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. № 6 Москва 2013

ИНФОРМАЦИОННЫЕ ЯЗЫКИ

УДК 811.9’36

В.В. Грибова, А.С. Клещев, Д.А. Крылов

Контекстно-зависимые грамматики искусственных языков*

Представлен математический аппарат для описания контекстно-зависимых грамматик искусственных языков – их синтаксиса, логической и порождающей семантик. Даются примеры, демонстрирующие применимость предложенного аппарата; модель формальных языков (порождающих грамматик Хомского) сравнивается с моделью искусственных языков, введенных в данной работе.

Ключевые слова: *искусственный язык, контекстно-свободная грамматика, контекстно-зависимая грамматика, логическая семантика, порождающая семантика*

ВВЕДЕНИЕ

В настоящей статье представлен математический аппарат для описания контекстно-зависимых грамматик искусственных языков. Напомним, что под искусственными языками понимаются языки, которые намеренно проектируются людьми для различных целей (языки математических и химических формул, языки математической логики, языки программирования, операционных систем, запросов к базам данных, языки представления знаний, онтологий и др).
В [1] введен логический метаязык для представления контекстно-свободных грамматик искусственных языков в виде (бесконечного) подмножества множества семантических сетей. Грамматика искусственного языка представляет собой логическую формулу этого языка. Логическими формулами являются размеченные графы (возможно, с циклами), в каждом из которых задана одна вершина, называемая начальной вершиной этой формулы. Выделены следующие группы логических формул: простая формула без переменных, простая кванторная формула без переменных, унарная формула без переменных, пропозициональная формула без переменных, структурная кванторная формула без переменных.

Для представления контекстно-зависимых грамматик в метаязык введены новый тип формул – конеч-

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

ное множество импликаций с взаимно-исключающими антецедентами, а также переменные, входящие в импликации. При этом контекст понимается в широком смысле: контекстом могут служить как любые фрагменты данной (порождаемой) сети, так и любые фрагменты других (ранее порожденных) сетей, принадлежащие тому же или другим искусственным языкам.

Цель нашей работы – описание математического аппарата для определения контекстно-зависимых искусственных языков.

КОНТЕКСТНО-ЗАВИСИМЫЕ ГРАММАТИКИ ИСКУССТВЕННЫХ ЯЗЫКОВ

Контекстно-зависимая грамматика искусственного языка есть формула общего вида метаязыка, начальная вершина которой имеет метку – название этого искусственного языка. Формула общего вида метаязыка есть либо формула без переменных, либо унарная формула общего вида, либо пропозициональная формула общего вида, либо структурная кванторная формула общего вида, либо конечное множество импликаций с взаимно-исключающими антецедентами.

Синтаксис формул общего вида

Унарная формула общего вида есть граф, состоящий из начальной вершины и дуги с меткой T (термином), выходящей из начальной вершины и входящей в начальную вершину некоторой формулы F общего вида (рис. 1).

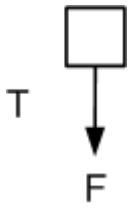
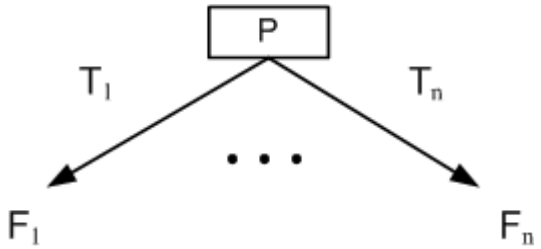


Рис. 1. Унарная формула общего вида

Пропозициональная формула общего вида есть граф, состоящий из начальной вершины с пропозициональной меткой P и выходящих из нее n дуг (не менее двух), каждая из которых имеет метку T_i (термин; i от 1 до *n*; все эти метки должны быть попарно различны) и входит в начальную вершину некоторой формулы F_i общего вида (рис. 2). Пропозициональными метками P являются:



& – конъюнкция,
∨ – дизъюнкция,
| – исключающее «или».

Рис. 2. Пропозициональная формула общего вида

Множество пропозициональных меток является расширяемым.

Любая из дуг, выходящая из начальной вершины с пропозициональной меткой &, может иметь метку факультативности «[]».

Структурная кванторная формула общего вида состоит из начальной вершины с меткой, имеющей вид QMT, где Q – знак квантора, M – описание (конечного или бесконечного) множества, а T – термин, и формулы общего вида F, начальная вершина которой изображается внутри начальной вершины структурной кванторной формулы (рис. 3). Описание конечного множества может иметь вид $\{c_1, \ldots, c_n\}$, где $c_1, \ldots, c_n$ – попарно различные константы, либо быть целым конечным интервалом – в этом случае знаком квантора может быть ∀, ∃, ∃2, ∃?. Описание бесконечного множества может быть названием сорта, неименованным множеством «*» или вещественным конечным интервалом – в этом случае знаком квантора может быть ∃, ∃2.

Рис. 3. Структурная кванторная формула общего вида

Конечное множество импликаций есть граф, состоящий из одной вершины с меткой $\{\Pi_1 \Rightarrow F_1, \ldots, \Pi_m \Rightarrow F_m\}$, где $\Pi_1, \ldots, \Pi_m$ – антецеденты, а $F_1, \ldots, F_m$ – консеквенты импликаций (рис. 4). Антецедент импликации есть конечное множество компонент – формул с переменными, каждая из которых может иметь префикс. Префикс есть название некоторого искусственного языка. Консеквентом импликации является формула с переменными общего вида. Любая переменная, входящая в консеквент импликации, должна входить и в ее антецедент.

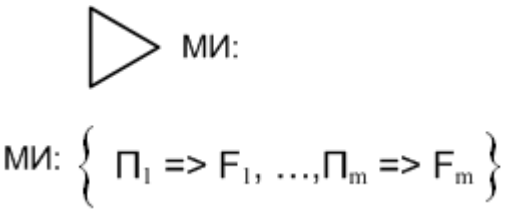


Рис. 4. Множество импликаций

Формулой с переменными (общего вида) может быть простая формула с переменной, простая кванторная формула с переменной, унарная формула с переменными (общего вида), пропозициональная формула с переменными (общего вида) и структурная кванторная формула с переменными (общего вида).

Простая формула с переменной есть граф, состоящий из единственной вершины с меткой C (рис. 5). Эту единственную вершину будем называть начальной вершиной простой формулы с переменной. Меткой может быть и переменная v, и элемент значения переменной v*.

Рис. 5. Простая формула с переменной

Простая кванторная формула с переменной есть граф, состоящий из единственной вершины с меткой, имеющей вид QMT, где Q – знак квантора, M – описание множества, а T – термин (рис. 6). Знаком квантора может быть: ∀ (для всех), ∃ (существует), ∃2 (существует не менее двух), ∃? (существует, но не для всех), ∃! (существует и единственен), ∃[] (существует подынтервал). Множество кванторов является расширяемым. Описание множества M может быть переменной v.

Рис. 6. Простая кванторная формула с переменной

Унарная формула с переменными (общего вида) имеет тот же вид, что и на рис. 1, но F_i является формулой с переменными (общего вида).

Пропозициональная формула с переменными (общего вида) имеет тот же вид, что и на рис. 2, но F_1, …, F_n являются формулами с переменными (общего вида). Если пропозициональная формула с переменными и пропозициональной меткой «&» входит в антецедент импликации, то дуги не могут иметь меток факультативности «[]», но могут иметь метки отрицания «¬».

Структурная кванторная формула с переменными (общего вида) имеет тот же вид, что и на рис. 3, но описание множества M может быть переменной, а F – формулой с переменными (общего вида).

Логическая семантика формул общего вида

Множество импликаций с меткой $\{\Pi_1 \Rightarrow F_1, \ldots, \Pi_m \Rightarrow F_m\}$ истинно на вершине V сети N, если существует единственное $i \in [1, m]$ такое, что только импликация $\Pi_i \Rightarrow F_i$ истинна на вершине V сети N, а антецеденты всех других импликаций из этого множества не являются истинными на вершине V сети N при любых подстановках значений вместо переменных, входящих в эти антецеденты.

Импликация $\Pi \Rightarrow F$ истинна на вершине V сети N, если существует такая подстановка значений вместо переменных, входящих в импликацию, при которой на вершине V сети N истинны и антецедент, и консеквент импликации. Антецедент импликации истинен на вершине V сети N при некоторой подстановке, если при этой подстановке истинна каждая компонента этого антецедента на вершине V сети N. Если компонента антецедента импликации не имеет префикса, то эта компонента истинна на вершине V сети N при некоторой подстановке, если в порождаемой сети N существует хотя бы одна вершина V_1, на которой истинна формула с переменными, являющаяся этой компонентой антецедента импликации, при этой подстановке. Если компонента антецедента импликации имеет префикс и этот префикс идентифицирует уже существующую другую сеть N_1, то эта компонента истинна на вершине V сети N при некоторой подстановке, если в сети N_1 существует хотя бы одна вершина V_1, на которой истинна формула с переменными, являющаяся этой компонентой антецедента импликации, при этой подстановке.

Простая формула с переменной F, меткой которой является переменная v, истинна на вершине V сети N при подстановке θ, если V является терминальной и простой, а ее метка есть константа C, являющаяся значением переменной v в подстановке θ. Если меткой простой формулы с переменной F является элемент значения переменной v*, а значением переменной v в подстановке θ является множество значений, то формула F истинна на вершине V сети N при подстановке θ, если V является терминальной и простой, а ее метка есть константа C, являющаяся одним из элементов значения переменной v в подстановке θ.

Если метка входной вершины ***простой кванторной формулы с переменной*** F имеет вид ∀ v T или метка входной вершины обобщенной простой кванторной формулы F, входящей в антецедент импликации, имеет вид ∃ v T, где v – переменная, а T – термин, то F истинна на вершине V сети N при подстановке θ, если V является структурной и терминальной вершиной, содержащей n сетей, каждая из которых состоит из одного корня с меткой класса – T и индивидуальными метками – элементами множества $\{d_1, \ldots, d_n\}$, являющегося значением переменной v в подстановке θ (заметим, что хотя логическая семантика этих формул в антецеденте импликации совпадает, их порождающая семантика различна).

Если метка входной вершины ***простой кванторной формулы с переменной*** F, входящей в консеквент импликации, имеет вид ∃ v T, где v – переменная, а T – термин, то F истинна на вершине V сети N при подстановке θ, если V является структурной и терминальной вершиной, содержащей непустое множество сетей, каждая из которых состоит из одного корня с меткой класса – T и индивидуальной меткой – одним из элементов множества, являющегося значением переменной v в подстановке θ (индивидуальные метки попарно различны).

Если метка входной вершины ***простой кванторной формулы с переменной*** F, входящей в консеквент импликации, имеет вид ∃2 v T, где v – переменная, а T – термин, то F истинна на вершине V сети N при подстановке θ, если V является структурной и терминальной вершиной, содержащей не менее двух сетей, каждая из которых состоит из одного корня с меткой класса – T и индивидуальной меткой – одним из элементов множества, являющегося значением переменной v в подстановке θ (индивидуальные метки попарно различны).

Если метка входной вершины ***простой кванторной формулы с переменной*** F, входящей в консеквент импликации, имеет вид ∃? v T, где v – перемен-

ная, а T – термин, то F истинна на вершине V сети N при подстановке θ, если V является структурной и терминальной вершиной, содержащей менее чем n сетей, каждая из которых состоит из одного корня с меткой класса – T и попарно различными индивидуальными метками – элементами множества $\{d_1, \dots, d_n\}$, являющегося значением переменной v в подстановке θ.

Если метка входной вершины ***простой кванторной формулы с переменной*** F, входящей в консеквент импликации, имеет вид ∃! v T, где v – переменная, а T – термин, то F истинна на вершине V сети N при подстановке θ, если V является простой и терминальной вершиной, а ее меткой является один из элементов множества, являющегося значением переменной v в подстановке θ.

Если метка входной вершины ***простой кванторной формулы с переменной*** F, входящей в консеквент импликации, имеет вид ∃[] v T, где v – переменная, а T – термин, то F истинна на вершине V сети N при подстановке θ, если V является простой и терминальной вершиной, а ее меткой является целый (вещественный) интервал, являющийся подынтервалом интервала I, являющегося значением переменной v в подстановке θ.

Унарная формула (с переменными) (общего вида) F истинна на вершине V сети N при подстановке θ, если V является простой вершиной, из нее выходит в точности одна дуга с меткой T и эта дуга входит в вершину V_1, на которой истинна формула F_1 (с переменными) (общего вида) при подстановке θ.

Если пропозициональной меткой начальной вершины W ***пропозициональной формулы (с переменными) (общего вида)*** F является &, то F истинна на вершине V сети N при подстановке θ, если V является простой вершиной, для каждой дуги с меткой T_i, выходящей из V, существует дуга с такой же меткой T_i, выходящая из W, для каждой дуги с меткой T_i, выходящей из W существует дуга с такой же меткой T_i, выходящая из V, а каждая формула (с переменными) (общего вида) F_i истинна на вершине, в которую входит дуга с меткой T_i, выходящая из вершины V при подстановке θ. Если F входит в антецедент импликации, а дуга с меткой T_i, выходящая из ее начальной вершины, имеет метку «¬», то в сети N дуга с меткой T_i, выходящая из V, должна отсутствовать. Если F входит в консеквент импликации, а дуга с меткой T_i, выходящая из ее начальной вершины, имеет метку «[]», то в сети N дуга с меткой T_i, выходящая из V, может отсутствовать.

Если пропозициональной меткой начальной вершины W ***пропозициональной формулы (с переменными) (общего вида)*** F, входящей в консеквент импликации, является ∨, то F истинна на вершине V сети N при подстановке θ, если V является простой вершиной, существует подмножество дуг, выходящих из W, между которым и множеством дуг, выходящих из V, существует взаимно-однозначное соответствие, при котором метки T_i соответствующих дуг совпадают, а формулы (с переменными) (общего вида) F_i, в начальные вершины которых входят дуги, выходящие из W, истинны при подстановке θ на вершинах, в которые входят соответствующие дуги с метками T_i, выходящие из V.

Если пропозициональной меткой начальной вершины W ***пропозициональной формулы (с переменными) (общего вида)*** F, входящей в консеквент импликации, является |, то F истинна на вершине V сети N при подстановке θ, если V является простой вершиной, из нее выходит единственная дуга, существует дуга, выходящая из W, такая что метки T_i этих дуг совпадают, а формула (с переменными) (общего вида) F_i, в начальную вершину которой входит дуга с меткой T_i, выходящая из W, истинна при подстановке θ на вершине, в которую входит дуга с меткой T_i, выходящая из V.

Если метка входной вершины ***структурной кванторной формулы (с переменными) (общего вида)*** F имеет вид ∀ v T или метка входной вершины ***структурной кванторной формулы (с переменными) (общего вида)*** F, входящей в антецедент импликации, имеет вид ∃ v T, где v – переменная, а T – термин, то формула F истинна на вершине V сети N при подстановке θ, если V является структурной вершиной, содержащей n сетей, каждая из которых имеет метку класса – T и индивидуальные метки – попарно различные элементы множества $\{d_1, \dots, d_n\}$, являющегося значением переменной v в подстановке θ, и на корне каждой из этих сетей истинна формула (с переменными) (общего вида) F_1 при подстановке θ (заметим, что хотя логическая семантика этих формул в антецеденте импликации совпадает, их порождающая семантика различна).

Если метка входной вершины ***структурной кванторной формулы (с переменными) (общего вида)*** F, входящей в консеквент импликации, имеет вид ∃ v T, где v – переменная, а T – термин, то F истинна на вершине V сети N при подстановке θ, если V является структурной вершиной, содержащей непустое множество сетей, каждая из которых имеет метку класса – T и индивидуальные метки – попарно различные элементы множества, являющегося значением переменной v в подстановке θ, и на корне каждой из этих сетей истинна формула (с переменными) (общего вида) F_1 при подстановке θ.

Если метка входной вершины ***структурной кванторной формулы (с переменными) (общего вида)*** F, входящей в консеквент импликации, имеет вид ∃2 v T, где v – переменная, а T – термин, то F истинна на вершине V сети N при подстановке θ, если V является структурной вершиной, содержащей не менее двух сетей, каждая из которых имеет метку класса – T и индивидуальные метки – попарно различные элементы множества, являющегося значением переменной v в подстановке θ, и на корне каждой из этих сетей истинна формула (с переменными) (общего вида) F_1 при подстановке θ.

Если метка входной вершины ***структурной кванторной формулы (с переменными) (общего вида)*** F, входящей в консеквент импликации, имеет вид ∃? v T, где v – переменная, а T – термин, то F истинна на вершине V сети N при подстановке θ, если V является структурной вершиной, содержащей менее чем n сетей, каждая из которых имеет метку

класса – T и индивидуальные метки – попарно различные элементы множества, являющегося значением переменной v в подстановке θ, и на корне каждой из этих сетей истинна формула (с переменными) (общего вида) F_1 при подстановке θ.

Порождающая семантика множества импликаций

Если активной вершине V сети N соответствует начальная вершина W ***множества импликаций*** с меткой $\{\Pi_1 \Rightarrow F_1, \ldots, \Pi_m \Rightarrow F_m\}$, где $\Pi_1, \ldots, \Pi_m$ – антецеденты, а $F_1, \ldots, F_m$ – консеквенты импликаций, то шаг порождения возможен лишь в случае, когда на вершине V сети N истинен антецедент только одной импликации (например, $\Pi_i \Rightarrow F_i$). В этом случае в каждой из сетей, соответствующих префиксам компонент антецедента Π_i, выполняется поиск по образцу, заданному формулой этой компоненты, строится согласованная подстановка θ значений, заменяющих переменные антецедента Π_i, а шаг порождения из вершины V осуществляется в соответствии с формулой консеквента F_i при подстановке θ (т.е. при замене в этих формулах переменных их значениями из подстановки).

Пример 3. (Примеры 1 и 2 см. в работе [1]). На рис. 7 приведена контекстно-зависимая грамматика языка описания простых историй болезни в контексте простой базы наблюдений. Простая история болезни состоит из множества наблюдений, имеющих различные названия. Каждое наблюдение имеет некоторое множество моментов наблюдения, обозначенных константами сорта «дата–время». С каждым моментом наблюдения связано одно значение. Контекстные зависимости состоят в том, что множество названий наблюдений в истории болезни должно быть подмножеством множества названий наблюдений в простой базе наблюдений, а значение наблюдения в любой момент наблюдения в истории болезни должно быть элементом множества значений наблюдения с тем же названием в простой базе наблюдений.

Пример 4. На рис. 8 приведена контекстно-зависимая грамматика языка представления медицинских знаний в контексте простой базы наблюдений. Простая база медицинских знаний состоит из описания нормы и описания заболеваний.

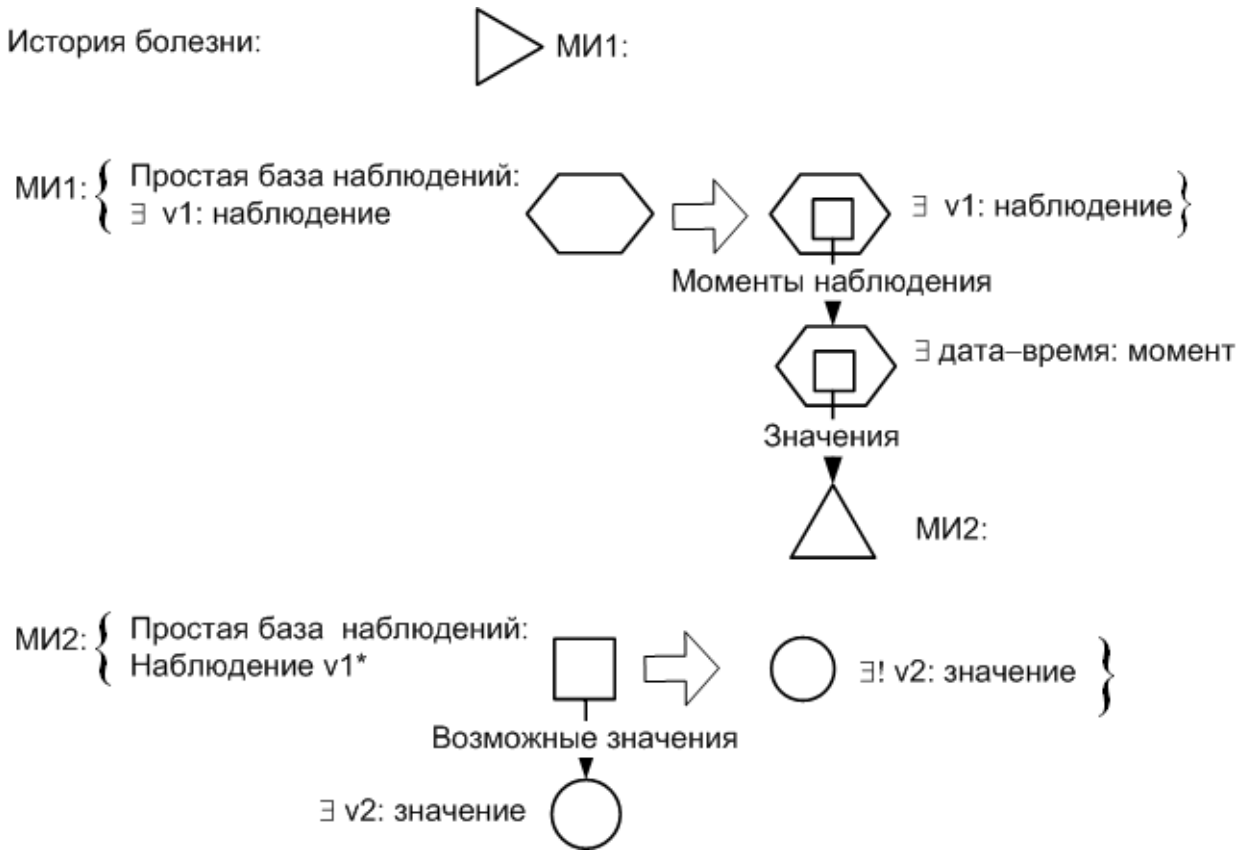


Рис. 7. Контекстно-зависимая грамматика языка описания простых историй болезни

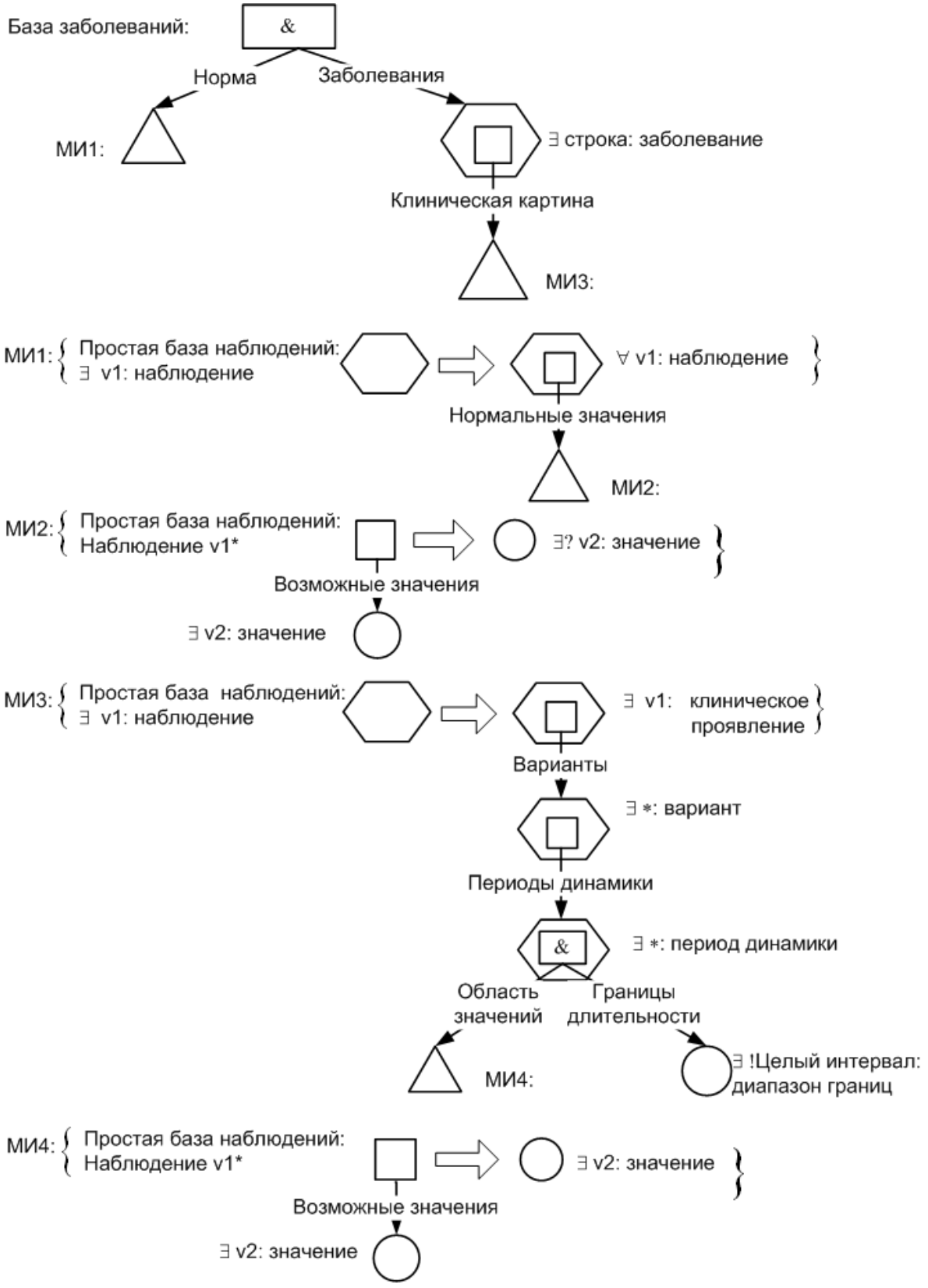


Рис. 8. Контекстно-зависимая грамматика языка представления медицинских знаний

Описание нормы каждому наблюдению из простой базы наблюдений сопоставляет нормальные значения – собственное подмножество множества возможных значений этого наблюдения в простой базе наблюдений. Описание заболеваний состоит из описаний отдельных заболеваний, имеющих названия. Описание каждого заболевания состоит из его клинической картины, содержащей описания клинических проявлений, названиями которых являются названия некоторых наблюдений в простой базе наблюдений. Описание каждого клинического проявления состоит из конечного множества вариантов, не имеющих названий, а описание каждого варианта – из конечного упорядоченного множества периодов динамики, не имеющих названий. Описание каждого периода динамики состоит из области значений (подмножество множества возможных значений наблюдения с тем же названием в простой базе наблюдений) и границ длительности (целого интервала).

Модульность

Для сокращения описаний грамматик и описания контекстно-зависимых грамматик рекурсивных языков в описание грамматик вводится модульность. Описание модуля имеет название, формальные параметры и тело. Тело представляет собой формулу, в которую могут входить формальные параметры модуля. Вызов модуля является формулой. Он состоит из названия модуля и фактических параметров, по порядку и числу соответствующих формальным параметрам вызываемого модуля. Порождающая семантика вызова модуля состоит в его текстуальной замене телом вызываемого модуля, в котором вхождения формальных параметров текстуально заменены на фактические.

ПРОЕКЦИЯ СЕТЕЙ НА ИХ ТЕКСТОВОЕ ПРЕДСТАВЛЕНИЕ

Проекция сетей на их текстовое представление описывается множеством импликаций (с возможным использованием модульности), но в этих импликациях консеквент содержит лишь операторы порождения строк. К ним относятся строковые константы (порождают сами себя), конкатенация (знак конкатенации опускается), переменные (при порождении заменяются значениями переменных), оператор вывода множества и универсальная кванторная формула. Оператор вывода множества имеет вид $\{S_1, S_2\}$, где S_1 – описание конечного множества (конечное множество констант, целый интервал или переменная, имеющая значением множество), а S_2 – знак разделителя между элементами множества. Порождением оператора вывода множества является последовательность элементов множества S_1, разделенных знаком разделителя S_2. Универсальная кванторная формула имеет вид $\forall$ v: F(v*), где v – переменная, имеющая значением множество, а F(v*) – формула, содержащая вхождения v*. Порождением универсальной кванторной формулы является последовательность порождений формулы F(v*), где v* пробегает все значения переменной v.

Пример 5. На рис. 9 приведено описание проекции реальной базы наблюдений на ее текстовое представление. В нем описано два модуля – МИ1 из трех импликаций и МИ2 из четырех импликаций. Заменяя в последнем предложении рис. 9 переменную v на индивидуальную метку корня сети, представляющей конкретную базу наблюдений, можно получить из этого описания текстовые представления различных баз наблюдений. Несмотря на то, что грамматика является контекстно-свободной, текстовое представление реальных баз наблюдений как формальный язык является контекстно-зависимым.

ОБСУЖДЕНИЕ

Сравним модель формальных языков (порождающие грамматики Хомского) с моделью искусственных языков, введенной в этой статье, с точки зрения возможности создания инструментальных средств, поддерживающих ввод в компьютер информации, представленной на этих языках, с целью ее последующей компьютерной обработки.

Когда информация преобразуется в форму, предписываемую некоторым формальным языком, то по отношению к этому виду деятельности (преобразованию формы информации, формализации) можно выделить две группы людей – носителей информации (людей, владеющих преобразуемой информацией) и носителей формального языка (людей, умеющих представлять информацию средствами этого формального языка). Для конкретного формального языка эти две группы людей не всегда совпадают, а иногда пересечение этих групп пусто. Впервые эта проблема была осознана в связи с попытками формирования баз знаний для экспертных систем. Как правило, носители знаний (например, врачи-эксперты) не входят в группу носителей языка представления этих знаний (например, языка описания продукционных систем OPS-5). Формирование баз знаний и управление ими через посредников (инженеров знаний, носителей языка) широко обсуждалось в литературе [2], но является, мягко говоря, неудачным решением этой проблемы. Усилия же, направленные на проектирование формальных языков, представляющих базы знаний в форме, близкой к их описанию на естественном языке, лишь облегчают восприятие содержимого этих баз знаний для экспертов, но не решают для них проблемы формирования баз знаний и управления ими без посредников.

Ошибки, возникающие при вводе в компьютер информации, представленной на формальном языке, могут быть разделены на ошибки в информации и ошибки в использовании формального языка. Не существует универсальных (проблемно-независимых) методов автоматического обнаружения ошибок первого из указанных классов. Ошибки второго класса являются следствием использования формального языка как средства формализации информации и совершаются носителями формального языка.

По многочисленным исследованиям, их количество пропорционально длине текста, представляющего информацию, хотя коэффициент пропорциональности индивидуален (зависит от носителя формального языка и от сложности этого языка). Поэтому важной задачей (до сих пор нерешенной)

при разработке инструментальных средств, поддерживающих ввод информации в компьютер, является минимизация и даже исключение ошибок этого класса.

Инструментальными средствами, поддерживающими ввод информации в компьютер и основанными на модели формальных языков (порождающих грамматик Хомского), являются синтаксические анализаторы, управляемые грамматиками формальных языков; инструментальными средствами, основанными на модели искусственных языков, – структурные редакторы, управляемые грамматиками искусственных языков. Сравним эти два класса инструментальных средств с точки зрения сложности работы, которую необходимо выполнить при вводе информации в компьютер с их помощью.

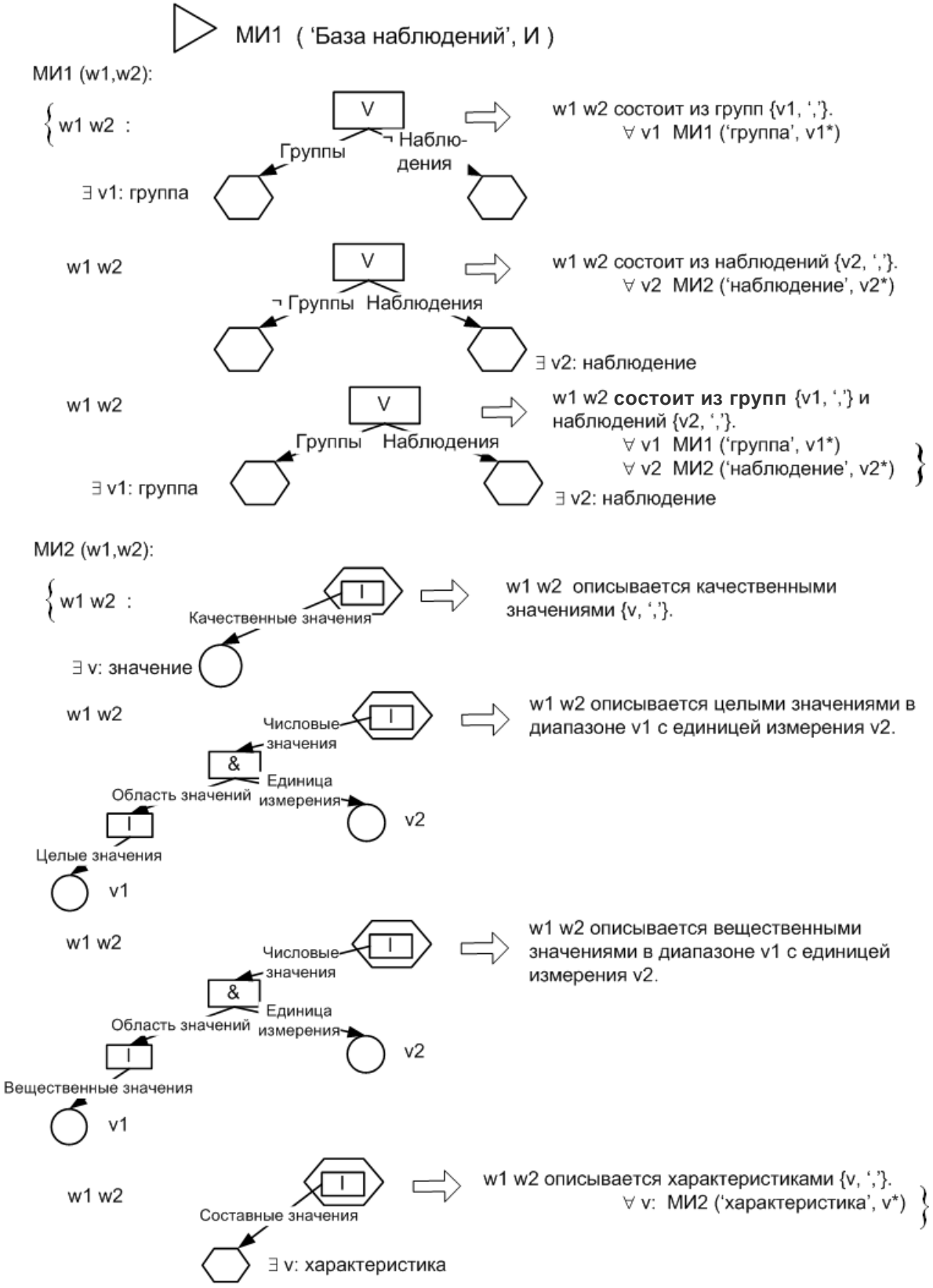


Рис. 9. Проекция реальной базы наблюдений на ее текстовое представление

При использовании синтаксических анализаторов формальных языков информация вводится в компьютер в форме текста. Далее выполняется синтаксический анализ этого текста и, если он содержит ошибки в использовании языка, синтаксический анализатор представляет пользователю сведения об этих ошибках. Эти сведения могут быть как недостаточными (когда одни ошибки маскируют другие), так и избыточными (когда правильные фрагменты текста воспринимаются анализатором как ошибочные вследствие наличия в этом тексте других ошибок). Как правило, причины ошибок определяются анализаторами весьма ненадежно (если вообще определяются). Поэтому устранение ошибок, обнаруженных анализатором, не гарантирует пользователю синтаксическую правильность исправленного текста, т.е. шаги анализа текста и исправления в нем ошибок часто должны повторяться многократно. Результатом работы анализатора является дерево грамматического разбора, которое обычно не совпадает с семантическим представлением информации, удобным для ее последующей обработки. Поэтому в дополнение к синтаксическому анализатору, как правило, приходится разрабатывать специализированную программу преобразования дерева разбора в семантическое представление. Такая технология ввода информации в компьютер связана с весьма значительными затратами времени как пользователя (он должен изучить формальный язык, преобразовать информацию в текст на нем и исправить все ошибки в этом тексте), так и компьютера (многократный синтаксический анализ и преобразование дерева грамматического разбора в семантическое представление), что позволило некоторым специалистам саркастически пошутить: компьютеры были изобретены для обработки информации, а на самом деле используются для трансляции программ.

Структурный редактор представляет собой программную реализацию порождающей семантики метаязыка (языка представления грамматик искусственных языков); в случае, когда для выполнения очередного шага порождения семантической сети информации недостаточно, редактор обращается за ней к пользователю. Поэтому, с точки зрения пользователя, такой редактор задает ему вопросы об информации и по его ответам порождает семантическую сеть. При этом редактор генерирует пользовательский интерфейс, который **не дает возможности** пользователю совершать ошибки в использовании языка. При работе с таким редактором пользователь **не должен** изучать какой-либо искусственный язык (он должен быть лишь носителем соответствующей информации), **не должен** заниматься каким-либо преобразованием информации (он должен лишь отвечать на вопросы редактора), **не может** совершить ошибку в использовании искусственного языка (может совершать ошибки лишь в информации), а результатом работы редактора является семантическое представление информации (отсутствуют дополнительные расходы компьютерного времени даже при исправлении ошибок в информации).

Примером структурного редактора, основанного на контекстно-свободных грамматиках искусственных языков, может служить редактор ИРУО [3], доступный через облачную платформу Многоцелевого банка знаний [4].

Единственным ограничением области применения структурных редакторов являются такие виды информации, для представления которых текстовое представление является наиболее естественным. Примерами могут служить выражения в языках программирования, математические формулы и т.п. Однако и в этом случае возможны решения, связанные со структурными редакторами, основанными на комбинации грамматик искусственных языков и порождающих контекстно-свободных грамматик.

СПИСОК ЛИТЕРАТУРЫ

1. Грибова В.В., Клещев А.С., Крылов Д.А. Грамматики искусственных языков. Часть 1. Контекстно-свободные грамматики // Научно-техническая информация. Сер. 2. – 2013. – № 4. – С. 9–17.
2. Уотермен Д. Руководство по экспертным системам / пер. с англ. – М.: Мир. 1989. – 388 с.
3. Клещев А.С., Орлов В.А. Компьютерные банки знаний. Универсальный подход к решению проблемы редактирования информации // Информационные технологии. – 2006. – № 5. – С. 25–31.
4. Клещев А.С., Орлов В.А. Компьютерные банки знаний. Многоцелевой банк знаний // Информационные технологии. – 2006. – № 2. – С. 2–8.

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

КРЫЛОВ Дмитрий Александрович - ведущий инженер-программист лаборатории интеллектуальных систем ИАПУ ДВО РАН, г. Владивосток
E-mail: dmalkr@gmail.com

ИНФОРМАЦИОННЫЙ АНАЛИЗ

УДК 001.82:001.103

А.В. Ревенко

Нахождение ошибок в бинарных таблицах данных*

Дается классификация возможных ошибок в строках бинарных таблиц данных (формальных контекстов), обсуждаются возможности их нахождения. Предложен подход к нахождению некоторых типов ошибок в новых строках (содержаниях объектов) бинарных таблиц данных (формальных контекстов). Этот подход основан на нахождении тех импликаций из базиса импликаций формальных контекстов, которые не удовлетворяются новым объектом. Отмечается, что такой подход может привести к вычислительно сложному решению. Предложен альтернативный подход, основанный на вычислении замыканий подмножеств содержания объекта, этот подход позволяет найти полиномиальный алгоритм решения задачи. Алгоритм также может быть использован для обоснования правильности (существования) объекта или для удаления ненужных и добавления недостающих признаков. Обсуждаются результаты экспериментов.

Ключевые слова: *формальный контекст, импликация, поиск ошибок*

ВВЕДЕНИЕ

Нахождение ошибок в данных исторически связано с такими областями науки, которые изучают корректирующие коды и базы данных [1, 2]. В процессе хранения и передачи информации по сетям связи неизбежно возникают ошибки. Контроль целостности данных и исправление ошибок – важные задачи на разных уровнях работы с информацией.

В системах связи имеются разные стратегии борьбы с ошибками:

- обнаружение ошибок в блоках данных и автоматический запрос повторной передачи повреждённых блоков;
- обнаружение ошибок в блоках данных и отбрасывание повреждённых блоков [3, 4];
- исправление ошибок [2].

В настоящей работе мы рассматриваем именно последнюю стратегию борьбы с ошибками, а именно – исправление ошибок. С кодами, исправляющими ошибки, тесно связаны коды обнаружения ошибок. В отличие от первых, последние могут только установить факт наличия ошибки в переданных данных, но не исправить её. Мы исследуем возможности реализации этих стратегий с помощью методов анализа формальных понятий (АФП), которые особенно хорошо зарекомендовали себя при работе с бинарными данными.

Никакие исследования по нахождению ошибок в сообществе АФП нам не известны. Однако тема нахождения и коррекции ошибок широко разрабатывается для баз данных и для извлечения данных. Известен целый ряд работ, в которых рассматривается задача нахождения пропущенного или неизвестного значения в данных. Интерес к этой проблеме вызван тем, что при работе с реальными данными очень часто возникает проблема пропущенных значений. Для решения предлагаются разные техники добавления пропущенных значений. В работах [4] и [5] предлагается обзор таких техник, среди которых есть методы игнорирования записей с пропущенными значениями и вставления вместо пропущенных значений некоторых средних по данным. Обсуждаются также и более сложные методы, основанные на деревьях решений, в работе [4] предлагается обзор использования различных нейронных сетей для нахождения пропущенных значений, а в работе [3] обсуждается возможность использования метода ближайших соседей для нахождения пропущенных значений. Как будет показано в настоящей работе, предлагаемые нами подходы могут в самом общем смысле рассматриваться как некоторая модификация метода ближайших соседей; обсуждаемые выше работы рассматривают несколько другие задачи. В случае пропущенного значения не стоит задача найти ошибку, так как из самой постановки задачи понятно какую часть данных необходимо исправить. Кроме того, в обсуждаемых работах не рассмотрен случай анализа исключительно бинарных данных, и многие методы (например, усреднение) не предполагают анализ бинарных данных; описанные некоторые модификации позволяют работать с частично категориальными данными, однако в случае

* Работа выполнена при финансовой поддержке Немецкой службы академических обменов (DAAD).

анализа исключительно бинарных данных необходим другой подход.

В работе [5] предлагается подход к нахождению и устранению ошибок в базах данных с использованием исправляющих правил. Если в этом случае и ставится задача собственно нахождения ошибок, то описанные подходы требуют предварительной работы эксперта с базой данных. Точнее, предполагается априорное задание некоторых исправляющих правил, основываясь на которых в дальнейшем будет осуществлено исправление ошибок. Такой подход не применим в том случае, если эксперт не может задать такие правила до начала проверки.

Наконец, необходимо отметить работу [6], в которой рассматривается задача объяснения фактов в Хорновских теориях. Хотя, на первый взгляд, ставится совсем другая задача, изложенные в этой работе методы и подходы оказываются похожими на используемые в нашей работе. В частности, наличие или отсутствие признака в содержании объекта можно рассматривать как факт, требующий объяснения. В работе [6] также отмечается, что можно находить объяснения наличия/отсутствия признака за полиномиальное время. Как бы то ни было, подход, изложенный в нашей работе, позволяет попытаться одновременно рассмотреть наличие каждого признака в содержании объекта как факта, требующего объяснения.

В настоящей работе мы обсуждаем возможные типы ошибок в содержаниях новых объектов, останавливаемся на двух наиболее общих типах и предлагаем эффективные методы нахождения этих типов ошибок. Мы предлагаем два разных подхода к нахождению ошибок: один основан на использовании базиса импликаций формального контекста, другой основан на вычислении замыканий подмножеств содержания объекта [7]. Так как замыкание может быть вычислено быстро, мы уделяем большее внимание этому подходу и обобщаем его для нахождения всех возможных ошибок выбранных типов. Эта процедура также может быть использована для обоснования корректности выбранных объектов. Как бы то ни было, для доказательства отсутствия ошибок все же необходимо проверить некоторые объекты вручную. Статья заканчивается обсуждением экспериментальных результатов.

Мы предполагаем, что изначально мы имеем проверенные данные, представленные контекстом (возможно пустым), и объекты, которые могут содержать ошибки. Эти данные (контекст и новые объекты) взяты из некоторой предметной области, и мы всегда можем задать вопрос эксперту, который дает только верные ответы. Тем не менее, мы должны задать как можно меньше вопросов.

Все множества, рассматриваемые в настоящей работе, предполагаются конечными.

ОСНОВНЫЕ ОПРЕДЕЛЕНИЯ

Пусть G и M - множества. Пусть $I \subseteq G \times M$ - бинарное отношение между G и M. Тройка $\mathbb{K}:=(G, M, I)$ называется *(формальным) контекстом*. Множество G называется множеством *объектов,* множество M называется множеством *признаков*.

Рассмотрим следующие отображения: $\varphi: 2^G \to 2^M$ и $\psi: 2^M \to 2^G$: $\varphi(X) := \{m \in M \mid gIm$ для всех $g \in X\}$, $\psi(A) := \{g \in G \mid gIm$ для всех $m \in A\}$. Для любых $X_1, X_2 \subseteq G$, $A_1, A_2 \subseteq M$ получим следующие свойства.

1. $X_1 \subseteq X_2 \Rightarrow \varphi(X_2) \subseteq \varphi(X_1)$.
2. $A_1 \subseteq A_2 \Rightarrow \psi(A_2) \subseteq \psi(A_1)$.
3. $X_1 \subseteq \psi\varphi(X_1)$ и $A_1 \subseteq \varphi\psi(A_1)$.

Отображения φ и ψ задают *соответствие Галуа* между $(2^G, \subseteq)$ и $(2^M, \subseteq)$, т. е. $\varphi(X) \subseteq A \Leftrightarrow \psi(A) \subseteq X$. Обычно вместо φ и ψ используют единое обозначение $(\cdot)'$. $(\cdot)'$ называют *штрих-оператором* или *оператором Галуа*. Множество X', где $X \subseteq G$, называется *содержанием X*. Аналогично, множество A', где $A \subseteq M$, называется *объемом A*.

Пусть $Z \subseteq M$ или $Z \subseteq G$.$(Z)''$ называется *замыканием Z* в $\mathbb{K}$. Применяя Свойство 1 и Свойство 2 последовательно получим свойство *монотонности*: для любых $Z_1, Z_2 \subseteq G$ или $Z_1, Z_2 \subseteq M$ имеем $Z_1 \subseteq Z_2 \Rightarrow Z_1'' \subseteq Z_2''$.

Пусть $m \in M, X \subseteq G$, тогда $\overline{m}$ называется *отрицательным* признаком. $\overline{m} \in X'$ в том и только в том случае, если ни один $x \in X'$ не удовлетворяет *xIm*. Пусть $A \subseteq M$; $\overline{A} \subseteq X'$ в том и только в том случае, если все $m \in A$ удовлетворяют $\overline{m} \in X'$.

Импликацией называется пара (A,B), которая обозначается $A \to B$, где $A, B \subseteq M$. A называется *посылкой*, B - *следствием* импликации. Импликация $A \to B$ *удовлетворяется множеством признаков* N, если $A \not\subseteq N$ или $B \subseteq N$. В противном случае говорят, что *множество признаков противоречит* импликации. Импликация $A \to B$ верна в $\mathbb{K}$, если она удовлетворяется всеми g', $g \in G$, т. е. каждый объект, обладающий всеми признаками из A, также обладает всеми признаками из B. Импликации удовлетворяют *правилам Армстронга*:

$$A \to A, \frac{A \to B}{A \cup C \to B}, \frac{A \to B, B \cup C \to D}{A \cup C \to D}.$$

Поддержкой импликации в контексте $\mathbb{K}$ называют все объекты контекста $\mathbb{K}$, чьи содержания включают как посылку, так и следствие импликации. *Атомарной импликацией* называют такую импликацию, в следствие которой содержится только один признак, т. е. $A \to b$, где $A \subseteq M$, $b \in M$. Любая импликация $A \to B$ может быть представлена как множество атомарных импликаций $\{A \to b \mid b \in B\}$. Для любых целей можно ограничиться рассмотрением только атомарных импликаций без потери общности.

Базисом импликаций контекста $\mathbb{K}$ называют множество $\mathfrak{L}$ импликаций, из которого любая верная в контексте $\mathbb{K}$ импликация может быть выве-

дена с помощью правил Армстронга и никакое собственное подмножество множества $\mathfrak{L}$ не обладает таким свойством.

Минимальный по числу импликаций базис был впервые представлен в [8] и известен как *канонический базис импликаций*. В работе [9] посылки этого базиса были описаны в терминах псевдо-содержаний. Подмножество признаков $P \subseteq M$ называется *псевдосодержанием*, если $P \neq P''$, и для каждого псевдосодержания Q такого, что $Q \subset P$, выполняется $Q'' \subset P$. Канонический базис импликаций тогда можно представить в виде:

$$\left\{P \to P'' \setminus P \middle| P - псевдосодержание\right\}.$$

Объект называется *редуцируемым* в контексте $\mathbb{K}:=(G, M, I)$ в том и только в том случае, если $\exists X \subseteq G : g' = \bigcap_{j \in X} j'$.

КЛАССИФИКАЦИЯ ОШИБОК

Обсудим *зависимости в предметной области*. Обычно объекты и признаки контекста являются представлением некоторых сущностей (например, физические объекты, математические сущности, товары и услуги и прочее). На характеристиках (описаниях таких сущностей) могут выполняться какие-то зависимости (например, если объект является выпуклым четырехугольником, то сумма всех улов должна быть равна π). Как бы то ни было, эти зависимости могут не являться верными импликациями контекста в результате некоторых ошибок в описаниях объектов. Таким образом, зависимости в предметной области - это такие правила, которые выполняются на сущностях, представляемых объектами контекста, но они могут ошибочно не быть верными импликациями контекста. Мы решаем проблемы нахождения ошибок в описаниях объектов через восстановление нарушенных зависимостей.

Каждый объект в контексте имеет определенное содержание. На признаках, входящих в эти содержания, могут выполняться зависимости в предметной области. Мы рассматриваем только такие зависимости, которые не имеют отрицательных признаков в посылке. Как уже отмечалось, нет необходимости рассматривать неатомарные импликации. Рассмотрим следующие **типы** зависимостей $(A \subseteq M, b, c \in M)$:

1. $A \to b$;
2. $A \to \bar{b}$;
3. $A \to b \vee c$;
4. $A \to \Phi$, где Φ - любая логические формула кроме написанных выше, например, $\Phi = a \vee (b \wedge \bar{c})$.

Если в контексте отсутствуют ошибки, то все зависимости Типа 1 выводимы из базиса импликаций контекста. Тем не менее, если в контекст добавлено недостаточно объектов, то некоторые импликации из базиса могут быть ошибочными. Как бы то ни было, гарантируется, что если все объекты в контексте не содержат ошибок, то никакие верные в предметной области зависимости не потеряны и с добавлением новых безошибочных объектов мы уменьшаем количество неверных импликаций.

Однако ситуация меняется при добавлении объектов, содержащих ошибки. Такие объекты могут нарушить зависимости, верные в предметной области. В таком случае, пока мы не исправим ошибки, мы не сможем вывести из базиса импликаций контекста все зависимости, верные в предметной области, независимо от того, какое количество безошибочных объектов мы добавим в будущем.

Рассмотрим подробнее возможные случаи. Если зависимость Типа 3 выполняется в предметной области, то должно быть ограничение на добавление редуцируемых объектов. Однако известно, что редуцируемые объекты не изменяют ни систему замыканий контекста, ни базис импликаций [7]. Этот случай является интересным для дальнейших исследований, однако в данной статье мы его не рассматриваем.

В Типе 4 формула Ф может быть представлена в конъюнктивной нормальной форме, что дает основание надеяться, что если задача нахождения ошибок Типа 3 будет решена, то будет найдено и решение для задачи нахождения ошибок Типа 4.

Ошибки Типа 1 и Типа 2 являются наиболее простыми и часто встречаемыми. В настоящей работе мы ищем пути нахождения зависимостей этих двух типов и исправления соответствующих ошибок.

НАХОЖДЕНИЕ ОШИБОК

Предлагаем два подхода к решению задачи нахождения ошибок. Первый основан на работе с базисом импликаций контекста. При добавлении нового объекта в контекст несложно найти все импликации из базиса импликаций, которым противоречит содержание нового объекта. Эти импликации затем выдаются в атомарном виде в качестве вопросов к эксперту. Если хотя бы одна импликация принимается экспертом как верная, то содержание объекта имеет ошибки. Так как канонический базис является наиболее компактным (по количеству импликаций) представлением множества всех верных импликаций контекста, то гарантируется минимальное количество вопросов к эксперту и ни одна из зависимостей Типа 1 не будет забыта.

Хотя такой подход и позволяет найти все зависимости Типа 1, он не лишен недостатков. Задача нахождения канонического базиса является сложной, известные на данный момент алгоритмы обладают высокой вычислительной сложностью. Последние результаты дают основания полагать, что алгоритмы с лучшей вычислительной сложностью найдены не будут [10-13]. Можно использовать и другие базисы импликаций (например, как в работе [14]), но известные для них алгоритмы также являются вычислительно трудными, кроме того нет гарантий, что эксперту будет задано минимальное число вопросов.

Как бы то ни было, раз мы интересуемся только импликациями, которые соответствуют одному

объекту, нет необходимости вычислять весь базис импликаций. Мы предлагаем другой подход. Пусть $A \subseteq M$ - содержание нового объекта, который еще не добавлен к контексту. $m \in A''$ в том и только в том случае, если $\forall g \in G : A \subseteq g' \Rightarrow m \in g'$, другими словами, A'' содержит только признаки, общие для всех содержаний объектов, содержащих A. Множество атомарных импликаций $\{A \to b | b \in A'' \setminus A\}$ может быть выдано как вопросы эксперту. Если все импликации отвергнуты, то никакие признаки не забыты в содержании нового объекта. В противном случае объект содержит ошибки. Такой подход позволяет находить ошибки Типа 1.

Пример

Рассмотрим пример с выпуклыми четырехугольниками. Формальный контекст на рис. 1 содержит выпуклые четырехугольники и их свойства. Контекст не полон, т. е. не все возможные примеры выпуклых четырехугольников находятся в контексте, а некоторые объекты контекста являются редуцируемыми (не приносят никакой новой информации в базис импликаций). Рассматриваются семь признаков. Признаки «есть равные стороны» и «есть равные углы» подразумевают наличие хотя бы двух равных сторон/углов в четырехугольнике. Некоторые зависимости из предметной области очевидны, Например, ясно, что если все углы равны, то найдутся и два равных.

Четыре ошибочных объекта показаны на рис. 2. Ошибки добавляются к контексту на рис. 1 по очереди. Ошибочный объект стоит рассматривать как новый объект, еще не добавленный к контексту.

Контекст без ошибок на рис. 1 обозначается $\mathbb{K}$, $(\cdot)'$ – есть соответствующий оператор Галуа.

Контекст ошибок на рис. 2 обозначается $\mathbb{K}_e$, $(\cdot)^e$ – есть оператор Галуа для $\mathbb{K}_e$.

Пример 1. Ошибка 2^e = {есть равные стороны, есть прямой угол, все стороны равны, все углы равны}.

Ошибка $2^{e''}$ = {есть равные стороны, есть прямой угол, все стороны равны, все углы равны, есть равные углы}.

Выпуклые четырехугольники	есть равные стороны	есть равные углы	есть прямой угол	все стороны равны	все углы равны	хотя бы 3 разных угла	хотя бы 3 разные стороны
Квадрат	×	×	×	×	×		
Прямоугольник	×	×	×		×		
Четырехугольник						×	×
Ромб	×	×		×			
Параллелограмм	×	×					
Прямоугольная трапеция		×	×			×	×
Четырехугольник с 2 равными сторонами и прямым углом	×		×			×	×
Равнобедренная трапеция	×	×					×
Прямоугольная трапеция с 2 равными углами	×	×	×			×	×
Четырехугольник с 2 равными углами		×				×	×
Четырехугольник с 2 равными сторонами	×					×	×
Четырехугольник с 2 равными сторонами и 2 равными углами	×	×				×	×

Рис. 1. Контекст выпуклых четырехугольников $\mathbb{K}$

Ошибки	есть равные стороны	есть равные углы	есть прямой угол	все стороны равны	все углы равны	хотя бы 3 разных угла	хотя бы 3 разные стороны
Ошибка1	×			×		×	
Ошибка2	×		×	×	×		
Ошибка3		×	×	×	×	×	×
Ошибка4	×	×		×			×

Рис. 2. Контекст ошибок $\mathbb{K}_e$

Канонический базис импликаций контекста на рис. 1 выглядит следующим образом:

1) хотя бы 3 разных угла $\rightarrow$ хотя бы 3 разные стороны,

2) все углы равны $\rightarrow$ есть равные углы, есть равные стороны, есть прямой угол,

3) все стороны равны $\rightarrow$ есть равные углы, есть равные стороны,

4) есть прямой угол, хотя бы 3 разные стороны $\rightarrow$ хотя бы 3 разных угла,

5) есть равные углы, есть равные стороны, хотя бы 3 разные стороны, все стороны равны $\rightarrow$ есть прямой угол, хотя бы 3 разных угла, все углы равны,

6) есть равные углы, есть равные стороны, хотя бы 3 разные стороны, все углы равны, есть прямой угол, хотя бы 3 разных угла $\rightarrow$ все стороны равны,

7) есть прямой угол, есть равные стороны, все стороны равны, есть равные углы $\rightarrow$ все углы равны.

Рассмотрим Ошибку 2. Ошибка 2^e = {есть равные стороны, есть прямой угол, все стороны равны, все углы равны}

При использовании первого подхода мы найдем, что этот объект противоречит Импликациям 2 и 3. Несложно заметить, что обе имликации верны в предметной области. Таким образом, эксперт заметит, что в описании объекта допущены ошибки.

Так как Ошибка 2^e" = {есть равные стороны, есть прямой угол, все стороны равны, все углы равны, есть равные углы}, второй подход даст нам следующие импликации: {есть равные стороны, есть прямой угол, все стороны равны, все углы равны} $\rightarrow$ {есть равные углы}. Эти импликации также верны в предметной области. Хотя эти импликации и менее общие, чем Импликации 2 и 3, их достаточно для нахождения ошибок.

УЛУЧШЕНИЯ

Очевидно, что вычисление замыкания намного проще, чем нахождение канонического базиса, и может быть выполнено за полиномиальное время. Однако возможен следующий случай: пусть $A \subseteq M$ такое содержание нового объекта, что $\nexists g \in G : A \subseteq g'$. В этом случае $A'' = M$ и импликация $A \rightarrow A'' \setminus M$ обладает пустой поддержкой. Такая ситуация может свидетельствовать об ошибке Типа 2, так как содержание объекта содержит набор признаков, невозможный в предметной области, но объект также может и не содержать ошибок. Можно задать в таком случае вопрос эксперту о том, возможна ли такая комбинация признаков в предметной области. Однако в таком вопросе не используется информация, которая уже находится в контексте. Более того, такой вопрос не позволит обнаружить ошибки Типа 1.

Утверждение. *Пусть* $\mathbb{K}:=(G, M, I)$, $A \subseteq M$.
Множество

$$I_A = \left\{ B \rightarrow d \middle| B \in \mathcal{MC}_A, d \in B'' \setminus A \cup \overline{A \setminus B} \right\},$$

где $\mathcal{MC}_A = \left\{ B \in C_A \middle| \nexists C \in C_A : B \subset C \right\}$ *и* $C_A = \left\{ A \cap g' \middle| g \in G \right\}$, *является таким множеством всех атомарных импликаций (или их следствий с некоторыми признаками, добавленными в посылку), имеющих вид зависимостей Типов 1 и 2, что они верны в контексте* $\mathbb{K}$, *не удовлетворяются A и имеют непустую поддержку.*

Доказательство.

Пусть $(E \rightarrow f) \in I_A$. Так как $E = A \cap g'$, то для некоторого $g \in G, f \in g'$. Рассмотрим возможные случаи:

1. $f \in E'' \setminus A$. Как следует из определения штрих-оператора, эта импликация верна и $f \in g'$, т. е. хотя бы g находится в поддержке этой импликации. Более того, $E'' \setminus A \not\subseteq A$ и импликация не удовлетворяется множеством A.

2. $f \in \overline{A \setminus E}$. Так как E является максимальным пересечением ($\nexists C \in C_A : E \subseteq C$), не существует объекта $\hat{g} \in G$ такого, что $E \cup m \in \hat{g}'$ для любого $m \in A \setminus E$. Этим доказывается, что импликация верна и хотя бы один объект g находится в поддержке этой импликации. Более того, из того что $A \setminus E \subseteq A$ следует, что A противоречит импликации.

Далее, пусть $E \rightarrow f$ верная импликация контекста, которая не удовлетворяется A и имеет непустую поддержку. Тогда $E \in A, f \notin A$ и существует $g \in G$ такой, что $E, f \in g'$. По построению существует $B \in \mathcal{MC}_A$ такое, что $E \subseteq A \cap g' \subseteq B$.

Рассмотрим возможные случаи.

1. $f \in M$. Из того что импликация верная и не удовлетворяется A следует, что $f \in (A \cap g')'' \setminus A$. По монотонности получим $f \in B''$. Это показывает, что $(B \rightarrow f) \in I_A$.

2. $f \in \overline{M}$. Пусть $f = \bar{\upsilon}$. Так как импликация верна, то не существует $\hat{g} \in G$ такого, что $\upsilon \in \hat{g}'$. Тогда $\upsilon \in A \setminus B$ и $(B \to f) \in I_A$.

Утверждение позволяет написать алгоритм для вычисления множества вопросов к эксперту, которые позволят найти все ошибки Типов 1 и 2. Псевдокод алгоритма весьма очевиден.

Псевдокод алгоритма

Input: $\mathbb{K} = (G, M, I),\ A \subseteq M$
Output: Импликации, которые противоречат A

1 **if** $A'' = A$ **then**
2 ⌊ **return** $\emptyset$
3 Candidates = $\{\text{object}' \cap A \mid \text{object} \in G\}$
4 Candidates = $\{C \in \text{Candidates} \mid \nexists B \in \text{Candidates}:\ C \subseteq B\}$
5 Result $\leftarrow \emptyset$
6 **for** *Candidate* ***in*** *Candidates* **do**
7 ⌊ Result.add($\{\text{Candidate} \to d \mid d \in (\text{Candidate}'' \setminus A \cup \overline{A \setminus \text{Candidate}})\}$)
8 **return** Result

A - содержание нового объекта. В третьей строке мы вычисляем множество всех подмножеств, которые могут дать искомые импликации. В четвертой строке мы избавляемся от всех немаксимальных элементов. В строках 7 и 8 мы вычисляем замыкания и добавляем к результату соответствующие импликации.

При добавлении нескольких новых объектов можно действовать следующим образом. После добавления очередного нового объекта необходимо задать дополнительные вопросы эксперту относительно прежде добавленных новых объектов. Действительно, после добавления очередного нового объекта, который не содержит ошибок, в контексте могут появиться новые импликации с непустой поддержкой. Однако такие вопросы выглядят не иначе, чем вопросы к очередному новому объекту с отрицательным следствием. На самом деле, пусть $B \to c$ - импликация, соответствующая вопросу к объекту. Если эта импликация отвергнута экспертом, как верная зависимость предметной области, и объект добавлен к контексту, то содержание объекта удовлетворяет импликации $B \to \bar{c}$. Так как импликация $B \to c$ была выдана как вопрос к эксперту, прежде не было ни одного объекта контекста, содержание которого удовлетворяло бы импликации $B \to \bar{c}$. Поэтому импликация $B \to \bar{c}$ не была спрошена до этого и должна быть выдана как вопрос в данный момент. Нахождение таких вопросов не требует затрат времени и гарантирует независимость до порядка добавления новых объектов.

Стоит заметить, что рассмотрение только импликаций с непустой поддержкой не всегда безопасно. С одной стороны, это позволяет избегать вопросов, не основанных на уже имеющейся информации, с другой стороны, такой подход не позволяет утверждать полное отсутствие ошибок рассматриваемых типов в содержаниях объектов. Однако для гарантии отсутствия ошибок в любой момент времени достаточно проверить только содержания максимальных объектов, так как только они могут иметь такие комбинации признаков, которые больше нигде в контексте не встречаются. Таким образом предлагается проверять максимальные объекты вручную, если необходимо доказать отсутствие ошибок Типов 1 и 2.

РЕЗУЛЬТАТЫ

Далее для компактности изложения импликации представлены в неатомарной форме. Название inspect_dg используется для обозначения функции, воплощающей первый описанный подход (с использованием канонического базиса).

Пример

Проверка Ошибки 1:

`inspect_dg`

хотя бы 3 разных угла ⟶ хотя бы 3 разные стороны

все стороны равны ⟶ есть равные углы, есть равные стороны

`inspect`

есть равные стороны, хотя бы 3 разных угла ⟶ хотя бы 3 разные стороны, все стороны равны

есть равные стороны, все стороны равны ⟶ есть равные углы. хотя бы 3 разных угла

Оба алгоритма выявляют ошибки похожим образом, хотя есть и очевидные различия. На выходе `inspect_dg` посылки меньше, чем на выходе `inspect`, хотя последний выявляет еще и ошибки Типа 2.

Легко видеть, что все выходные импликации верны как зависимости предметной области. Например, если все стороны четырехугольника равны, то у него должны быть равные углы и не должно быть трех разных углов. Как следствие, объект должен быть распознан как ошибка.

Проверка Ошибки 2:

`inspect_dg`

все углы равны ⟶ есть равные углы, есть равные стороны, есть прямой угол

все стороны равны ⟶ есть равные углы, есть равные стороны

`inspect`

есть прямой угол, есть равные стороны, все стороны равны, все углы равны ⟶ есть равные углы.

В этом примере, используя второй подход, удается задать даже меньше вопросов к эксперту, чем в первом подходе. Меньшее количество вопросов является следствием использования импликаций, порожденных максимальными пересечениями с содержанием объекта. И в этом случае все импликации являются верными зависимостями предметной области. Содержание объекта Ошибки 2 уже встречается в одном из содержаний контекста (содержания квадрата), поэтому на выходе `inspect` нет импликаций с отрицательными признаками в следствии.

Проверка Ошибки 3:

`inspect_dg`

все углы равны $\rightarrow$ есть равные углы, есть равные стороны, есть прямой угол

все стороны равны $\rightarrow$ есть равные углы, есть равные стороны

`inspect`

есть равные углы, есть прямой угол, хотя бы 3 разные стороны, хотя бы 3 разных угла $\rightarrow$ все углы равны, все стороны равны

есть равные углы, есть прямой угол, все стороны равны, все углы равны $\rightarrow$ есть равные стороны. хотя бы 3 разных угла, хотя бы 3 разные стороны

В случае Ошибки 3 обе импликации из выхода `inspect_dg` скомбинированы в одной импликации с большей посылкой в выходе `inspect`. Кроме того, мы получаем несколько импликаций с отрицательным следствием. Легко видеть, что все импликации верны как зависимости предметной области.

Проверка Ошибки 4:

`inspect_dg`

есть равные углы, есть равные стороны, хотя бы 3 разные стороны, все стороны равны $\rightarrow$ есть прямой угол, хотя бы 3 разных угла, все углы равны

`inspect`

есть равные углы, есть равные стороны, все стороны равны $\rightarrow$ все углы равны

есть равные углы, есть равные стороны, хотя бы 3 разные стороны $\rightarrow$ все стороны равны .

Ошибка 4 является как раз таким специальным случаем, когда все соответствующие импликации из канонического базиса обладают пустой поддержкой. На выходе `inspect_dg` мы получаем все возможные вопросы для данного содержания. Как уже указывалось, такие вопросы не основаны на какой-либо информации, введенной до этого. Даже если мы добавим признаки «хотя бы 3 разных угла» и «все углы равны» и отвергнем последнюю импликацию, мы не заметим ошибку в содержании объекта. Напротив, выход `inspect` позволяет найти ошибку Типа 2.

Эксперимент

Приведем результаты тестов на больших объемах данных. Тесты проводились следующим образом: объекты отделяются от контекста один за другим и затем добавляются как новые объекты; находятся все возможные ошибки при использовании обоих подходов.

Пакет FCA, написанный на языке Python, использовался для имплементации [15]. Для вычисления канонического базиса был использован оптимизированный алгоритм, основанный на `Next Closure` [16]. Все тесты проводились на компьютере с процессором Intel Core i7 1.6GHz и 4 Gb оперативной памяти под Linux Ubuntu 11.10 x64.

На рис. 3 показаны результаты тестов на случайных контекстах. Каждый контекст содержит по 50 объектов. Параметр d представляет собой плотность контекста, т. е. вероятность нахождения любой пары (g,m) в бинарном отношении I. Результаты представлены в полулогарифмической шкале, так как сложность вычисления канонического базиса растет по закону, близкому к экспоненциальному. Нетрудно заметить, что с ростом плотности и количества признаков в контексте, разность времен работы двух алгоритмов также возрастает.

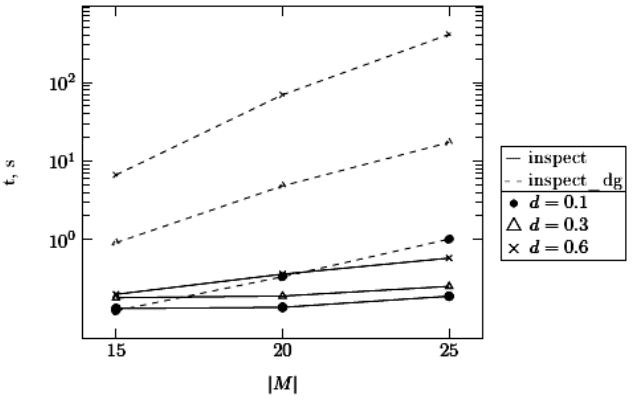


Рис. 3. Сравнение времен работы на случайных контекстах в полулогарифмической шкале

Результаты работы алгоритмов на реальных данных показаны в таблице. Эксперимент проводился таким же образом, как и на случайных контекстах. Данные взяты из хранилища данных UCI [17]. В тестах алгоритм inspect работает значительно быстрее алгоритма inspect_dg; с ростом количества признаков разница по времени становится более значительной.

Сравнение времен работы (в с) на реальных данных, взятых из UCI

Название контекста	|G|	|M|	***inspect(s)***	***inspect_dg(s)***
wine	178	68	4,674	13627,952
house-votes-84	435	18	1,048	64,735
SPECT	266	23	0,636	672,942

ДРУГОЕ ПРЕДСТАВЛЕНИЕ МЕТОДА

Формальным понятием формального контекста *(G, M, I)* называют пару (X,A), где $X \subseteq G$, $A \subseteq M$, $X' = A$ и $A' = X$. Множество X называют объемом, а множество A - содержанием формального понятия (X,A).

В контексте (G, M, I) понятие $K = (X, A)$ *менее общее или равно* понятию $J = (Y, B)$ (иначе $K \leq J$) если $X \subseteq Y$ или эквивалентно $B \subseteq A$. Таким обра-

зом определяется частичный порядок на формальных понятиях.

Частично упорядоченное множество $(L,\leq)$ называют *(полной) решеткой*, если для любого подмножества $S \subseteq L$ определены наибольшая нижняя грань (*инфимум*, обозначается $\wedge$) и наименьшая верхняя грань (*супремум*, обозначается $\vee$).

Можно показать, что множество формальных понятий контекста образует *решетку понятий* по отношению к введенному выше частичному порядку $\leq$.

В решетке элемент $K \in L$ называют *супремум-неразложимым*, если этот элемент не может быть представлен как супремум некоторых других элементов.

Пусть $I, K \in L, I \leq K$. Элемент I называют *нижним соседом K*, если не существует элемента $J \in L$ такого, что $J \neq K, J \neq 1, I \leq J \leq K$. Мы обозначаем наименьший элемент решетки через $\mathbb{O}$, наибольший - через $\mathbb{1}$. Элемент $K \in L$ называют *атомом*, если его нижним соседом является $\mathbb{O}$.

Предложенный метод нахождения и правки ошибок можно рассматривать как классификацию по ближайшим соседям в решетке понятий (точнее, в ее верхней подрешетке, исключая нулевой элемент). Рассмотрим решетку понятий для исходного контекста $\mathbb{K}$ вместе с содержанием нового объекта A. Все рассматриваемые в этом разделе операторы Галуа $(\cdot)'$ действуют в исходном контексте $\mathbb{K}$, но рассматриваемая решетка относится к расширенному контексту вместе с новым объектом.

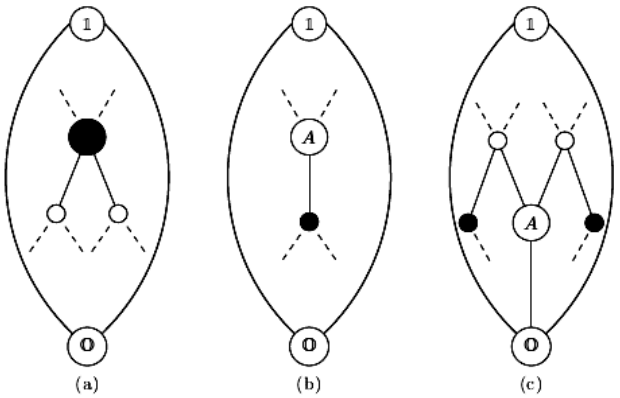


Рис. 4. Поиск ближайших соседей в решетке понятий

Может случиться так, что понятие (X,B), в объеме которого впервые появляется новый объект, является супремум-разложимым, т. е. его можно получить как супремум некоторых понятий (X_1,B_1) и (X_2,B_2), как показано на рис. 4a. В этом случае содержание этого понятия является пересечением содержаний B_1 и B_2. Т. е. $B = A = B_1 \cap B_2$. Тогда $(A \cap X_1')'' = A$, т. е. замыкание максимального пересечения содержания нового объекта с уже имеющимися содержаниями равняется самому содержанию нового объекта. Тогда объект не содержит ошибок Типа 1 и Типа 2.

В другом случае понятие (X,B) является супремум-неразложимым. Рассмотрим ситуацию, когда это понятие не является атомом, как показано на рис. 4b. Пусть нижним соседом будет понятие (X_1,B_1), тогда $A \cap B_1 = A$ и $A'' = B_1$. Таким образом, классификация нового объекта будет происходить по нижнему понятию.

В последнем случае понятие (X,B) является атомом. Тогда мы не можем классифицировать его по нижнему соседу, так как им является нулевой элемент. Максимальные пересечения с содержаниями других объектов приводят к верхним соседям по решетке. Однако по этим понятиям классифицировать нельзя, так как их объемы включают и новый объект, а их содержания зависят от содержания A. Замыкая эти пересечения в исходном контексте, мы получим нижних соседей этих понятий, которые уже не содержат в своем объеме новый объект, так как его нет в исходном контексте. Этому случаю соответствует рис. 4c.

ЗАКЛЮЧЕНИЕ

Нами предложен алгоритм нахождения ошибок Типа 1 и Типа 2 в содержаниях новых понятий в формальных контекстах. В отличие от нахождения ошибок по каноническому базису импликаций контекста, предложенный алгоритм выдает результат за полиномиальное время. Алгоритм интерактивен, т. е. работает с экспертом, который дает правильные ответы на выдаваемые алгоритмом вопросы. Алгоритм является динамическим в том смысле, что с накоплением данных предсказания алгоритма становятся более точными. Кроме нахождения ошибок, алгоритм предлагает пути их устранения. Если алгоритм используется при добавлении всех объектов в контекст для доказательства отсутствия ошибок Типа 1 и Типа 2, то в любой момент достаточно проверить вручную только максимальные объекты.

* * *

Автор благодарит Сергея Олеговича Кузнецова, Сергея Александровича Объедкова и Бернарда Гантера за обсуждения работы и ценные замечания.

СПИСОК ЛИТЕРАТУРЫ

1. Fan W., Li J., Ma S., Tang N., Yu W. Towards certain fixes with editing rules and master data // Proceedings of the VLDB Endowment. – 2010. – Vol. 3, № 1. – P.173–184.
2. Huffman W.C., Pless V.S. Fundamentals of Error-Correcting Codes. – Cambridge, University Press, 2003.
3. Kumar Y., Suryawanshi V. A New Approach for Handling Null Values in Web Log Using KNN and Tabu Search KNN // International Journal of Data Mining & Knowledge Management Process. – 2011. – Vol. 1, № 5.
4. Silva-Ramrez E.-L., Pino-Mejas R., Lopez-Coello M., Cubiles-de-la Vega M.-D. Missing value imputation on missing completely at random data using multilayer perceptrons // Neural

networks : the official journal of the International Neural Network Society. – 2011. – Vol. 24, № 1. – P.121–129.

5. Song Q., Shepperd M. A new imputation method for small software project data sets // Journal of Systems and Software. – 2007.
6. Kautz H.A., Kearns M.J., Selman B. Reasoning with characteristic models // The Eleventh National Conference on Artificial Intelligence (AAAI-93), 1993.
7. Ganter B., Wille R. Formal Concept Analysis: Mathematical Foundations.-1999.
8. Guigues J.-L., Duquenne V. Familles minimales dimplications informatives rsultant dun tableau de donnes binaires // Math. Sci. Hum. – 1986. – Vol. 24, № 95.
9. Ganter B. Two basic algorithms in concept analysis.-1984.
10. Distel F., Sertkaya B. On the complexity of enumerating pseudo-intents // Discrete Applied Mathematics. – 2011. – Vol. 159, № 6. – P. 450–466.
11. Babin M.A., Kuznetsov S.O. Computing premises of a minimal cover of functional dependencies is intractable // Discrete Applied Mathematics, 2012 (doi 10.1016/j.dam.2012.10.026).
12. Kuznetsov S.O., Obiedkov S.A. Counting Pseudo-intents and #P-completeness // Formal Concept Analysis, 4th International Conference, 2006. – P. 306–308.
13. Kuznetsov S.O., Obiedkov S.O. Some Decision and Counting Problems of the Duquenne-Guigues Basis of Implications // Discrete Applied Mathematics. – 2008. – Vol.156, №. 11. – P. 1994–2003.
14. Ryssel U., Distel F., Borchmann D. Fast computation of proper premises // International Conference on Concept Lattices and Their Applications. – 2011. – P. 101–113.
15. Romashkin N. Python package for formal concept analysis. – URL: https://github.com/jupp/fca.
16. Obiedkov S.A., Duquenne V. Attribute-incremental construction of the canonical implication basis //Annals of Mathematics and Artificial Intelligence. – 2007. – Vol. 49. – P.1–4.
17. Frank A., Asuncion A. UCI machine learning repository. – 2010. – URL: http://archive.ics.uci.edu/ml/

Материал поступил в редакцию 19.03.13.

Сведения об авторе

РЕВЕНКО Артём Викторович – аспирант Национального исследовательского университета – Высшая школа экономики (НИУ ВШЭ), Москва
E-mail: artreven@gmail.com

УДК [002:004] : 621.311(470)

Е.Л. Логинов

Информационная платформа, объединяющая телематические, вычислительные и информационные сервисы в ЕЭС России

Проанализированы проблемы формирования конвергентной информационной платформы, объединяющей телематические, вычислительные и информационные сервисы в ЕЭС России с итоговым выходом российской электроэнергетики на новое качество управления на основе принципа самоорганизующейся интеграции.

Ключевые слова: *информационная система, управление, критическая энергетическая инфраструктура*

В электроэнергетике мира и России идет активная работа по внедрению «интеллектуальных сетей» (smart grid). В США и Западной Европе уже реализуется ряд проектов по переходу электроэнергетики и ЖКХ на «интеллектуальные сети».

В России формирование комплекса «интеллектуальных сетей» в различных секторах электроэнергетики закономерно ведет к созданию нового системно-структурного образования, которое можно назвать конвергентной информационной платформой, так как она объединяет телематические, вычислительные и информационные сервисы (рис.1). Различные проявления конвергентной информационной платформы уже реально наблюдаются с середины первого десятилетия XXI в., однако ее комплексное научное исследование еще только начинается. Особенно ярко черты конвергентной информационной платформы в системах управления критической энергетической инфраструктуры в нашей стране видны в Единой энергетической системе России (ЕЭС России).

Перспективой развития конвергентной информационной платформы, объединяющей телематические, вычислительные и информационные сервисы в ЕЭС России, исходя из сложившихся схемно-режимных потребностей, является тенденция, которая в итоге должна привести к упорядоченной взаимосвязанности функционирования и взаимодействия распределенных информационных объектов, информационных сетей и потребителей информации (пользователей данных) за счет интеллектуальных возможностей и многостороннего обмена данными на территориально-организационном уровне на основе принципов их самоорганизующейся интеграции.

Преимуществом конвергентной информационной платформы являются качественно более широкие возможности сбора, обработки, хранения, распределения информации, т. е. способность адаптироваться к динамике информационного спроса и потребления и обеспечение электроэнергетики (с ее технологической составляющей) информацией при современном уровне удовлетворения запросов потребителей.

Сложившаяся информационная инфраструктура в ЕЭС России с ее традиционной, оправданной практикой решения информационно-коммуникационных проблем в сложных условиях информационного, технического, природно-климатического и т.п. характера, ориентацией на крупные объекты и сети информационного назначения, требует новых подходов с учетом задач повышения надежности управления электроэнергетическими объектами.

Такие подходы в нашей стране должны значительно отличаться от практикуемых в большинстве зарубежных информационных образований, так как информационная система ЕЭС России требует качественно иного – более высокого – уровня интегрированности и должна развиваться на основе принципов функционирования больших систем со значительно более высоким уровнем сложности системных взаимосвязей и, соответственно, задач принципиального построения и функционирования.

Решение этой задачи осложняет наличие слабых, но в то же время протяженных информационно-управленческих связей, что ограничивает возможность сбора и анализа больших потоков информации [1].

На основе развития информационных систем критической энергетической инфраструктуры с сегментом конвергентной информационной платформы в ЕЭС России возможен итоговый выход российской электроэнергетики на новое качество управления путем формирования многоуровневой совокупности программно-технических комплексов оперативно-диспетчерского и автоматического управления энергосистемами с гибкими управляемыми элементами активно-адаптивной сети с увеличением объемов автоматизации и повышением количественных и качественных характеристик сбора, обработки, хранения и распределения информации, используемой для принятия управленческих решений.

Формирование конвергентной информационной платформы в ЕЭС России является качественно новым техническим уровнем развития отечественной сферы информационно-коммуникационных технологий (ИКТ) и создает положительный мультипликативный эффект для таких сфер деятельности:

1. Развитие новых информационных технологий (освоение нового поколения инфо-коммуникационных технологий).

Рис.1. Общая схема формирования конвергентной информационной платформы, объединяющей телематические, вычислительные и информационные сервисы

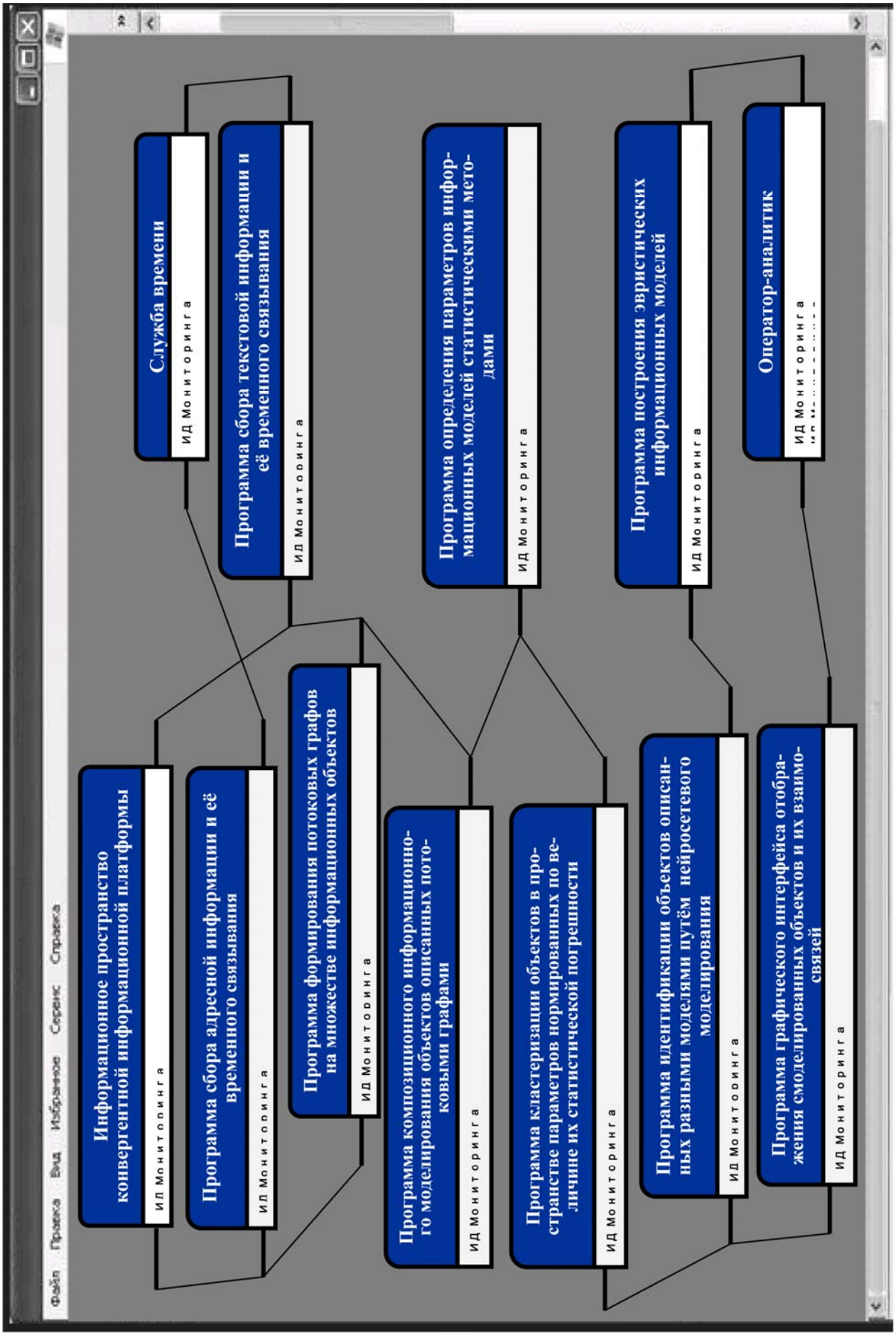


Рис. 2. Схема программного комплекса мониторинга управляющих транзакций в ЕЭС России

2. Разработка новых актуальных направлений по НИОКР, фундаментальным исследованиям, научно-исследовательским работам.

3. Развитие энергетики страны и смежных отраслей, обеспечивающих разработку нового поколения энерго-сетевых технологий с качественно более высокими характеристиками.

4. Повышение эффективности использования данных, ликвидация «информационного разрыва» с применением опыта ведущих стран мира.

5. Востребованность и развитие отечественного научного потенциала, подготовка и профессиональное развитие квалифицированных кадров.

6. Появление новых данных и источников информации о сложных системах критической энергетической инфраструктуры.

Таким образом, общим результатом развития конвергентной информационной платформы является повышение эффективности управления на основе качественно более высокого уровня сбора, обработки, хранения, распределения информации, используемой для принятия управленческих решений на основе сетецентрической парадигмы ее функционирования.

Надо подчеркнуть, что на сегодня механизма (организационного, аппаратного, программного и т.п.), эффективно решающего проблемы обеспечения надежности и безопасности в отношении конвергентной информационной платформы в рамках имеющихся информационно-аналитических моделей деятельности государственных органов и энергетических корпораций, в России не существует [2].

Поэтому, несмотря на внешнюю аналогию традиционных задач создания информационных систем, имеется ряд принципиальных отличий, как в их содержании, так и в методах решения, что вынуждает рассматривать задачу мониторинга электронных управляющих транзакций по интеллектуальным (активно-адаптивным) сетям с учетом деятельности ранее не существовавшего фактора - конвергентной информационной платформы - как существенно новую.

Возможность решения перечисленных задач и проблем заключается в создании информационно-технического комплекса мониторинга электронных управляющих транзакций, встроенного в системы автоматического регулирования и управления ЕЭС России, для выявления электронных управляющих транзакций, опосредующих попытки информационных атак по интеллектуальным (активно-адаптивным) сетям. Глобальной целью автоматизации в сфере мониторинга функционирования объектов оперативно-диспетчерского и автоматического управления энергосистемами в рамках конвергентной информационной платформы в ЕЭС России является обеспечение комплексной информационной, методологической и программно-технологической поддержки процессов обеспечения возможности принятия решений руководством и специалистами государственных органов и энергетических компаний в рамках возложенных на них функций.

Обеспечение мониторинга электронных управляющих транзакций для выявления попыток информационных атак по интеллектуальным (активно-адаптивным) сетям предполагает выбор модели конфигурации информационной системы управления энергообъектами в рамках конвергентной информационной платформы в ЕЭС России. Требуется достижение рационального компромисса между всеми принимаемыми локальными решениями при конфигурировании информационных взаимосвязей отдельных агентов информационных воздействий в интересах выбранной глобальной функциональной цели безопасного и устойчивого развития «интеллектуальных сетей» в аспекте оперативно-диспетчерского и автоматического управления энергосистемами с особо точным поддержанием координации процессов генерации, передачи и распределения электроэнергии в ЕЭС России.

Таким образом, наиболее рациональным подходом к решению проблем управления организацией и реализацией взаимосвязей агентов информационных воздействий в рамках конвергентной информационной платформы, объединяющей телематические, вычислительные и информационные сервисы в ЕЭС России, является разработка и практическое внедрение новой управленческой технологии, базирующейся на методах интеллектуального мониторинга для поддержания координации процессов генерации, передачи и распределения электроэнергии в рамках конвергентной информационной платформы в ЕЭС России.

С точки зрения обеспечения эффективного и надежного функционирования, в архитектуру конвергентной информационной платформы следует включить компоненты, составляющие инвариантное ядро:

1) подсистему мониторинга параметров, которые обеспечивают формирование текущего наблюдаемого состояния конвергентной информационной платформы в ЕЭС России;

2) подсистему диагностики попыток информационных атак по интеллектуальным (активно-адаптивным) сетям, которая на основании данных системы мониторинга может идентифицировать причину аномалии и определять текущее состояние критической энергетической инфраструктуры в ЕЭС России;

3) подсистему анализа текущих характеристик, которая на основании информации, полученной от подсистемы диагностики попыток информационных атак, производит оценку функциональных возможностей конвергентной информационной платформы, объединяющей телематические, вычислительные и информационные сервисы в ЕЭС России, в изменившейся ситуации;

4) подсистему оценки текущего плана управления телематическими, вычислительными и информационными сервисами в ЕЭС России, которая анализирует влияние ухудшения функциональных характеристик систем управления на достижимость поставленной цели и, в случае непригодности существующего плана, выдает соответствующее предупреждение;

5) интеллектуальный агент, который с учетом возникших изменений в состоянии конвергентной информационной платформы в ЕЭС России, осуще-

ствляет перепланирование или корректировку плана работы телематических, вычислительных и информационных систем.

Прикладная часть системы мониторинга должна включать следующие основные модули:

- систему мониторинга в реальном времени элементов и процессов в конвергентной информационной платформе в ЕЭС России;
- систему моделирования, распознавания и анализа глобальной текущей ситуации и поддержки принятия решений при попытках информационных атак по интеллектуальным (активно-адаптивным) сетям и в нормальных режимах;
- систему прогнозирования в реальном времени развития нарушений во времени и распространения нарушений по взаимодействующим подсистемам в конвергентной информационной платформе в ЕЭС России;
- интеллектуальный агент реального времени для корректировки плана работ при возникновении попыток информационных атак по интеллектуальным (активно-адаптивным) сетям и оптимизации решений на основе аппаратов планирования для оперативно-диспетчерского управления;
- интерфейс для выдачи рекомендаций операторам с целью стабилизации технологического процесса управления в критической энергетической инфраструктуре в ЕЭС России;
- решатель реального времени для выдачи оперативному персоналу рекомендаций по парированию возникших попыток информационных атак по интеллектуальным (активно-адаптивным) сетям, а также заключения о возможности (правильности) применения управляющих воздействий в данной текущей ситуации;
- подсистему отображения текущей ситуации в конвергентной информационной платформе в ЕЭС России;
- интерфейсы пользователей в конвергентной информационной платформе в ЕЭС России.

Ожидается, что информация о результатах мониторинга управляющих транзакций предупредит о приближении критической ситуации, выявив внешнее электронное деструктивное воздействие. Индикаторы системной ситуации помогут избежать принятия операторами избыточных или недостаточных мер. Использование предлагаемых алгоритмов позволяет приблизиться к оптимальным предельным значениям параметров сетевых ситуаций в режиме реального времени таким образом, что позволит подойти «ближе к краю» предельных значений параметров энергосистемы, вследствие увеличения возможностей наблюдения и управления [3].

На рис. 2 приводится схема программного комплекса мониторинга управляющих транзакций в информационном пространстве конвергентной информационной платформы.

Выявление скрытых «ядер» систем распределенных событий или факторов операций производится по попыткам информационных атак по интеллектуальным (активно-адаптивным) сетям и связанных с ними субъектов агрессии. В том числе, когда эти «ядра» имеют подвижный (блуждающий) характер в массе (поле) анализируемых событий или факторов и реализуются латентным образом по различным активно-адаптивным каналам, чье наличие в ЕЭС России другими способами контроля выявить крайне проблематично.

В предлагаемом комплексе мониторинга использованы передовые методики анализа деятельности формализованных и неформализованных структур, имеющие принципиальные отличия от ранее существовавших информационно-аналитических систем, поскольку функционируют не по традиционным методикам, базирующимся на методах анализа по интегральным показателям, а на основе построения информационной модели связей мультиагентных компонентов критической энергетической инфраструктуры на основе неявных (случайных, хаотических, несистемных) причинно-следственных зависимостей, имеющих в своей основе латентно организованный характер.

Таким образом, нейросетевое моделирование деятельности мультиагентных компонентов критической энергетической инфраструктуры в ЕЭС России с использованием информации, полученной путем анализа трафика электронных управляющих транзакций, позволяет прогнозировать деятельность как формализованных, так и неформализованных структур, в том с числе с применением сценарного моделирования. При этом выявляются системные параметры сходств различных структур и процессов, ассоциативное подобие конфигурации связей или динамики процессов, референциальные или квазиреференциальные единицы (при анализе совпадений моделируемой динамики с динамикой реальных субъектов, событий, территорий, цифровых пороговых границ и т.п.).

СПИСОК ЛИТЕРАТУРЫ

1. Бугаев А.С., Логинов Е.Л., Райков А.Н., Сараев В.Н. Семантика сетевых контактов // Научно-техническая информация. Сер. 1. – 2009. – № 2. – С. 33-36.
2. Иванов Т.В., Иванов С.Н., Логинов Е.Л., Наумов Э.Б. Интеллектуальная электроэнергетика. – М.: Изд-во «Спутник+», 2012. – 304 с.
3. Логинов Е.Л. Проблемы повышения надежности управления объектами критической инфраструктуры на основе методов композиционного и нейросетевого моделирования. – М.: НИЭБ, 2011.– 241 с.

Материал поступил в редакцию 23.02.13.

Сведения об авторе

ЛОГИНОВ Евгений Леонидович – доктор экономических наук, зам. генерального директора АНО «Научно-исследовательский институт экономических стратегий» (АНО «НИИ ИНЭС», Москва)
E-mail: evgenloginov@gmail.com

АВТОМАТИЗАЦИЯ ОБРАБОТКИ ТЕКСТА

УДК 81’322.2

Ю. С. Акинина, И. О. Кузнецов, С. Ю. Толдова

Сравнение двух методов автоматического извлечения участников события из неструктурированных источников*

Описывается одна из задач извлечения информации из текста, а именно – извлечение фактов (событий разного типа) из неструктурированных источников. Рассматриваются различные методы определения существительных, которые являются типовыми наименованиями участников события. Для решения данной задачи предлагается использовать статистические методы выделения коллокаций. Исследуются два подхода: выделение коллокаций на основе простой контекстной близости существительных к глаголу и выделение коллокаций на основе синтаксических связей существительного с глаголом. В результате сравнения двух методик авторы приходят к выводу, что вопреки первоначальной гипотезе о том, что информация о синтаксических связях должна обеспечить более точное и полное выделение участников и других характеристик событий, первая методика дает о них более полное представление. Анализ результатов показывает, что для успешного применения синтаксического фильтра необходимо учитывать опосредованные синтаксические связи.

Ключевые слова: *коллокации, глагольная сочетаемость, автоматический синтаксический анализ, корпусные методы*

ВВЕДЕНИЕ

Проблема извлечения информации из текста для автоматического/ автоматизированного пополнения онтологии

В последнее время в сфере информационных технологий все более острой становится проблема работы с большим объемом разнородных и слабоструктурированных данных. На передний план выходит задача упорядочивания, систематизации накопленных в различных областях деятельности знаний. В этой связи акцент в обработке контента смещается с простых задач информационного поиска на задачу предоставления пользователю некоторой обобщенной структурированной информации, полученной на основе агрегации информации из первичных источников. Для решения этой задачи были разработаны принципы и технологии семантического Веба. В основе идеологии семантического Веба лежит идея записи информации в виде семантической сети с помощью онтологий. Данная задача особенно актуальна для развития разных областей науки и техники, поскольку представление о важнейших достижениях и инновациях в некоторой научной сфере является неотъемлемым условием получения научных результатов в этой области.

В рамках развития направления семантического Веба во всем мире ведется активная разработка онтологий различных научно-технических областей. С одной стороны, такие онтологии разрабатываются при участии экспертов в данных областях. С другой стороны, поскольку в сети появляются все новые текстовые сообщения о каких-то событиях, достижениях, мероприятиях в соответствующих отраслях науки и техники, встает задача автоматизированного или автоматического пополнения таких онтологий информацией, извлеченной из неструктурированных источников. Таким образом, задача извлечения информации о некотором событии из текста оказывается востребованной для автоматического пополнения онтологий.

* Работа проводилась при финансовой поддержке Министерства образования и науки Российской Федерации в рамках государственного контракта № 07.524.11.4005 от «20» октября 2011 г., заключенного между Министерством образования и науки Российской Федерации и ЗАО «Эвентос».

Если мы имеем дело с неструктурированным текстом, то достаточно часто основным носителем информации о типе события является глагол (1) либо устойчивое словосочетание ‘Глагол+Существительное’ (2), а находящиеся рядом с ними существительные нередко обозначают типовых участников события (1):

(1) *Новый* ***материал*** *из углерода* ***синтезировали*** ***ученые*** *из университета Тунцзи в Шанхае.*

(2) *Потенциально нанотрубки* ***имеют*** *невероятно широкую* ***сферу применения.***

В примерах выделение используется для обозначения названий событий, а выделение и подчеркивание – для обозначения типовых участников или существительных, входящих в устойчивый оборот. А существительное *Тунцзи*, например, не является ни типичным актантом или обстоятельством места для события из (1), ни второй частью устойчивого словосочетания, как в (2).

Таким образом, выделение именно “типических” пар ‘V–N’ (глагол-существительное), во-первых, поможет автоматизировать построение онтологии экспертом, поскольку позволит выявить типовых участников того или иного события (а также их роли), во-вторых, оказывается полезным в задаче тезаурусного расширения соответствующих узлов онтологии: выделения синонимов, гипонимов и т.п.

Задача выделения ‘V–N’ коллокаций

Представляется, что задача нахождения именно типичных, а не случайных участников события может быть решена с использованием технологий выделения коллокаций. Задачу выделения коллокаций можно условно разделить на два этапа:

1) отбор кандидатов в коллокаты;

2) ранжирование пар по степени связанности.

Существует много методов как для отбора кандидатов, так и для их ранжирования. В настоящей работе используется один из стандартных методов ранжирования (PMI). Вопрос о критериях выбора метода остается за рамками настоящего исследования. В центре внимания находится вопрос о выборе кандидатов в коллокаты.

С одной стороны, существительные, обозначающие типичных участников, должны устойчиво встречаться достаточно близко к соответствующему глаголу. То есть можно предположить, что для извлечения интересующих нас ‘V–N’-коллокационных пар при сравнительно большом объеме корпуса достаточно простой информации о совместной встречаемости существительного с глаголом в пределах некоторого контекста. При этом полученное множество типичных существительных может содержать много «шума».

Альтернативная гипотеза заключается в том, что выделение типичных участников может основываться на информации о синтаксических связях глагола вида ‘глагол–существительное’. Эта гипотеза базируется на представлении о том, что основные участники события являются актантами, а в некоторых случаях и сирконстантами соответствующего глагола (ср., например, теорию концептуальных схем (Р. Шенк, Р. Абельсон [1]), фреймовую теорию Филлмора [2]). Здесь и далее для характеристики синтаксических связей используется синтаксическое представление в терминах грамматики зависимостей. Синтаксически-ориентированный подход, предположительно, должен давать лучшие результаты. С одной стороны, среди существительных, расположенных близко к глаголу, учитываются только синтаксически связанные с ним, с другой, в кандидаты попадают связанные с глаголом существительные, находящиеся от него на большом расстоянии.

Проведенные нами серии экспериментов показали, что предположение о том, что синтаксически-ориентированный подход должен дать безусловно лучшие результаты, не оправдалось. Ниже остановимся на описании и анализе результатов эксперимента более подробно.

МЕТОДЫ ВЫДЕЛЕНИЯ КОЛЛОКАЦИЙ

Понятие коллокации

Прежде всего, необходимо определить, что представляют собой коллокации, какие методы для их выделения используются.

Во многих теоретических работах этот термин используется по отношению к несвободным устойчивым словосочетаниям. С одной стороны, они не обладают некомпозициональностью, как, например, фразеологизмы. С другой стороны, второй компонент коллокации не может быть свободно заменен на синоним (ср., например, *strong tea* vs. **powerful tea* [3], см. также [4]). При определении коллокаций для некоторой лексемы учитывается ее типичное и постоянное окружение (см., например, работы Дж. Р. Ферс [5], Х. Джексона [6] и др., а также работы Е. Г. Борисовой [7]).

Теоретические определения коллокаций, основанные на понятии контекстной предсказуемости [5], к сожалению, дают слишком размытые критерии для выделения таких единиц (например, не вполне понятно, является ли словосочетание *произнести речь* связанным). Исследование коллокаций является также актуальным направлением корпусной лексикографии. В рамках этого направления предлагается некоторая «объективизированная» процедура оценки контекстной «предсказуемости»: коллокациями считаются два или более слова, которые встретились рядом в некотором корпусе чаще чем случайно. То есть используются некоторые статистические процедуры оценки «неслучайности» того, что две лексемы оказались рядом. Кроме того, использование статистических критериев позволяет ранжировать словосочетания по степени «коллокационной» связи. Базовые методы ранжирования пар слов по степени «связанности» описаны в [8].). Среди этих методов наиболее упоминаемыми в литературе являются взаимная информация (PMI), критерий Стъюдента (T-score), мера LogLikelihood и др. (подробнее см. [8–11] и др.). Такой подход позволяет подойти к понятию «устойчивости» более гибко (обсуждение проблем традиционного лексикографического vs. статистического методов определения коллокаций см., в частности, [9, 12]).

Поскольку большинство статистических методов ранжирования чувствительны к частотности пары в исследуемом корпусе текстов, то в верху такого ранжированного списка оказываются как фразеологизированные словосочетания типа *ломать голову*, так и глаголы с их наиболее частотными и типичными актантами, такие как *ломать руку*.

В описываемом ниже эксперименте в качестве целевых коллокаций рассматриваются оба класса случаев.

Методы отбора кандидатов в коллокации

Как было сказано, необходимым этапом в процедуре выделения коллокаций является этап отбора кандидатов. Принципиальным является противопоставление двух подходов: подхода, основанного на простой контекстной близости (далее: контекстный подход), и подхода, основанного на синтаксических связях существительных с глаголом (синтаксический подход). Первый подход предполагает нахождение кандидатов в коллокаты для некоторого глагола в числе существительных, находящихся на расстоянии не более n (n слов справа и n слов слева) от этого глагола [10, 13–15]. В данном случае говорят об окне от –n до +n словоупотреблений. Поскольку нас интересуют типичные участники события, для выбора кандидатов используется также частеречный фильтр, т.е. рассматриваются только существительные (ср. также [16, 14]).

Альтернативный метод отбора кандидатов в коллокаты для некоторого глагола – это отбор на основе синтаксического анализа: во множество кандидатов попадают существительные, связанные с глаголом синтаксической связью [17–19]. В этом отношении особый интерес представляет система Sketch Engine [18, 19]. В рамках этого проекта разработана система создания лексического портрета слова на основе его лексико-грамматической сочетаемости, т.е. на основе выделения коллокатов слова с учетом определенного типа синтаксической связи. В [18] и целом ряде других работ утверждается, что список коллокатов, полученный простым контекстным методом, содержит много «шума». Опора на синтаксические связи существительного позволяет этого избежать, а также позволяет учитывать существительные, которые линейно расположены достаточно далеко от глагола [18]. При этом учет конкретных типов синтаксических связей существенным образом повышает точность.

Предыдущие исследования по извлечению V–N коллокаций

Методы выделения ‘V–N’ словосочетаний занимают особое место в исследованиях, посвященных коллокациям. Во-первых, выделение несвободных ‘V–N’ коллокаций играет важную роль при исследовании так называемых “light verb constructions”. Именно такие словосочетания представляют интерес как для лексикографов, так и в теоретическом плане. Они имеют большое значение при изучении иностранного языка, их необходимо включать в словари. Возможность автоматического или полуавтоматического выделения таких конструкций статистическими методами активно исследовалась для английского, французского, немецкого языков (например, в работах [13–15] и др.). Авторы этих работ отмечают, что привлечение информации о синтаксических связях влияет на повышение точности (см., например, [14]). Необходимо, однако, отметить, что в задачи исследователей входило извлечение несвободных конструкций с глаголами. Соответственно, интерес представлял ограниченный набор синтаксических отношений, например, отношение ‘Глагол–Объект’, как в *to make a suggestion* [15]. Выявление некомпозициональных V–N коллокаций является актуальной задачей в системах машинного перевода [20, 21].

Другое направление исследований ‘V–N’ пар – это нахождение типичных участников ситуации, в том числе существительных, которые могут занимать позицию объекта при некотором глаголе. Выделение лексем, ассоциированных с определенными глаголами, например, с глаголами *to drink* и *to eat*, позволяет получить семантические классы существительных (‘жидкости’ vs. ‘еда’, см. “What can you drink?” [10]), а также позволяет задавать сочетаемостные ограничения для некоторой лексемы или конструкции, которые помогут разрешить многозначность (см., например, [10]). Информация о типичных актантах глагола позволяет более точно извлекать модели управления глаголов, в том числе и в задаче определения семантических ролей (semantic role labeling [22]), а также при разрешении семантической неоднозначности глаголов [23].

Исследование ‘V–N’ пар статистическими методами для русского языка представлено, например, в работе [16]. В ней исследуются возможности корпуса сверхбольшого объема (более 1 млрд словоупотреблений) для составления словаря глагольной сочетаемости. Синтаксический анализ при обработке корпуса не проводился: вместо этого было сделано предположение, что группы слов, удовлетворяющие некоторым шаблонам, с большой вероятностью представляют собой синтаксически связанные сочетания (например, «следующая за единственным глаголом группа существительного синтаксически подчиняется данному глаголу»). Действительно, гипотеза подтверждается на большом объеме текстов: авторы говорят о 99% точности результата. Однако использование корпуса исключительно большого объема, как в [16], в большинстве случаев невозможно, а для менее объемных корпусов все перечисленные ранее проблемы остаются актуальными.

В рамках задачи, поставленной в настоящей работе, нас интересуют любые существительные, семантически связанные с глаголом. В частности, при использовании синтаксического метода отбора коллокатов именные коллокаты не дифференцируются по типу синтаксической связи, учитываются все существительные, синтаксически связанные с глаголом.

ЭКСПЕРИМЕНТ

Постановка задачи

Как уже отмечалось, целью эксперимента является сравнение коллокаций ‘глагол–существительное’, извлеченных из большого корпуса с использованием синтаксического метода и контекстного метода.

Корпус

Для получения достаточно надежных статистических данных в исследовании использовался корпус достаточно большого объема – приблизительно 9 млн словоупотреблений[1]. Корпус состоит из случайных предложений, извлеченных из различных новостных статей, опубликованных в период с апреля 2011 по апрель 2012 года.

Поскольку корпус охватывает достаточно компактный период времени и представляет собой тексты о событиях, произошедших в этот период, то он не является сбалансированным и дает картину, несколько смещенную относительно употребления тех или иных лексем. Тем не менее, разумно предположить, что при сравнении двух методов на одном и том же материале такой смещенностью можно пренебречь.

Предварительная обработка корпуса

Для обработки корпуса использовался набор инструментов, разработанных С. Шаровым И. Нивром [24]. Токенизация и морфологический анализ производился с использованием инструмента TreeTagger [25]. Параметры для русского языка были получены Шаровым на основе подкорпуса Национального корпуса русского языка со снятой омонимией. Была также произведена лемматизация – лемматизатором на основе CSTLemma [26]. Синтаксический анализ в формализме грамматики зависимостей производился парсером для MaltParser [27], обученным на корпусе SynTagRus [28].

Обработанные тексты были помещены в реляционную базу и проиндексированы. В итоговой базе содержатся словоформы; каждой словоформе приписан набор морфологических характеристик и лемма, и для каждого предложения представлен набор отношений-зависимостей. Шаров и Нивр сообщают о 95–97% точности частеречной разметки [24]. Синтаксический анализатор также демонстрирует достаточно высокую точность. По результатам внеконкурсного участия парсера в Форуме по оценке работы систем автоматического синтаксического анализа, точность синтаксического анализа по неразмеченным связям составила 91%.

В эксперименте не учитывались типы синтаксической связи, поскольку отсутствуют данные о точности маркирования синтаксических связей. Представляется, что большая часть ошибок компенсируются большим объемом анализируемого корпуса.

Выбор экспериментальных глаголов

В эксперименте рассматривались не все глаголы, встретившиеся в корпусе. Во-первых, рассматривалась сочетаемость только финитных форм глагола. Это связано с тем, что финитные и нефинитные формы имеют разные наборы активных и пассивных синтаксических валентностей: так, например, в причастном обороте существительное может выступать вершиной, а не зависимым, как при финитной форме глагола. В финитной предикации агенс, выраженный именной группой, – зависимое от глагола, в причастном обороте агенс становится вершиной, в деепричастном обороте агенс выражен нулем, именная группа, соответствующая агенсу, оказывается в другой предикации (сравним, например: (а) *принятое <– решение*; (б) *суд <– принял решение*; (с) *приняв решение, суд …*). К тому же нефинитные формы глагола нередко омонимичны отглагольным прилагательным или субстантивированным причастиям/прилагательным: например, *данные*, *арестованный*.

Во-вторых, рассматривались только относительно частотные глаголы, чья абсолютная частота превышает в корпусе 100 словоупотреблений. Таких глаголов на корпус объемом 10 млн оказалось около 500.

Процедура извлечения коллокаций

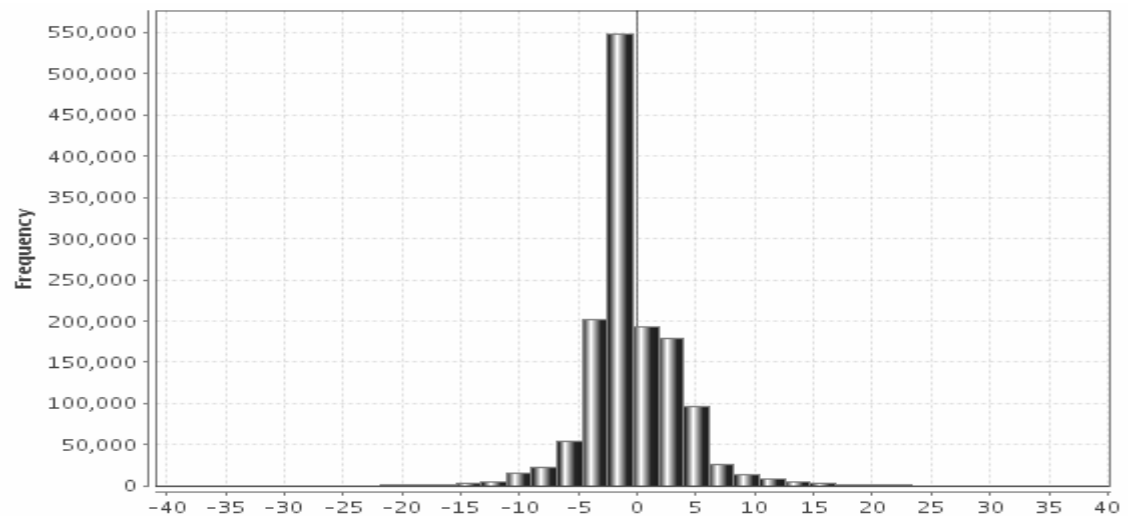
Как уже говорилось выше, коллокации извлекались из синтаксически размеченного корпуса с использованием двух разных стратегий формирования исходных списков кандидатов. Нас интересовали только существительные независимо от конкретного типа синтаксической связи и падежного оформления.

Первая стратегия состояла в том, чтобы формировать потенциальные пары коллокатов, извлекая зависимости ‘глагол–существительное’ безотносительно к типу синтаксической связи. Учитывались также актанты и сирконстанты глагола, оформленные предложными группами. То есть в случае управления предлогом он «пропускался» и в кандидаты попадало соответствующее существительное: так, кандидатом для глагола держать является пара ‘держать–рука’ в примере (3). Всего было извлечено 358 915 ‘V–N’ пар. Далее мы будем называть коллокации, извлеченные с использованием синтаксической информации, синтаксическими коллокациями (3).

Вторая стратегия заключалась в использовании окна заданной длины. Чтобы выбрать подходящий размер окна, было рассчитано распределение расстояний между глаголами и их зависимыми существительными. Распределение расстояний представлено на графике, который наглядно показывает, что при расстоянии больше 5 в обе стороны количество существительных, связанных с глаголом, резко падает.

[1] Корпус был собран H. Christensen и доступен по адресу http://corpora.heliohost.org.

(3) В руках участники акции держали плакаты " Народу нужна справедливость , а не фронт бюрократов " .



Распределение расстояния от глагола до его аргументов

Таким образом, было решено ограничить длину окна пятью словоупотреблениями справа и пятью словоупотреблениями слева. Не-слова, например, знаки препинания и числа, игнорировались. Для всех извлеченных пар были сформированы списки коллокационных кандидатов, состоящие из леммы глагола, леммы существительного и частоты совместной встречаемости коллокации. Методом окна была извлечена 708 131 пара коллокаций. Далее мы будем называть коллокации, полученные этим методом, **контекстными коллокациями**.

Кандидаты на коллокации ранжировались с помощью метрики PMI, которая рассчитывается по формуле 1:

$$PMI(x,y) = \log \frac{P(x,y)}{P(x) * P(y)}$$

Формула 1. Pointwise mutual information

Фрагмент верхней части списка коллокаций приведен в табл.1.

Таблица 1

Значения PMI для синтаксических пар ‘глагол–существительное’

verb	*noun*	*PMI*
произвести	фурор	14.6247
сойти	конвейер	13.4103
нанести	урон	13.2984
внести	лепта	13.2297
сойти	рельс	13.2050
удовлетворить	ходатайство	12.9685
пропасть	весть	12.9394
внести	корректива	12.9366
потерпеть	крушение	12.8547
выдать	ордер	12.6185
потерпеть	поражение	12.5566

PMI как статистическая мера связанности слов имеет несколько недостатков, в частности, она переоценивает значимость редких словосочетаний ([29] и др.). С этим недостатком обычно справляются, используя пороги отсечения по частоте. Однако, если порог частоты излишне завышен, то в коллокационные кандидаты попадет слишком мало пар, а если он слишком низкий, то возникнет много «шума». Поскольку заранее трудно определить, какой порог по частоте оптимален, была проведена серия экспериментов с различным порогом отсечения пар по частоте: 10, 5 и 2. Зависимость состава кандидатов от порога отсечения проиллюстрирована примером (4).

(4) сломать

c10wc10 syntax, **window**: *рука, нога*

c5wc5 syntax, window: *нога, нос, ребро, рука*

c2wc2 syntax: *нога, нос, ребро, результат, рука, челюсть*

c2wc2 window: *андрей, бедро, год, женщина, камера, лицо, мальчик, матч, нога,*

нос, падение, палец, побои, раз, ребро, результат, рука, челюсть, шея

Нотация *Cn* используется для обозначения порога совместной встречаемости величиной *n* в синтаксической модели, а *WCn* – для порога величиной *n* в контекстной модели. Например, *c10wc5* обозначает комбинацию порогов, при которой применялись пороги 10 и 5 для синтаксиса и окна соответственно.

Как видно из примера, порог отсечения 10 для синтаксических коллокаций оставляет только двух кандидатов – наиболее частотные существительные (ср. $fr_{нога}$=1187, $fr_{рука}$=3420 vs. $fr_{ребро}$=54). При этом существительные в списке для порога отсечения 10 и в списке для порога отсечения 5 не различаются семантическим классом.

Методы сравнения

Списки кандидатов для каждого глагола были ранжированы по PMI; для дальнейшего анализа были выбраны только 20 коллокаций из верхней части списка. Чтобы сравнить степень пересечения списков для каждого глагола, полученных двумя методами, были вычислены два значения взвешенной меры пересечения по формуле 2:

$$WI(A,B) = \frac{|x \in (A \cap B)|}{|x \in A|}$$

Формула 2. Взвешенная мера пересечения

Пусть Window – список коллокаций в пределах окна, syntax – список синтаксических коллокаций. Взвешенная мера пересечения WI(Window,Syntax) показывает, насколько список, полученный контекстным методом, включен в список, полученный синтаксическим методом. Мера WI(Syntax,Window), наоборот, отражает пропорцию слов из синтаксического списка, представленную в контекстном списке. Эти меры можно рассматривать как аналогию с традиционными оценками, принятыми в информационном поиске, – точностью (Presicion) и полнотой (Recall), при условии, что синтаксический список является эталоном. По аналогии с точностью и полнотой по стандартной формуле было рассчитано гармоническое среднее F между этими двумя переменными:

$$F_1 = \frac{2 * WI(window, syntax) * WI(syntax, window)}{WI(window, syntax) + WI(syntax, window)}$$

Формула 3. F-мера для WI

Результаты

В результате проведенных экспериментов были получены меры для степени совпадения контекстного списка с синтаксическим (WI(Window,Syntax) и степени пересечения синтаксического списка с контекстным (WI(Syntax,Window)). Сравнение результатов представлено в табл. 2.

Анализ результатов показывает невысокое пересечение списков. Наивысшее значение F_1 достигается при использовании порога, равного 10 для обоих алгоритмов (см. табл. 2). Однако при более подробном анализе результатов видно, что высокая F-мера при одинаковых порогах для синтаксических и контекстных кандидатов (10–10) достигается не столько за счет сближения результатов двух списков, сколько за счет того, что синтаксические списки во многих случаях полностью вложены в контекстные, но имеют существенно меньший объем. То есть усредненное значение WI(Window,Syntax) значимо выше, чем усредненное значение WI(Syntax,Window). Это отражает тот факт, что большинство слов, полученных при помощи синтаксического списка, включены в контекстные списки, в то время как обратное неверно. Так, например, для соотношения порогов синтаксических и контекстных списков 10–10 количество глаголов, для которых синтаксический список полностью вкладывается в контекстный, составляет почти 60% (для 247 глаголов из 418). Наивысшее значение WI(Syntax,Window) достигается при значении порога 5 для синтаксиса и значении порога 10 для оконного метода отбора кандидатов.

Таблица 2

Сравнение списков синтаксических и контекстных коллокаций

WI(window,syntax)				*WI(syntax,window)*			
	wc10	**wc5**	**wc2**		**wc10**	**wc5**	**wc2**
c10	0.621086	0.278444	0.117607	**c10**	0.937296	0.840384	0.604937
c5	0.79878	0.550382	0.20042	**c5**	0.605834	0.880747	0.641428
c2	0.678665	0.666004	0.496304	**c2**	0.228424	0.449963	0.696687
average=0.48974				*average=0.65396*			

F-measure			
	wc10	**wc5**	**wc2**
c10	0.718474	0.384284	0.174289
c5	0.663675	0.647521	0.269283
c2	0.30926	0.511223	0.555424

В соответствии с первоначальной гипотезой, использование синтаксиса должно было дать немного «шума», списки же коллокаций, полученные контекстным методом, – существенно больше и страдать от недостатка точности. Однако экспертный анализ показывает, что коллокации, извлеченные контекстным методом и отсутствующие в синтаксических списках, в значительном числе случаев отвечают поставленной задаче и могут считаться правильными.

ОБСУЖДЕНИЕ

Состав списков

Коллокации и типичные актанты.

Рассмотрим более подробно, какие существительные в результате попали в списки коллокатов для соответствующих глаголов по PMI. Примеры из табл. 1 показывают, что методы, использованные в работе, позволяют находить устойчивые несвободные словосочетания с глаголом. В таблице представлены пары ‘глагол–существительное’ с самым высоким PMI. Как видим, все 10 первых словосочетаний являются идиоматичными (*произвести фурор, сойти с конвейера* и пр.).

Как и ожидалось, в списках представлены как существительные, образующие с глаголом несвободные словосочетания, так и существительные, обозначающие типичных для данного глагола участников ситуации или обстоятельства места и времени. Рассмотрим пример (5):

(5) прочитать c10wc10
syntax: *интернет, книга* **window**: *интернет, книга, лекция*

В данном примере представлено как несвободное сочетание *прочитать лекцию*, так и типичный актант глагола *прочитать – книга*. В контекстном списке также представлен другой актант, отражающий реалии XXI века, и связанный с глаголом через предлог: *прочитать в интернете*.

Пример (6) иллюстрирует тот факт, что предложенный в работе метод позволяет выделять всех участников ситуации, а также обстоятельства. Так, в списках для глагола *выехать* представлены участники ситуации и обстоятельства места, типичные для новостных текстов и для события ‘действия внутренних войск’: агенсы – *группа, полиция, сотрудник (управления)*; средства – *автомобиль, машина*; цель – *полоса (встречная полоса), место (место происшествия), область*.

(6) выехать c5wc5
syntax: *автомобиль, группа, место, полоса, раз* **window**: *автомобиль, глава, год, группа, движение, дом, машина, место, область, полиция, полоса, происшествие, сотрудник, управление, человек*

Случаи «шума».

Как видно из примера (6), в списки могут попадать высокочастотные существительные, типичные для сирконстантных позиций, такие как *год*. Они могут встречаться практически с любым глаголом и, следовательно, не являются «типичными» для конкретного глагола участниками ситуации. Их можно отнести к классу «шума» с точки зрения поставленной в работе задачи. Данные существительные значительно чаще встречаются в контекстных списках.

Второй источник «шума» – это частотные одушевленные существительные, такие как *человек*. … Это связано с тем, что у достаточно большого процента глаголов в качестве подлежащего выступают одушевленные агенсы.

С другой стороны, некоторые классы одушевленных существительных, а именно имена собственные, попадают в списки из-за “смещенности” корпуса, связанной с его новостной тематикой, для которой характерна высокая упоминаемость первых лиц государств, некоторых политиков и т.д. (см. примеры (7) и (8)).

(7) уволить c5wc5
syntax: *работа* **window:** *год, медведев, работа, тренер*

(8) посоветовать c5wc5
syntax: врач **window:** *внимание, врач, знакомый, путин*

В то же время именованные сущности, хотя и в гораздо меньшем объеме, встречаются также и в синтаксических списках (см. пример (9)). Это свидетельствует о том, что попадание имен в список обусловлено не недостатками контекстного метода, а лексической смещенностью корпуса. Вопрос о том, как проявился бы этот эффект на более сбалансированном корпусе, остается открытым.

(9) заметить c10wc10
syntax: *александр, андрей, глава, депутат, заместитель, министр, нарушение, председатель* **window:** *александр, андрей, виктор, владимир, г-н, глава, губернатор, депутат, директор, дмитрий, женщина, заместитель, медведев, министр, нарушение, председатель, сергей, улица, эксперт*

Еще одним источником так называемого «шума» в контекстных списках являются части устойчивых словосочетаний. Так, например, существительное *движение* из (6) является частью устойчивого словосочетания – именной группы *полоса встречного движения* (ср. *выехать на полосу встречного движения*), аналогично существительное *происшествие* – *выехать на место происшествия*. Такой тип шума отсутствует в синтаксических списках. Для выделения компонентов некоторого события интерес представляют также устойчивые сочиненные глагольные группы, как в примере (10), где представлена статистически устойчивая цепочка событий: *не справился с управлением и выехал*. В такой ситуации в контекстные списки попадают актанты ситуации, заданные другим глаголом:

(10) **не справился с управлением и выехал**

Сравнение результатов контекстного и синтаксического методов

Соотношение синтаксических и контекстных списков.

Как показывают результаты сравнения (см. табл. 2), контекстные списки включают больше существительных, чем синтаксические. Более того, для порогов отсечения 10–10 наборы существительных, выделенных синтаксическим методом, полностью вкладываются в наборы, полученные контекстным методом, для 60% глаголов. При этом 104 глагола имеют только одно существительное в синтаксическом списке. В табл. 3 приведены фрагменты списков для порогов 10–10.

Предварительный анализ отдельных примеров, казалось бы, говорит в пользу гипотезы о том, что синтаксический метод должен быть более точным: в соответствующих списках должно оказаться меньше нерелевантных существительных, в списки должны попасть существительные, отделенные от своих вершин-глаголов придаточными предложениями и другими конструкциями.

В примере (11) пара *уменьшиться–доля* разделена придаточным предложением длиной 7 словоупотреблений. В пределах 5 словоупотреблений рядом с глаголом встречаются существительные *процесс, курс*, которые никак семантически с глаголом не связаны.

Однако более подробный анализ результатов показывает, что:

1) контекстные списки больше, чем синтаксические, не только и не столько из-за того, что в них много не связанных с глаголом существительных;

2) в синтаксических списках нередко отсутствуют существительные, образующие с глаголом несвободные словосочетания – лексические функции типа *принять решение*.

Сравним списки существительных, полученные двумя способами при пороге отсечения 5 (пример (12)). Пересечения списков выделены, релевантные коллокации из оконного списка подчеркнуты.

(12) снимать c5wc5

syntax: *год, квартира, комната, фильм*

window: *видео, видеокамера, время,* ***год, квартира****, кино,* ***комната****, оператор,* ***фильм***

Таблица 3

Сравнение списков синтаксических и контекстных коллокаций

Глагол	*Синтаксический метод*	*Контекстный метод*
Снять	год, фильм	год фильм
Идти	Бой, борьба, война, время, год, дело, игра, место, процесс, работа, разговор, речь, человек	бой, борьба, война, время, год, город, дело, день, игра, место, процесс, путь, работа, разговор, речь, строительство, театр, улица, ход, человек
Понять	человек	время, год, деньги, жизнь, момент, человек
Продлить	арест, год, контракт, срок, суд	арест, год, контракт, срок, суд
Приобрести	год, компания, миллион, популярность, характер	акция, год, доля, компания, миллион, опыт, популярность, характер
Привести	возникновение, дефицит, изменение, итог, качество, мнение, падение, повышение, порядок, последствие, потеря, появление, пример, пример, результат, рост, рост, снижение, увеличение, удорожание	возникновение, война, дефицит, изменение, конкуренция, падение, повышение, последствие, потеря, появление, пример, рост, снижение, сокращение, тариф, топливо, увеличение, удорожание, уменьшение, цифра

Как видно из примера, контекстный список полностью включает синтаксический. В контекстном списке в число объектов, которые обычно ‘снимают’, помимо ‘фильма’, попадает ‘кино’, в списке оказываются также типичный инструмент съемки *видеокамера* и типичный агенс *оператор*. При этом существительное *год,* которое можно было бы в данном случае счесть «шумом», встречается в обоих списках. Это существительное – высокочастотное, встречается в именной группе – обстоятельстве времени, т.е. потенциально частотно при достаточно широком множестве глаголов. Это подтверждается корпусными данными: при пороге совместной частоты 10 это существительное попадает в синтаксические списки 160 из 400 глаголов и в контекстные списки 240 глаголов.

Проблемы лакун в синтаксических списках

На наш взгляд, особого внимания заслуживает вопрос, почему синтаксические списки менее полные.

Безусловно, отчасти это связано с неточностью работы синтаксического анализатора. В ряде случаев существительное «не добирает вес» для того, чтобы попасть в синтаксический список в силу того, что, например, на 10 предложений, в которых оно встречается с глаголом, в двух оно не связывается с глаголом из-за ошибок синтаксического анализа.

Однако подробный анализ примеров показывает, что достаточно большое количество расхождений в списках обусловлено с тем, что участник ситуации связан с глаголом опосредованной синтаксической связью, а не напрямую. Такое возможно в сложном предложении, когда глагол и семантически связанное с ним существительное находятся в разных предикациях или в сочинительных конструкциях, как в примере (13) (глагол в придаточном относительном, коллокат – в главном) или в примере (14) (участники ситуации входят в сочинительную группу, с глаголом связан только один). Также это возможно, если существительное является зависимым в группе, где главное слово обозначает количество (*множество* ***<u>демонстрантов</u>***) или является вершиной некоторой коллокационной конструкции ‘ход Х-а’ в (15).

Еще одна ситуация, которая важна для извлечения фактов из текста, – это случаи, когда в качестве непосредственного актанта глагола выступает имя собственное или местоимение, а «типовое» существительное (играющее роль конкретного референта в ситуации) содержится в приложении, как, например, в (16).

Отметим, что в задачах извлечения фактов список таких существительных играет принципиальную роль, поскольку они могут служить маркерами ролей именованных сущностей в событии.

Еще одна причина расхождений – это то, что в контекстных списках участники ситуации, не занимающие позиции подлежащего и прямого дополнения, представлены более полно:

(17) утвердить

syntax: *депутат, правительство, программа, совет, список*

window: *бюджет, год, депутат, директор, заседание, кандидатура, компания, москва, план, правительство, президент, программа, рф, собрание, совет, список*

В примере, в контекстном списке, помимо основных участников ситуации представлено и типичное место (событие), где (в ходе которого) происходит ‘утверждение’): *собрание, заседание*.

(13) После лекции , которую он прочитал нам в школе .

(14) по их следам во Владимир выехали полицейские и сотрудники военкомата .

(15) следить c5wc5

syntax: *ход*
window: *<u>ход, голосование</u>*

Как член комиссии , следил за ходом голосования на дому .

(16) ехать c5wc5

syntax: *вагон машина*
window: *автобус, вагон, водитель, год, машина, минута, человек*

Вот он , Средний Московский Водитель , едет на своей «девятке»

Пример (17) также иллюстрирует тот факт, что контекстные списки дают более полные наборы типичных участников ситуации с некоторой ролью: например, по контекстному списку видно, что утверждают, кроме списка программы, еще и план, кандидатуру и бюджет. При этом, как уже отмечалось, «лишними» в контекстном списке оказываются лексемы, входящие в состав именных групп, которые представляют собой устойчивые словосочетания и являются актантами глаголов: *привести к сокращению доходов, бюджет на год*. Однако не всегда это действительно лишняя информация. Если исходить из задачи «собрать» типичные признаки некоторого события, то оказывается, что зависимые в именной группе, обозначающей участников ситуации, задаваемой глаголом, также играют немаловажную роль:

(18) *...приговорен к лишению свободы с отбыванием наказания в колонии строгого режима*

В (18) глагол *приговорить* имеет такую валентность, как именная группа, обозначающая участника ситуации ‘содержание приговора’, выраженного именной группой *лишение свободы с отбыванием наказания в колонии строгого режима*. С глаголом связана (через предлог) только вершина именной группы – *лишению.* Однако для полноты картины события ‘приговорить’ то, что в контекстный список попадают существительные *колония, наказание, свобода,* оказывается нелишним.

В результате, как показывает подробный анализ списков, оказывается, что контекстные списки более полные с точки зрения характеристики некоторого события. Однако они требуют дополнительного анализа для того, чтобы извлечь из такого списка характеристик именно существенные. С другой стороны, у синтаксического метода, возможно, есть перспективы, если учитывать не только ближайшие синтаксические связи, но и определенные типы опосредованных связей, не только отдельные существительные, но и типичные зависимые в именной группе, обозначающей некоторого участника ситуации.

ЗАКЛЮЧЕНИЕ

Таким образом, было проведено исследование по сравнению двух методов извлечения коллокационных пар ‘глагол–существительное’. Материалом исследования послужил корпус новостных текстов достаточно большого объема (примерно 9 млн словоупотреблений). В первом случае кандидаты на коллокации извлекались на основе контекстной близости, во втором – извлекались пары ‘глагол-существительное’, между которыми существовала синтаксическая связь. В обоих случаях использовалась мера PMI для ранжирования пар. Сравнение результатов показало, что степень совпадения списков результирующих коллокаций невысокая. Списки коллокаций, полученные синтаксическим методом, оказались неполными по сравнению с контекстными списками. Результат показывает, что для выделения типичных участников события необходимо учитывать не только непосредственные синтаксические связи существительных с глаголом, но и опосредованные синтаксические отношения. Анализ результатов, полученных контекстным методом, показал, что при достаточно большом объеме корпуса данный метод дает вполне приемлемые результаты.

СПИСОК ЛИТЕРАТУРЫ

1. Шенк Р., Абельсон Р. П. Скрипты, планы и знание // Труды IV Международной конференции по искусственному интеллекту. Т. 6. – М.: Научный совет по компл. пробл. «Кибернетика» АН СССР, 1975. – С. 208–220.
2. Филлмор Ч. Фреймы и семантика понимания // Новое в зарубежной лингвистике: Когнитивные аспекты языка. Вып. XXIII. – М.: Прогресс, 1988. – С. 52–92.
3. Halliday M. A. K. Lexis as a linguistic level // CE Bazell et al.: In memory of JR hrth. London: Longman. – 1976. – P. 150–61.
4. Виноградов В. В. Русский язык. – М.: Государственное учебно-педагогическое изд-во, 1947.
5. Firth J. R. et al. A synopsis of linguistic theory, 1930–1955 // Studies in Linguistic Analysis. Special volume of the Philological Society. – Oxford: Blackwell, 1957. –P. 1–32.
6. Jackson H. Words and their Meaning. – London and New York: Longman, 1995.
7. Борисова Е. Г. Коллокации. Что это такое и как их изучать? – М: Филология, 1995.
8. Manning M., Christopher D., Schütze H. Foundations of statistical natural language processing. – Boston: MIT Press, 1999.
9. Хохлова М. Экспериментальная проверка методов выделения коллокаций. // Slavica Helsingiensia, 34. – Helsinki: Helsinki University Press, 2008. – P. 343–357.
10. Church K. W., Hanks P. Word association norms, mutual information, and lexicography // Computational Linguistics. – 1990. – Vol. 16(1). – P. 22–29.
11. Khokhlova M. Extracting Collocations in Russian: Statistics vs. Dictionary // JADT 2008: actes des 9es Journées Internationales d’Analyse Statistique des Données Textuelles / ed. S. Heiden , B. Pincemin. – P. 613–624.
12. Ягунова Е. В., Пивоварова Л. М. Природа коллокаций в русском языке. Опыт автоматического извлечения и классификации на материале новостных текстов // Научно-техническая информация. Сер. 2. – 2010. – № 6. – С. 30–40.
13. Breidt E. Extraction of V–N-collocations from text corpora: A feasibility study for German // Proc. of the Workshop on Very Large Corpora: Academic and Industrial Perspectives. – Columbus, 1993. – P. 74–83.

14. Todirascu A., Tufis D., Heid U., Gledhill C., Stefanescu D., Weller M., Rousselot F. A Hybrid Approach to Extracting and Classifying Verb+Noun Constructions // Proceedings of LREC'2008, Marrakesh, Morocco. – URL: http://www.lrec-conf.org/proceedings/lrec2008/pdf/500_paper.pdf (дата обращения: 02. 04. 2013).
15. Todirascu A., Gledhill C. Extracting Collocations in Context: The case of Verb–Noun Constructions in English and Romanian // Recherches Anglaises et Nord-Américaines (RANAM). – Strasbourg: Université Marc Bloch, – P. 107–122.
16. Клышинский Е., Кочеткова Н., Ливинов М., Максимов В. Автоматическое формирование базы сочетаемости слов на основе очень большого корпуса текстов // Труды международной конференции ДИАЛОГ'2010. – Бекасово, 2010. – С. 181–185.
17. Lin D. Automatic Retrieval and Clustering of Similar Words // COLING-ACL98. – Canada: Montreal, 1998.
18. Kilgarriff A., Tugwell D. Sketching words, lexicography and natural language processing: A Festschrift in Honour of B. T. S. Atkins // EURALEX / ed. Marie-Hélène Corréard. – 2002. – P. 125–137.
19. Khokhlova M. Applying Word Sketches to Russian. In: Proceedings of Raslan 2009. Recent Advances in Slavonic Natural Language Processing. – Brno: Masaryk University, 2009. – P. 91–99.
20. Orliac B., Dillinger M. Collocation extraction for machine translation // Proceedings of Machine Translation Summit IX. – LA: New Orleans, 2003. – P. 292–298.
21. Pado S., Lapata M. Dependency-based Construction of Semantic Space Models // Computational Linguistics. – 2007. – Vol. 33(2). – P. 161–199.
22. Gildea D., Jurafsky D. Automatic Labeling of Semantic Roles // Computational Linguistics. – 2002. – Vol. 28(3).
23. Кустова Г., Толдова С. НКРЯ: семантические фильтры для разрешения многозначности глаголов // Национальный корпус русского языка 2006-2008. Новые результаты и перспективы. – Санкт-Петербург: Нестор–История, 2009.
24. Sharoff S., Nivre J. The proper place of men and machines in language technology: Processing Russian without any linguistic knowledge // Компьютерная лингвистика и интеллектуальные технологии: по материалам ежегодной Международной конференции «Диалог» (Бекасово, 25–29 мая 2011 г.). Вып. 10 (17). – М.: РГГУ, 2011. – С. 591–604.
25. Schmid H. Probabilistic Part-of-Speech Tagging Using Decision Trees // Proceedings of International Conference on New Methods in Language Processing. –Manchester. – 1994. – Vol. 12, Issue 4. – P. 44–49.
26. Jongejan B., Dalianis H. Automatic training of lemmatization rules that handle morphological changes in pre-, in- and suffixes alike // Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP. – Singapore: Association for Computational Linguistics, 2009. – P. 145–153.
27. Nivre J., Hall J., Nilsson J. Maltparser: A data-driven parser-generator for dependency parsing // Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC). – 2006. – P. 2216–2219.
28. Boguslavsky I., Grigorieva S., Grigoriev N., Kreidlin L., Frid N. Dependency treebank for Russian: concept, tools, types of information // Proceedings of the 18th conference on Computational linguistics (COLING '00). – 2000. – Vol. 2. – P. 987–991.
29. Evert S., Krenn B. Methods for the qualitative evaluation of lexical association measures // Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics. – Toulouse, France, 2001. – P. 188–195.

Материал поступил в редакцию 19.03.13.

Сведения об авторах

АКИНИНА Юлия Сергеевна – научный сотрудник Центра Семантических Технологий НИУ ВШЭ, Москва

E-mail: jakinina@hse.ru

КУЗНЕЦОВ Илья Олегович – младший научный сотрудник Центра Семантических Технологий НИУ ВШЭ, Москва, Россия, аспирант филологического факультета ВШЭ

E-mail: iokuznetsov@hse.ru

ТОЛДОВА Светлана Юрьевна – кандидат филологических наук, доцент кафедры теоретической и прикладной лингвистики филологического факультета МГУ им. М. В. Ломоносова, старший научный сотрудник Центра Семантических Технологий НИУ ВШЭ, Москва

E-mail: toldova@yandex.ru

УВАЖАЕМЫЕ ЧИТАТЕЛИ!

ЦЕНТР НАУЧНО-ИНФОРМАЦИОННОГО ОБСЛУЖИВАНИЯ ВИНИТИ РАН

ПРЕДОСТАВЛЯЕТ КОПИИ ПЕРВОИСТОЧНИКОВ

ВИНИТИ РАН осуществляет обслуживание копиями первоисточников, хранящихся в фонде научно-технической литературы ВИНИТИ, в фондах других библиотек, а также в доступных ВИНИТИ электронных ресурсах.

Фонд научно-технической литературы ВИНИТИ включает более 2 млн изданий по точным, естественным и техническим наукам, в т.ч.:

– отечественные и иностранные периодические и продолжающиеся издания – с 1987 г.;

– отечественные книги – с 1987 г.;

– иностранные книги – с 1991 г.;

– рукописи, депонированные в ВИНИТИ, – с 1962 г.

Заказы на бумажные или электронные копии первоисточников принимает Центр научно-информационного обслуживания (ЦНИО) ВИНИТИ. ЦНИО ВИНИТИ обслуживает коллективных (организации и учреждения) и индивидуальных пользователей.

Формы обслуживания:

– абонементная (на основе договоров и предоплаты);

– разовые заказы (с предоплатой заказа по счету);

– индивидуальная форма обслуживания в читальном зале ЦНИО ВИНИТИ.

На сайте ВИНИТИ (http://www.viniti.ru) представлен полный Электронный каталог научно-технической литературы (http://catalog.viniti.ru), зарегистрированной в ВИНИТИ с 1994 г. Доступ для просмотра и поиска по Каталогу свободный. Постоянные абоненты ЦНИО ВИНИТИ, имеющие логин и пароль для работы с Каталогом, могут делать заказ копий непосредственно через Каталог.

Услуги по изготовлению копий первоисточников из фондов других библиотек предоставляются только постоянным абонентам. Место хранения первоисточников указывается в Электронном каталоге.

За подробной информацией обращаться по адресу:

125190, Россия, Москва, ул. Усиевича, 20, ВИНИТИ РАН. ЦНИО

Телефоны: 8 (499)155-42-43, 155-42-09, 152-54-59

Факс: 8 (499) 943-00-60

E-mail: cnio@viniti.ru; **URL:** http://www.viniti.ru

БАЗА ДАННЫХ ВИНИТИ РАН

ВИНИТИ предлагает к использованию через WWW-сервер (**http://www.viniti.ru**) крупнейшую Федеральную базу отечественных и зарубежных публикаций по естественным, точным и техническим наукам. БД ВИНИТИ РАН генерируется с 1981 г., обновляется ежемесячно, пополнение составляет около 1 млн документов в год. БД ВИНИТИ представлена ретроспективными тематическими фрагментами и единой политематической БД (ретроспектива с 2001 г.), объединяющей все тематические фрагменты БД ВИНИТИ.

БД ВИНИТИ РАН в сети INTERNET

Сервер ВИНИТИ – http://www.viniti.ru – обеспечивает on-line доступ к Базе данных ВИНИТИ РАН круглосуточно без выходных.

На основе БД ВИНИТИ РАН предоставляются следующие услуги:

- Диалоговый поиск научно-технической информации **в режиме on-line**;
- **Демо-версия**, позволяющая ознакомиться с основными функциями поисковой системы, составом данных, формами представления документов и получить навыки работы с системой;
- **Поисковые эксперты** ВИНИТИ выполнят тематический поиск по разовым или постоянным запросам, а также окажут **консультационные услуги**.

БД ВИНИТИ РАН на CD-ROM

Любые наборы тематических фрагментов БД ВИНИТИ или их разделов могут быть предоставлены **на CD-ROM в поисковой системе (ИПС) "Сокол"**, обеспечивающей все поисковые функции, доступные в режиме on-line:

- Поиск можно вести в годовом или ретроспективном массиве (за несколько лет сразу) в одном или нескольких тематических фрагментах .
- Поиск по словам и любым словосочетаниям из заглавия, реферата, ключевых слов.
- Использование года, языка, рубрик, шифров тематических разделов БД для уточнения поиска.
- Поиск по словарю, выполняющему функции многоаспектного указателя, в том числе авторского, предметного, источников, индексов МПК, номеров патентных документов и депонированных рукописей и т.д.
- Возможность запоминания запросов для последующего использования и/или редактирования их.
- Чтение документов не только как в РЖ (последовательный просмотр документов одного номера за другим), но и чтение документов нужных тематических фрагментов (разделов) по оглавлению за весь период заказанной ретроспективы.

ИПС "Сокол" является прикладной программой Microsoft Windows.

Любые наборы тематических фрагментов БД ВИНИТИ или их разделов могут быть подготовлены **в коммуникативных форматах** ISO-2709, МЕКОФ, txt на любых видах электронных носителей.

Продукты предоставляются на договорной основе.

Информационная служба БД ВИНИТИ: 125190, Москва, ул. Усиевича 20, ВИНИТИ
Телефон: (499) 155-45-01, 155-45-02, **Факс:** (499) 152-62-31 **e-mail:** csbd@viniti.ru