в структурированный .txt файл, а результат доопределения (τ)-примеров заносится в соответствующие поля .xls файла на вкладке с респондентами.

Для обработки социологических данных была создана отдельная интеллектуальная система, в которую вошел модуль «ДСМ-Решатель». Для системы был разработан модуль для автоматической конвертации файлов, созданных в профессиональной программе для обработки социологических данных Statistical Package for the Social Sciences (SPSS), формата .sav в формат Microsoft Office Excel.

Таблица 2

Описание таблиц базы данных

Название таблицы	Характеристика
Patients	Пациенты: идентификатор, имя, тип примера (+, -, 0 или ? для (τ)-примера), выбран ли для эксперимента (логический тип данных), объяснен ли пример полученными гипотезами (логический тип данных; принимает значение истина, если не объяснен).
Description	Содержит все возможные словесные описания признаков (названия и значения). Каждому признаку присваивается идентификатор.
Properties	Признаки: идентификатор названия, идентификатор типа, включен ли признак в эксперимент, является ли целевым свойством-эффектом, в какие фильтры входит, длина признака.
PropertyTypes	Типы признаков и их краткое описание.
NormaValues	Значение нормы (порядковый номер в последовательности) для типа «интервал отклонения».
PropertItems_Links	Хранит связи между идентификаторами признаков и их описаний. IdParent необходим для иерархических структур (деревьев). В случае неиерархических структур IdMainParent = IdParent.
PropertyValues	Все возможные значения строк, состоящих из «+», «x» и «t».
ItemsContent	Хранит связи между идентификаторами признаков и их возможных значений. Каждой паре IdProp – IdValue присваивается общий идентификатор.
Patient-Content	Хранит связи между идентификаторами пациентов и их описаний.
Hypotheses	Гипотезы: идентификатор, такт ДСМ-рассуждений, на котором была получена, тип гипотезы (+,–, 0) и число родителей (поле необходимо для повышения скорости вывода данных на форме).
Hypothesis-Content	Хранит связи между идентификаторами гипотез и их описаний.
Hypothesis-Parents	Хранит связи между идентификаторами гипотез и их родителей.
T_Result	Результат доопределения (τ)-примеров: идентификатор (τ)-примера, такт ДСМ-рассуждений, на котором последний раз была попытка доопределения, значение доопределения, число доопределивших гипотез (поле необходимо для повышения скорости вывода данных на форме).
Object-Hypothesis	Хранит связи между идентификаторами (τ)-примеров и доопределивших их гипотез.
Total	Общие: параметры запуска ДСМ-метода: выбранные стратегии, пороговые значения родителей гипотез и др.
Determine_ByOne	Доопределение по одному. Здесь содержится информация, как были доопределены ДСМ-Решателем исходные (+), (–) и (0)-примеры с выбранными для эксперимента параметрами. Таблица содержит идентификатор примера и тип (сорт) – результат доопределения.
HelpInfoTable	Вспомогательная таблица. Хранит справочную информацию о содержании вкладок главной формы.

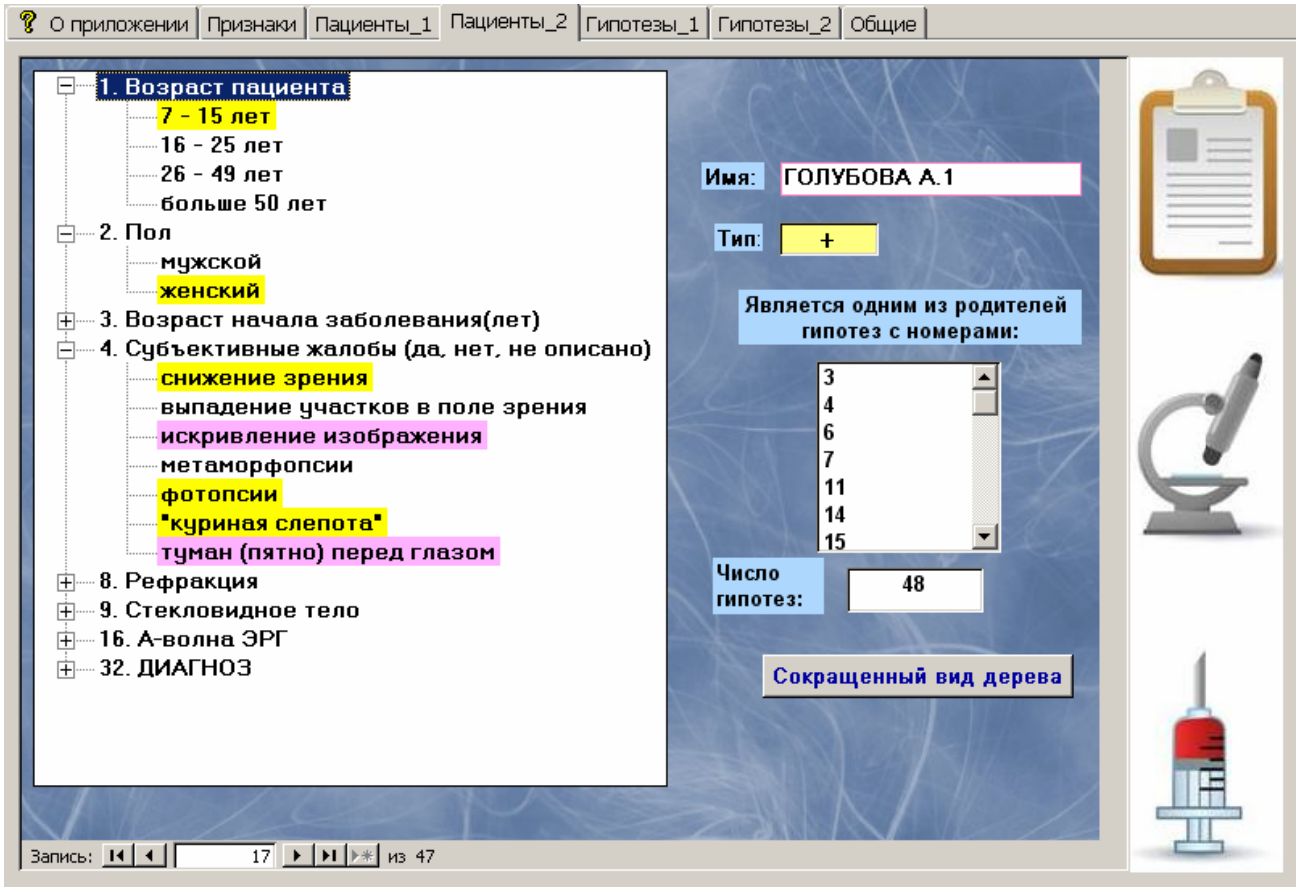

Рис. 3. Вкладка «Пациенты_2». Полный вид дерева признаков для выбранного пациента

4. ГРАФИЧЕСКИЙ ИНТЕРФЕЙС ПОЛЬЗОВАТЕЛЯ

Графический интерфейс системы был создан на языке программирования C++ с использованием библиотеки классов MFC (Microsoft Foundation Classes).

Интерфейс поддерживает следующие функциональные возможности:

1) выбор режима работы с атомарной версией ДСМ-метода;

2) выбор режима работы с неатомарной версией ДСМ-метода: прямой метод;

3) выбор режима работы с неатомарной версией ДСМ-метода: обратный метод;

4) выбор предметной области: для атомарного ДСМ-метода доступны «Фармакология», «Медицина», «Социология»; для неатомарного – «Медицина», «Социология»;

5) выбор режима работы с усеченным методом остатков и установка настроек в режиме «Усеченный метод остатков»;

6) выбор источника данных для проведения эксперимента в формате .xls (или .mdb для медицинских данных);

7) выбор файла для проведения эксперимента с описанием признаков для медицинских данных в формате .txt;

8) настройка ДСМ-Решателя на эксперимент: выбор стратегий и установка параметров;

9) вычисление противоречивости для двух массивов гипотез (выбор предметной области, выбор файлов с гипотезами и указанием допустимого порога непротиворечивости, вывод полученных степеней противоречивости для (+), (–) и (0)-гипотез);

10) вычисление противоречивости для N массивов гипотез (выбор предметной области, выбор файлов с гипотезами в порядке применения к ним алгоритма и указанием допустимого порога непротиворечивости, вывод полученных степеней противоречивости для (+), (–) и (0)-гипотез).

Опции 1-3, 5 поддерживаются с помощью главного меню приложения, также для удобства созданы кнопки на панели элементов управления (см. рис. 4-8). При наведении на какую-либо из кнопок появляется всплывающая подсказка (атомарный ДСМ-метод; неатомарный ДСМ-метод; обратный ДСМ-метод), а также подсказка в строке состояния окна приложения.

Опции 9, 10 доступны из главного меню приложения (рис. 9).

Допустимые параметры настройки решателя на эксперимент представлены на рис. 10, 11. Вид диалогового окна зависит от выбора версии ДСМ-метода. На иллюстрациях представлен диалог для неатомарного прямого ДСМ-метода.

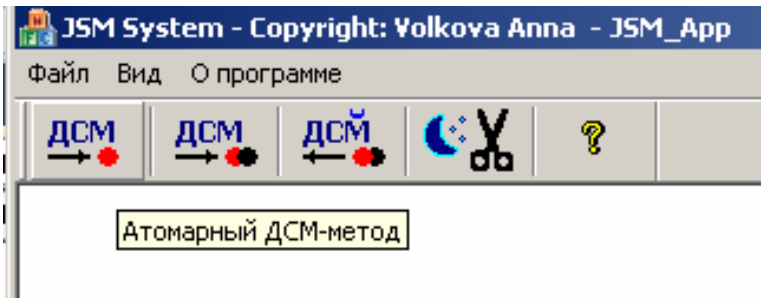

Рис. 4. Панель инструментов. Выбор атомарного ДСМ-метода.

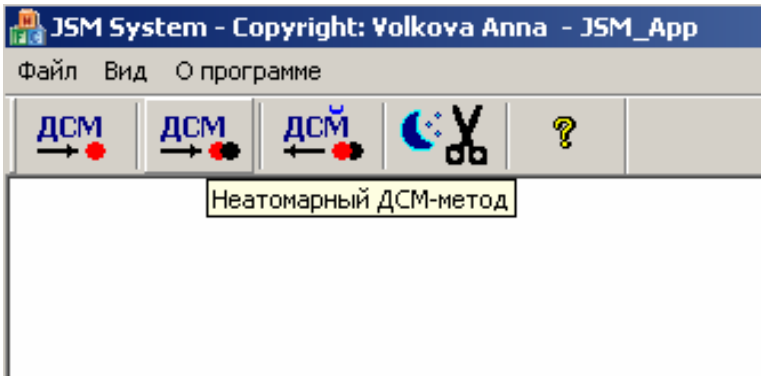

Рис. 5. Панель инструментов. Выбор неатомарного ДСМ-метода.

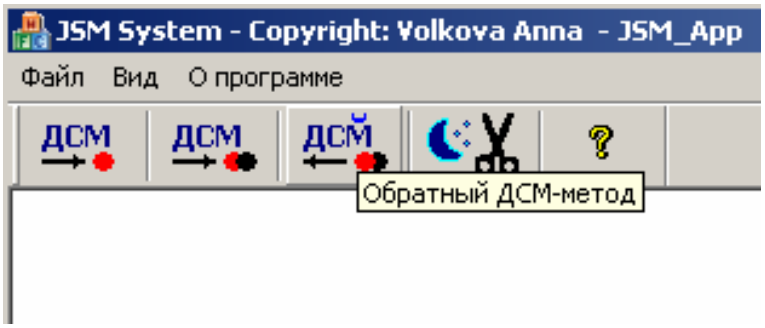

Рис. 6. Панель инструментов. Выбор обратного ДСМ-метода.

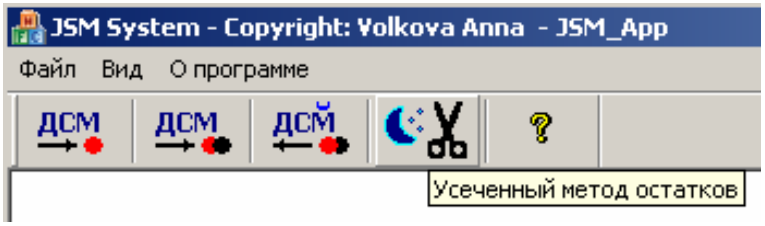

Рис. 7. Панель инструментов. Выбор режима "Усеченный метод остатков".

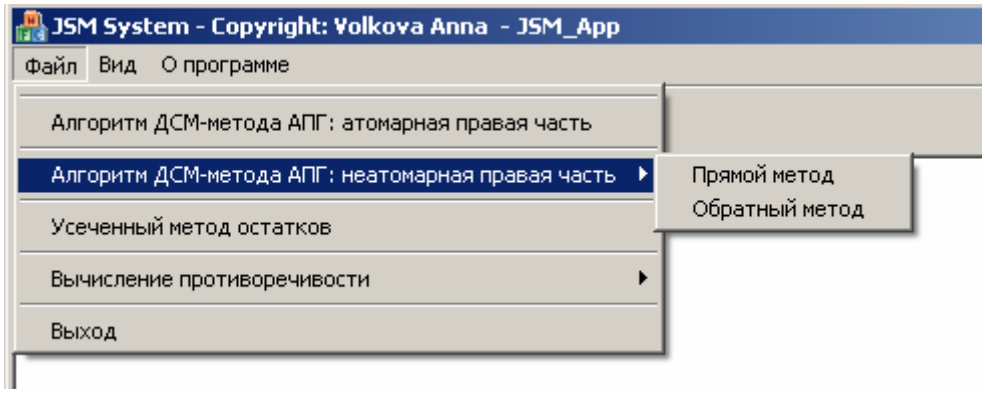

Рис. 8. Главное меню приложения. Выбор версии алгоритма.

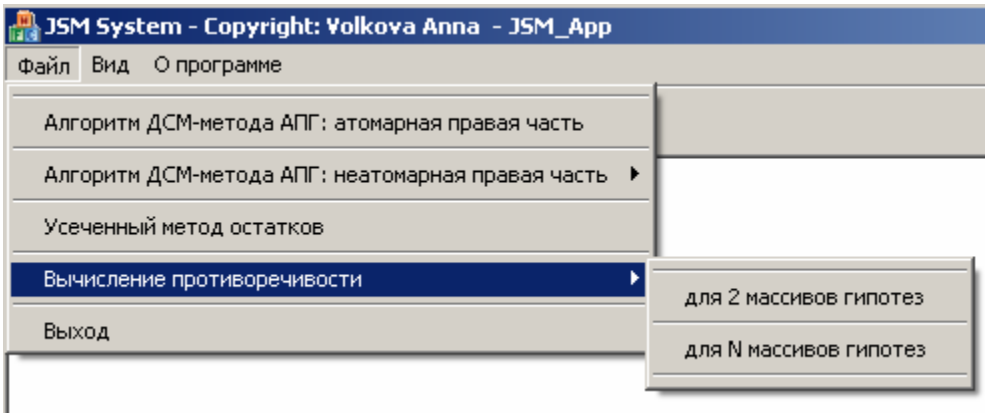

Рис. 9. Главное меню приложения. Выбор режима «Вычисление противоречивости»

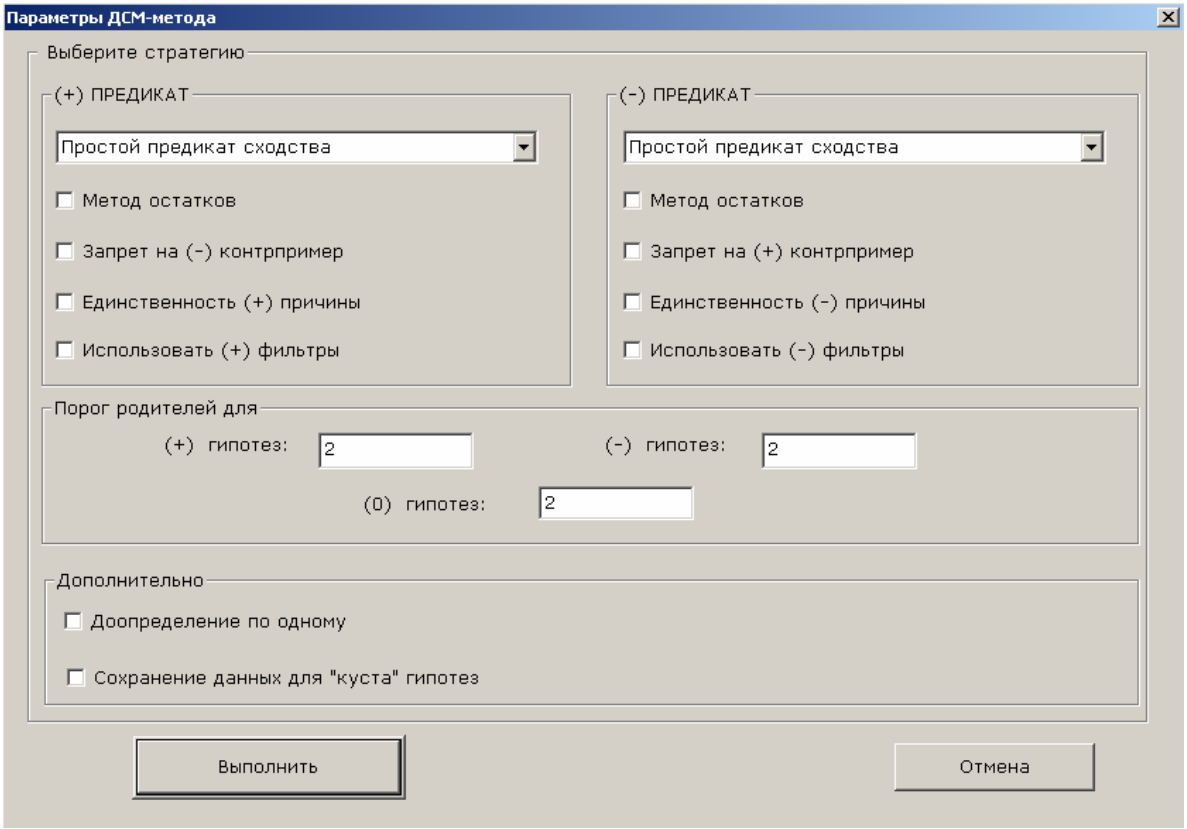

Рис. 10. Диалоговое окно настройки параметров ДСМ-метода

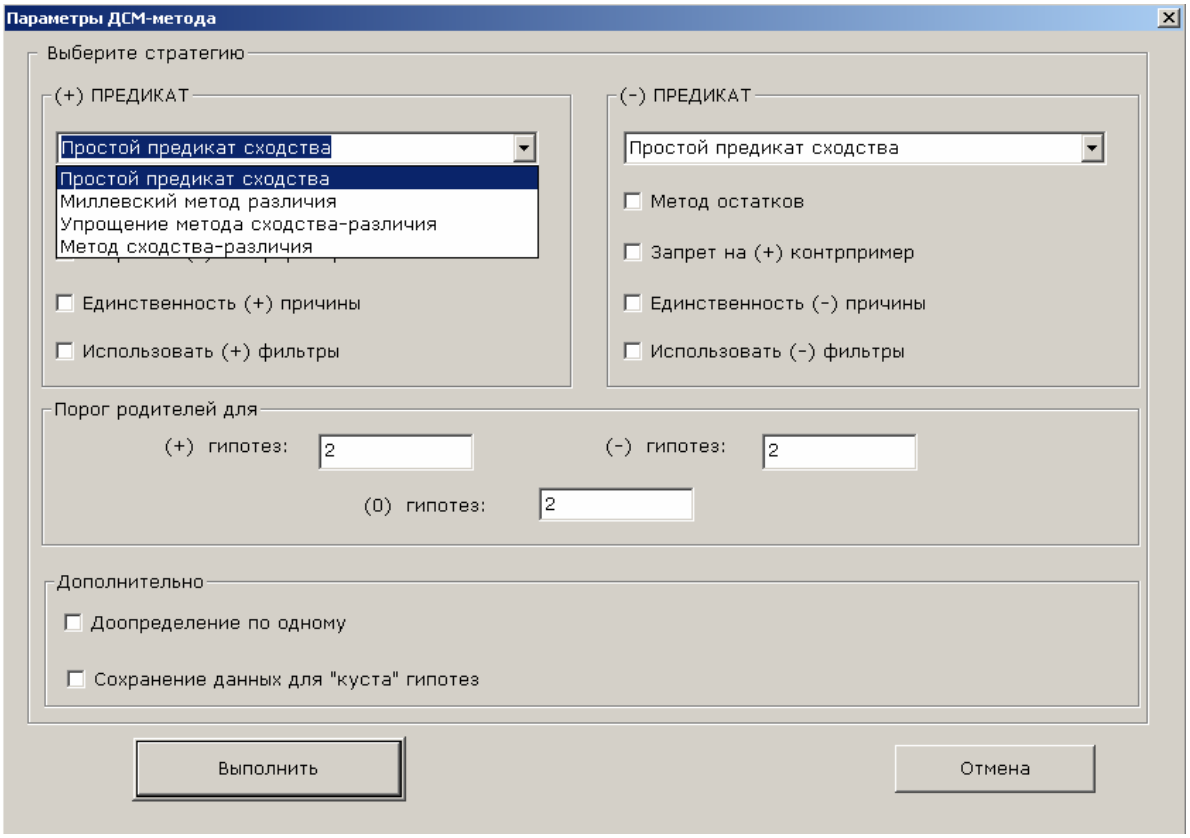

Рис. 11. Диалоговое окно настройки параметров ДСМ-метода. Список выбора допустимых стратегий.

В ходе работы с приложением пользователю выдаются необходимые сообщения, в том числе сообщения об ошибках в случае неудачного завершения процедур ДСМ-Решателя или считывания/записи данных.

В целом, стоит отметить, что основная сложность при работе с системой состоит в заполнении входных данных. Далее пользователю остается только выбрать версию ДСМ-метода или дополнительные процедуры (усеченный метод остатов/ вычисление противоречивости), выбрать файлы с данными для обработки и получить результат. Пользователь должен быть знаком с ДСМ-методом, понимать, подходят ли имеющиеся у него данные для обработки системой, знать структуру заполняемых файлов, а также понимать специфику процедур ДСМ-метода при заполнении параметров экспериментов.

Несмотря на то, что система универсальна для разных предметных областей, очевидно, что её недостаток сказывается в том, что при анализе полученных данных эксперту не хватает удобного интерфейса для просмотра результатов. Этот недостаток может устраняться только путём создания дополнительных программ уже для конкретной предметной области.

5, ПРАКТИКА ПРИМЕНЕНИЯ СИСТЕМЫ

На первом этапе созданная ДСМ-система применялась при обработке данных фармакологии и медицинской диагностики (диагностика двух заболеваний глаз: дегенеративного ретиношизиса и наследственных витреоретинальных дистрофий [22] и прогнозирование продолжительности жизни больных меланомой и оценка прогностического биохимического маркера - протеина S100).

В результате анализа исследуемых баз фактов (массив фармакологических соединений, данные пациентов с болезнями глаз и данные больных меланомой) были получены следующие важные результаты:

- при применении миллевского варианта метода различия не было получено ни одной гипотезы ни на одной из баз фактов;
- с помощью метода различия – упрощенного метода соединенного сходства-различия – гипотезы были порождены на всех массивах данных;
- при применении метода соединенного сходства-различия были получены (+)-гипотезы на фармакологических данных;
- с помощью усеченного метода остатков была подтверждена важная причинно-следственная связь данных, взятых из сведений о больных меланомой между протеином S100 и продолжительностью жизни больного: при значении уровня S100 меньше 0,12 нг/мл продолжительность жизни больного – больше пяти лет, при значении уровня S100 больше 0,12 нг/мл – меньше пяти лет. Впервые эта связь была обнаружена Д.А. Добрыниным и Е.С.Панкратовой в результате компьютерных экспериментов с использованием ДСМ-системы, созданной в ВИНИТИ РАН [20, 23];
- стратегия упрощенного метода соединенного сходства-различия с запретом на (±)-контрпример

подтвердила гипотезы, полученные стандартной стратегий запрета на (±)-контрпример, на данных больных меланомой. Эти гипотезы доказывают наличие указанной связи между белком S100 и продолжительностью жизни больного меланомой. При этом процедура «доопределение по одному», используемая как критерий оценки подбора параметров и стратегии эксперимента, показала хороший результат.

Важно отметить, что на всех рассматриваемых базах фактов наилучшей стратегией по результатам процедуры «доопределение по одному» и проверки степеней абдуктивной сходимости в целом можно считать упрощенный метод соединенного сходства-различия с запретом на (±)-контрпример.

Полученные выше результаты были описаны в [10, 21].

Исследования данных больных меланомой были продолжены в 2011-2012 гг. [24, 25]. Наряду с белком S100 были изучены закономерности, включающие показатели CD44, IL-12, IL-02, IFNy. Основные результаты были получены в результате экспериментов, проводимых Е.С. Панкратовой, Д.А. Добрыниным и О.П.. Шестерниковой. Настройка системы на эксперимент включала выбор стратегии (простой метод сходства, метод сходства с запретом на контрпримеры), подбор нужного количества родителей гипотезы, настройка фильтров для конъюнктивных и дизъюнктивных признаков. В качестве инструмента для подтверждения и проверки найденных гипотез использовалась и программа, представленная в настоящей статье. Наряду с уже использованными стратегиями проводились эксперименты с применением методов сходства-различия, упрощенного и соединенного, в сочетании с фильтрами для признаков. Закономерности, выявленные на разных системах, совпали.

Так как ДСМ-система является расширяемой, было принято решение реализовать возможность обработки социологических данных (ноябрь 2011 года). Так в модуле представления данных были созданы классы, необходимые для описания респондентов, а в модуле ДСМ-Решателя реализованы процедуры обратного ДСМ-метода.

Разработка указанных средств позволила применить систему для исследования данных конкретных социологических задач, таких как сравнения стран по уровню одобрения населением протестного поведения и анализ трудовых отношений на российских предприятиях.

В рамках диссертационного исследования А.В. Кученковой [26] при изучении протестного поведения одним из используемых логико-комбинаторных методов стал ДСМ-метод АПГ. В качестве эмпирической базы были взяты данные исследования 2006 года «Роль правительства в государстве», проведенного в рамках Международной программы социальных исследований [см. также 27].

Эксперименты проводились в январе 2012 г. с двумя выборками данных по 32 странам. Объект представлял собой страну, описанную рядом характеристик – детерминант протестного поведения. (+)-примеры – это страны с высоким уровнем одобрения права на акции политического протеста, (–)-примеры – с низким. В первой базе для каждой страны было определено пять детерминант (показатели, определяющие специфику политической культуры в стране [26]), во второй – 59.

При помощи системы были проведены эксперименты прямым атомарным ДСМ-методом с запретом на контрпримеры.

А.В. Кученковой были переданы обработанные данные, а также одна из первых версий программы для просмотра полученных результатов для социологии, созданная в ходе решения второй рассматриваемой социологической задачи. В результате анализа полученных гипотез эксперту удалось выявить классы стран («типологические синдромы»») отдельно для стран с высоким и отдельно для стран с низким уровнем одобрения права на протестное поведение.

Другой важной задачей стал анализ анкет социологического опроса на материале анкетирования работников промышленных предприятий.

В ходе исследование в рамках проекта 10–06–00033 «Методология и методы ситуационного анализа на предприятии», финансируемого РФФИ[1], Самарским Государственным Университетом и Институтом Социологии РАН был проведен социологический опрос работников самарских предприятий. Полученные данные были переданы для компьютерной обработки, в процессе которой применялся и ДСМ-метод АПГ. Исследовательская задача и полученные результаты подробно описаны в [28]. «Содержательно задача формулировалась как выяснение вопросов: кто, какие работники и в какой ситуации готовы отстаивать свои трудовые права и кто, в какой ситуации решает отказаться от их защиты» [28, с. 6] .

Данные были обработаны с помощью описываемой ДСМ-системы, и полученные гипотезы были проанализированы экспертами. Для удобства была также создана отдельная программа – интерфейс для просмотра результатов. Обработка данных и создание окончательной версии программы-«вьюера» результатов проводились в период ноябрь 2011 г. – сентябрь 2012 г. Описание программы представлено в [19].

Результаты исследования показали, что существует острая необходимость предоставить социологу-эксперту возможность не только оценивать уже полученные результаты, но и самостоятельно проводить эксперименты, используя ДСМ-метод АПГ. К сожалению, на период изучения данных общей системы, которая могла бы предоставить такие возможности, не существовало. Было принято решение разработать такую систему в виде компьютерного приложения, интегрировав в него созданный ранее интерфейс для просмотра результатов.

Итогом работы стала компьютерная интеллектуальная система для качественного анализа социологических данных (JSM-Socio), реализующая стратегии ДСМ-метода АПГ. Система была официально зарегистрирована Федеральной службой по интеллектуальной собственности (РОСПАТЕНТ) и внесена

[1] Руководитель проекта – С. Г. Климова. Участники проекта: Н.В. Авдошина, В.Ю. Бочаров, И.А. Климов, М.А. Михееенкова, Б.Г. Тукумцев, В.К. Финн, В.А. Ядов.

в Реестр программ для ЭВМ (№ 2013614978, от 24 мая 2013 г.).

Система представляет собой практический инструмент для социолога-эксперта и позволяет исследовать данные социологического опроса посредством формализованного качественного анализа.

Система предназначена для решения следующих задач:

1. Качественный анализ анкет социологического опроса с использованием стратегий ДСМ-метода АПГ:
   - ✓ извлечение причинно-следственных зависимостей, содержащихся в данных;
   - ✓ прогнозирование значения изучаемого свойства;
   - ✓ оценка полученных гипотез посредством процедуры объяснения исходного множества фактов.
2. Представление данных опроса в форме, удобной для исследования.

Система включает следующие функциональные возможности:

Предобработка данных: перекодировка данных, перевод в другой формат (из файлов Statistical Package for the Social Sciences (SPSS) формата .sav в формат Microsoft Office Excel).

Возможность классификации признаков, описывающих респондента, экспертом:

признаки, соответствующие описанию объекта,

признаки, описывающие мнение,

признаки, описывающие ситуацию.

Выделение целевого признака (свойства для изучения) и автоматическое разделение респондентов на группы в соответствии с выделенным свойством .

Введение дополнительных ограничений на данные в виде фильтров (выборка признаков, которые требуется исследовать).

Выбор используемой стратегии анализа данных.

Реализация стратегий ДСМ-метода АПГ: простой метод сходства, запрет на (±) контрпример; атомарная/неатомарная, прямая/обратная версии; простой ДСМ-метод /ситуационный ДСМ-метод АПГ.

Выдача результата, который представляет собой гипотезы, полученные различными стратегиями, и прогнозирование значения целевого свойства у респондентов, для которых это значение изначально не определено.

Представление полученных результатов: представление данных респондентов и гипотез, удобное для анализа экспертом (таблица / «дерево»).

Система состоит из четырех функциональных модулей: модуль предобработки данных, интерфейс пользователя подготовки данных для эксперимента, ДСМ-Решатель и интерфейс пользователя для просмотра полученных результатов.

Социолог-эксперт регистрирует полученные в результате опроса данные, используя программу Statistical Package for the Social Sciences (SPSS), в файле формата .sav.

Графический интерфейс пользователя подготовки данных для эксперимента позволяет загрузить файл формата .sav и вызвать модуль предобработки данных.

Модуль предобработки данных переводит файл формата .sav в файл формата Microsoft Office Excel, который содержит все необходимые поля для хранения информации о параметрах эксперимента.

Модуль, реализующий обработку входных данных, создан на языке программирования C++ с использованием интерфейса для доступа и манипулирования внешними данными ADO и языка запросов SQL.

Для извлечения данных из файлов формата .sav используется библиотека, бесплатно поставляемая на сайте продукта Statistical Package for the Social Sciences (SPSS Statistics Input-Output Module).

Модуль, реализующий стратегии ДСМ-метода АПГ, был выделен из описываемой «универсальной» ДСМ-системы. На его основе было создано консольное приложение, которое вызывается из Web-интерфейса.

Приложения для настройки параметров эксперимента и представления полученных результатов созданы с использованием Web-технологий (HTML, CSS, язык программирования JavaScript), а также технологии для доступа и манипулирования внешними данными ADO и языка запросов SQL.

ЗАКЛЮЧЕНИЕ

Представленная в статье система обладает многофункциональным ДСМ-Решателем, который позволяет проводить интеллектуальный анализ данных с помощью различных процедур ДСМ-метода применительно к различным предметным областям – фармакологии, медицине и социологии.

Как уже было отмечено, такая «гибридная» система была создана впервые.

Все классы программы четко разделены на три модуля: «Представление данных», «ДСМ-Решатель» и «Графический интерфейс пользователя». Такая архитектура программы позволяет свободно добавлять реализацию новых алгоритмов, а также представления других предметных областей.

Система лишена развитого предметно-ориентированного пользовательского интерфейса, что явилось следствием динамичной структуры программы и стремлением к «универсальности». Однако, как было показано на примере исследования социологических данных, при необходимости модули и субмодули программы отчуждаемы и могут быть встроены в систему для конкретной предметной области.

СПИСОК ЛИТЕРАТУРЫ

1. Автоматическое порождение гипотез в интеллектуальных системах / под общ. ред. В. К. Финна. – М.: Книжный дом «Либроком», 2009.
2. ДСМ-метод автоматического порождения гипотез: логические и эпистемологические основания / под общ. ред. О. М. Аншакова. – М.: Книжный дом «Либроком», 2009; гл. 4 переведена на англ. яз.: Finn V. K. The Synthesis of Cognitive Procedures and the Problem of Induction // Automatic Documentation and Mathematical Linguistics. – 2009. – Vol. 43, № 3. – P. 149-195.

3. Финн В.К. Правдоподобные рассуждения в интеллектуальных системах типа ДСМ // Итоги науки и техники. Сер. Информатика. Т. 15 . – М.: ВИНИТИ, 1991. – С.54-101.
4. Финн В.К., Блинова В.Г., Панкратова Е.С., Фабрикантова Е.Ф. Интеллектуальные системы для анализа медицинских данных // Врач и информационные технологии. Ч.1 – 2006. – № 5. – С. 62-70; Ч.2. – 2006. – № 6. – С. 50-60; Ч.3 – 2007. – №1. – С. 51-57.
5. Финн В.К. Интеллектуальные системы и общество. – М.: РГГУ, 2001. – С. 252-272.
6. Михеенкова М.А. О логических средствах интеллектуального анализа социологических данных // Искусственный интеллект и принятие решений. – 2010. – № 1. – С. 20 – 32.
7. Климова С.Г., Михеенкова М.А., Панкратов Д.В. ДСМ-метод как метод выявления детерминант социального поведения // Научно-техническая информация. Сер. 2. – 1999. – № 12. – С. 3-14.
8. Волкова Т.А., Добрынин Д.А. Сравнение системы нечеткого вывода и обучаемой ДСМ-системы при планировании движения мобильного робота. // V-я Международная научно-практическая конференция "Интегрированные модели и мягкие вычисления в искусственном интеллекте" (28-30 мая 2009) Сб. научных трудов. В 2-т. – М: Физматлит, 2009. – Т.1. – С. 473-481.
9. Добрынин Д.А., Карпов В.Э. Моделирование некоторых форм адаптивного поведения интеллектуальных роботов // Информационные технологии и вычислительные системы. – 2006. – № 2. – С.45-56
10. Волкова А.Ю. Анализ данных различных предметных областей с помощью процедур ДСМ-метода автоматического порождения гипотез // Научно-техническая информация. Сер. 2. – 2011. – № 6. – С. 9-18; Volkova A. Yu. Analyzing the Data of Different Subject Fields Using the Procedures of the JSM Method for Automatic Hypothesis Generation // Automatic Documentation and Mathematical Linguistics. – 2011. – Vol. 45, № 3. – P. 127-139.
11. Финн В.К. Индуктивные методы Д.С. Милля в системах искусственного интеллекта // Часть I. Искусственный интеллект и принятие решений. – 2010. – № 3. – С. 3- 21; Часть II. Искусственный интеллект и принятие решений. – 2010. – № 4. – С. 14-40.
12. Финн В. К. Индуктивный метод соединенного сходства – различия и процедурная семантика ДСМ-метода // Научно-техническая информация. Сер. 2. – 2010. – № 4. – С.1-17.
13. Финн В.К. Своевременные замечания о ДСМ-методе автоматического порождения гипотез // Научно-техническая информация. Сер. 2. – 2009. – № 8. – С. 15-26; Finn V. K. Timely Notes about the JSM Method for Automatic Hypothesis Generation // Automatic Documentation and Mathematical Linguistics. – 2009. – Vol. 43, № 5. – P. 257-269.
14. Гусакова С.М., Михеенкова М.А., Финн В.К. О логических средствах автоматизированного анализа мнений // Научно-техническая информация. Сер. 2. – 2001. – № 5. – С. 4-24.
15. Финн В.К., Михеенкова М.А. О логических средствах концептуализации анализа мнений // Научно-техническая информация. Сер. 2. – 2002. – № 6. – С. 4-22.
16. Страуструп Б. Язык программирования С++. – М.: Издательство Бином, 2012. – 1136 с.
17. Волкова А.Ю. Создание программных средств анализа базы фактов для применения ДСМ-метода автоматического порождения гипотез: дипломная работа / науч. рук. В.К.Финн. – М.: Ин-т лингвистики Рос. гос. гуманитарного ун-та, 2010.
18. Волкова А.Ю. Алгоритмизация процедур ДСМ-метода автоматического порождения гипотез // Научно-техническая информация. Сер. 2. – 2011. – № 5 – С. 6-12; Volkova A. Yu. Algorithmization of Procedures of the JSM Method for Automatic Hypothesis Generation // Automatic Documentation and Mathematical Linguistics. – 2011. – Vol. 45, № 3. – P. 113-120.
19. Михеенкова М.А., Волкова А.Ю. Спецификация интеллектуальной системы типа ДСМ // Научно-техническая информация. Сер. 2. – 2013. – № 7. – С. 5-19.
20. Финн В.К. Об определении эмпирических закономерностей посредством ДСМ- метода автоматического порождения гипотез // Искусственный интеллект и принятие решений. – 2010. – № 4. – С. 41 – 48.
21. Волкова А.Ю., Шестерникова О.П. О создании интеллектуальных систем, реализующих ДСМ-метод автоматического порождения гипотез, и результатах их применения для анализа медицинских данных // Научно-техническая информация. Сер. 2. – 2012. – № 5. – С. 10-15.
22. Панкратова Е.С., Добрынин Д.А., Цапенко И.В., Зуева М.В. Интеллектуальная ДСМ-система для диагностики заболеваний органа зрения на примере дегенеративных и наследственных форм ретиношизиса // Научно-техническая информация. Сер. 2. – 2007. – № 3. – С. 14-18.
23. Михайлова И.Н., Панкратова Е.С., Добрынин Д.А., Самойленко И.В., Решетникова В.В., Шелепова В.М., Демидов Л.В., Барышников А.Ю., Финн В.К. О применении интеллектуальной компьютерной системы для анализа клинических данных больных меланомой // Российский терапевтический журнал. – 2010. – Т. 9, № 2 – С.54.
24. Панкратова Е.С. Интеллектуальная система типа ДСМ для анализа клинических данных // Научно-техническая информация. Сер 2. – 2011. – № 4. – С. 8-16.

25. Панкратова Е.С., Добрынин Д.А., Михайлова И.Н. Интеллектуальная компьютерная система для анализа клинических данных больных меланомой / Тринадцатая национальная конференция по искусственному интеллекту с международным участием КИИ-2012 (16-20 октября 2012 г., г. Белгород, Россия): Труды конференции в 3 томах. Белгород: Изд-во БГТУ, 2012. – Т. 1. – С. 128–134.

26. Кученкова А.В. Логико-комбинаторные методы анализа социологических данных: эвристический потенциал и методическая специфика : диссертация на соискание ученой степени кандидата социологических наук по специальности 22.00.01 – теория, методология и история социологии / Минобрнауки России, Федер. гос. бюджет. образоват. учреждение высш. проф. образования "Рос. гос. гуманитарный ун-т"; науч. рук. Г. Г. Татарова. – М., 2012.

27. Кученкова А.В. Логико-комбинаторные методы анализа социологических данных: эвристический потенциал и методическая специфика : Автореферат диссертации на соискание ученой степени кандидата социологических наук по специальности 22.00.01 – теория, методология и история социологии / Минобрнауки России, Федер. гос. бюджет. образоват. учреждение высш. проф. образования "Рос. гос. гуманитарный ун-т"; науч. рук. Г. Г. Татарова. – М., 2012.

28. Климова С.Г., Михеенкова М.А. Формальные средства ситуационного анализа: опыт применения // Научно-техническая информация. Сер. 2. – 2012. – № 10. – С. 1-13.

Материал поступил в редакцию 03.07.13.

Сведения об авторе

ВОЛКОВА Анна Юрьевна – аспирант Российского Государственного Гуманитарного Университета, Москва
e-mail: anna-.-volkova@mail.ru

ИНФОРМАЦИОННЫЕ ЯЗЫКИ

УДК 1 : 004.42-027.21

В.В. Грибова, А.С. Клещев

Онтологическая парадигма программирования*

Описана новая парадигма программирования – онтологическая, которая является дальнейшим развитием парадигмы декларативного программирования, а также описана модель данных, лежащая в основе парадигмы, основные конструкции языка, приведены примеры.

Ключевые слова: *парадигмы программирования, модель данных, семантические сети, онтология*

ВВЕДЕНИЕ

Трудоемкость разработки программных систем сопоставима со сложностью разработки крупных материальных объектов, а порой и превышает ее [1]. При этом сопровождение таких систем является еще более дорогим процессом [2], что требует от исследователей поиска новых подходов, направленных на решение этих проблем. В качестве одного из таких подходов была предложена парадигма декларативного программирования и разработанные в ее рамках языки декларативного программирования.

Программа на языке декларативного программирования в общем случае представляет собой описание абстрактной модели решаемой задачи или, согласно [3], исполнимую спецификацию результата вычислений по программе. Программисту не требуется описывать процесс управления вычислениями, он полностью возлагается на процессор языка. Среди множества преимуществ парадигмы декларативного программирования в качестве основных называются: более простое написание программы, более легкое ее понимание по сравнению с соответствующими императивными программами, упрощение ее сопровождения и модификации [4].

К языкам декларативного программирования относят, как правило, языки функционального и логического программирования. Однако, как отмечено в [4], провозглашенная идея декларативного программирования в современных языках логического и функционального программирования в полной мере не реализована; в результате практические программы на Прологе или Лиспе ненамного легче разрабатывать, понимать и сопровождать, чем программы на императивных языках. Среди других недостатков языков декларативного программирования называют скудные средства взаимодействия с пользователем и сложность внесения в программы императивных операторов, если их использование более удобно по сравнению с декларативными конструкциями.

Цель данной работы – предложить новую парадигму программирования, названную парадигмой онтологического программирования, как дальнейшее развитие парадигмы декларативного программирования, в рамках которой программа и результат ее выполнения связаны между собой более непосредственно, чем в парадигмах императивного, функционального и логического программирования, а также предложить визуальный язык программирования OPL (Ontological Programming Language), удовлетворяющий определению декларативного программирования, в котором программа является онтологией результата вычислений; для этого языка также вводятся механизмы организации интерфейса с пользователем и включения в язык императивных конструкций.

1. ПОСТАНОВКА ЗАДАЧИ

При решении любой задачи на компьютере результат может быть получен лишь как итог некоторого процесса вычислений, который мы будем называть процессом получения результата. Сложность написания программы для решения некоторой задачи в значительной степени состоит в том, что разработчик этой программы должен представить себе все множество процессов получения результатов решения задачи (экстенсионал задачи) при различных допустимых исходных данных и определить это множество средствами языка программирования в виде программы [5]. Сложность понимания программы сопровождающим программистом в значительной степени состоит в том, что он должен по программе представить себе экстенсионал задачи и понять, почему процессы получения результата ведут именно к результату решения этой задачи. Сложность модификации программы в значительной степени состоит в том, что сопровождающий программист, понимая, как должны измениться результаты решения задачи, должен понять, как должны измениться процессы

* Работа выполнена при финансовой поддержке ДВО РАН в рамках Программы ОНИТ РАН (проект 12-I-ОНИТ-04) и РФФИ (проект 12-07-00179)

получения результатов, чтобы они приводили именно к этим изменениям результатов, а затем понять, как эти изменения в процессах получения результатов могут быть выражены через изменения в программе. Рассмотрим с этой точки зрения, как связаны процессы получения результата и программы с результатами вычислений в парадигмах императивного, функционального и логического программирования.

В основе парадигмы императивного программирования лежат вычислительные модели [6, 7]. В этих моделях процесс получения результата представляет собой последовательность состояний, первое из которых получается из исходных данных с помощью входной процедуры, каждое следующее состояние получается из предыдущего с помощью оператора непосредственной переработки, а из последнего состояния извлекается результат вычислений с помощью выходной процедуры. Все состояния вычислительного процесса, кроме последнего, лишь косвенно связаны с результатом вычислений (решения задачи). В современных императивных языках состоянием вычислительного процесса является множество значений переменных, а оператором непосредственной переработки – оператор присваивания, с помощью которого следующее состояние процесса получается изменением значения одной переменной, а все остальные переменные сохраняют свои значения. В программе на императивном языке последовательности выполнения операторов присваивания определяются с помощью линейных участков, условных операторов и операторов цикла (а также вызовов процедур).

В основе парадигмы функционального программирования лежит ламбда-исчисление [6, 7]. Процесс получения результата может быть представлен ориентированной размеченной сетью вызова функций, в которой метками терминальных вершин являются исходные данные, а метками нетерминальных вершин – значения некоторой функции, аргументами которой являются метки концов дуг, выходящих из этой вершины (ориентация дуг – от результата к аргументу). Результатом вычислений является метка корня сети (вершины, в которую не входит ни одна дуга). Все промежуточные значения (метки нетерминальных вершин), кроме метки корня, лишь косвенным образом связаны с результатом вычислений. В современных функциональных языках имеются некоторые наборы базовых функций, а также средства определения (конструирования) новых функций из базовых и уже определенных (в том числе условные термы и рекурсия).

В основе парадигмы логического программирования лежит исчисление предикатов первого порядка [6, 7]. Процесс получения результата может быть представлен ориентированной размеченной сетью вывода результата, в которой метками терминальных вершин являются кортежи отношений, представляющие исходные данные, а метками нетерминальных вершин – кортежи отношений, являющиеся результатом применения некоторого правила к посылкам, которые суть метки концов дуг, выходящих из этой вершины (ориентация дуг – от заключения к посылкам). Результатом вычислений (вывода) является метка корня сети. Все промежуточные значения (метки нетерминальных вершин), кроме метки корня, лишь косвенным образом связаны с результатом вычислений. В современных логических языках программа представляет собой некоторый запрос и множество правил (импликаций) и фактов (кортежей отношений).

Таким образом, в каждой из трех рассмотренных парадигм результат вычислений получается лишь на последнем шаге процесса получения результата, а все остальные шаги этого процесса имеют к результату вычислений лишь косвенное отношение, на них получаются лишь некоторые промежуточные значения. Такие процессы получения результата можно назвать косвенными по отношению к результату.

Следовательно, для снижения сложности разработки программ, их понимания и модификации необходимо предложить такую парадигму программирования, в которой процессы получения структурных результатов вычислений, представляемых в виде ориентированных графов, были бы прямыми, т.е. состояли из шагов, на каждом из которых формируется некоторая часть результата. В этом случае сам процесс получения результата может быть представлен в виде графа, совпадающего с этим результатом, а значит, программа на языке является (исполнимой) спецификацией множества результатов вычислений (а не множества косвенных процессов их получения). Разрабатывая такую программу, программист должен лишь специфицировать множество результатов вычислений; анализируя такую программу, программист должен представить себе лишь множество результатов вычислений, специфицируемых этой программой; модифицируя такую программу, программист должен лишь понять, как следует изменить спецификацию множества результатов вычислений, чтобы получить требуемые изменения этих результатов, причем каждое такое изменение в программе является локальным, связанным с изменением соответствующих частей результатов вычислений. Такая спецификация множества результатов вычислений, в соответствии с современными представлениями в области искусственного интеллекта, называется онтологией результата вычислений. Учитывая это, назовем предлагаемую парадигму ***парадигмой онтологического программирования***. Ее можно рассматривать в качестве более полного воплощения парадигмы декларативного программирования по сравнению с парадигмами функционального и логического программирования.

2. МОДЕЛЬ ДАННЫХ, ЛЕЖАЩАЯ В ОСНОВЕ ПАРАДИГМЫ

Одно из положений парадигмы онтологического программирования состоит в том, что все данные имеют вид семантических сетей. Для определенности рассматриваются иерархические семантические сети (что не является принципиальным), определяемые следующим образом. Иерархическая семантическая сеть есть связный ориентированный граф без циклов, в котором каждая дуга имеет метку, а вершины могут

быть одного из двух типов – простые и структурные; вершина сети, в которую не входит ни одна дуга, называется ее корнем. Каждая простая терминальная вершина сети имеет в качестве метки константу некоторого сорта. Корень сети имеет две метки-термина, из которых первая метка есть метка класса (имя функции, в результате выполнения которой была создана эта семантическая сеть), а вторая – индивидуальная метка этой сети. Каждая структурная вершина сети является контейнером, содержащим упорядоченное конечное множество иерархических семантических сетей с одной и той же меткой класса и попарно различными индивидуальными метками.

Семантическая сеть может быть как сетью постоянного хранения (хранимой независимо от приложения) – для обработки всеми приложениями, реализованными в рамках парадигмы онтологического программирования, так и временной сетью, создаваемой при запуске приложения и прекращающей свое существование по окончании этого запуска.

Использование семантических сетей в качестве модели данных означает, что обрабатываются только целостные информационные структуры, которые выделяются в качестве объектов обработки, в отличие от других парадигм, поддерживающих обработку информационных объектов хотя и произвольной структуры, но не обязательно объединенных в целостную информационную структуру (как правило, целостные информационные структуры разделяются на различные фрагменты, являющиеся значениями различных переменных, связь между которыми известна и понятна только разработчику программы). Например, в экспертной системе медицинской диагностики такими целостными информационными структурами являются история болезни, база знаний и объяснение, представленные в форме семантических сетей.

3. ПРИЛОЖЕНИЕ

В общем случае приложение имеет одну или несколько выходных семантических сетей (результатов), может иметь (или не иметь) одну или несколько входных семантических сетей (входных данных), а также иметь (или не иметь) промежуточные семантические сети (промежуточные результаты). Каждая промежуточная или выходная семантическая сеть приложения вычисляется с помощью функции (название которой является меткой класса корня этой семантической сети).

Каждая функция может иметь (или не иметь) входные семантические сети, в точности одну выходную семантическую сеть и не может иметь промежуточных семантических сетей.

Входные семантические сети как приложения, так и функций, могут быть постоянного хранения (созданными и существующими независимо от приложения), так и временными, созданными во время запуска приложения (как результат одной из его функций). Соответственно, выходные семантические сети как приложения, так и функций могут быть как постоянного хранения, так и временными.

Между всеми семантическими сетями приложения (входными, промежуточными и выходными) задается частичный порядок, в котором вычисляются промежуточные или выходные семантические сети как результаты соответствующих функций. Поэтому программу приложения можно рассматривать как суперпозицию этих функций, связанных с вычислением всех семантических сетей приложения, не являющихся входными семантическими сетями постоянного хранения.

Таким образом, выполнение приложения состоит в вычислении выходных семантических сетей по входным семантическим сетям постоянного хранения, созданным до запуска приложения (их может и не быть) через промежуточные семантические сети, создаваемые при запуске приложения.

Каждая функция приложения представляет собой онтологию (описание структуры) выходной семантической сети этой функции на языке онтологического программирования OPL (Ontological Programming Language).

4. ЯЗЫК ОНТОЛОГИЧЕСКОГО ПРОГРАММИРОВАНИЯ

Язык OPL является визуальным логическим языком программирования. Описание функции – онтология выходной семантической сети – представляет собой ориентированный граф с размеченными вершинами и дугами. Метками дуг являются термины, которые в ходе выполнения программы становятся метками дуг выходной семантической сети функции. Метками вершин являются логические формулы.

Процесс вычисления выходной семантической сети функции сводится к построению этой сети, начиная с корня. В процессе построения семантической сети устанавливается однозначное соответствие между вершинами и дугами этой сети и вершинами и дугами ее онтологии (описания функции). Выходная семантическая сеть, в зависимости от описания функции, может формироваться как автоматически, без участия пользователя, так и интерактивно, с участием пользователя, который в процессе диалога управляет процессом построения этой сети.

Логическими формулами языка являются: простая формула, простая кванторная формула, унарная формула, пропозициональная формула, структурная кванторная формула, множество импликаций. Синтаксис логических формул подробно описан в работах [8, 9].

Операционная семантика каждой логической формулы включает команду построения семантической сети и, возможно, команду абстрактного интерфейса.

Командами построения семантической сети являются:

- команда построения простых терминальных вершин графа (входит в операционную семантику простых формул),
- команда построения структурных вершин графа (входит в операционную семантику всех кванторных формул)
- команда построения множества дуг (входит в операционную семантику всех пропозициональных формул).

Специфическим оператором языка, расширяющим множество логических функций, является импликация. Она предназначена для проверки условий при формировании отдельных фрагментов семантической сети, при этом антецедент и консеквент импликации допускают использование переменных. С точки зрения операционной семантики импликация представляет собой поиск по образцу (антецеденту, являющемуся логической формулой), при котором переменные получают значения, обеспечивающие истинность антецедента.

Для организации сложных эффективных вычислений средств логического языка может оказаться недостаточно. Для решения этой проблемы язык OPL дополнен вычисляемыми предикатами. Вычисляемые предикаты предназначены для описания вычислительных алгоритмов, описание которых средствами декларативного логического языка слишком громоздко. Сами вычисляемые предикаты описываются средствами императивного языка в соответствующей среде создания приложений. Вычисляемый предикат имеет имя и множество формальных параметров. Вызов вычисляемого предиката с фактическими параметрами может быть лишь компонентой антецедента импликации. При поиске по образцу, когда в антецеденте импликации встречается вызов вычисляемого предиката, происходит его выполнение: в тело предиката подставляются значения, указанные в его аргументах, и производятся вычисления.

Команды абстрактного интерфейса определяют функциональность пользовательского интерфейса. Их можно разделить на четыре класса:

- команда вывода (процессором языка) сформированного фрагмента семантической сети (выполняется при построении семантической сети в интерактивном режиме или по запросу пользователя);

- команды непосредственного ввода с клавиатуры одного значения заданного типа и изменения этого значения (входят в операционную семантику кванторных формул с квантором существования и единственности, если множество, по которому пробегает квантор, бесконечно);

- команды непосредственного ввода с клавиатуры нескольких значений заданного типа и изменения этих значений (входят в операционную семантику кванторных формул с различными вариантами квантора существования, если множество, по которому пробегает квантор, бесконечно);

- команды выбора значения либо подмножества значений из предлагаемого множества и изменения этого выбора (входят в операционную семантику дизъюнкции и кванторных формул с различными вариантами квантора существования, если множество, по которому пробегает квантор, конечно).

Команды абстрактного интерфейса ввода и выбора значений совпадают для кванторных формул с различными вариантами квантора существования, если квантор пробегает конечное множество, и пропозициональных формул дизъюнкции и исключающего «или».

5. ПРИМЕР ЭКСПЕРТНОЙ СИСТЕМЫ МЕДИЦИНСКОЙ ДИАГНОСТИКИ

Входными данными приложения «Экспертная система медицинской диагностики» являются семантические сети постоянного хранения: S1, сформированная с помощью функции «Простая база наблюдений» (пример 1 из [8]), и S2, сформированная с помощью функции «База заболеваний» (пример 4 из [9]) с аргументом S1. Приложение состоит в последовательном выполнении вызовов двух функций – «История болезни» (пример 3 из [9]) (ввод истории болезни) с аргументом S1 и результатом S3 и «Объяснение» (формирование объяснения) (рис. 1) с аргументом S3 и результатом S4, который является результатом всего приложения. Ниже даются некоторые пояснения к программе функции «Объяснение» на OPL.

Семантическая сеть S4 представляет объяснение, формируемое экспертной системой медицинской диагностики. Объяснение состоит из двух частей – объяснения гипотезы о том, что пациент здоров, и объяснения гипотез о том, что пациент болен одним из заболеваний, представленных в базе заболеваний S2.

Объяснение гипотезы о том, что пациент здоров, состоит из объяснения гипотез о том, что все наблюдаемые признаки пациента находятся в норме, и из оценки гипотезы о том, что пациент здоров.

Импликация МИ1 (рис. 2) формирует значение переменной v1, которым является множество всех наблюдаемых признаков в истории болезни S3; для каждого из этих признаков объяснение гипотезы о том, что он находится в норме, состоит из объяснения гипотез о том, что все наблюдавшиеся значения этого признака являются нормальными, и из оценки гипотезы о том, что этот признак находится в норме.

Импликация МИ2 (рис. 3) для каждого значения v1* переменной v1 формирует значение переменной v2, которым является множество моментов наблюдения признака v1* в истории болезни S3; для каждого из этих моментов наблюдения формируется оценка гипотезы о том, что значение признака v1*, наблюдавшееся в этот момент, является нормальным.

Первая импликация (рис. 4) из множества МИ3 для каждого значения v1* переменной v1 и для каждого значения v2* переменной v2 формирует значение переменной v3, которым является значение, наблюдавшееся в момент v2* у признака v1* в истории болезни S3, а также формирует значение переменной v4, которым является множество нормальных значений признака v1* в базе заболеваний S2, а затем выполняется вычисляемый предикат «принадлежность» (принадлежность элемента v3 множеству v4); если значение этого предиката есть «истина», то формируется оценка «подтверждена» гипотезы о том, что значение признака v1*, наблюдавшееся в момент v2*, является нормальным. Вторая импликация, если значение этого предиката есть «ложь», формирует оценку «опровергнута» гипотезы о том, что значение признака v1*, наблюдавшееся в момент v2*, является нормальным.

Первая импликация (рис. 5) из множества МИ4 для значения v1* переменной v1 формирует оценку

«подтверждена» гипотезы о том, что признак v1* находится в норме, если в объяснении S4 оценка гипотез о том, что значения признака v1*, наблюдавшиеся в моменты, равные всем значениям переменной v2, являются нормальными, является «подтверждена». Вторая импликация для значения v1* переменной v1 формирует оценку «опровергнута» гипотезы о том, что признак v1* находится в норме, если в объяснении S4 оценка гипотез о том, что значения признака v1*, наблюдавшиеся в моменты, равные некоторым элементам множества значений переменной v2, являются нормальными, является «опровергнута».

Первая импликация из множества МИ5 (рис. 6) формирует оценку «подтверждена» гипотезы о том, что пациент здоров, если в объяснении S4 оценка гипотезы о том, что признаки, равные всем значениям переменной v1, является «подтверждена». Вторая импликация формирует оценку «опровергнута» гипотезы о том, что пациент здоров, если в объяснении S4 оценка гипотезы о том, что признаки, равные некоторым элементам множества значений переменной v1, является «опровергнута».

Гипотеза «пациент болен» описывается аналогичным образом.

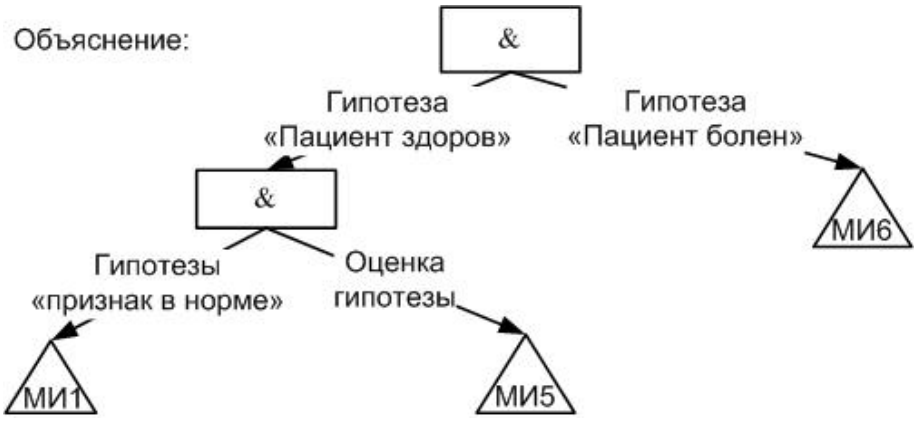

Рис. 1. Объяснение экспертной системы медицинской диагностики

Рис. 2. Описание гипотезы «признак в норме»

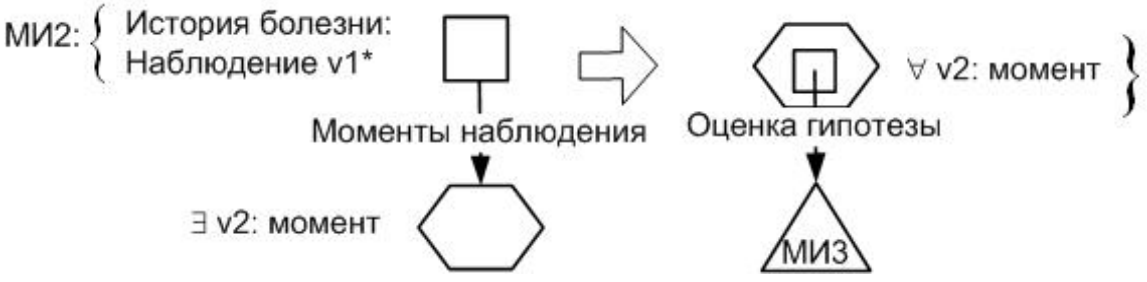

Рис. 3. Описание гипотезы «значение нормальное»

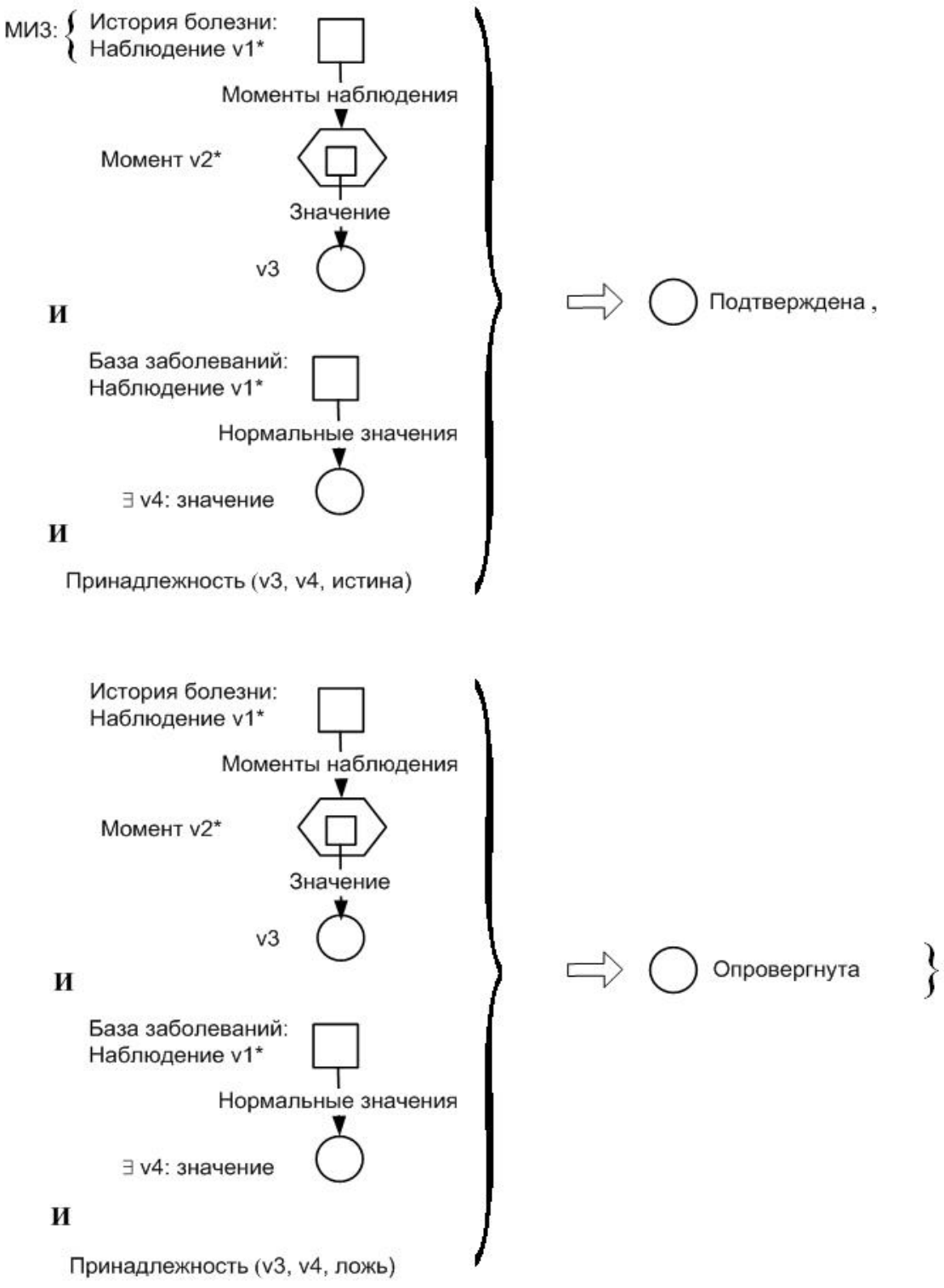

Рис. 4. Описание оценки гипотезы «значение нормальное»

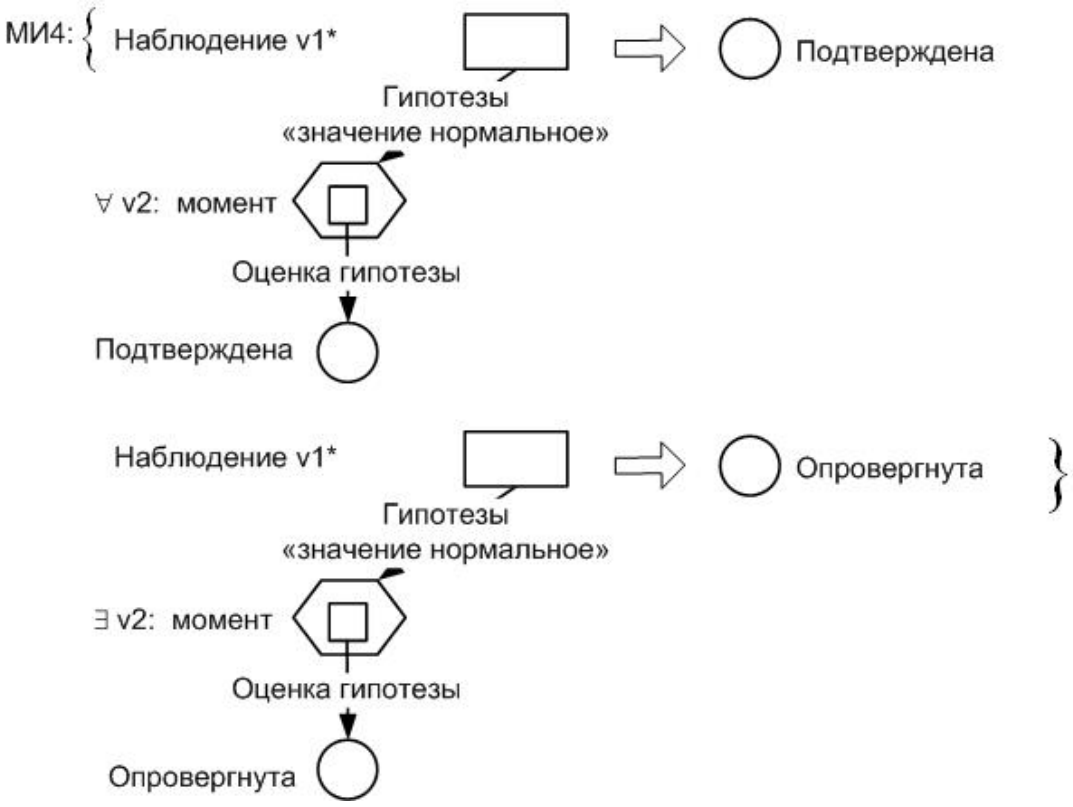

Рис. 5. Описание оценки гипотезы «признак в норме»

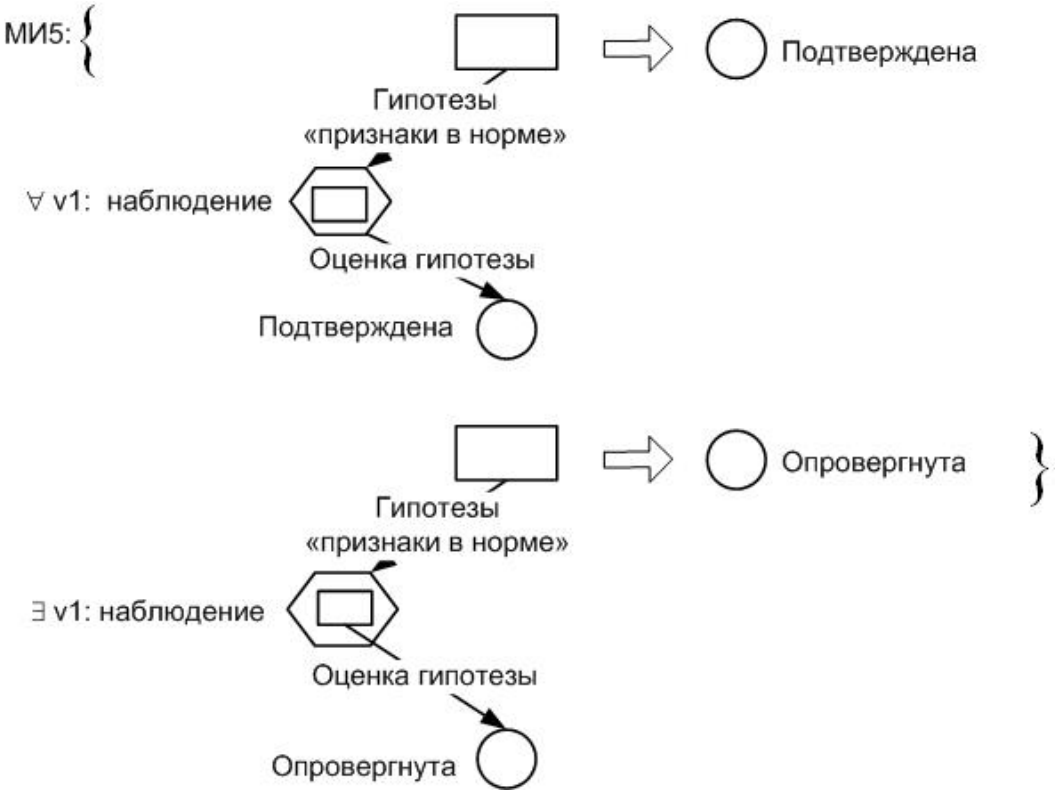

Рис. 6. Описание оценки гипотезы «пациент здоров»

6. ОБСУЖДЕНИЕ

Основные задачи, которые были поставлены при разработке новой онтологической парадигмы программирования и поддерживающего ее языка программирования OPL, – это снижение сложности разработки и сопровождения программ. Анализ существующих парадигм и языков программирования показал, что основными сложностями, с которыми сталкиваются программисты, разрабатывающие программы, и программисты, которые их сопровождают, являются:

- сложность для программиста, разрабатывающего программу, понять, формализовать и описать средствами языка программирования множество процессов получения результатов решения задачи при различных исходных данных; эта сложность имеет место независимо от того, что представляет собой этот процесс – последовательность состояний, сеть вызова функций либо сеть вывода результата;

- сложность для программиста, сопровождающего программу, понять, как устроено описанное программистом-разработчиком множество решений, а в случае если его необходимо изменить – понять как внести все необходимые изменения в программу;

- сложность для программиста реализовать пользовательский интерфейс, обеспечивающий ввод информации от пользователя и вывод результатов, и встроить его в процесс получения результата; в итоге программа становится еще более сложной для понимания и сопровождения, а изменение процесса вычислений приводит к необходимости внесения изменений в интерфейс и наоборот.

Парадигма онтологического программирования предлагает следующие решения указанных выше проблем.

1. Использование семантических сетей в качестве модели данных означает, что обрабатываются только целостные информационные структуры. Это упрощает сопровождение приложений – программисту легче понять целостную информационную структуру, нежели множество разрозненных структур, связь между которыми известна и понятна только разработчику программы.

2. Приложение в рамках предложенной парадигмы может быть представлено как частично упорядоченное множество вызовов функций; каждая функция формирует в качестве результата и, соответственно, обрабатывает в качестве входных данных только целостные информационные структуры, которых, как правило, с приложением связано не слишком много. Соответственно, основная «нагрузка» ложится на описание каждой функции. Описание функции представляет собой онтологию некоторого ансамбля семантических сетей – результата функции. Таким образом, при разработке приложения необходимо: выделить целостные информационные структуры, которые должны быть сформированы в ходе выполнения приложения, определить частичный порядок между задачами их формирования; описать функции для вычисления каждой такой информационной структуры (промежуточного или окончательного результата приложения) как онтологию этой структуры. Сопровождение приложения сводится к добавлению новых целостных информационных структур и функций, изменению частичного порядка вызовов функций приложения и сопровождению самих функций – онтологий их результатов. Описание онтологии результата, как правило, более компактно и наглядно, чем описание процесса его получения.

3. Онтология ансамбля семантических сетей представляется на визуальном языке в виде ориентированного графа. Графовое (визуальное) представление, согласно многим исследованиям, например [10], также упрощает понимание программы. Таким образом, программа на языке OPL в общем случае является более понятной, чем программы, представленные на языках других парадигм.

4. Модель данных, поддерживаемая парадигмой, включает данные (семантические сети) постоянного хранения (подобно базам данных). Это, во-первых, упрощает их повторное использование; во-вторых, упрощает разработку программ, поскольку не требуется дополнительного программирования для чтения информации и ее записи в поддерживаемый формат хранения.

5. Семантические сети, являющиеся результатами некоторой функции, образуют ансамбль (концептуализацию), для которого программа функции является онтологией. Тем самым, многие функции оказываются повторно используемыми. При этом, если процесс формирования информационной структуры является интерактивным (с участием пользователя), то каждая функция может рассматриваться как редактор формирования такой структуры. Время разработки приложения за счет повторного использования функций может быть сокращено. Например, если функция представляет собой редактор историй болезни, то она может быть использована для ввода историй болезни в экспертных системах медицинской диагностики, в медицинских диагностических тренажерах, во многих других медицинских информационных системах. При этом для каждой такой системы не надо разрабатывать редактор для формирования историй болезни, что в общем случае является трудоемкой работой.

6. Абстрактный пользовательский интерфейс входит в операционную семантику логических формул языка. Таким образом, не требуется программирование интерфейса функций.

7. Процесс получения результата в предлагаемой парадигме до некоторой степени аналогичен такому же процессу в логической парадигме. Однако в логической парадигме на каждом шаге процесса вывода должно определяться множество применимых правил и множество значений их посылок. В онтологической парадигме процесс порождения является детерминированным, для каждого шага процесса построения семантической сети определена единственная формула, операционная семантика которой определяет действия на этом шаге.

В настоящее время сложно обозначить весь спектр применений парадигмы онтологического программирования. На сегодня видны две основные возможности использования – создание интеллектуальных систем и редакторов информационных ресурсов любого типа – онтологий, знаний, данных.

СПИСОК ЛИТЕРАТУРЫ

1. Обзор Microsoft. – URL: http://www.ict.edu.ru/ft/005126/intro_net.pdf.
2. Промыслов В.Г., Жарко Е.Ф., Промыслова О.А. Практические аспекты сопровождения и модификации сложных программных систем // Труды IV Международной конференции «Идентификация систем и задачи управления» SICPRO'05. Москва, 25–28 января 2005 г. – М. : Институт проблем управления им. В.А. Трапезникова РАН, 2005. - С. 1151 – 1163.
3. Дехтяренко И.А. Программирование в повествовательном наклонении // SoftCraft разноликое программирование. 2003.. – URL: http://www.softcraft.ru/paradigm/dp/dp01.shtml.
4. Lloyd J.W. Practical Advantages of Declarative Programming // Proc. of Joint Conference on Declarative Programming. GULD-PRODE`94. Pe~niscola (Spain), September 19–22, 1994.
5. Успенский В.А., Семенов А.Л. Теория алгоритмов. Основные открытия и приложения. – М. : Наука, 1987. – 288 с.
6. Себеста Роберт В. Основные концепции языков программирования / Пер. с англ. – 5-е изд. – М. : Вильямс, 2001. – 672 с.
7. Floyd R.W. The Paradigms of Programming // Communications of the ACM. – 1979. – Vol. 22(8). – P. 455 – 460.
8. Грибова В.В., Клещев А.С., Крылов Д.А. Контекстно-свободные грамматики искусственных языков // Научно-техническая информация. Сер. 2. – 2013. – № 4. – С. 9-17.
9. Грибова В.В., Клещев А.С., Крылов Д.А. Контекстно-зависимые грамматики искусственных языков // Научно-техническая информация. Сер. 2. – 2013. – № 6. – С. 1-9.
10. Касьянов В.Н., Евстигнеев В.А. Графы в программировании: обработка, визуализация и применение. – СПб. : БХВ-Петербург, 2003. – 1104 с.

Материал поступил в редакцию 23.11.12.

Сведения об авторах

ГРИБОВА Валерия Викторовна - доктор технических наук, старший научный сотрудник, заведующий лабораторией интеллектуальных систем Института автоматики и процессов управления Дальневосточного отделения Российской академии наук (ИАПУ ДВО РАН), г. Владивосток
e-mail: gribova@iacp.dvo.ru

КЛЕЩЕВ Александр Сергеевич - доктор физико-математических наук, профессор, главный научный сотрудник ИАПУ ДВО РАН, г. Владивосток
e-mail: kleschev@iacp.dvo.ru

УДК 81’322.2’367

Э.С. Клышинский, Н.А. Кочеткова, В.К. Логачева

Метод кластеризации слов с использованием информации об их синтаксической связности*

Излагаются результаты экспериментов в области кластеризации слов русского языка. Разработан новый метод кластеризации, позволяющий разделить слова по семантическим классам в соответствии с их синтаксическими связями с другими словами. Для проведения экспериментов был взят большой (около 7,2 млрд слов) неразмеченный корпус текстов, из которого были извлечены синтаксически связанные группы слов различного вида. Для извлечения подобных связей использовался набор конечных автоматов, обрабатывающих контактные группы слов, не омонимичных по части речи. Для экспериментов были использованы связи вида «существительное + прилагательное» и «глагол (+ предлог) + существительное». Для разделения слов по кластерам использовался модифицированный метод группового среднего. Качество разделения слов по кластерам было проверено с использованием «Русского семантического словаря» под общей редакцией Н. Ю. Шведовой. В итоге были получено значение NMI, равное 0,457, и F_1-мера, равная 0,607.

Ключевые слова: *кластеризация слов, синтаксическая связанность, семантическая близость*

ВВЕДЕНИЕ

В ряде случаев при анализе текстов, исследуя наблюдаемые факты и применяя к ним методы извлечения фактов, можно выделить скрытые законы, которым подчиняется язык. Одной из областей, к которой можно применить подобный подход, является кластеризация слов по смыслу с использованием информации об их сочетаемости. Общеизвестно, что синтаксическая роль слова в предложении зависит от семантических характеристик как самого этого слова, так и окружающих его слов, а различия в синтаксической структуре сходных предложений отражает разницу в их смыслах. Подобная связь заставляет думать, что мы можем извлечь семантическую информацию, используя информацию о синтаксической структуре текстов (обратное уже активно используется системами синтаксического анализа). В соответствии с этим предположением З. Харрис в 1954 г. сформулировал гипотезу о том, что слова, употребляющиеся в сходных контекстах, должны иметь сходный смысл [1]. К сожалению, данное утверждение не является истинным для всех слов языка. Существует большое количество примеров, соответствующих данной гипотезе, однако имеется и определенное число контрпримеров. Анализ результатов экспериментов, проводимых в этой области (см., например, [2–5]), позволяет утверждать, что в соответствии с данной гипотезой можно получить положительные результаты, качество и объем которых наталкиваются на некоторые ограничения (вероятно, вызванные особенностями естественных языков и сходством применяемых методов).

В настоящей статье мы изложим метод кластеризации глаголов, существительных и прилагательных, основанный на выделении из текста на русском языке связей вида «существительное + прилагательное» и «глагол (+ предлог) + существительное» с помощью набора конечных автоматов. Для выделения подобных групп был использован большой неаннотированный корпус текстов различной стилистики. Нашей задачей являлось создание метода автоматической кластеризации сходных слов русского языка в семантически однородные группы. При этом использовалась информация о синтаксической сочетаемости этих слов. Кластер считался семантически однородным в случае, когда все его слова относились к одному семантическому классу (в идеальном случае являлись синонимами).

СУЩЕСТВУЮЩИЕ ПОДХОДЫ К РЕШЕНИЮ ЗАДАЧИ

Успешное решение задачи классификации слов по семантическим группам позволяет повысить качество результатов в различных областях автоматической обработки текстов. К таким областям относятся, например, автоматизация составления и пополнение онтологий и тезаурусов [6, 7], проведение синтаксического анализа [8, 9], выделение сочетаемостных свойств слов языка, используемых, например, при обучении иностранному языку, снятие омонимии [10] и целый ряд других задач. Одной из первых в данной области была работа Ф. Перейры и др. [11]. При построении онтологий и тезаурусов предварительная группировка слов позволяет проводить опе-

* Работа выполнена при финансовой поддержке грантов РГНФ № 12-04-00060 и РФФИ № 11-01-00793.

рации не над отдельными словами, а над их группами, что существенно повышает скорость работы. Наши эксперименты показали, что предварительная кластеризация слов позволяет повысить скорость работы специалиста, работающего над созданием тезауруса предметной области, в 2–3 раза. Кроме того, разделение слов на кластеры может использоваться как дополнительная сторонняя информация, при помощи которой проверяется связанность и корректность уже внесенной информации. При выполнении синтаксического анализа с помощью информации о сочетаемости слова или о его модели управления проверяется возможность установления связи между словами, а также уточняется роль слова в предложении. При снятии омонимии проверяется возможность разных вариантов слова формировать связи с другими словами. Использование данного метода как дополнения к уже существующим позволяет повысить качество снятия омонимии на 1–3% (что является хорошим результатом, так как современные системы, решающие эту задачу, преодолели порог в 95% и стремятся к 98%) [10]. Информация о принадлежности слова к некоторой семантической группе или о модели его управления используется как дополнительная оценка для каждого из вариантов, полученного от модуля лексического анализа слова. Также группирование слов проводится для выделения их модели управления [3, 4, 12], которая помогает в дальнейшем разделить семантические классы слов, устранить полисемию и выделить их возможные синтаксические связи. В работе [5] приведен весьма объемный и качественный обзор различных задач, которые могут быть решены при помощи анализа информации о сочетаемости слов.

Для проведения кластеризации необходимо определить меру близости между словами. В качестве данных, на основе которых рассчитывается мера близости, чаще всего берется информация о связях определенного слова с другими словами. М. Барони и А. Ленчи в работе [5] выделяют несколько видов связности между словами или, скорее, понятиями, обозначаемыми словами:

- атрибутивная связность, когда два слова, находящиеся, например, в отношении синонимии или гиперонимии, имеют одинаковые признаки или набор действий;
- связность по отношениям, когда два слова не принадлежат одному классу, а связываются между собой отношением, например, «часть–целое» (в этом случае они должны относительно часто встречаться в тексте вместе).

Следует заметить, что в случае атрибутивной связности два слова, являющиеся синонимами, но употребляющиеся в текстах разного стиля, могут иметь различный набор действий или объектов и субъектов. Так, например, слова «доктор» и «эскулап» обозначают сходные предметы, но второе слово, в отличие от первого, употребляется обычно в возвышенном или, напротив, уничижительном смысле. И наоборот, существуют слова с разным значением, но со сходным употреблением. Например, как «резервирование», так и «санация» могут быть «дополнительными», «заблаговременными», «стопроцентными», «полными» и «частичными». А для слов «газета» и «полиция» в русском языке нами было найдено более 300 прилагательных, употребляемых как с одним, так и с другим словом (в основном относящихся к географическим названиям или обозначающих принадлежность к тем или иным органам власти, эмоциональным оценкам). Вследствие этого мы не сможем полностью избавиться от ошибок при автоматической группировке слов. Однако, как было замечено выше, сходство выявляется корректно для большого списка слов.

Вне зависимости от типа используемой информации различают несколько методов ее выделения. Самыми простыми из них являются методы, основанные на модели «набор слов» (bag of words model). Методы этой группы предполагают, что если два слова появляются в тексте недалеко друг от друга достаточно часто, то они связаны друг с другом по смыслу [13]. Также в качестве входных данных здесь могут использоваться слова, входящие в одно предложение [14]. Последний метод прост в реализации, позволяет получить связи различного вида между словами, находящимися достаточно далеко друг от друга. С другой стороны, метод не позволяет получить результаты приемлемого качества. В связи с этим чаще используются результаты синтаксического анализа [5, 15]. В этом случае информация о связях характеризуется более высоким качеством, но времени на разбор текста требуется гораздо больше. Кроме того, при определении связей в тексте синтаксический анализ совершает 5–10% ошибок (что достаточно много), покрывая при этом практически 100% текста. Помимо информации о наличии связей между словами часть исследователей используют информацию о роли слова в предложении [5], выдаваемую семантическим анализом. Однако, как отмечалось выше, в ряде случаев роль слова не может быть корректно выявлена без привлечения семантической информации, получение которой и является нашей целью. Таким образом, мы приходим к замкнутому кругу.

Промежуточное положение занимают методы, основанные на лексических шаблонах, когда с помощью, например, регулярных выражений выделяется часть текста, синтаксическая связность которой может быть определена очевидным образом без привлечения синтаксического анализа. Данный метод дает более высокое качество определения синтаксических связей (около 95–98%), но при этом обеспечивает более низкое покрытие (около 20–40% текста). Кроме того, в большинстве языков встает проблема частеречной омонимии, которая не позволяет выделять связанные конструкции напрямую. Применение здесь автоматического снятия омонимии снижает качество результатов. Однако, как будет показано ниже, в русском языке имеется возможность преодолеть эту проблему.

Синтаксические связи выделяются для слов, относящихся к различным частям речи: глаголов [16], существительных [17] и прилагательных [18]. Большинство исследований в данной области посвящены английскому языку, но имеются работы, выполненные для русского [19], немецкого [20], каталонского [21] и других языков.

МЕТОД КЛАСТЕРИЗАЦИИ СЛОВ

Особенности входных данных

В качестве входных данных для метода кластеризации мы использовали информацию, извлеченную из большого неразмеченного корпуса (коллекции) текстов. В наших предыдущих работах [22] была собрана база данных, содержащая в себе информацию о синтаксической сочетаемости слов. Для того чтобы собрать указанную базу данных, мы применили поверхностный синтаксический анализатор, извлекающий именные, предложные и глагольные группы, используя для этого набор регулярных выражений. Из текста извлекались лишь контактные группы слов, для которых можно было достоверно получить синтаксические связи, не прибегая при этом к полному синтаксическому анализу. Приведем несколько примеров используемых регулярных выражений.

<s> (NP|PP) (adv)? VP – единственная предложная или именная группа, стоящая в начале предложения перед единственным глаголом в предложении (перед глаголом может идти причастие),

prep (adj)+ noun – одно или более прилагательных, перед прилагательными находится предлог, после прилагательных – существительное, согласующееся с каждым из прилагательных.

Приведенные выражения обеспечивают высокую вероятность корректного определения синтаксических связей в предложениях на русском языке. В нашем проекте мы использовали лишь несколько видов полученных связей: <глагол, существительное>, <глагол, предлог + существительное> и <существительное, прилагательное>. В приведенных кортежах второй элемент подчиняется первому, но сами слова в исходном тексте могут следовать в произвольном порядке. Так, например, из предложения «В школу пришли новые учителя» будет извлечено три группы следующего вида <приходить + в + школа>, <приходить + учитель>, <учитель + новый>. Созданная база данных содержит полученные коллокации (при этом сами слова приведены к нормальной форме) и их частоты встречаемости.

На практике омонимичность слов не позволяет извлекать подобные сочетания с высокой точностью. Определение границ групп здесь становится затруднительным, так как отсутствует точная информация о части речи, которую представляет слово. Так, например, зачастую сложно определить, является ли рассматриваемое слово прилагательным, существительным или прилагательным в роли существительного. Как следствие, становится непонятно, следует ли рассматривать данное слово как прилагательное и продолжать выделять группу или необходимо принять, что оно существительное, и закончить разбор. Наши предыдущие исследования показали, что природа омонимии в русском языке позволяет использовать в данном случае группы неомонимичных слов. Дело в том, что в русском языке около 75–85% слов не омонимичны по части речи. По этой причине при разборе текста существует высокая вероятность обнаружить неомонимичную группу слов, отвечающую одному из синтаксических шаблонов, использовавшихся для сбора информации в базу данных зависимостей. В табл. 1 приведено сравнение омонимии в русском и английском языках (подробнее см. [23]). За счет высокого процента слов, не омонимичных по части речи, мы имеем возможность извлекать синтаксически связанные конструкции из большой части корпуса. Слова не обязаны быть неомонимичными по параметрам. Вне зависимости от значений лексических параметров мы можем констатировать факт наличия синтаксической связи и поместить сочетания в базу.

Применение методов снятия омонимии, основанных на использовании n-грамм [24, 25] или деревьев принятия решений [26], в данном случае не оправдано. Это связано с тем, что, несмотря на высокий уровень точности работы (до 97%), системы снятия омонимии вносят достаточное количество ошибок, чтобы повлиять на качество итоговых результатов. Наши эксперименты показали, что предложенный метод, основанный на частичном синтаксическом анализе неомонимичных групп слов, позволяет собирать информацию с высоким качеством: сочетания глагола с существительным содержат около 3% ошибок, существительного с прилагательными – около 1% ошибок. Самый большой процент ошибок наблюдался в сочетаниях <существительное + существительное в родительном падеже> – около 8% ошибок.

Заметим, что нам удалось понизить процент ошибок за счет применения модели управления глагола, а также проверки согласования между прилагательным и существительным. Несмотря на высокий уровень качества, покрытие в предложенном методе является низким (20–40% текста в зависимости от его стиля и предметной области). В связи с этим для получения представительных результатов нам пришлось использовать большой корпус текстов (несколько млрд слов).

Под ошибкой мы понимаем здесь ошибку работы предложенного метода или явные ошибки в тексте. Небольшая часть (около 1–2%) корректных с точки зрения метода сочетаний может вызывать сомнения с точки зрения стилистики или физической реализуемости. При этом извлеченные сочетания могут быть обусловлены окружающим текстом, являться выразительными приемами. Таким образом, полученная база в большей мере отражает узус, а не грамматические особенности языка.

Таблица 1

Сравнение видов омонимии в русском и английском языках

Вид омонимии	Лента.ру 2005–2009	Компьюлента 2001–2009	Reuters 2009
Неомонимичные	48,28%	46,55%	38,87%
Неизвестные	4,38%	4,28%	7,65%
Омонимичные по части речи	5,26%	6,33%	50,35%
Омонимичные по лексическим параметрам	27,68%	28,22%	2,79%
Омонимичная начальная форма	4,67%	4,01%	0,32%
Омонимичные по нескольким параметрам	9,92%	10,61%	0,96%

Мера сходства слов и вектор признаков кластеризации

Полученная база данных синтаксической сочетаемости слов может быть формально описана как множество кортежей $\mathbf{B}=\{<w_m, w_s, f>\}$, где w_m – главное слово, w_s – зависимое слово или часть предложной группы (предлог + существительное), а f – частота встречаемости данного сочетания в рассматриваемом тексте. В этом случае можно построить словарь главных слов $\mathbf{D}^+ = <w>$ (прямой словарь). Он будет содержать в себе множество всех главных слов w_m из базы данных **B**. Кроме того, можно построить словарь зависимых слов $\mathbf{D}^- = <w>$ (обратный словарь), который будет содержать в себе множество всех зависимых слов w_s из базы данных **B**. Словари $\mathbf{D}^+$ и $\mathbf{D}^-$ будут использоваться ниже как пространство признаков в ходе кластеризации (т. е. для расчета меры сходства будут использоваться частоты встречаемости слов из этих словарей).

Вектор признаков $\mathbf{v}^+_a$ слова a содержит в себе частоты встречаемости слова a со словами из обратного словаря. Данный вектор строится следующим образом: $\mathbf{v}^+_a = <v_1, v_2, \ldots, v_n>$, $n=|\mathbf{D}^-|$: $v_i = \{f_i$, если в **B** имеется запись вида $<a, d^-_i, f_i>$; 0 – в противном случае$\}$, здесь d^-_i это i-й элемент $\mathbf{D}^-$. Вектор признаков $\mathbf{v}^-_a$ для слова a определяется аналогичным образом, но с использованием частот для слов из обратного словаря $\mathbf{D}^+$. Мера сходства слов a и b определяется в таком случае с использованием косинусной меры сходства между векторами признаков $\mathbf{v}^+_a$ и $\mathbf{v}^+_b$ или между векторами признаков $\mathbf{v}^-_a$ и $\mathbf{v}^-_b$.

Эксперименты показали, что использование собственно частоты встречаемости приводит к большому количеству ошибок. Так, например, слова «Орлеан» и «Зеландия» будут весьма похожи, так как оба чаще всего встречаются в сочетании со словом «Новый» и гораздо реже с другими словами. Для того чтобы устранить подобную ситуацию предлагается использовать логарифм частоты встречаемости вместо самой частоты. Это уменьшает разницу между редко и часто используемыми словами, в результате чего они вносят примерно равный вклад в определение меры сходства.

Фильтрация данных и метод кластеризации

В целях снижения числа ошибок рекомендуется отфильтровать часть объектов, вносящих шум во входные данные. В связи с этим в своих экспериментах мы использовали только те сочетания, у которых частота встречаемости превосходила заданный порог. Кроме того, мы использовали сочетания только с теми главными словами, которые встретились более чем в заданном количестве различных комбинаций, т. е. число зависимых слов для которых превышало определенный порог. В терминах вектора признаков можно сказать, что число ненулевых значений в векторе превышает заданный порог. Математически формирование векторов признаков описывается следующим образом. Пусть $\mathbf{S}^+ = \{\mathbf{v}^+_a\}$, $a \in \mathbf{D}^+$. Тогда $\mathbf{v}^+_a = <v_1, v_2, \ldots, v_n>$, $n=|\mathbf{D}^-|$: $v_i = \{f_i$, если в **B** имеется запись вида $<a, d^-_i, f_i>$ и $f_i \geq \alpha$, 0 – в противном случае$\}$, здесь d^-_i это i-й элемент $\mathbf{D}^-$. При этом $\mathbf{v}^+_a \in \mathbf{S}^+$ только если $NZCount(\mathbf{v}^+_a) > \beta$, где функция $NZCount(\mathbf{v}^+_a)$ возвращает количество ненулевых элементов $\mathbf{v}^+_a$. Аналогичным образом строится множество векторов $\mathbf{v}^-_a$: $\mathbf{S}^- = \{\mathbf{v}^-_a\}$, $a \in \mathbf{D}^-$. Тогда $\mathbf{v}^-_a = <v_1, v_2, \ldots, v_n>$, $n=|\mathbf{D}^+|$: $v_i = \{f_i$, если в **B** имеется запись вида $<d^+_i, a, f_i>$ и $f_i \geq \alpha$, 0 – в противном случае$\}$, здесь d^+_i это i-й элемент $\mathbf{D}^+$. При этом $\mathbf{v}^-_a \in \mathbf{S}^-$ только если

$$NZCount(\mathbf{v}^-_a) > \beta.$$

На первом шаге метода кластеризации рассчитывается матрица расстояний, основанная на расчете косинусной меры сходства. В зависимости от используемого словаря для расчета используются различные векторы признаков: $cos(\mathbf{v}^+_a, \mathbf{v}^+_b)$ или $cos(\mathbf{v}^-_a, \mathbf{v}^-_b)$, $a \neq b$. Из данной матрицы извлекается n самых близких (максимальных) значений или все значения, превосходящие заданный порог γ. Для определения расстояния между кластерами применяется модифицированный метод средней связи [2]. Исходный метод средней связи заключается в том, что расстояние между кластерами рассчитывается как среднее расстояние между всеми элементами двух кластеров. В нашем случае рассчитывалось среднее расстояние между каждым элементом каждого кластера и центроидом второго кластера.

На начальном шаге кластеризации каждое слово считается отдельным кластером. Далее на каждом шаге кластеризации выбираются два самых близких кластера, после чего проводится их объединение в один с пересчетом расстояний до других кластеров. Алгоритм кластеризации останавливается, когда расстояние между объединяемыми кластерами становится меньше заданного порога (напомним, что меньшее значение косинусной меры соответствует большему расстоянию между кластерами). Данное пороговое значение неявно задает количество получаемых кластеров. В случае слишком малого значения порога все слова будут объединены в кластеры, среди которых будут встречаться достаточно большие (до 10–15 слов). Большие кластеры при этом могут объединять в себе несколько более мелких, не связанных между собой по смыслу. Слишком большое значение порога приведет к тому, что будут кластеризованы не все слова, однако качество кластеров будет выше. Тогда возможно применение иерархических методов кластеризации [27]. В этом случае слова образуют бинарное дерево (так как на каждом шаге объединяется ровно два кластера), и перед нами встанет новая проблема – определение границ кластеров.

ОПИСАНИЕ ВХОДНЫХ ДАННЫХ И МЕТОДА ОЦЕНКИ РЕЗУЛЬТАТОВ

Для своих экспериментов мы использовали неразмеченный корпус текстов на русском языке объемом 7,2 млрд слов, состоящий из беллетристики (около 6,6 млрд слов), новостных (около 550 млн слов) и научных текстов (статьи, тексты диссертаций, отчеты – всего около 50 млн слов). Как это было описано выше, синтаксически связанные комбинации слов собирались при помощи набора регулярных выражений, действующих по заданным шаблонам. Морфологическая разметка проводилась с помощью

анализатора «Кросслейтор» [28]. База данных синтаксически связанных комбинаций слов для русского языка составила более 23 млн различных глагольных групп (<глагол + существительное> или <глагол + предлог + существительное>) и около 5,5 млн различных именных групп (существительное + прилагательное). В базу данных попали группы как с прямым, так и с обратным порядком слов, однако все они были нормализованы.

Как и в работе [29], для оценки результатов плоской кластеризации мы использовали формулу нормализованной взаимной информации NMI(A, B) и F_1-меру [2].

$$NMI(A,B) = \frac{I(A,B)}{\left[H(A)+H(B)\right]/2};$$

$$I(A,B) = \sum_k \sum_j \frac{\left|v_k \cap c_j\right|}{N} \log \frac{N\left|v_k \cap c_j\right|}{\left|v_k\right|\left|c_j\right|},$$

где $|v_k \cap c_j|$ – количество общих элементов в кластере v_k и классе золотого стандарта c_i, $H(A)$ – энтропия кластеров, $H(B)$ – энтропия классов золотого стандарта. F_1-мера представляет собой удвоенное среднегармоническое значение от покрытия (recall) и точности (precision). Энтропия определяется по формуле, предложенной Шенноном: $H(x) = -\sum_{i=1}^{n} p(i) * \log_2 p(i)$, где $p(i) = n / N$ – вероятность вхождения элемента в кластер, n – число элементов в кластере, N – общее число элементов.

Под золотым стандартом обычно понимается набор классов, разбиение для которых задано заранее, например, вручную [2]. В нашей работе в качестве золотого стандарта мы использовали словарь [30]. Нами было взято 97 глаголов движения, приобретения и найма. Кроме того, для того чтобы внести некоторый шум в исходные данные, было добавлено еще 49 глаголов из других групп. Все глаголы были разделены на 25 классов: 14 из них были взяты из [30], 9 были получены вручную по таким естественным параметрам, как переходность/непереходность или совершённость/несовершённость действия.

РЕЗУЛЬТАТЫ ЭКСПЕРИМЕНТОВ ДЛЯ РУССКОГО ЯЗЫКА

В ходе экспериментов с корпусом текстов на русском языке мы использовали следующие пороговые значения: частота встречаемости сочетания в корпусе не ниже 5; количество слов, подчиняющихся данному – не менее 7; алгоритм останавливается при расстоянии между кластерами менее 0,3. Для выбранных 146 глаголов значение NMI составило 0,457, а F_1-мера составила 0,607.

В ходе экспериментов мы не могли оценить качество всех полученных данных с помощью золотого стандарта, так как расхождения между лексикой, полученной в экспериментах, и лексикой золотого стандарта слишком велики. Так, например, из рассматриваемых текстов было извлечено достаточно много специфичной лексики, отсутствующей в выбранном в качестве золотого стандарта словаре. В связи с тем, что для русского языка отсутствует достаточно большая (50–70 тыс. слов) открытая онтология (такая, как, например, WordNet [31]), мы приняли решение провести проверку полученных результатов при помощи асессора.

Вместо того чтобы выбирать все пары слов с мерой сходства, превышающей заданное значение, мы отобрали 7000 пар с максимальным сходством. Прямой словарь для глаголов составил в этом случае 4200 различных слов, распределенных по 1550 кластерам. В данном эксперименте все слова были распределены по кластерам, т. е. покрытие составило 1. Оцененная асессором точность составила 0,79. Таким образом, F_1-мера равна 0,88.

Мы также провели эксперименты для существительных и прилагательных. Для существительных имелась возможность применить как прямой (существительное + прилагательное), так и обратный (глагол + предлог, существительное) словарь. Полученные кластеры также были проверены асессором. Для обратного словаря покрытие также равнялось 1, точность составила 0,85, F_1-мера была равна 0,92. Для прямого словаря мы получили покрытие, равное 0,85, точность – 0,88 и F_1-меру – 0,85. Эксперименты с прилагательными показали покрытие 0,85, точность 0,96 и F_1-меру 0,9. Результаты приведены в табл. 2.

Результаты, полученные другими авторами, приведены в табл. 3 (курсив означает, что значение F_1-меры было рассчитано нами на основе приведенных в работах данных). Как видно из приведенных данных, полученные нами результаты вполне согласуются с данными аналогичных работ (здесь мы не учитываем разницу между языками).

Таблица 2

Результаты экспериментов с прямым и обратным словарями

Вид связи	Покрытие	Точность	F_1-мера
<Глагол + предлог, существительное>	1	0,79	0,88
<Существительное + прилагательное>	0,85	0,88	0,85
< Существительное + предлог, глагол >	1	0,85	0,92
<Прилагательное + существительное >	0,85	0,96	0,9

Таблица 3

Сравнение результатов экспериментов

Работа	NMI	F_1-мера
[29]	0,378 – 0,573	0,367 – 0,4
[16]	–	*0,78*
[32]	–	0,46
[33]	–	*0,36*
Данная статья	0,457	0,607

Л. Сан и А. Корхонен в работе [29] получили значение меры NMI, находящееся между 0,378 и 0,573, а также значение F_1-меры, находящееся между 0,367 и 0,4. В работе [16] не приведено значение F_1-меры, однако, согласно приведенным данным, она может быть оценена как равная 0,78. Для арабского языка в работе [32] получена F_1-мера, равная 0,46. В работе [33], где F_1-мера также не рассчитывалась, она равна примерно 0,36. Для всех рассмотренных языков средний размер кластера достаточно мал (2–4 слова). В нашей работе мы получили значение меры NMI равное 0,457 и F_1-меру, равную 0,607. Заметим, что во всех экспериментах для сравнения использовался золотой стандарт. Относительно высокое значение F_1-меры, полученное в наших экспериментах, может быть объяснено значительно бóльшим размером использованного корпуса, а также особенностями русского языка. Также заметим, что методы кластеризации разнятся от работы к работе: начиная с простой косинусной меры и заканчивая выделением ведущих элементов при помощи метода LSA.

ЗАКЛЮЧЕНИЕ

Главным минусом изложенного подхода к кластеризации слов является малый размер получаемых кластеров. Это может быть объяснено ограничениями, накладываемыми на начальную гипотезу о корреляции семантического сходства слов с лексикой, связываемой с этими словами синтаксическими связями, а также особенностями естественных языков.

1. В языке присутствуют слова, которые имеют общий смысл, но обладают различной сочетаемостью с другими словами. Так, например, слова «врач» или «доктор» обладают очень широким и сходным словоупотреблением. Однако слово «эскулап» встречается гораздо реже и в последнее время носит иронический оттенок, если употребляется с определениями. Количество присоединяемых к нему прилагательных значительно меньше и степень сходства со словами «врач» и «доктор» в связи с этим невысока. Сходная ситуация наблюдается и при анализе присоединяемых глаголов, хотя здесь для слова «эскулап» преобладает возвышенная тональность. Как следствие, указанные три слова не могут быть объединены в один кластер предложенным методом. Справедливо и обратное. Так для слов «газета» и «полиция» в русском языке нами было найдено более 300 прилагательных, употребляемых как с одним, так и с другим словом (в основном относящихся к эмоциональным оценкам, географическим названиям, принадлежности тем или иным органам власти).

2. В случае анализа полисемичных слов множество слов, зависимых от данного, может быть разделено на несколько подмножеств. В связи с этим два полисемичных слова, сходных лишь по одному из значений, будут иметь лишь по одному совпадающему подмножеству. Как следствие, их мера сходства окажется весьма мала. С другой стороны, слова, имеющие несколько сходных значений, будут объединяться в первую очередь. Подобный подход позволяет, однако, решить две другие задачи: выделение сходных значений в полисемичных словах и выделение групп сходных признаков.

3. Значения лексических параметров играют очень важную роль в русском языке. Как уже отмечалось выше, все слова в нашей модели приводились к нормальной форме. Однако в модели глагольного управления синтаксическая роль в русском языке задается, помимо глагола, к которому присоединяется слово, или предлога, при помощи которого проводится присоединение, еще и посредством падежей. В связи с этим в нашем случае два глагола, имеющие различный набор валентностей, могут быть объединены в один кластер. Аналогичное можно утверждать и для других частей речи.

Следует заметить, что все проанализированные методы обладают сходными недостатками (с учетом специфики рассматриваемого языка). В большинстве рассмотренных работ средний размер кластера равен примерно 3, увеличение размера кластера приводит к потере точности. Тот факт, что слово присоединяет к себе различные слова в зависимости от того, в каком значении оно используется, успешно используется рядом авторов для автоматического выделения этих значений [34, 35].

Указанные выше фундаментальные особенности языков не позволяют решить задачу автоматического построения кластеров семантически связанных слов. Получаемые кластеры слишком малы. Размер кластеров, как и объем обрабатываемой лексики, по всей видимости, может быть немного увеличен за счет увеличения количества анализируемых текстов и расширения их тематики. При этом, однако, не следует рассчитывать на существенное изменение ситуации.

Возможным представляется применение данной группы методов для начальной разметки кластеров с ее последующим постредактированием. Проведенные нами эксперименты показывают, что при классификации сходных слов вручную обработка предварительно выделенных кластеров позволяет повысить скорость работы оператора в 2–3 раза.

В дальнейшем мы планируем использовать нечеткие методы кластеризации, которые позволяют поместить одно и то же слово в несколько кластеров (см., например, [36]). Это должно увеличить размер кластеров и помочь учитывать полисемию слов. Учет лексических параметров должен улучшить качество кластеризации слов русского языка. Однако нам представляется, что совокупность указанных улучшений не позволит вывести качество решения задачи на кардинально иной уровень. Для этого требуется разработка новых методов обработки имеющейся информации.

СПИСОК ЛИТЕРАТУРЫ

1. Harris Z. Distributional structure // Word. – 1954, 10 (23). – P. 146–162.
2. Маннинг К., Рагхаван П., Шютце Х. Введение в информационный поиск. – М.: Вильямс, 2008. – 528 с.
3. Korhonen A., Krymolowski Yu., Marx Z. Clustering Polysemic Subcategorization Frame Distributions Semantically // Proceedings of the 41st an-

nual meeting of the Association for Computational Linguistics, 2003.

4. Korhonen A., Krymolowski Yu., Briscoe T. A Large Subcategorization Lexicon for Natural Language Processing Applications // Proceedings of the 5th international conference on Language Resources and Evaluation, 2006.
5. Baroni M. and Lenci A. Distributional Memory: A general framework for corpus-based semantics // Computational Linguistics. – 2010. – Vol. 36 (4). – P. 673–721.
6. Волкова Г.А. Создание «онтологии всего». Проблемы классификации и решения // Сб. трудов научно-практического семинара «Новые информационные технологии в автоматизированных системах». – 2013. – С. 293–300.
7. Kipper K., Korhonen A., Ryant N., and Palmer M. Extending VerbNet with Novel Classes // Proceedings of the 5th International Conference on Language Resources and Evaluation. Genoa. Italy.
8. Гельбух А. Разрешение синтаксической неоднозначности и извлечение словаря моделей управления из корпуса текстов // Материалы VIII Международной конференции KDS-99.
9. Мальковский М.Г., Арефьев Н.В. Сочетаемостные ограничения в системе автоматического синтаксического анализа // Программные продукты и системы. № 1. – Тверь, 2012. – С. 28–31.
10. Клышинский Э.С., Кочеткова Н.А., Литвинов М.И., Максимов В.Ю. Метод разрешения частеречной омонимии на основе применения корпуса синтаксической сочетаемости слов в русском языке // Научно-техническая информация. Сер. 2. – 2011. – № 1. – С. 31–35.
11. Pereira F., Tishby N., Lee N. Distributional Clustering of English Words // Proceedings of ACL-93.
12. Manning C. Automatic acquisition of a large subcategorization dictionary from corpora // Proceedings of the 31st annual meeting on Association for Computational Linguistics.
13. Li H. and Abe N. Word Clustering and Disambiguation Based on Co-occurrence Data // Proceedings of COLING-ACL'98. – P. 749–755.
14. Bassiou N., Kotropoulos C. Long distance bigram models applied to word clustering // Pattern Recognition. – 2011. – Vol. 44, Issue 1. – P. 145–158.
15. Preiss J., Briscoe T., and Korhonen A. A system for large-scale acquisition of verbal, nominal and adjectival subcategorization frames from corpora // Proceedings of ACL, 2007. – P. 912–919.
16. Schulte im Walde S. Clustering Verbs Semantically According to their Alternation Behaviour // Proceedings of the 18th International Conference on Computational Linguistics, 2000.
17. Hindle D. Noun classification from predicate-argument structures // Proceedings of ACL-90.
18. Boleda Torrent G. and Alonso i Alemany L. Clustering Adjectives for Class Acquisition // Proceedings of the tenth conference on European chapter of the Association for Computational Linguistics, 2003.
19. Митрофанова О.А., Мухин А.С., Паничева П.В. Автоматическая классификация лексики в русскоязычных текстах на основе латентного семантического анализа // Компьютерная лингвистика и интеллектуальные технологии : Труды международной конференции «Диалог-2007». – М.: Изд-во РГГУ, 2007.
20. Bohnet B., Klatt S., and Wanner L. A Bootstrapping approach to automatic annotation of functional information to adjectives with an application to German // Proceedings of the Workshop on Linguistic Knowledge Acquisition and Representation at the 3rd LREC Conference, 2002.
21. Boleda G., Badia T., Schulte im Walde S. Morphology vs. Syntax in Adjective Class Acquisition // Proceedings of the ACL-SIGLEX Workshop on Deep Lexical Acquisition, 2005. – P. 77–86.
22. Клышинский Э.С., Кочеткова Н.А., Литвинов М.И., Максимов В.Ю. Автоматическое формирование базы сочетаемости слов на основе очень большого корпуса текстов // Компьютерная лингвистика и интеллектуальные технологии : По материалам ежегодной Международной конференции «Диалог» (Бекасово, 26–30 мая 2010 г.). Вып. 9 (16). – М.: Изд-во РГГУ, 2010. – С. 181–185.
23. Клышинский Э.С., Кочеткова Н.А. Метод автоматической генерации модели управления глаголов русского языка // Сб. трудов 12 национальной конференции по искусственному интеллекту КИИ-2012 (Белгород, 16–20 сентября 2012 г.). Т. 1. – Белгород: изд-во БГТУ, 2012. – С. 227–235.
24. Brill E. Unsupervised Learning of Disambiguation Rules for Part of Speech Tagging // Proceedings of the Third Workshop on Very Large Corpora. Cambridge, USA, 1995.
25. Зеленков Ю.Г., Сегалович И.В., Титов В.А. Вероятностная модель снятия морфологической омонимии на основе нормализующих подстановок и позиций соседних слов // Компьютерная лингвистика и интеллектуальные технологии. Труды международного семинара «Диалог-2005».
26. Schmid H. Probabilistic Part-of-Speech Tagging Using Decision Trees // Proceedings of International Conference on New Methods in Language Processing. Manchester, 1994.
27. Sokal R.R. and Rohlf F.J. The comparison of dendrograms by objective methods. – Taxon, 1962.
28. Система морфологического анализа “Crosslator”. [Электрон. ресурс]. – URL: http://clschool.miem.edu.ru/ Материалы-школы.html.

29. Sun L. and Korhonen A. Hierarchical Verb Clustering Using Graph Factorization // Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing (EMNLP). Edinburgh, UK.

30. Русский семантический словарь. Толковый словарь, систематизированный по классам слов и значений / Ин-т рус. яз. им. В. В. Виноградова; под общ. ред. Н. Ю. Шведовой. – М.: Азбуковник, 1998.

31. Fellbaum F. WordNet: An Electronic Lexical Database. – Cambridge, MA: MIT Press, 1998.

32. Snider N. and Diab M. Unsupervised Induction of Modern Standard Arabic Verb Classes Using Syntactic Frames and LSA // Proceedings of ACL, 2006. – P. 795–802.

33. Sass B. First Attempt to Automatically Generate Hungarian Semantic Verb Classes // Proceedings of the 4th Corpus Linguistics conference. Birmingham, 2007.

34. Yarowsky D. Unsupervised Word Sense Disambiguation Rivaling Supervised Methods // Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics. Cambridge, MA, 1995. – P. 189–196.

35. Толдова С.Ю., Кустова Г.И., Ляшевская О.Н. Семантические фильтры для разрешения многозначности в Национальном корпусе русского языка // Компьютерная лингвистика и интеллектуальные технологии. Диалог-2008. – М., 2008. – С. 522–529.

36. Yang M.-S. A Survey of Fuzzy Clustering // A survey of fuzzy clustering, Mathematical and Computer Modelling. – 1993, 18(11). – P. 1–16.

Материал поступил в редакцию 10.07.13.

Сведения об авторах

КЛЫШИНСКИЙ Эдуард Станиславович – кандидат технических наук, доцент Московского института электроники и математики Национального исследовательского университета «Высшая школа экономики» (МИЭМ НИУ ВШЭ); старший научный сотрудник Института прикладной математики им. М. В. Келдыша РАН, Москва
e-mail: eklyshinsky@hse.ru

КОЧЕТКОВА Наталия Александровна – аспирант Московского института электроники и математики Национального исследовательского университета «Высшая школа экономики», Москва
e-mail: natalia_k_11@mail.ru

ЛОГАЧЕВА Варвара Константиновна – кандидат физико-математических наук, научный сотрудник Института прикладной математики им. М. В. Келдыша РАН, Москва
e-mail: logacheva_vk@mail.ru

РОССИЙСКАЯ АКАДЕМИЯ НАУК

Федеральное государственное бюджетное учреждение науки
ВСЕРОССИЙСКИЙ ИНСТИТУТ НАУЧНОЙ И ТЕХНИЧЕСКОЙ ИНФОРМАЦИИ
РОССИЙСКОЙ АКАДЕМИИ НАУК

предлагает научным работникам, аспирантам и другим специалистам в области естественных, точных и технических наук, желающим быстро и эффективно опубликовать результаты своей научной и научно-производственной деятельности, использовать способ публикации своих работ через *систему депонирования.*

«Депонирование (передача на хранение) – особый метод публикации научных работ (отдельных статей, обзоров, монографий, сборников научных трудов, материалов научных мероприятий – конференций, симпозиумов, съездов, семинаров) узкоспециального профиля, разрешенных в установленном порядке к открытому опубликованию, которые нецелесообразно издавать полиграфическим способом печати, а также работ широкого профиля, срочная информация о которых необходима для утверждения их приоритета. Депонирование предусматривает прием, учет, регистрацию, хранение научных работ и обязательное размещение информации о них в специальных информационных изданиях».

Подготовка и передача на депонирование научных работ происходит в соответствии с «Инструкцией о порядке депонирования научных работ по естественным, техническим, социальным и гуманитарным наукам» (М., 2003).

Результатом депонирования является публикация информации о депонированных научных работах в информационных изданиях ВИНИТИ РАН – Реферативном журнале и аннотированном библиографическом указателе «Депонированные научные работы».

В соответствии с “Положением о порядке присуждения ученых степеней”, утвержденным Постановлением Правительства Российской Федерации от 30.01.2002 № 74 (в ред. Постановлений Правительства РФ от 20.04.2006 № 227, от 02.06.2008 № 424, от 20.06.2011 № 475), научные работы, депонированные в организациях государственной системы научно-технической информации, признаны публикациями, учитывающимися при защите кандидатских и докторских диссертаций.

Подать научную работу на депонирование можно обратившись в Отдел депонирования ВИНИТИ РАН по адресу:

125190, Москва, ул. Усиевича, 20.
ВИНИТИ РАН, Отдел депонирования научных работ.
Тел.: 8 (499) 155-43-28, Факс: 8 (499) 943-00-60.
e-mail: dep@viniti.ru

С инструкцией о порядке депонирования можно ознакомиться на сайте ВИНИТИ РАН: **http://www.viniti.ru**