Министерство внутренних дел Российской Федерации
Омская академия

А. А. Гайдамакин

ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ В ЮРИДИЧЕСКОЙ АНАЛИТИКЕ

Учебное пособие

Омск
ОмА МВД России
2019

УДК 340(075.8)
ББК 67.062
Г14

Рецензенты:
А. Г. Коротов (ИЦ УМВД России по Омской области);
доктор философских наук, профессор *А. Ш. Руди*
(Омская юридическая академия)

Гайдамакин, А. А.

Г14 Искусственный интеллект в юридической аналитике : учеб. пособие. — Омск : Омская академия МВД России, 2019. — 132 с.

ISBN 978-5-88651-720-0

В учебном пособии рассматриваются проблемы и предпосылки применения методов искусственного интеллекта в правовой аналитике, демонстрируются основные подходы к решению задач юридического анализа и прогнозирования с позиций кибернетики, даются рекомендации по выбору программного обеспечения.

Предназначено для курсантов, слушателей образовательных организаций МВД России, сотрудников аналитических подразделений органов внутренних дел Российской Федерации.

УДК 340(075.8)
ББК 67.062

ISBN 978-5-88651-720-0

ВВЕДЕНИЕ

Современное общество требует более совершенных способов обработки информации, извлечения знаний из информационных массивов, технологий генерации новых гипотез на основе этих знаний, а также методов аргументации решений, принимаемых на основе полученного знания. Понимание сути происходящих событий, выявление тенденций развития социальных систем и выработка научной основы для управленческих решений является задачей аналитика. Отставание в развитии интеллектуального инструментария в области аналитической методологии влечет задержку политических и экономических процессов в обществе. В то же время разработка общей методологии, объединяющей отдельные аналитические технологии в целостную систему, могла бы дать весомые конкурентные преимущества (при условии их использования в практике управления).

Аналитические технологии, т. е. устойчивые совокупности методов оперирования данными, нацеленные на производство информационной продукции заданного качества, могут быть как инструментальными, так и неинструментальными. Неинструментальные технологии имеют историю и широко применяются в аналитической работе. Инструментальные технологии, под которыми понимаются главным образом машинные методы обработки данных, традиционно используются для автоматизации рутинных операций. Однако в последнее время степень интеллектуальности компьютерных систем растет, это позволяет подступиться к таким творческим и, казалось бы, не поддающимся формализации задачам, как классификация информации, ее обобщение, выдвижение гипотез и др.

Сказанное выше относится к аналитической деятельности применительно к любому аспекту и уровню социального управления. В данном пособии инструментальные методы, основанные на технологиях искусственного интеллекта (далее — ИИ), рассматриваются применительно к юридической сфере. Как показывает анализ, ИИ активно внедряется в те сферы правовой деятельности, где еще недавно, по убеждению большинства исследователей, его применение было невозможно в принципе. Например, существуют систе-

мы, достаточно эффективные в области прогнозирования судебных решений. Понимание принципов, на которых построены эти и другие системы, способствует осознанию как перспектив, так и ограничений, характерных для современных интеллектуальных инструментов обработки данных.

Искусственный интеллект и аналитика: основные понятия

Существуют десятки определений понятия «аналитика». Например, аналитика — это:

— методологическая основа процесса обработки информации;

— форма научного знания, применяемая в процессах управления (прежде всего для выработки управленческих решений);

— совокупность методов, с помощью которых можно выявлять скрытые смыслы в текстах и реальных социально-политических и экономических процессах;

— синоним системного анализа;

— процесс выявления причинно-следственных зависимостей и пространственно-временных связей в каких-либо объектах;

— процесс систематизации содержания посредством схематизации, конструирования и моделирования сущностных элементов и связей и т. д.[1]

Иногда понятие «аналитика» отождествляется с научно-исследовательской деятельностью вообще (аналитика в широком смысле слова). В любом случае аналитическая деятельность предполагает знание эффективных приемов организации мыслительных процессов и использование некоторого технологического инструментария.

Подходов к определению ИИ и интеллектуальной деятельности тоже несколько:

— автоматизация действий, которые мы ассоциируем с человеческим мышлением, таких как принятие решений, решение задач, обучение (системы, которые думают как люди. Их, наряду с экспериментальными методами психологии, исследует междисциплинарная наука когнитология);

— наука о том, как научить компьютеры делать то, в чем люди в настоящее время их превосходят (системы, которые действуют как люди. Согласно тесту Тьюринга, система интеллектуальна, если ее поведение нельзя отличить от поведения человека);

— изучение таких вычислений, которые позволяют чувствовать, рассуждать и действовать (системы, которые думают рационально, конструируются в традициях логицизма);

---

[1] *Курносов Ю. В., Конотопов П. Ю.* Аналитика: методология, технология и организация информационно-аналитической работы. М., 2004. С. 27–28.

— наука, посвященная изучению интеллектуального поведения артефактов (системы, которые действуют рационально, обычно используют рациональных агентов — автономные программы, способные анализировать окружающую обстановку, обучаться, адаптироваться к изменениям и брать на себя реализацию заданных целей).

Наиболее известен критерий, предложенный Тьюрингом: система интеллектуальна, если в процессе общения с ней человек не в состоянии понять, что общается с машиной. В этом случае компьютерная система как минимум должна обладать:

— возможностями обработки текстов на естественных языках;
— средствами представления знаний;
— средствами автоматического формирования логических выводов;
— средствами машинного обучения.

Кроме этого, полный тест Тьюринга предполагает еще наличие машинного зрения и средств робототехники для манипулирования предметами. Современные исследователи ИИ не стремятся строить системы, удовлетворяющие полному тесту, так как задача изучения основополагающих принципов интеллекта намного важнее полной имитации его носителей. Примером может служить авиация, которая своими успехами обязана изучению законов аэродинамики, а не копированию полета птиц. Вместе с тем реализация четырех вышеперечисленных аспектов интеллектуальной деятельности существенно расширяет инструментальный арсенал аналитика.

Конструирование систем в традициях логицизма имеет долгую историю и восходит к логике Аристотеля. Построено большое количество логических систем, открыт алгоритм автоматического доказательства теорем. Структура силлогизма и логика высказываний Аристотеля позволяют моделировать любые, в том числе и юридические утверждения, а упомянутый алгоритм дает возможность автоматизировать их доказательство в некоторой системе норм (аксиом). Вместе с тем специфика человеческого мышления и языка обусловливает трудности процесса формализации знаний; методологические требования логики оказываются чрезвычайно жесткими. Кроме того, переборный характер поиска решения приводит к «комбинационному взрыву» — исчерпанию вычислительных ресурсов системы при отсутствии механизма управления приоритетами выбора.

Проекты, созданные в рамках когнитологического направления, весьма разнообразны и направлены на познание механизма человеческого мышления. Выделяют два способа изучения мышления: интроспек-

ция — попытка проследить за ходом собственных мыслей, психологические эксперименты. Однако точной теории мышления еще не создано, хотя первые шаги в этом направлении предпринимались. Так, программа GPS (универсальный решатель задач), предложенная А. Ньюэллом и Г. Саймоном, успешно моделировала ход человеческих рассуждений при решении задач.

Несмотря на то что в настоящее время указанные выше направления исследований в области ИИ четко оформлены, они активно обмениваются между собой идеями. Например, мультиагентные системы часто используют как фреймовые представления знаний, разработанные в рамках когнитологического направления, так и правила, сформированные в рамках логицистского подхода. Реальная информационная система может иметь гибридный характер и содержать в себе как комбинации «элементов ИИ», присущих различным направлениям, так и каждый из этих элементов в отдельности (подсистемы распознавания образов, рациональные агенты, алгоритмы кластеризации, технологии формального анализа понятий, различные способы представления знаний и семантического анализа, системы логического вывода, модели аргументации, правдоподобные рассуждения, статистические методы).

Термин «правовая аналитика» в узком смысле, как правило, означает анализ юридической составляющей деятельности некоторого предприятия, а также составление юридических прогнозов в области взаимоотношений предприятий и их контрагентов, компаний и контролирующих органов. Такая юридическая экспертиза призвана оценить юридическую и экономическую судьбу предприятия в долгосрочной перспективе. Оценка основывается на некотором наборе заранее известных маркеров, по которым можно судить о том, какое событие произойдет в будущем и какие последствия могут быть при позитивном и негативном сценарии. В более широком смысле под правовой аналитикой понимают профессиональную аналитическую деятельность в сфере законотворчества и правоприменения, правовой культуры, правового информирования, образования и воспитания.

Как видим, понятия «аналитическая деятельность» и «ИИ» не вполне определены. Это в той или иной степени общее свойство понятия, так как оно (понятие) есть образ, ассоциируемый с неким термином, — образ, обращение к которому в процессе межличностной коммуникации требует использования некоторой модели, которая зависит от контекста и, в свою очередь, опирается на другие понятия и т. д. В данном учебном пособии иллюстрируется одна из основных проблем ИИ — проблема естествен-

ного языка, которая часто не осознается начинающими исследователями. Другой язык — это не просто другая лексика, это другой понятийный аппарат, а значит, и иной взгляд на мир.

Технологии решения интеллектуальных задач

Сфера применения информационных технологий в деятельности современного юриста достаточно широка, однако не все информационные системы, даже весьма полезные и эффективные, можно отнести к интеллектуальным. Так, например, не имеет отношения к ИИ использование средств телекоммуникации для проведения удаленного заседания суда в режиме телеконференции. Разными авторами выделены следующие направления использования ИИ в юридической деятельности:

— методология планирования деятельности;

— модели формирования мнения;

— модели юридического доказательства;

— моделирование процесса аргументации (в том числе генерация версий или вариантов аргументации по заданным схемам);

— моделирование документа (например, текста договора);

— моделирование и мониторинг законодательства (в том числе выявление противоречий в тексте, а также поиск релевантной правовой нормы);

— анализ социальных сетей и интернет-контента;

— моделирование социальных процессов;

— выявление цепочек межличностных связей;

— составление статистики массивов данных;

— реферирование неструктурированных текстов;

— интеллектуальный анализ данных;

— диагностика на основе правил и эвристик (экспертные системы);

— оптимизация алгоритмов действий (определение наилучшего порядка действий для достижения цели, т. е. поиск пути по графу применительно к минимизации рисков, времени или материальных затрат, в том числе при разработке или выявлении юридических процедур, действий по квалификации деяний и др.).

Основой для решения указанных задач являются следующие базовые технологии:

— концептуализация, формирование онтологий;

— метод автоматического доказательства теорем (метод резолюций);

— распознавание образов;

— кластерный анализ;
— имитационное моделирование;
— семантическое моделирование;
— методы статистического анализа.

Рассмотрим некоторые из них. С другими статистическими методами можно ознакомиться, например, в учебнике С. Г. Олькова «Аналитическая юриспруденция» [2].

[2] *Ольков С. Г.* Аналитическая юриспруденция : учебник. Сургут, 2012. 1125 с.

ГЛАВА I. АЛГОРИТМЫ И ЭКСПЕРТНЫЕ СИСТЕМЫ

§ 1. Алгоритмы в деятельности юриста

Одним из аргументов противников формализации и автоматизации юридических суждений является утверждение о том, что интерпретация и приложение закона должны учитывать особенности человеческой природы. Действительно, компьютер вряд ли можно научить справедливости или беспристрастности. Однако во многих случаях закон достаточно формален и не учитывает этих особенностей [1]. Многие юридические рассуждения поддаются алгоритмизации. Так, в «шпаргалке для умных» Д. В. Панова [2] можно найти полезные юридические алгоритмы. Их общие черты сводятся к следующему:

— общеобязательность правил, составляющих содержание алгоритма;

— последовательность реализации предписанных действий (правил поведения);

— формализация процесса реализации правил;

— наличие указания на достижение цели, заложенной в алгоритме или норме права.

Действительно, алгоритм — это пошаговое описание действий, необходимых для получения заданного результата. Норму права можно рассматривать как алгоритм правового поведения, реализуемый действиями субъектов, связанных некоторыми обязанностями. Правоотношение — это та же норма права, но уже конкретизированная применительно к определенным лицам, которые стали участниками этого правоотношения. Таким образом, правоотношение можно рас-

---

[1] *Гайдамакин А. А.* Формальные модели в юридической науке и технике. Омск, 2017. С. 136.

[2] *Панов Д. В.* Решатель юридических проблем: скорая правовая помощь на все случаи жизни. М., 2012. 256 с.

сматривать как механизм, воспринимающий и выполняющий правила поведения, т. е. как алгоритмический процесс. Дискретность процесса реализации правовой нормы, а тем более нормативного акта, очевидна. Он состоит из совокупности правовых норм, последовательная реализация которых (особенно процессуальных) заложена в самом праве.

Алгоритмизация применима и к процессу расследования преступлений, так как эта деятельность предполагает планирование, организацию и проведение определенной совокупности действий и операций. В ходе накопления опыта расследования конкретных видов преступлений давно очерчен как круг таких действий, так и их направленность, целевые функции. И при всем многообразии следственных ситуаций их можно подразделить на несколько характерных групп, выделив типичные следственные ситуации и основные направления расследования и следственные версии. Последние, в свою очередь, влекут определенные типичные следственные или оперативно-розыскные действия.

Деятельность суда тоже подчинена алгоритму. Вот что сказал по этому поводу Председатель Конституционного Суда России В. Д. Зорькин на VI Всероссийском съезде судей: «Суд гарантирует равенство сторон. Способ, которым он это достигает, — предельная формализация правосудия, сведение всего судебного процесса исключительно к логическому состязанию (т. е. формальной процедуре, лишенной всяческого субъективного содержания и допускающей столь же строгую формализованную проверку). В этом — суть юриспруденции, на этом зиждется профессиональное сознание юристов, и это именно то качество, которого все еще не хватает российскому правосудию» [3]. Судья не решает задачу, а проверяет условия для ее решения. Алгоритм решения заложен в законе. Судья исполняет требования закона, т. е. исполняет алгоритм. Законное судебное решение зависит от: доказанности условий, необходимых для удовлетворения иска; незаинтересованности судьи; знаний судьи.

В своей книге «Как стать хорошим юристом» В. И. Добровольский пишет: «По сути, у юриста нет свободы действий. Он связан правовым алгоритмом». Так, алгоритм судебного спора требует ответа на пять вопросов.

1. Чем подтверждается наличие права, нуждающегося в защите?
2. Чем подтверждается нарушение права?
3. Не пропущен ли срок для защиты права?

[3] *Добровольский В. И.* Как стать хорошим юристом. М., 2017. С. 136.

4. Присутствуют ли обстоятельства, исключающие ответственность правонарушителя?

5. Какой способ защиты, соразмерный характеру и последствиям нарушения, предусмотрен законом?

По уголовному, административному и гражданскому делу юрист делает одно и то же: доказывает наличие условий, необходимых для судебной защиты[4].

Безусловно, образность нашего мышления и обусловленная ею нечеткость терминологии, а также наличие субъективных оценочных категорий и многообразия внешних факторов ограничивают возможности формально-логического подхода. Эти проблемы рассмотрены, к примеру, в монографии автора настоящего пособия[5]. Однако они были успешно решены «обходным путем», во взаимодействии машины и человека. С 80-х гг. XX в. широкое распространение получили экспертные системы — программы, хранящие опыт экспертов в виде фактов, правил и онтологий и реализующие на их основе логический вывод в диалоге с пользователем. В таком «тандеме» компьютер отвечает за знания и логику, а человек — за распознавание ситуаций и интерпретацию терминов.

§ 2. Юридические экспертные системы

Экспертная система — это человеко-машинный автоматизированный аппаратно-программный комплекс (далее — АПК), использующий и моделирующий знания, квалификацию и опыт эксперта в решении интеллектуальных задач в определенной предметной области. Экспертные системы (далее — ЭС) основаны на применении различных правил к исходной информации, т. е. знаниям, используемым для решения конкретной интеллектуальной задачи. Некоторые из этих правил устанавливают отношения между уже известными знаниями о сущности предметной области, а другие позволяют получить о ней новые знания. На основе этих отношений могут быть сделаны выводы (по эффективности близкие к выводам и рассуждениям эксперта), приводящие к решению поставленной задачи.

Распространено мнение, что ЭС могут делать не более того, что может эксперт, на основе знаний и опыта которого создавалась данная система. Такое мнение представляется ошибочным: вполне можно постро-

---

[4] Там же. С. 135.

[5] *Гайдамакин А. А.* Указ. соч. С. 202.

ить самообучающуюся ЭС в области, в которой вообще нет экспертов, либо объединить в одной ЭС знания нескольких экспертов и получить в результате систему, которая может то, чего ни один из ее создателей сам не сделает. Ошибочно и представление о том, что ЭС никогда не заменит эксперта. Это заблуждение опровергается широкой практикой применения данных систем.

В отличие от человека, ЭС:

— не имеют предубеждений;

— не делают поспешных выводов;

— способны обрабатывать базы знаний очень большого объема;

— работают систематизированно, рассматривая все детали, часто выбирая наилучшую альтернативу из всех возможных;

— сохраняют приобретенные знания навсегда;

— устойчивы к помехам.

ЭС наряду с документационными информационно-справочными и гипертекстовыми системами на равных правах входят в более общий класс, именуемый «системами поддержки принятия решений». ЭС используются прежде всего в системах управления, а также для идентификации и диагностики объектов различной природы, в том числе и в праве. Важным назначением правовых ЭС является предоставление пользователю консультаций в той или иной области права на основе обобщенного опыта, навыков и интуиции экспертов, принимавших участие в формировании ее базы знаний.

ЭС наилучшим образом отвечают задачам, требующим принятия решений в сложных ситуациях, когда критическим фактором является не только достоверный диагноз, но и время. Вторым классом задач, хорошо отвечающих технологии ЭС, является тиражирование опыта высококвалифицированных экспертов. Положительные аспекты данного подхода состоят и в том, что затраты на обслуживание не возрастают при необходимости увеличения числа экспертов либо при их увольнении. К третьему классу задач, стимулирующих создание ЭС, относятся те, которые требуют постоянного и длительного принятия решений в трудных или экстремальных условиях. Приведенный список задач, применение ЭС в которых может давать существенный и разноплановый эффект, конечно, не является исчерпывающим.

В то же время следует иметь в виду ограничения, присущие современным ЭС:

— диалоговый режим, обычно принятый в таких системах, иногда замедляет процесс получения решений (пользователю, юристу или, на-

пример, врачу уже ясна общая картина, а система все еще разворачивает логику диалога);

— многие системы кажутся удобными только разработчику (важно, чтобы формулировки вопросов были понятны всем, а это труднодостижимо);

— их использование ограничивается узкими предметными областями.

Основными компонентами ЭС являются база знаний и машина логического вывода — универсальный решатель задач.

База знаний является информационной моделью предметной области. Декларативная компонента базы знаний содержит информацию об известных свойствах сущностей программного обеспечения (далее — ПО) и об отношениях между ними, а ее процедурная компонента — правила, применяемые для преобразования и обработки декларативной информации в ходе решения определенного класса (или классов) интеллектуальных задач.

Машина логического вывода — это реализованный на ЭВМ АПК, предназначенный для формирования определенной последовательности правил (взятых из процедурной части базовых знаний) и ее применения к фактам (взятым из декларативной части базовых знаний) в целях получения вывода, приводящего к решению конкретной задачи.

Помимо двух основных частей любая ЭС содержит:

— оперативную память;

— модуль интеллектуального редактирования базы знаний;

— модуль объяснения решений;

— интерфейс пользователя.

При создании ЭС нашли применение методы ИИ, разработанные ранее: методы представления знаний, логического вывода, эвристического поиска, распознавания предложений на естественном языке и др.

§ 3. Реализация алгоритма в экспертной системе продукционного типа

Для юриста представляла бы интерес система, позволяющая, например, моделировать процесс логического вывода и на его основе осуществлять предварительную квалификацию действий субъекта. Структура такой системы может быть различной. Так, широко используются фреймовые структуры и нейронные сети. Однако наиболее просты и популярны ЭС на базе продукционных моделей, т. е. правил вида «*если—то*». Именно такую систему мы и построим.

Выберем в качестве объекта моделирования некоторое деяние, например, изъятие собственности. Выражая понятия *хищения* и *кражи* через понятие *изъятия* в соответствии с логикой ст. 158 УК РФ, в итоге получим набор правил такого вида:

Если субъект изъятия не является владельцем изъятого имущества
и нанесен материальный ущерб
и изъятие совершено противоправно, корыстно и безвозмездно,
то субъект совершил **хищение** имущества.
Если хищение совершено тайно,
то субъект совершил **кражу** имущества.
Если размер ущерба, нанесенного кражей, более 250 тыс. руб.,
то кража квалифицируется по ч. 3 ст. 158.

Представим правила, составляющие базу знаний системы, в виде ориентированного графа (орграфа), вершины которого отображают состояние ЭС и соответствуют либо ее вопросам к пользователю, либо ее ответам. Если вершина помечена вопросом ЭС, то из нее к другим вершинам выходят дуги, соответствующие возможным вариантам ответа пользователя. В нашем случае таких дуг может быть более двух, так как, например, в ответ на вопрос: «Оцените размер ущерба», необходимо выбрать ответ из следующих вариантов: *менее 250 тыс. руб, более 250 тыс. руб* (в крупных размерах) и *более 1 млн руб.* (в особо крупных размерах). Пример графа, отображающего логику ст. 158 УК РФ, приведен на рисунке 1[6].

Работа с системой состоит из последовательности однотипных шагов, на каждом из которых пользователь должен решить, по какой дуге он пойдет из очередной вершины. Перемещение по орграфу происходит до тех пор, пока система не перейдет в состояние ответа, т. е. в вершину без исходящих дуг.

Для реализации такого алгоритма можно использовать различные инструментальные средства: готовые оболочки ЭС (например, «Рапана» или «Малая Экспертная Система»), логическое программирование на языке Prolog, системы программирования на языках высокого уровня и т. д. В этом параграфе мы воспользуемся системой управления базами данных (далее — СУБД) «Access» — стандартным приложением из пакета «MS Office»[7].

---

[6] *Гайдамакин А. А.* Новые образовательные стандарты и роль информационно-правового блока в подготовке юристов // Вестник Волгоградской академии МВД России. 2012. № 3. С. 117–122.

[7] Там же.

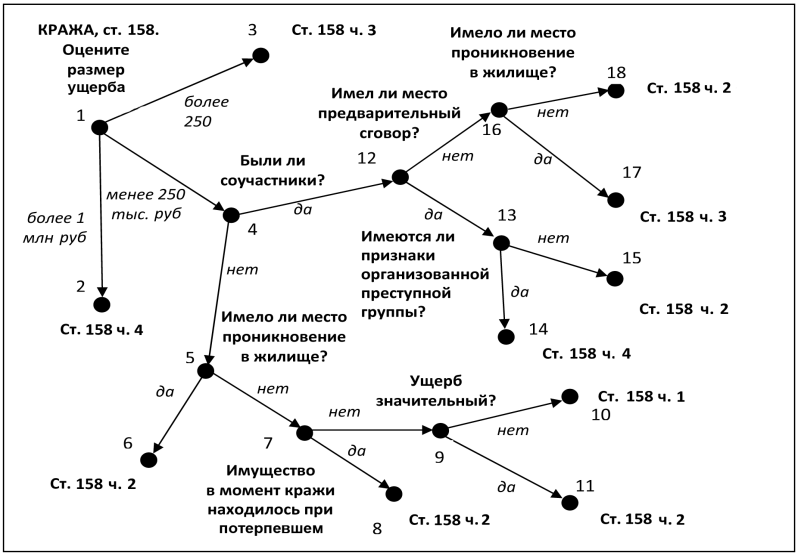

Рис. 1. Одна из возможных реализаций реляционного графа с вершиной в состоянии «КРАЖА»

Следуя реляционному подходу, опишем представленный выше орграф подходящими таблицами.

1. Дуги будем описывать в таблице, которую назовем «Ребра». Каждую дугу можно описать номером ее начала и конца. Кроме столбцов с номерами дуг добавим в таблицу столбец «Ответ пользователя». Обратите внимание: *во всех таблицах ключевые поля создавать* ***не нужно!*** (например, поле КОД типа Счетчик).

Ребра : таблица

Начало_дуги	Конец_дуги	Ответ_пользователя
1	2	Ущерб более 1 млн. руб.
1	3	Ущерб более 250 тыс. руб.
1	4	Ущерб менее 250 тыс. руб.
4	5	соучастников нет
5	6	имело место проникновение в жилище
5	7	без проникновения в жилище
7	8	изъятое находилось при потерпевшем
7	9	изъятое не находилось при потерпевшем
9	10	нанесен незначительный ущерб (менее 2500 руб.)
9	11	нанесен значительный для потерпевшего ущерб
4	12	есть соучастники
12	13	имел место предварительный сговор
13	14	имеются признаки организованной преступной группы
13	15	нет признаков организованной преступной группы
12	16	без предварительного сговора
16	17	имело место проникновение в жилище
16	18	проникновения в жилище не было
0	0	

Рис. 2. Таблица «Ребра»

2. Для удобства пользования системой создадим еще две таблицы: «Вопросы» и «Ответы».

Вопросы : таблица

Состояние	Вопрос
1	Оцените размер ущерба
4	Были ли соучастники?
5	Имело ли место проникновение в жилище?
7	Находилось ли имущество при потерпевшем в момент хищения?
9	Является ли ущерб значительным для потерпевшего?
12	Имел ли место предварительный сговор?
13	Имеются ли признаки организованной преступной группы?
16	Имело ли место проникновение в жилище?
0	

Рис. 3. Таблица «Вопросы»

Ответы : таблица

Состояние	Ответ
2	Ст. 158 часть 4
3	Ст. 158 часть 3
6	Ст. 158 часть 2
8	Ст. 158 часть 2
10	Ст. 158 часть 1
11	Ст. 158 часть 2
14	Ст. 158 часть 4
15	Ст. 158 часть 2
17	Ст. 158 часть 3
18	Ст. 158 часть 2
0	

Рис. 4. Таблица «Ответы»

Последние две таблицы создавать не было необходимости, так как из вариантов ответов, приведенных в таблице «Ребра», нетрудно понять текущий вопрос (если дана полная его формулировка), а окончательный ответ можно включить в ту же таблицу. Но для улучшения интерфейса мы сделаем это.

3. Теперь нужно определить взаимодействие таблиц. Для этого нам в каждый момент работы ЭС потребуется знать, в каком состоянии она находится. Номер этого состояния будем хранить еще в одной таблице, которую назовем «Текущее» (тип данных — числовой).

Состояние
1

Рис. 5. Таблица «Текущее»

Как видим, в этой таблице всего одно поле, и в данный момент в него вписано начальное состояние ЭС.

4. Система будет работать следующим образом.

По текущему состоянию она генерирует запрос к таблице «Ребра» по полям КОНЕЦ_ДУГИ и ОТВЕТ_ПОЛЬЗОВАТЕЛЯ. Результатом его работы будет список возможных вариантов ответа пользователя на вопрос, соответствующий текущему состоянию системы. При этом должно быть реализовано соединение таблиц «Ребра» и «Текущее» по атрибутам Начало_дуги = Состояние.

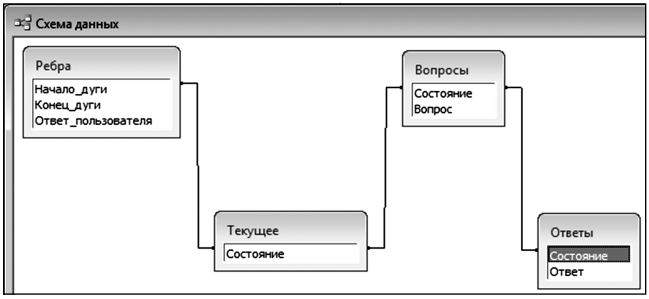

Рис. 6. Связи между таблицами

В зависимости от реакции на этот запрос пользователя ЭС переходит в следующее состояние и либо выдает ответ, либо генерирует очередной запрос.

5. Запрос выглядит следующим образом:

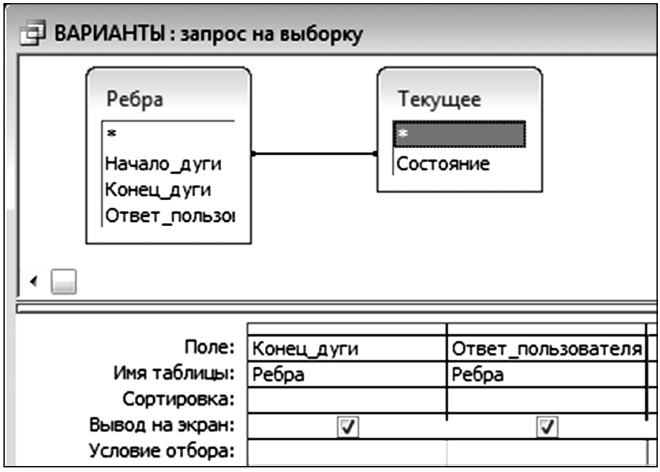

Рис. 7. Запрос «Варианты» для получения вариантов ответов

Например, в начальном состоянии системы (Состояние=1) выдает такой результат:

ВАРИАНТЫ : запрос на выборку

Конец_дуги	Ответ_пользователя
2	Ущерб более 1 млн руб.
3	Ущерб более 250 тыс. руб.
4	Ущерб менее 250 тыс. руб.

Рис. 8. Пример выборки по запросу «Варианты»

6. Теперь нужно предложить пользователю выбрать один из вариантов ответа (т. е. одну из дуг) и установить в таблице «Текущее» состояние, соответствующее метке КОНЕЦ выбранной дуги. Все это мы реализуем с помощью формы с полем со списком. Основным элементом этой формы является поле, в котором будет выпадать список дуг, доступных в данном состоянии. Кроме того, на форме необходимо поместить поле Состояние из таблицы «Текущее» (оно получается из списка доступных полей, открывающегося в меню «Конструктор/Добавить поля»).

Порядок действий следующий:

6.1. В режиме конструктора создаем форму «Варианты». В свойствах формы в качестве источника записей указываем таблицу «Текущее».

6.2. Выбираем из набора доступных инструментов «*Поле со списком*» и размещаем его в центре формы. В меню свойств поля со списком на вкладке «*Данные*» в качестве источника строк выбираем запрос «Варианты», а в качестве данных — поле «Состояние». На вкладке «*Макет*» поля со списком устанавливаем *Число столбцов*=2, чтобы вывести на экран не только номера вершин, но и текст вопросов. Таким образом, выбранный из списка вариант должен определять новое состояние системы.

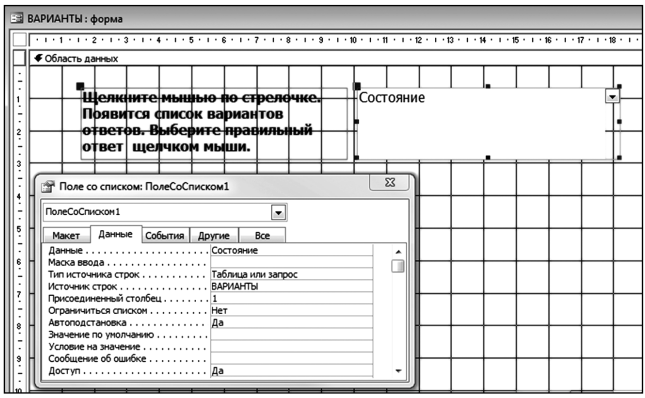

Рис. 9. Форма «Варианты»

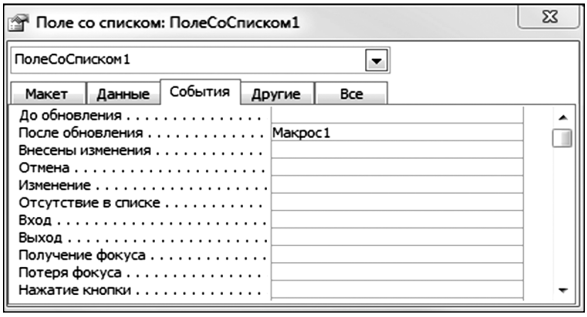

Рис. 10. Создание поля со списком в форме «Варианты»

6.3. Для удобства работы с системой создаем запрос «Вопросы» и «Ответы», используя одноименные таблицы (рис. 11).

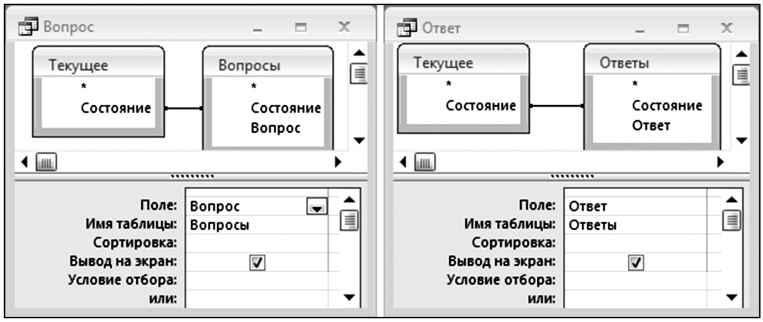

Рис. 11. Запросы для форм «Вопросы» и «Ответы»

6.4. На основе этих запросов создаем формы с теми же названиями. Для этого открываем запрос в режиме таблицы и меню «Создание/Форма», после чего редактируем форму в режиме Конструктора. В макетах обеих форм следует убрать заголовки, примечания, полосы прокрутки, область выделения, кнопки окна и кнопки перехода (Свойства/Макет или «Страница свойств/Макет»). Также следует установить шрифт 14 для полей «Вопрос» и «Ответ» и убрать у них границы. Созданные формы разместить как подчиненные на форме «Варианты» в соответствии с рисунком 14, удалив поля с названиями этих форм.

7. Для изменения состояния ЭС будем использовать макрокоманды. Нужно заготовить макрос, который будет обновлять форму «Варианты».

Окно для создания макроса выглядит так, как представлено на рис. 12.

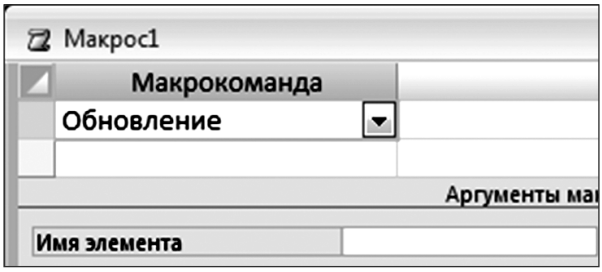

Рис. 12. Создание макроса для обновления формы «Варианты»

8. Чтобы при выборе значения в поле подстановок данные заносились в таблицу «Текущее», макрос должен выполняться после обновления поля со списком. Это должно быть отмечено на вкладке «События» меню свойств поля со списком.

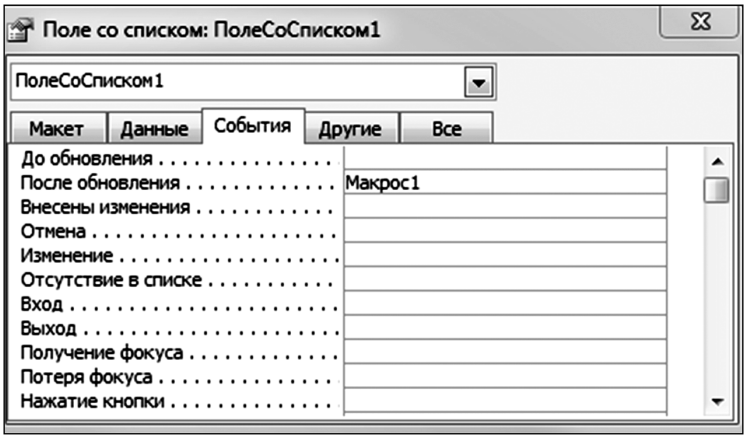

Рис. 13. Организация ссылки на макрос в поле со списком (форма «Варианты»)

9. Теперь система работоспособна. На рисунке 14 представлен окончательный вид формы «Варианты»; здесь для удобства перезапуска системы дополнительно выведены поле «*Состояние*» (меню «Конструктор/Добавить поля») и кнопка «*Обновить*» (она запускает созданный Макрос 1, обновляющий форму «Варианты»). Поле *Состояние* позволяет перезапускать систему заново (состояние=1), а также выбирать начальное состояние для случая, когда предварительная квалификация уже проведена и нет необходимости начинать поиск с нуля.

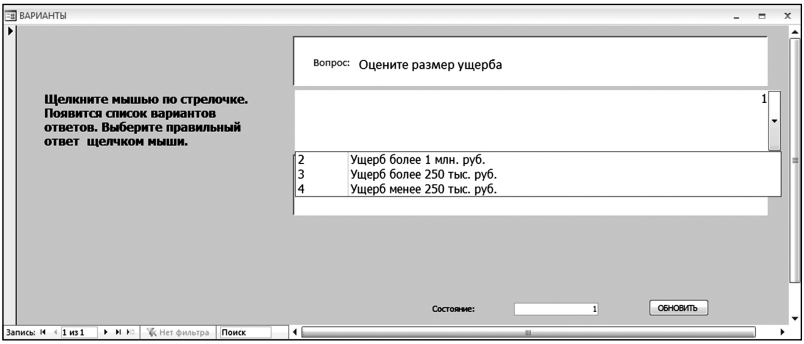

Рис. 14. Окончательный вид формы «Варианты»

Возможности ЭС при такой ее организации, конечно, весьма ограничены. Так, при сложном составе преступления придется осуществлять «спуск» по орграфу несколько раз по разным путям. Однако возможно дальнейшее совершенствование; например, есть возможность ввести простейший блок объяснений, как это делается в настоящих ЭС.

§ 4. Создание экспертной системы на основе готовой оболочки

Для создания более универсальной и мощной ЭС используем оболочку «Рапана».

При разработке базы знаний в ЭС «Рапана» не требуется написание каких-либо кодов или скриптов. ЭС «Рапана» в первую очередь ориентирована на экспертов — людей, обладающих знаниями в какой-либо области и желающих сделать эти знания доступными другим. ЭС «Рапана» может использоваться как для создания простых баз знаний локального применения, так и быть основой для решения глобальных задач любой сложности.

Прежде всего определимся с теми сущностями, с которыми предстоит работать, чтобы четко сформулировать задачу. Сущности — это и физические объекты, и общие представления; обычно с сущностями связываются атрибуты, которые описывают некоторые свойства сущностей. Вообще удачное выделение сущностей является одним из условий успеха в деле построения любой ЭС. В программе «Рапана» понятие «сущность» объединено с понятием «атрибут», а для представления *факта* используется пара *сущность — значение*. Значение является строковой величиной (например, «несовершеннолетний» или «до 18» применительно к атрибуту

«возраст»). Для проблемной области раздела бизнеса, например, в случае расторжения брака, мы выделим следующие сущности (факты):

Таблица 1. **Сущность — значение**

Сущность		**Принимаемые значения**
Бизнес возник до заключения брака	A	Да/Нет
Бизнесом владеют оба супруга	B	Да/Нет
Брачный договор	C	Да/Нет
Высокое участие супруга в бизнесе	D	Да/Нет
Есть несовершеннолетние дети	E	Да/Нет
Согласие о разделе имущества	F	Да/Нет
Согласие суда на раздел имущества	G	Да/Нет
Вывод (резюме)	R1	Имущество разделено без суда
	R2	Условия нужно пересмотреть с учетом требований суда
	R3	Бизнес делится согласно доле в уставном капитале и коэффициенту участия
	R4	Суд может не признать претензии истца на долю в бизнесе
	R5	Получит право на долю в бизнесе, размер которой установит суд
	R6	Суд учитывает условия договора, если он не противоречит закону
	R7	Владелец имеет преимущественное право на предприятие
	R8	Предприятие рассматривается как совместно нажитое имущество, возможно, 50 на 50
	R9	Суд учтет мировое соглашение

Работу с оболочкой «Рапана» следует начать с создания новой темы (в меню «Редактор — Темы» добавить запись с новым названием), задачи (меню «Редактор — Задачи») и подзадачи.

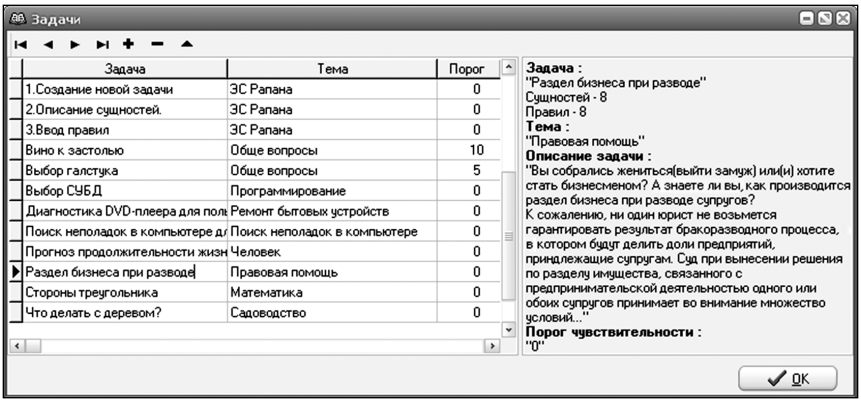

Рис. 15. Меню «Задачи»

Далее в меню «Редактор» вводятся выделенные ранее сущности и их характеристики.

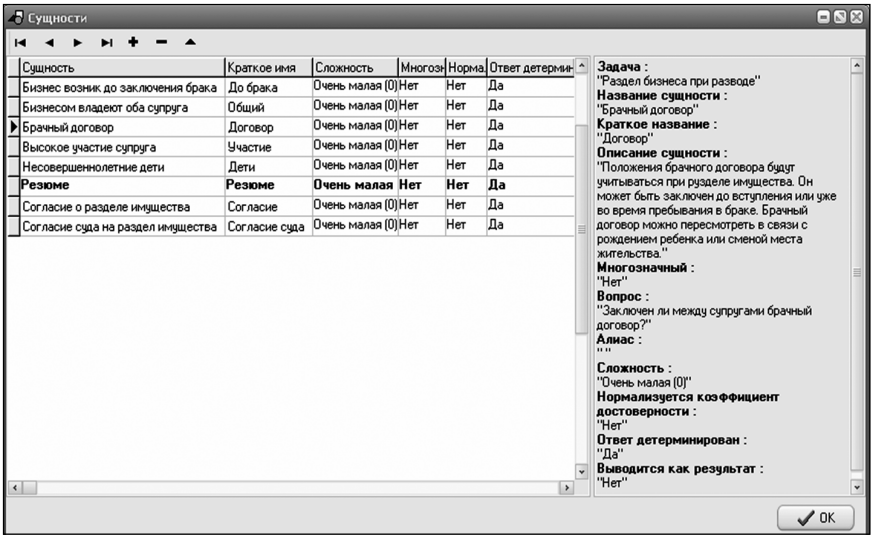

Рис. 16. Меню «Сущности»

В нашем примере все сущности однозначные. Однако данная программа предусматривает также возможность работы с многозначными сущностями, которые могут одновременно принимать несколько значений с различной степенью уверенности.

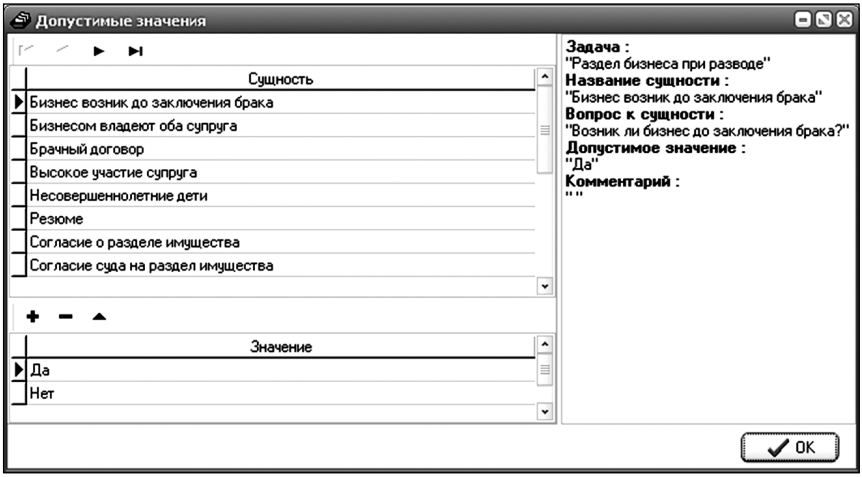

Рис. 17. Меню «Добавление сущностей»

После ввода сущностей и их возможных значений необходимо определить правила, связывающие факты. В нашем случае для этого следует просмотреть соответствующие статьи Гражданского кодекса Российской Федерации. Например, одно из правил имеет вид:

Правило1:

если

«Согласие о разделе имущества имеется» = Да

и

«Несовершеннолетние дети» = Нет,

то

«Имущество разделено без суда».

С учетом приведенных в таблице обозначений это правило можно записать в символической форме, что значительно короче:

Правило 1: F∧¬E→ R1.

Хотя для работы в ЭС «Рапана» перехода к символической записи не требуется, для экономии места мы приведем здесь оставшиеся правила в символическом виде, предоставив читателю самостоятельно произвести обратное преобразование:

Правило 2: F∧E∧¬G → R2.

Правило 3: ¬F∧B → R3.

Правило 4: ¬F∧¬ A∧¬C∧¬D → R4.

Правило 5: ¬F∧¬B∧¬ C∧D → R5.

Правило 6: ¬F∧¬B∧C → R6.

Правило 7: ¬F∧¬B∧¬ A∧¬C → R8.

Правило 8: F∧E∧G → R9.

В нашем случае правила (нормы) в явном виде можно получить из текста законодательства. Однако следует заметить, что при проектировании реальных ЭС редко приходится иметь дело со столь очевидными правилами. Правилоподобные знания приходится «извлекать» из экспертов с помощью специально разработанных методик, и это достаточно трудная задача. Причин тому немало: многие знания интуитивны и не осознаются самим экспертом именно как знания, некоторые из них трудно сформулировать в виде правила с использованием известных терминов, а иные весьма нечеткие или требуют учета довольно обширного контекста, который представляется само собой разумеющимся только эксперту. Многие эксперты, успешно используя в повседневной деятельности свои обширные знания, испытывают затруднения при попытке сформулировать и представить в системном виде хотя бы основную часть этих знаний: иерархию используемых понятий, эвристики, алгоритмы, связи между ними. Оказывается, что для подобной формализации знаний необходим определенный систематический стиль мышления, более близкий математикам и программистам, чем, например, юристам. Кроме того, не все эксперты желают делиться своим опытом. В итоге из эксперта удается «извлечь» от двух до пяти элементов знания (например, правил) в день, что, конечно, немного.

Ввод правил, условий их применения, а также коэффициентов доверия (далее — КД) (или коэффициентов уверенности) осуществляется через пункт меню «Редактор/Правила». КД выражает относительную уверенность в факте и представляет собой число в диапазоне от 0 (минимальное значение) до 100 (абсолютная уверенность). Методы работы с КД, отличными от единицы, исследуются нечеткой логикой. Их использование значительно расширяет возможности ЭС, позволяя принимать решения в условиях недостаточной определенности. В нашем примере, однако, эта возможность не используется, и всем правилам мы присвоим максимальные значения КД.

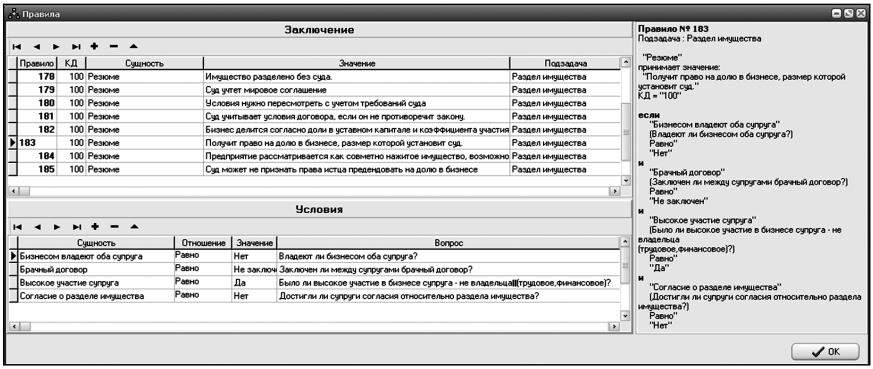

Рис 18. Меню «Правила»

После ввода правил обычно необходим процесс отладки базы знаний, который, как правило, носит итерационный характер. База проверяется на *полноту, непротиворечивость и связанность* введенных правил и сущностей. Проверка на *полноту* позволяет выявить сочетания значений сущностей, при которых сущность, определяемая через правила, не может быть определена. *Противоречивость* означает наличие групп правил, дающих разные результаты при одних и тех же входных данных, а отсутствие *связанности* означает наличие правил, не влияющих на итоговый результат.

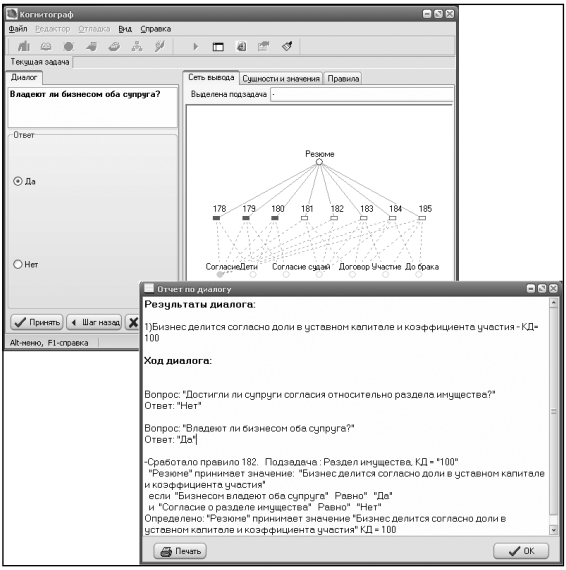

Рис. 19. Процесс и результаты работы экспертной системы

На рисунке 19 представлен процесс и результаты работы ЭС. Процесс отображается в виде графа базы знаний. Вершины, соответствующие сработавшим правилам, а также смежные с ними ребра окрашиваются в красный цвет. Вершины, соответствующие правилам, которые не могут сработать, заливаются серым. Текущие значения сущностей можно посмотреть, кликнув мышкой по соответствующей вершине. Порядок заданных вопросов и сработавших правил протоколируется; пример протокола представлен на этом же рисунке.

§ 5. Учебные задания

Самостоятельно постройте граф, отображающий логику произвольной статьи уголовного или уголовно-процессуального законодательства. Реализуйте ЭС продукционного типа на основе этого графа, используя средства Access.

Ознакомьтесь с книгой Д. В. Панова «Решатель юридических проблем: скорая правовая помощь на все случаи жизни». Выберите один из предложенных в книге алгоритмов и постройте на его основе ЭС, используя оболочку «Рапана».

Изучите следующую инструкцию по уплате налога на доходы физических лиц при продаже автомобиля: «Продавая автомобиль, следует иметь в виду, что сумма, вырученная вами от этой сделки, является вашим доходом и, возможно, с нее придется заплатить налог на доходы физических лиц. Сумма менее 250 тыс. налогом не облагается. Если вы владели автомобилем более трех лет, то налог платить не нужно, так как законом предусмотрен налоговый вычет в размере стоимости вашего автомобиля. Если же вы продаете автомобиль, купленный менее трех лет назад, то следует предоставить документы, подтверждающие стоимость вашего автомобиля при его покупке. Налог придется заплатить с разницы между суммой продажи и стоимостью покупки, за вычетом имущественного налогового вычета 125 тыс. руб. Если документов, подтверждающих стоимость покупки, нет, то придется платить налог со всей суммы продажи минус 125 тыс.». Изобразите алгоритм вычисления суммы налога в виде дерева решений и постройте консультирующую ЭС на основе оболочки «Малой Экспертной Системы», предварительно ознакомившись с ее руководством пользователя.

Выберите одну из предложенных задач и самостоятельно реализуйте продукционную ЭС:

а) определите причины неисправности компьютера;

б) определите причины неисправности автомобиля по заданным отклонениям в его работе;

в) определите вид транспорта по его внешним признакам и значениям определенных параметров;

г) определите наличие вирусов в компьютере по их характерным проявлениям, если они наблюдаются. База знаний содержит сведения о характерных проявлениях вирусов, об определении их принадлежности какому-либо классу вирусов и другие сведения.

ГЛАВА II. ДЕКЛАРАТИВНОЕ ПРОГРАММИРОВАНИЕ

§ 1. Факты, свойства, отношения

Среди инструментов ИИ особое место занимает Prolog — язык программирования, предназначенный главным образом для обработки текстовых символов и списков.

В отличие от таких известных языков программирования, как Pascal, C, Java или Fortran, Prolog является языком декларативного программирования. Он имеет встроенную машину логического вывода, которая просматривает заданную пользователем базу знаний на предмет достижения цели, заданной пользователем. База знаний включает в себя факты и правила, на основе которых цель доказывается как теорема, при этом для автоматического доказательства используется предложенный в 1960-х годах метод резолюций.

Для первоначального знакомства с языком Prolog используем Prolog Inference Engine — компонент, входящий в комплект Visual Prolog.

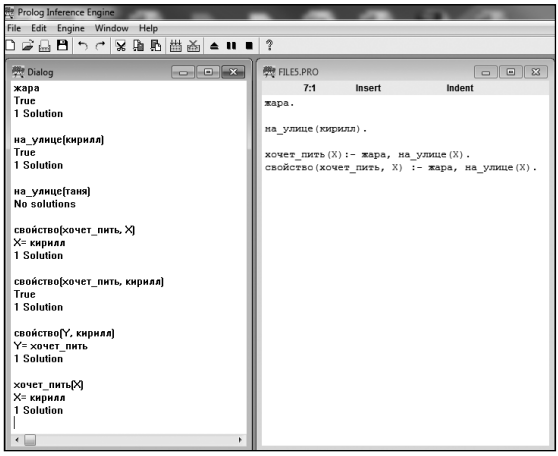

Рис. 20. Окна диалога и базы знаний Prolog

При запуске файла *pie.exe* открываются окно Dialog, а после выбора пункта меню File/New — еще одно окно, по умолчанию названное «FILES.PRO» (рис. 20). В окне «Dialog» пользователь может ставить перед системой цели и получать от нее ответы. Второе окно предназначено для формирования базы знаний системы.

Введем в окне «FILES.PRO» базы знаний текстовую строку строчными буквами, заканчивающуюся точкой, например, так:

```
жара.
```

Если теперь дать команду Engine/Reconsult и перейти в окно «Dialog», система перейдет в режим консультирования. После ввода в окне «Dialog» той же строки в ответ получим:

```
True
1 Solution
```

(результаты некоторых наших экспериментов представлены на рисунке 20).

Таким образом, система распознала текст «жара» как константу, принимающую значение «Истина». Иными словами, установлена истинность факта «жара». Так получилось, потому что он присутствует (так как только что введен нами) в базе знаний. Легко проверить, что для любой другой, не указанной в базе константы, мы получим ответ No solutions, т. е. «нет решений». Константы в Prolog всегда обозначаются строчными символами.

Теперь вернемся к базе знаний и дополним ее следующей строкой:

```
на_улице(кирилл).
```

Как и в первом случае, проверка истинного значения этого факта в режиме диалога будет успешной. Однако в данном случае мы имеем дело уже не с константой, а с отношением, в котором, например, *кирилл* является объектом, а *на_улице* — предикатным словом, обозначающим в данном случае свойство или унарное отношение, т. е. то, что говорится об объекте. Таким образом, данную строку можно рассматривать как выражение следующего факта: *кирилл* [имеет свойство быть] *на улице*.

Если теперь несколько изменить текст запроса в диалоговом окне и ввести:

```
на_улице(Who),
```

то в ответ немедленно получим:

```
Who = кирилл.
```

Who здесь обозначает переменную (по правилам Prolog, переменные должны начинаться с прописных латинских букв; это мешает нам дать переменной более удобное имя `Кто`)[1]. Приняв предикат с переменной в качестве цели, система логического вывода Prolog просмотрела все факты в базе знаний и подобрала такие значения переменной *Who*, при которых целевой предикат согласуется с этими фактами. В итоге в ответ на запрос вида *`на_улице(Who)`* мы получили список всех известных нашей базе знаний лиц, находящихся на улице.

Теперь введем в базу знаний правило:

```
хочет_пить(X) :- жара, на_улице(X).
```

Символ « :- » представляет упрощенное изображение символа «←» и выражает идею логического следования. В данном случае правило утверждает, что если истинны оба факта в правой части выражения (запятая между ними эквивалентна логической функции ∧, или AND[2]), то высказывание в его левой части тоже истинно. Иными словами, если налицо факт жары и одновременно некто X находится на улице, то этот X хочет пить. После ввода этого правила можно задавать системе вопросы типа:

```
хочет_пить(X),
```

т. е. «кто хочет пить?» и

```
хочет_пить(кирилл),
```

т. е. «хочет ли пить Кирилл?»; система их обработает, хотя соответствующие факты в явном виде в базе данных отсутствуют. Однако вопросы вроде «какие свойства имеет Кирилл?» или «кто какие свойства имеет?» задать не получится, потому что значение свойства в данном случае задано не аргументом, а предикатным словом, поэтому не может быть представлено переменной.

Использованный нами предикат имеет всего один аргумент и называется унарным (одноместным). Значительно большие возможности предоставляют предикаты большей местности, или, как говорят, арности. Так, *n*-арный предикат содержит *n* аргументов и, соответственно, может описывать отношение между *n* объектами. В рассматриваемом примере можно ввести бинарный (двухместный) предикат «свойств (X,Y)», в котором роль первого аргумента X играет значение свойства, а роль второ-

---

[1] Область действия имени переменной ограничена одним предложением (выражением, заканчивающимся точкой), поэтому в новом предложении имена переменных можно использовать повторно.

[2] В Prolog логическая функция дизъюнкции ∨ обозначается как «;», импликация ← как «:-», отрицание ¬ как «not» и коньюнкция ∧ как «,».

го — Y — имя объекта. Тогда, например, факт и правило можно записать следующим образом:

```
свойство (на_улице, кирилл).
свойство (хочет_пить, X) :-  жара, свойство (на_улице, X).
```

Теперь системе можно задавать вопросы относительно свойств, что очень удобно. Так, ввод в диалоговое окно выражения:

```
свойство (Y, кирилл)
```

приведет к перечислению всех известных свойств Кирилла, включая вычисленные с помощью правил:

```
Y = на_улице.
Y = хочет_пить.
```

А запрос свойство (Y, X) выдаст перечень всех свойств Y для всех объектов X, для которых эти свойства определены в базе знаний:

```
Y= на_улице, X= кирилл.
Y= хочет_пить, X= кирилл.
```

§ 2. Фреймы

Одним из эффективных способов представления знаний являются фреймы. Фрейм представляет собой именованную табличную структуру, содержащую набор слотов (аналогичны полям базы данных) и их значения. Каждый фрейм представляет некий объект базы знаний. Некоторые слоты у фрейма могут быть пустыми, а могут содержать ссылки на имена других фреймов. Таким образом организуется сеть фреймов. Слоты фрейма могут также содержать присоединенную процедуру, которую иногда называют «демоном». При таком подходе легко реализуется механизм наследования, характерный для объектного программирования. Наследование означает, что потомок объекта заимствует у своего предка его свойства — атрибуты (т. е. значения слотов). При этом один объект может иметь несколько предков, а значения унаследованных слотов могут быть переопределены.

Рассмотрим одну из возможных реализаций фреймовой структуры и наследования на языке Prolog.

Создадим фрейм, описывающий понятие юридического лица. Для этого отразим его следующим образом:

```
юр_лицо(вид, организация).
юр_лицо(полномочия, [предъявлять_иск, отвечать_по_иску,
заключать_договор]).
юр_лицо(обязанность, отвечать_имуществом).
```

```
юр_лицо(владеет, имущество).
юр_лицо(ответственность_участников,имуществом_организации).
```

Данный фрейм называется «юр_лицо» и имеет слоты: вид, полномочия, обязанности, владеет, ответственность_участников.

Теперь по аналогии создадим еще два фрейма:

```
ооо(вид, юр_лицо).
ооо(ответственность_участников, ограничена_долями).
рога_и_копыта (экземпляр,ооо).
рога_и_копыта (род_деятельности, заготовки).
```

Итак, здесь мы грубо описали следующую классификацию (таксономию): организация → юридическое лицо → (ООО) → рога_и_копыта. Здесь «организация» является родовым понятием, а «юридическое лицо» и «ООО» — его подвидами. Наименьшим объектом этой таксономии является конкретный экземпляр общества с ограниченной ответственностью, а именно ООО «рога_и_копыта».

Как имена предикатов, так и имена аргументов выбраны нами относительно произвольно и пока не определены. Дополним нашу программу следующими строками:

```
значение(Frame, Slot, Value):-
Query=..[Frame,Slot,Value],
call(Query),!.
значение(Frame, Slot, Value):-
род(Frame,ParentFrame), значение(ParentFrame, Slot, Value).
/* т. е. фрейм Frame наследует все слоты предка ParentFrame*/
род(Frame,ParentFrame):-
(Query=..[Frame, вид, ParentFrame];
Query=..[Frame, экземпляр, ParentFrame]),
call(Query).
```

/* т. е. предок — это такой фрейм, при котором фрейм Frame является его потомком (видом либо экземпляром*/).

Теперь мы имеем возможность узнавать имена и значения слотов любого фрейма с помощью трехместного предиката «значение» (Frame, Slot, Value). Например, запрос:

```
«значение (рога_и_копыта, полномочие, Power)».
```

В качестве ответа мы получим список полномочий, унаследованных от юридического лица, а именно:

```
Power = [предъявлять_иск, отвечать_по_иску, заключать_
договор]
1 Solution,
```

а результатом запроса

```
«значение (рога_и_копыта, ответственность_участников, L)»
```

будет:

```
L = ограничена_долями
1 Solution,
```

т. е. значение слота «ответственность_участников» унаследовано не от юридического лица, а из фрейма ООО, поскольку оно было переопределено в описании последнего.

Фреймовые структуры используются не только для описания сущностей, но и для описания действий и ситуаций.

§ 3. Поиск пути в графе

Графовые структуры широко используются для представления отношений, проблем и ситуаций. Графическими схемами мы описываем маршруты, процедуры, алгоритмы действий, а также семантические, физические, родственные и другие многочисленные связи между объектами.

Граф — это некоторое множество узлов и соединяющих их ребер. Ориентированные ребра называют дугами и представляют упорядоченными парами узлов. На рисунке 21 приведен пример ориентированного графа, содержащего девять узлов и девять дуг. Описать каждую из дуг можно двухместным предикатом, указав узлы начала и конца дуги соответственно в качестве первого и второго аргумента. Тогда описание графа на рисунке 21 будет следующим:

```
c(a,h). c(c,i). c(d,f). c(e,a). c(f,b). c(g,b). c(g,h). c(h,c). c(i,e).
```

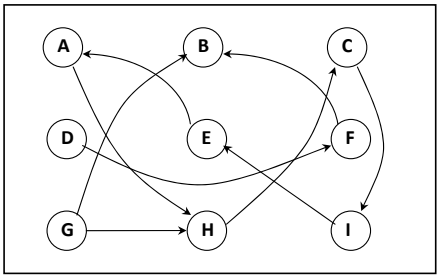

Рис. 21. Поиск пути в ориентированном графе

Поиск пути от одного узла к другому по направлению ребер графа является типовой задачей для систем ИИ. Для ее решения с помощью си-

стемы Prolog нам потребуется небольшая программа, текст которой приведен ниже:

```
member (X,[X|_]). member (X,[_|L]):- member (X,L).
путь (A, Z, Path) :- путь1 (A, [Z], Path).
путь1 (A, [A | Path1], [A | Path1]).
путь1 (A, [Y | Path1], Path):-
      c ( X, Y ),
      not member (X, Path1),
      путь1 (A, [X, Y | Path1], Path).
```

Эта короткая программа (чтобы понять ее суть, нужно более глубоко познакомиться с Prolog[3]) позволяет быстро найти путь Path между произвольными узлами Start и Finish с помощью предиката вида:

```
путь (Start, Finish, Path).
```

Введя, например, в диалоговом окне запрос путь (a, b, Path), в ответ немедленно получим: No solutions. И действительно, из рисунка можно обнаружить, что пути между узлами A и B нет. Аналогичный запрос для пары узлов G и A даст:

```
PATH= [g,h,c,i,e,a].
```

В ответ на целевой запрос PATH(X,Y, Path) получим список всех возможных путей в исследуемом графе:

```
X= a, Y= h, PATH= [a,h]
X= e, Y= h, PATH= [e,a,h]
X= i, Y= h, PATH= [i,e,a,h]
X= c, Y= h, PATH= [c,i,e,a,h]
X= c, Y= i, PATH= [c,i]
X= h, Y= i, PATH= [h,c,i]
X= a, Y= i, PATH= [a,h,c,i]
X= e, Y= i, PATH= [e,a,h,c,i]
X= g, Y= i, PATH= [g,h,c,i]
X= d, Y= f, PATH= [d,f]
...
```

и т. д.

```
...
X= g, Y= e, PATH= [g,h,c,i,e]
30 Solutions.
```

Чтобы в полной мере оценить скорость решения последней задачи, можно попробовать найти все 30 путей самостоятельно. Даже при столь небольшом числе узлов задача оказывается весьма утомительной.

---

[3] *Алгоритмы* искусственного интеллекта на языке Prolog / пер. с англ. И. Братко. М., 2004. 640 с.

Дугам графа могут быть поставлены в соответствие некоторые имена, обозначения или весовые коэффициенты (например, стоимости, расстояния или вероятности). И тогда актуальной становится задача поиска наиболее короткого, наименее затратного или наиболее вероятного пути. Решение такой задачи потребует незначительной модификации приведенной выше программы.

Очень актуальной для криминалистики задачей является выявление связей между лицами и/или объектами. Исходные данные для ее решения могут быть взяты из социальных сетей, разного рода учетов, детализации телефонных переговоров. При этом виды связей между людьми могут быть самыми разнообразными: родственными, по месту работы, месту жительства или учебы, общие знакомые, общие увлечения или посещаемые места. Покажем, как Prolog справляется с задачей построения цепочек таких связей.

Пусть в некоторой базе знаний о персоналиях связи некоего Сергеева зафиксированы следующим образом:

```
c(место_работы,патп16, сергеев).
c(посещает,боулинг,сергеев).
c(посещает,ипподром,сергеев).
c(место_рождения,ковров, сергеев).
c(знаком,лапин,сергеев).
c(судимость,158, сергеев).
c(сын, александр, сергеев).
```

Такое описание связей отличается от предыдущей задачи только использованием трехместных предикатов. Первый аргумент в данном случае содержит информацию о характере связи, а второй и третий представляют связываемые объекты. Заметим, что отобразить эту информацию можно было и иначе, например, предикатами вида — знаком (лапин, сергеев), однако такое представление затрудняет групповые операции с видами связей, поскольку последние представлены не аргументами, а именами предикатов.

В нашем трехместном предикате с (Z, X, Y) аргумент Z определяет характер связи между объектами X и Y. Очевидно, что эти связи разнообразнее связей вида с (X, Y), — тех, что были в графе на рисунке 1. Кроме того, некоторые из них ориентированы, а некоторые — нет. Так, связь с местом работы, явно выражаемая предикатом с(место_работы, патп16, сергеев), является ориентированной: ПАТП-16 является местом работы Сергеева, но Сергеев, очевидно, не является местом работы ПАТП=16. Однако, если нас интересуют опосредованные связи — между людьми через третьих лиц, через общие интересы, предметы и другие объекты, то необходимо

учитывать связи в любом направлении, т. е. дуги нашего графа не должны быть ориентированы. Для учета связей в любом направлении введем дополнительное правило:

```
связаны (J, Y, X) :- c(J, Y, X); c(J, X, Y).
```

Затем заменим используемый ранее предикат c(X, Y) только что созданным предикатом связаны (J, Y, X). Окончательно программа поиска пути по графу приобрела следующий вид:

```
путь (A, Z, Path):- путь1 (A, [Z], Path).
путь1 (A, [A | Path1], [A | Path1]).
путь1 (A, [Y | Path1], Path):- связаны (J, Y, X), not member (X, Path1),
путь1 (A, [X,J,Y | Path1], Path).
```

Теперь можно проиллюстрировать построение цепочек. Дополним нашу базу знаний записями о других персонах, в том числе:

```
c(место_учебы, школа112, александр). c(посещает,брейкданс,александр).
c(знаком,валерия,александр).

c(место_учебы, школа112, татьяна). c(посещает,фитнес,татьяна).
c(место_рождения,петропавловск, татьяна).c(знаком,валерия,татьяна).

c(место_работы,патп16, прошкин). c(посещает,ипподром,прошкин).
c(знаком,лапин,прошкин). c(знаком,литвина,прошкин).
c(родств,оксана,дочь,прошкин).

c(место_работы,аптека8,литвина). c(посещает,мсч12,литвина).
c(знаком,лапин,литвина). c(дочь,татьяна,литвина).
```

На введенный в диалоговое окно запрос

«путь(сергеев, валерия)»

система предложит 14 решений, причем кратчайшей цепочкой будет:

PATH= [сергеев,сын,александр,знаком,валерия],

т. е. с Валерией знаком Александр, сын Сергеева. Одна из более длинных цепочек имеет вид:

PATH=[сергеев,посещает,ипподром,посещает,прошкин,знаком, лапин,знаком,литвина,дочь, татьяна, знаком,валерия].

Иными словами, Сергеев посещает ипподром, посещаемый также Прошкиным, который знаком с Лапиным, который знаком с Литвиной, дочь которой Татьяна знает Валерию.

Имеется, конечно, и возможность вывести перечень всех связей для любого объекта базы знаний путем запроса: путь (сергеев, X, Path). Однако в реальных базах знаний при этом следует принять меры для того, чтобы максимально ограничить количество звеньев в искомых це-

почках: при увеличении числа узлов в графе общее количество возможных путей лавинообразно возрастает, требуя от вычислительной системы все больше ресурсов. Для поиска в больших базах знаний применяются такие приемы, как упорядочение строк, индексация и другие специальные алгоритмы, которые здесь не рассматриваются. К счастью, на практике слишком длинные цепочки и не требуются. Популярна теория «пяти рукопожатий», согласно которой любые два жителя Земли связаны друг с другом цепочкой, включающей не более пяти посредников.

§ 4. Правила и нормы

В качестве примера использования семантических отношений и выделенных концептов для описания реальной ситуации рассмотрим следующую фабулу: «Смирнова, желая отомстить Петрову, предоставила Иванову ключ от квартиры Петрова. Воспользовавшись ключом, Иванов из корыстных побуждений 15 ноября 2003 г. тайно проник в квартиру Петрова и изъял принадлежащий тому телевизор. В результате Петрову нанесен ущерб 5 тыс. рублей». Все события описанного сценария являются аргументами стандартных отношений, перечисленных выше. С их помощью указанная фабула может быть записана в следующем виде[4]:

агент (смирнова, предоставление_средств).	объект (ключ, предоставление_средств).
цель (ущерб (0), предоставление_средств).	реципиент (петров, ущерб (0)).
адрес(иванов,предоставление_средств).результат(изъятие,предоставление_средств).	
агент (иванов, изъятие (1)).	дата ([15,11,2003], изъятие(1)).
объект (телевизор, изъятие (1)).	место (жилище, телевизор).
принадл (жилище, петров).	принадл (телевизор, петров).
цель (выгода, изъятие (1)).	объект (ключ, использование_средств).
следствие (ущерб (1), изъятие (1)).	величина (5000, ущерб (1)).
способ ([противоправно, безвозмездно, тайно, использование средств], изъятие (1)).	

Здесь факт «ущерб (0)» отражает идеальный результат развития событий в представлении Смирновой, а ущерб (1) — последствия реальных действий Иванова по изъятию телевизора (в общем случае эти результаты могут быть различны). Вообще большинство аргументов отношений необходимо снабжать уникальными идентификационными номерами, чтобы различать одноименные объекты — действия, предметы и т. д.

[4] *Гайдамакин А. А.* О формальном описании семантических связей в статьях Уголовного кодекса // Юристъ-правоведъ. Ростов н/Д, 2008. № 4. С. 99–104.

Далее машиной логического вывода активируются общие правила, полученные при построении формально-логической модели ст. 158 УК РФ. Вот некоторые из них, записанные на Prolog:

```
хищение (X,Y):- агент (X,изъятие(Z)), объект (Y,изъятие(Z)),
            not (владелец(X,Y)), способ (D,изъятие(Z)),
            подмножество ([противоправно, корыстно, безвозмездно],D),
            величина (V,ущерб(Z)),V>0.
кража (X,Y):- хищение (X,Y), способ (D,изъятие (Z)), элемент (тайно, D),
            величина (V, ущерб (Z)),V>0.
квалиф (ст_158_2в_КРАЖА,X,Y):-
            кража (X,Y), величина (V,ущерб(Z)), V>2500.
квалиф (ст_158_3_КРАЖА,X,Y):-
            кража (X,Y),место(Place,Y),member (Place,[жилище]).
```

Здесь используются предикаты «подмножество и элемент», истинные в случае, когда их первые аргументы (список или переменная соответственно) содержатся во втором списке. По запросу вида «квалиф (St, X, изъятие (Z)) (т. е. указать статью St, квалифицирующую действия субъектов X по изъятию (Z))», система на основе приведенных и других правил выдаст следующее решение:

```
ST= ст_158_2в_кража, X = иванов, Y= телевизор
ST= ст_158_3_кража, X = иванов, Y= телевизор
```

На запрос об оценке действий Смирновой система сделает вывод о соучастии Смирновой в краже в качестве пособника.

Кроме приведенной выше формы записи ту же фабулу можно описать и другими способами. Варианты описания могут быть как более подробными (на уровне элементарных действий — *взял, изъял, вынес, продал* и т. п.), так и менее (на уровне метапонятий более высокого уровня — *содействие, хищение, проникновение в помещение* и т. п.). Проектируемая система должна допускать любые варианты, это требует серьезной работы по структурированию знаний предметного поля, по выявлению основных понятий и отношений между ними.

Рассмотренный в настоящей статье способ представления знаний, конечно, не является единственно возможным. Формируя семантическую сеть на основе бинарных отношений, мы, по существу, перечисляем признаки различных действий, лиц и предметов, поэтому, например, характеристика некоторого действия оказывается эквивалентной заполнению записи в некоторой таблице «Действие реляционной базы данных» с соответствующими полями. Имеет свои достоинства и применение многоместных предикатов. В этом случае отношения могут иметь более двух аргументов, и тот факт, что Иванов продал телевизор

посреднику, может быть записан в виде: «*продал (Иванов, телевизор, посредник)*». Такая запись более лаконична, чем бинарная, к тому же здесь возможно введение операторов, позволяющих приблизить форму записи к естественному языку и осуществлять ввод, например, в таком виде: *иванов продал телевизор*.

Многоместные предикаты продуктивно используются вычислительной системой в процессе логической обработки и при выводе информации (выше, например, был введен предикат «квалиф (St, Person, Action))». Однако описание сценария событий с их использованием затрудняется, поскольку при такой записи событие является уже не вершиной семантического графа, а отношением. Во всяком случае, утрачивается единообразие этого описания. Другой недостаток — большое количество отношений и их аргументов. Чтобы обеспечить корректный ввод и исключить ошибки, ввод лучше осуществлять с помощью форм — шаблонов и полей со списком. В бинарной же форме записи число отношений можно ограничить, причем каждое из них имеет только два аргумента, что упрощает работу. Кроме того, такая форма удобна для автоматической обработки текстовой информации и используется, например, в системах компьютерного перевода и анализа текста. Таким образом, при описании сценариев в процессе разработки юридической ЭС имеет смысл отдать предпочтение двухместным предикатам семантических отношений как более универсальной форме записи [5].

§ 5. Учебные задания

1. Определив необходимые константы, функциональные и предикатные символы и задав их интерпретацию, представьте на языке логики предикатов следующую информацию:

а) слышит звон, да не знает, где он;

б) нет дыма без огня;

в) некоторые граждане России имеют двойное гражданство;

г) не каждый курсант, поступивший на первый курс академии, получает диплом.

2. Используя инструментарий Prolog, проверьте правильность суждения: «Если вокруг огнестрельной раны имеются следы внедрения пороха, значит выстрел был близкий. В данном случае выстрел был дальний, так как вокруг раны на теле нет следов внедрения пороха».

[5] Там же. С. 99–104.

3. Совершена кража в ювелирном магазине. Это мог сделать либо уголовник «Слон», либо гастролер «Артист», либо Павел Смышляев. Следователь получил информацию, что: а) ювелирный магазин ограбил не «Артист»; б) магазин «взял» Смышляев. Однако позже выяснилось, что только одно из этих сообщений соответствует действительности. Кто совершил кражу? Решите задачу средствами Prolog и опишите ход рассуждений.

4. В деле имеются два подозреваемых — Лившиц и Потапенко. Опрошено четыре свидетеля. Первый свидетель сообщал: «Лившиц не виноват», второй свидетель говорил: «Потапенко не виноват», третий свидетель заявлял: «Из двух показаний по крайней мере одно истинное», а четвертый свидетель утверждал: «Показания третьего свидетеля ложные».

Четвертый свидетель оказался прав. Кто совершил убийство? Решите задачу средством Prolog.

5. Средствами Prolog проверьте истинность рассуждения:

Если Джонс не встречал этой ночью Смита, то либо Смит был убийцей, либо Джонс лжет. Если Смит не был убийцей, то Джонс не встречал Смита этой ночью, и убийство имело место после полуночи. Если убийство имело место после полуночи, то либо Смит был убийцей, либо Джонс лжет. Следовательно, Смит был убийцей.

6. Известно, что: либо злоумышленник уехал на автомобиле, либо свидетель ошибся; если злоумышленник не имел сообщника, то он уехал на автомобиле; либо у злоумышленника был сообщник, либо у него был ключ; у злоумышленника был ключ. К какому заключению можно прийти, имея эти данные?

7. Создайте в Prolog базу знаний для описания родственных отношений в конкретной семье (рис. 22).

База знаний должна состоять из совокупности фактов (свойств сущностей) и правил. Факты: мужчина, женщина, родитель, в браке. Правила определяют отношения: брат/сестра, муж/жена.

8. В городе, где был ограблен банк, живут 5 подозрительных личностей: Джек, Том, Сэм, Энди и Лиз. Известно, что грабитель — мужчи-

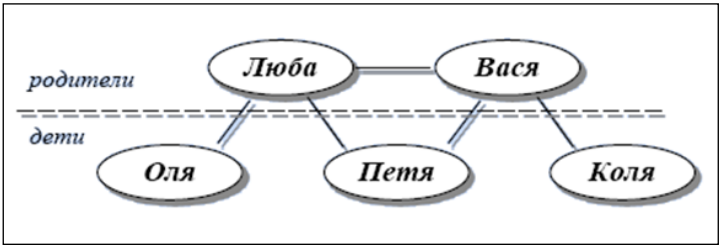

Рис. 22. Схема родственных отношений

на, приехавший на грузовике и одетый в темную одежду. Другие факты, также известные полиции: Том в манере одеваться копирует Энди; Сэм и Джек — заядлые курильщики; Джек ездит на Volvo, а Энди на грузовике; Лиз гоняет по городу на велосипеде; Сэм перемещается на том же транспорте, что и Энди; Лиз любит красные костюмы и куртки; Энди, как правило, одевается в зеленое; все курящие жители города N носят темную одежду. Определите имя (имена) преступника, представив известную информацию в виде фактов и правил языка Prolog и задав программе соответствующий вопрос [6].

[6] *Корухова Ю. С.* Управление знаниями : учеб. пособие. М., 2012. С. 13.

ГЛАВА III. НЕЙРОННЫЕ СЕТИ

§ 1. Юристы и нейроны

Среди систем ИИ особую популярность в последнее время получили искусственные нейронные сети (далее — ИНС) — компьютерные программы, архитектура которых похожа на строение нервной системы живого организма. В 2017 г. Г. Греф заявил: «Товарищи юристы, забудьте свою профессию. В прошлом году 450 юристов, которые у нас (т. е. в Сбербанке. — *А. Г.*) готовят иски, ушли в прошлое, были сокращены. У нас нейронная сетка готовит исковые заявления лучше, чем юристы, подготовленные Балтийским федеральным университетом»[1]. Это, конечно, преувеличение. ИНС не могут готовить иски и вообще пока еще очень плохо генерируют логически связные тексты. Анализировать и толковать правовые нормы они тоже пока не в состоянии. Однако они способны оценить вероятные перспективы принятия судом благоприятного решения по некоторому классу исков. По результатам их работы возможна корректировка содержания искового заявления, а исследование обученной нейросети позволяет выявлять новые зависимости и иногда вводить новые понятия.

Основой нейронной сети является, естественно, нейрон. Биологический нейрон, как и любая клетка, состоит из ограниченного мембраной тела, внутри которого располагается ядро. Нейроны имеют многочисленные отростки: короткие древовидные дендриты можно рассматривать как входы нейрона, а длинные (до 1 м) аксоны — как его выходы. Аксоны используются для обмена информацией с другими нейронами. Аксон одного нейрона соединяется с дендритом другого посредством синапсов, выполняющих функции специальных контактов. При поступлении нервного импульса на такой контакт

[1] *URL:* http://www.rbc.ru/business/23/07/2017/5974b7a69a79477896b6708d (дата обращения: 20.04.2019).

происходит химическая реакция с выбросом из аксона биологически активного вещества (нейромедиатора), которое попадает в клетку, в той или иной степени стимулируя ее электрическое возбуждение. Степень возбуждения нейрона зависит от суммарного количества нейромедиатора, выделенного на всех синапсах.

Отдельный нейрон не является элементарной единицей обработки информации, а выполняет функции нервного центра. Дендриты и аксоны могут вступать в связи с участками мембран других нейронов, образуя сети. Эти сети и служат системами обработки информации. Входное воздействие в виде набора (вектора) входных сигналов одновременно подается на вход сети, образованной нейронами человеческого мозга, и это воздействие параллельно, т. е. волной распространяется по всей сети.

Классической задачей для ИНС является распознавание (классификация) образов, например, буквенно-цифровых символов (как в программе FineReader) или лиц людей (как при поиске фотографий в Google или Yandex). Однако сегодня ИНС все чаще используются в задачах диагностики и прогнозирования, которые имеют много общего с задачами классификации. Незаменимы нейронные сети при решении многих неформализуемых или трудно формализуемых задач — там, где обычные алгоритмические решения оказываются неэффективными или вовсе невозможными.

Основные типы задач, решаемых с помощью нейронных сетей:

1) разделение на классы или классификация, таксономия или кластеризация — если без учителя (сети естественной классификации);

2) предсказание (действительного) числа или предикция (иначе называемая нейросетевой регрессией);

3) распознавание образов;

4) оптимизация;

5) прогнозирование;

6) моделирование.

Дополнительные задачи:

7) прогноз («что будет завтра?»);

8) условный прогноз («что будет завтра, если ..?»);

9) определение значимости входных параметров.

Из конкретных задач, решение которых доверяют ИНС, можно к упомянутым выше добавить прогнозирование курса валют и котировки ценных бумаг; оценку финансового состояния предприятий; оценку стоимости недвижимости; прогнозирование риска (например,

невозврата кредитов); извлечение знаний (Data Mining) из больших объемов данных в бизнесе, финансах и научных исследованиях. Для решения указанных задач разработан и используется математический аппарат ИНС.

Важнейшим достоинством ИНС является их способность к обучению. В процессе обучения нейронная сеть выявляет закономерности между входными и выходными данными, а после обучения может дать верный результат даже для тех данных, которых не было в выборке, использованной для ее обучения, т. е. она способна к обобщению и прогнозированию. Если обучить нейронную сеть на проверенных практикой договорах, то можно доверить ей поиск проблемных мест и пробелов в новых договорах. Нейронная сеть, освоившая типовые судебные решения, может предсказать результат решения по иску, аналогичному тем, на которых она обучена. Адвокату она подскажет наиболее перспективный иск для защиты, а клиенту посоветует выбрать адвоката в зависимости от того, какой именно правовой интерес нуждается в защите.

К другим преимуществам нейронных сетей можно отнести:

— относительную простоту производимых вычислений и используемых структур данных;

— легкость аппаратной реализации;

— возможность выделения наиболее характерных особенностей входного сигнала;

— возможность построения дерева решения по обученной нейронной сети[2].

§ 2. Нейронные сети в задачах классификации

Если в «обычных» программах полученная компьютером информация обрабатывается по определенному алгоритму последовательно, шаг за шагом, то в ИНС вычисления производятся параллельно: фрагменты полученной компьютером информации обрабатываются одновременно во множестве различных процессов, и лишь на выходе итоги вычислений объединяются в общий результат. Теоретическая (математическая) модель отдельного нейрона создана уже достаточно давно: ее в 1943 г. предложили У. Мак-Каллок и В. Питтс. Искусственный нейрон имитирует в первом приближении свойства биологического нейрона.

---

[2] *Осипов Г. С.* Методы искусственного интеллекта. М., 2011. С. 246–248.

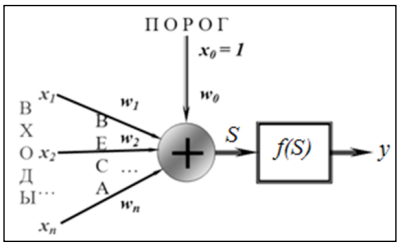

Рис. 23. Математический нейрон как сумматор с пороговой функцией

Для того, чтобы корректно учитывать влияние каждого сигнала на состояние нервной клетки, в математической модели нейрона входные сигналы умножаются на коэффициенты w_i, пропорциональные количеству нейромедиатора, выделяемого на соответствующем синапсе при разовом воздействии нервного импульса. Все произведения суммируются, определяя уровень активации нейрона (рис. 23, где n — число входов нейрона, x_i — значение i-го выхода нейрона, w_i — вес i-го синапса):

$$S = w_1 x_1 + w_2 x_2 + \ldots + w_i x_i + \ldots w_n x_n = \sum_{i=1}^{n} \mathrm{WiXi}.$$

Синаптические веса w_i могут принимать как положительные, так и отрицательные значения. В первом случае синапс оказывает возбуждающее, а во втором тормозящее действие, препятствующее возбуждению клетки другими сигналами. После выполнения суммирования математический нейрон формирует выходной сигнал y согласно правилу:

$$f(x) \begin{cases} 1, & S \geq \theta \\ 0, & S < \theta, \end{cases}$$

где $\theta = w_0 x_0$ — порог чувствительности нейрона. Из формулы видно, что в случае, когда взвешенная сумма входных сигналов меньше некоторой пороговой величины θ, то математический нейрон не возбужден. Если же суммарное значение входных сигналов превышает θ, то на выходе нейрона формируется сигнал y=1 (рис. 24).

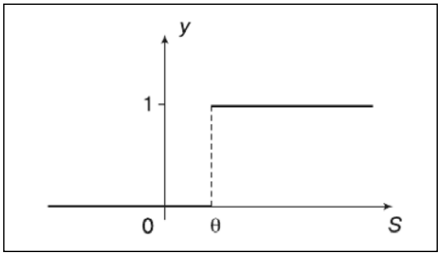

Рис. 24. Пороговая функция типа «ступенька»

В данном случае порог определяется ступенчатой единичной функцией, однако часто применяются и другие, более «гладкие» виды пороговых функций, например, логарифмическая, сигмоидальная или гиперболический тангенс.

Нейрон успешно выполняет функции линейного классификатора. Так, если у него имеется единственный вход, то все значения сигналов на этом входе делятся на два класса: ниже порога (0 на выходе) и выше него (1 на выходе). Если имеется два входа (или, иными словами, вектор входных значений имеет размерность, равную n=2), то множество возможных состояний входных сигналов можно геометрически представить в виде плоскости. В этом случае точки, соответствующие порогу активации, образуют линию на этой плоскости, которая отграничивает область значений входного вектора, соответствующих активному состоянию нейрона, от области его неактивного состояния.

Рассмотрим математический нейрон с двумя входами (n=2). Значения сигналов x_1 и x_2, соответствующие порогу активации нейрона, лежат в данном случае на прямой, которая описывается уравнением:

$$S = w_0 + w_1 x_1 + w_2 x_2 = 0.$$

Эта прямая представлена на рисунке 25. На графике для векторов приняты следующие значения: $w_0 = -2$; $w_1 = 1$; $w_2 = 1$, так что $x_1 + x_2 = 2$.

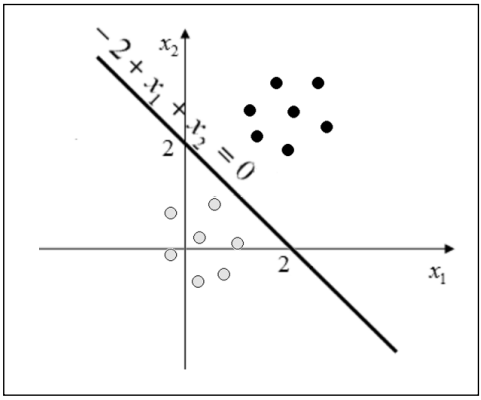

Рис. 25. Линия активации в линейном классификаторе

Точки, лежащие выше прямой (например, точка *a*), относятся к первому классу, а точки, лежащие ниже (точка *b* и другие светлые точки), — ко второму.

Интерпретация сигналов на входах нейрона может быть самой разной. Если рассматривать только положительные значения координат и поставить в соответствие x_1, например, оценку потребности в приобретении некоторого товара, а x_2 — его качество, то данный линейный классификатор поделит пространство векторов (x_1, x_2), характеризующих потенциальную покупку, на классы приемлемых и неприемлемых характеристик. Меняя с помощью весовых коэффициентов наклон прямой, можно регулировать соотношение между x_1 и x_2 на пороге активации нейрона, а сдвиг прямой по осям позволит установить требуемую величину порога активации.

При увеличении числа входов мерность векторного пространства растет, и оно делится на два класса уже не точкой или прямой, а гиперплоскостью.

Простейший линейный классификатор можно использовать не только для обработки непрерывных сигналов. Если ограничить множество значений x_i нулями и единицами, нейрон с успехом реализует такие логические функции, как OR (рис. 26 а) или AND (рис. 26 б). Но, например, функция XOR (исключающее OR, рис. 26 в) не может быть реализована одним нейроном, так как соответствующие значения занимают несвязные области векторного пространства и поэтому не могут быть разделены одной прямой. Поэтому реальные нейронные сети представляют собой совокупность нескольких линейных классификаторов — нейронов.

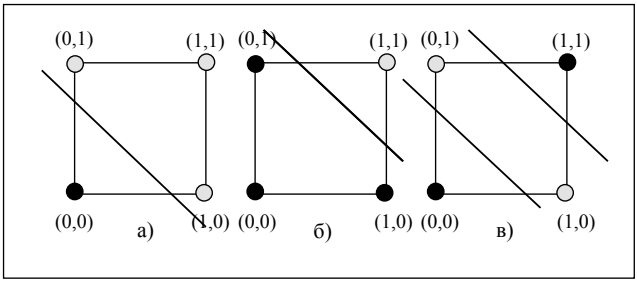

Рис. 26. Реализация логических функций математическим нейроном

Совокупность нейронов образует слой нейронной сети, при этом выходы нейронов предыдущих слоев обычно являются входами нейронов следующих слоев, образуя многослойную ИНС. Входной вектор сети и выход последнего нейрона называются внешними слоями, все остальные слои рассматриваются как внутренние (скрытые). Двухслойная сеть позволяет соотносить входной сигнал с областью пространства, ограниченной выпуклым многогранником, а трехслойная — произвольным многогранником.

§ 3. Обучение нейронной сети

Для своей работы нейронная сеть требует определенной подготовки данных. Ее успех во многом зависит от подбора параметров предметной области и обучающих примеров. Эти параметры могут иметь разнообразный характер, однако в любом случае должны быть закодированы числами, как и желаемые выходные сигналы.

Если удается выделить репрезентативный набор факторов, всесторонне характеризующий некоторую ситуацию (деяние), и имеется достаточно статистического материала для формирования векторов обучающей выборки (как с положительными, так и с отрицательными примерами), то вполне возможно обучить нейронную сеть распознаванию (квалификации) этой ситуации (деяния) и оценке ее правомерности. Регулярно переобучая сеть на материалах последних решений суда некоторой конкретной юрисдикции, можно с высокой вероятностью предсказать его решение по новому делу. И такие системы существуют. Проблема, однако, в том, что многие факторы, описывающие положение вещей, являются оценочными категориями. Они субъективны, их оценка сама по себе требует либо построения отдельной ИНС, обученной на решениях конкретного судьи, либо увеличения числа факторных признаков в основной ИНС. Поэтому, определяя состав факторных признаков, следует избегать оценочных категорий, что весьма непросто. Кроме того, выбранный разработчиком набор факторных признаков всегда может оказаться неполным, например, вследствие изменения политической конъюнктуры. Видимо, поэтому американская система ROSS (программа, работающая на когнитивном компьютере IBM Watson) прогнозирует решения Верховного суда с вероятностью лишь около 70%. Здесь, правда, следует заметить, что эксперты из числа бывших членов этого суда верно предсказывают его решения только в 53% случаев.

Обучение сети — это итерационный процесс подстройки весов нейронов таким образом, чтобы группе входных сигналов, принадлежащих к одному классу, соответствовал один и тот же выходной сигнал. Обучение может быть автоматическим, при этом алгоритмы обучения нейронных сетей построены таким образом, что наиболее характерным признакам (компонентам) входного вектора в процессе обучения автоматически приписываются большие веса. При обучении с учителем нейронной сети предъявляются значения как входных, так и желательных выходных сигналов, и она подстраивает веса своих синоптических связей по некоторому внутреннему алгоритму. При удачном обучении в результате работы алгоритма происходит переход системы в новое устойчивое состояние,

т. е. весовые коэффициенты стабилизируются. Обученная сеть приобретает способности к обобщению и прогнозированию; можно, например, представить ИНС в качестве обучающих примеров ряд строк из таблицы умножения, и она обретет способность «предсказывать» результаты примеров, не вошедших в обучающую выборку. Рассмотрим два примера, используя для моделирования разные программные средства.

Пример 1. В этом примере мы *научим нейронную сеть арифметическим операциям* — сложению и умножению. При работе будем использовать нейронную сеть из состава аналитической платформы *Deductor*.

1. Прежде всего необходимо подготовить исходные данные, из состава которых позже будут выделены обучающее и тестирующее подмножества. С этой целью в табличном процессоре *Excel* создадим таблицу, состоящую из 4 столбцов (A, B, C, D) и 200 строк. Столбец A содержит первый аргумент функции (т. е. слагаемое или сомножитель), столбец B — второй аргумент, столбец C — вид операции (текст «сложение» или «умножение»), а столбец D — желаемый результат. Таблица должна выглядеть так, как на рисунке 27.

Подготовленные данные следует сохранить в текстовом формате с разделителями табуляции (назовем файл *обучающие_данные.txt*), поскольку используемая нами демонстрационная версия программы *Deductor* имеет некоторые ограничения и не позволяет импортировать данные в других форматах.

2. Запустите *Deductor Studio Academic* и вызовите мастер импорта данных (клавиша F6 или пиктограмма ). Укажите тип импортируемых данных (Text), а затем имя файла *обучающие_данные.txt*. Задавая параметры столбцов, укажите для операндов *A*, *B* и *D* вещественный тип данных, для столбца *C* (для «операции») — строковый (они предлагаются по умолчанию). После этого кнопкой «Пуск» завершаем импорт. Окно должно приобрести вид, показанный на рисунке 28.

	A	B	C	D
1	**A**	**B**	**Операция**	**D**
2	0	0	сложение	0
3	1	0	сложение	1
4	2	0	сложение	2
5	3	0	сложение	3
6	4	0	сложение	4
7	5	0	сложение	5
8	6	0	сложение	6
9	7	0	сложение	7
10	8	0	сложение	8
11	9	0	сложение	9
12	0	1	сложение	1
13	1	1	сложение	2
14	2	1	сложение	3
99	7	9	сложение	16
100	8	9	сложение	17
101	9	9	сложение	18
102	0	0	умножение	0
103	1	0	умножение	0
104	2	0	умножение	0
197	5	9	умножение	45
198	6	9	умножение	54
199	7	9	умножение	63
200	8	9	умножение	72
201	9	9	умножение	81

Рис. 27. Обучающие данные

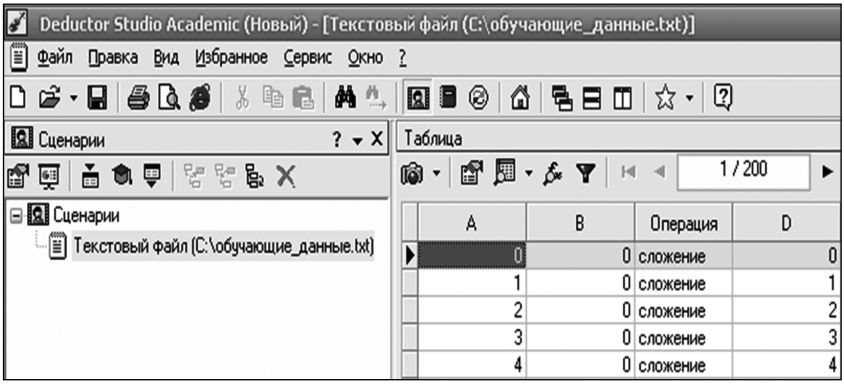

Рис. 28. Импорт обучающих данных в Deductor Studio

3. Запустите мастер обработки (клавиша F7 или пиктограмма). В появившемся окне выберите в разделе Data Mining режим «Нейросеть». Установите назначение полей *A*, *B* и *Операция* как входное, а поле *D* — как выходное.

4. Разделите исходное множество данных на обучающее и тестовое. При этом используйте параметры, предлагаемые «по умолчанию»: размер обучающего множества 95%, тестового — 5%. Это означает, что из всего набора данных 5% случайным образом выбранных значений используется для контроля работоспособности построенной нейронной сети, а остальные 95% данных используются для обучения нейросети подбором синоптических весов межнейронных связей с помощью специальных алгоритмов. Укажите способ разбиения исходного множества данных «Случайно».

5. Далее программа предложит определиться со структурой нейронной сети. Укажите 1 скрытый слой в 24 нейрона. Тип активационной функции, алгоритм обучения и другие параметры алгоритма примите по умолчанию.

6. При настройке параметров остановки обучения нейронной сети в качестве критерия остановки выберите ошибку менее 0,01 или достижение 25 000 эпох обучения.

7. Запустите процесс обучения нейронной сети, нажав кнопку «Пуск». Во время процесса обучения нейросети происходит автоматический подбор весовых коэффициентов связей между ее отдельными нейронами согласно алгоритму обратного распространения ошибки. Примерный результат представлен на рисунке 29. Заметим, что результаты могут отличаться, поскольку начальные веса межнейронных свя-

зей устанавливаются случайно. Если процесс обучения сети не сходится (ошибки в ходе обучения не уменьшаются, а возрастают), следует повторить его, для чего установить флажок в поле «Рестарт» и вновь нажать «Пуск».

8. По окончании обучения следует нажать кнопку «Далее» и в открывшемся окне выбрать визуализаторы Data Minig («Граф нейросети», «Диаграмма рассеяния», «Что — если», «Обучающий набор»). Если выбрать все указанные визуализаторы, то получим окно с четырьмя соответствующими вкладками. Наиболее интересны граф нейросети и инструмент «Что — если».

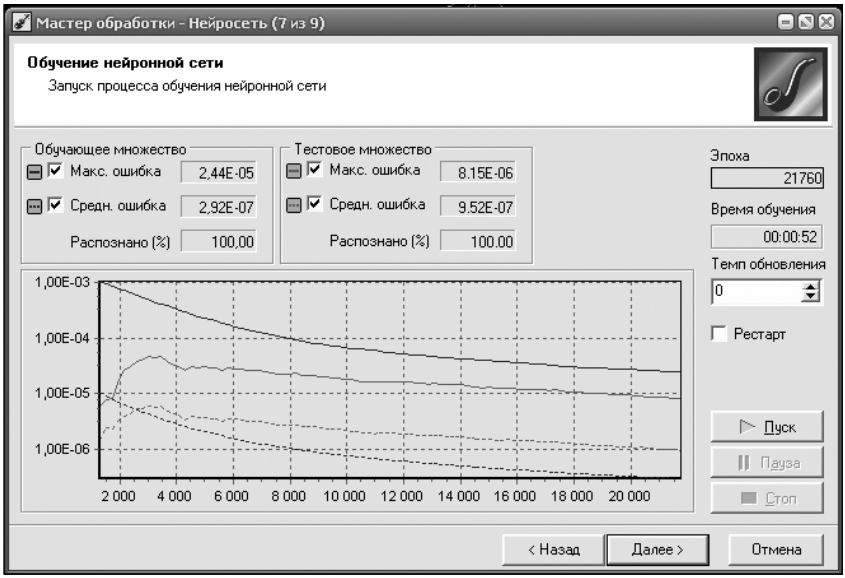

Рис. 29. Процесс обучения нейронной сети (Deductor)

Граф нейросети визуально отображает ее структуру. При этом сила синаптических связей кодируется цветом: самые сильные связи имеют красный цвет, самые слабые — синий (цветовой код весов связей приведен на специальной шкале, размещенной под графом сети) (рис. 30).

Визуализатор «Что—если» представляет собой инструмент для проведения вычислений. Вводя в соответствующие ячейки значения входных факторов (в нашем случае — операндов A и B), можно рассчитать результат (сумму или их произведение) (рис. 31).

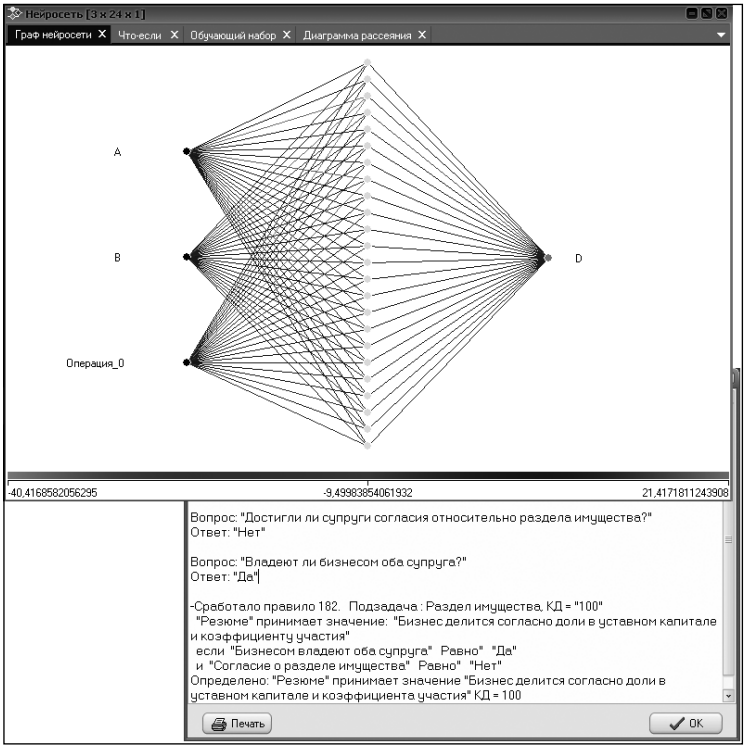

Рис. 30. Структура нейросети, реализующей арифметические операции

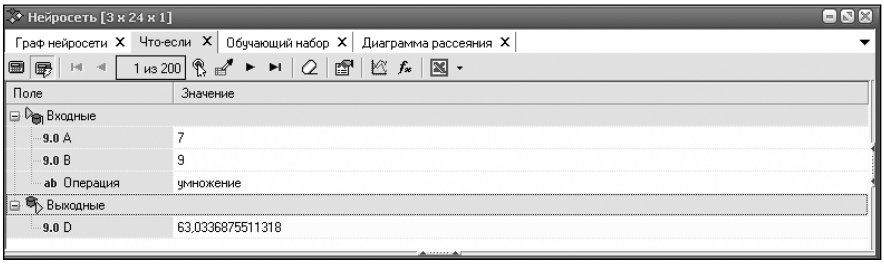

Рис. 31. Проведение вычислений с помощью нейросети (Deductor)

Экспериментируя с этим режимом, можно обнаружить, что сеть достаточно точно вычисляет результаты не только для операндов, представленных в обучающей выборке, но и тех, которые в ней не представлены, например, для дробных чисел. Таким образом, *нейронная сеть обладает способностью к обобщению.*

Другие вкладки полезны для оценки точности работы полученного линейного классификатора, т. е. расхождения прогнозируемых и эталонных данных.

9. Полученный сценарий можно сохранить в файле (меню «Файл/Сохранить как/Сохранить»). Файл проекта в программе *Deductor* имеет расширение .DED. На этом исследование нашей нейросети можно считать законченным.

Математически доказано, что нейронные сети могут аппроксимировать любую функцию. Сложность состоит в выборе набора значимых параметров: часто на практике бывает трудно установить, какие из параметров предметной области являются значимыми, а какие нет. Для достижения успеха в вектор входных сигналов сначала можно включить как можно больше параметров, исключая лишь очевидно незначимые, а в дальнейшем отбросить те входные нейроны, синаптические веса которых значительно меньше, чем у других, либо входные сигналы которых слабо влияют на работу сети. Обучающие выборки можно подготовить, используя в качестве источника необработанных данных, например, материалы сайта kad.arbitr.ru — картотеку арбитражных дел. Результаты работы соответствующей нейронной сети были бы интересны и полезны для прогноза и аналитики, однако для учебных целей такая работа является достаточно трудоемкой.

Гораздо проще подготовить исходные данные, если распознаваемая сетью ситуация задана логической нормой. В этом случае входные и выходные векторы однозначно задаются наборами из множества {1,0}. Следует отметить, что использование нейронных сетей для логических исчислений является не самым эффективным примером их использования: они проявляют свои преимущества как раз в тех случаях, когда формализация затруднена. Однако в учебных целях такой подход оправдан. К тому же набор характеристик, сформированный на основе логической нормы, может быть впоследствии расширен и дополнен, например, для детализации понятий тяжких последствий, существенного нарушения интересов или степени превышения полномочий. С учетом этих замечаний рассмотрим следующий пример.

Пример 2. *Распознавание правовой ситуации*. Эта задача, как и предыдущая, с успехом может быть решена с помощью той же программы *Deductor*. Однако для разнообразия мы будем использовать простую и удобную программу «Нейросимулятор», разработанную в Пермском государственном педагогическом университете и специально ориентированную на учебный процесс вузов.

Для описания состава преступления по ст. 286 УК РФ (превышение должностных полномочий) сформируем вектор из следующих входных параметров:

X1 Субъект вменяем.

X2 Субъект — должностное лицо.

X3 Субъект занимает государственную должность, должность субъекта РФ или должность главы местного самоуправления.

X4 Совершение действий, выходящих за пределы полномочий.

X5 Действие повлекло существенное нарушение прав и интересов граждан.

X6 Действие повлекло тяжкие последствия.

X7 Применение насилия, угрозы насилием, оружия или специальных средств.

X8 Наличие прямого или косвенного умысла.

Далее сформулируем вектор выходных сигналов:

Y1 Ст. 286 — 1.

Y2 Ст. 286 — 2.

Y3 Ст. 286 — 3.

Выборку для обучения этой нейронной сети получим из таблиц истинности, построенных для выражений, отражающих логику статьи:

Y1=X1∧X2∧¬X3∧X4∧X5∧¬X6∧¬X7∧X8.

Y2=X1∧X3∧X4∧X5∧¬X6∧¬X7∧X8.

Y3=X1∧(X2∧X3)∧X4∧X5∧(X6∧X7)∧X8.

Таблицы истинности генерируем с помощью *Excel* (в данном случае — всего 256 строк; на рисунке 32 приведены лишь строки с ненулевыми значениями на выходе) и будем использовать в качестве обучающей выборки нейронной сети.

Для проведения нашего исследования выполним следующие действия.

Запускаем «Нейросимулятор» (файл Nsim5sc.exe).

На вкладке «Проектирование сети» фиксируем структуру сети, задавая число нейронов в слоях: входном (8), скрытом (3) и выходном (3). Здесь же выбираем функцию активации — тангенс гиперболический.

На вкладке «Обучение/Данные обучения» с помощью специальной кнопки загружаем из файла предварительно подготовленные в *Excel* данные (рис. 32).

На этой же вкладке выбираем алгоритм обучения (например, алгоритм упругого распространения). В качестве условия остановки алгоритма укажите 500 итераций. Остальные параметры оставьте по умолчанию.

A	B	C	D	E	F	G	H	I	J	K
X1	X2	X3	X4	X5	X6	X7	X8	Y1	Y2	Y3
1	0	1	1	1	0	0	1	0	1	0
1	0	1	1	1	0	1	1	0	0	1
1	0	1	1	1	1	0	1	0	0	1
1	0	1	1	1	1	1	1	0	0	1
1	1	0	1	1	0	0	1	1	0	0
1	1	0	1	1	0	1	1	0	0	1
1	1	0	1	1	1	0	1	0	0	1
1	1	0	1	1	1	1	1	0	0	1
1	1	1	1	1	0	0	1	0	1	0
1	1	1	1	1	0	1	1	0	0	1
1	1	1	1	1	1	0	1	0	0	1
1	1	1	1	1	1	1	1	0	0	1

Рис. 32. Фрагмент обучающей выборки (положительные примеры)

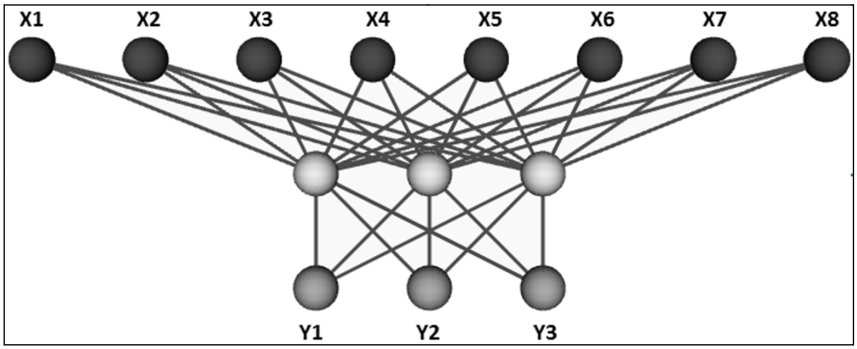

Рис. 33. Структура проектируемой сети

Запускаем процесс обучения нажатием кнопки  . Результатом работы будет отчет, представленный на рисунке 34.

Проведенное таким образом обучение сети позволило в одном из экспериментов уже после 80 итераций получить практически нулевое значение среднеквадратичной ошибки обучения.

Выбор количества и мощности скрытых слоев, а также выбор алгоритма обучения чаще всего осуществляются методом проб и ошибок. Так, в нашем случае обучение сети с двумя скрытыми нейронами потребовало для достижения приемлемого результата более 1000 итераций. А с помощью обучения на основе алгоритма обратного распространения тех же результатов получить не удалось, погрешность была суще-

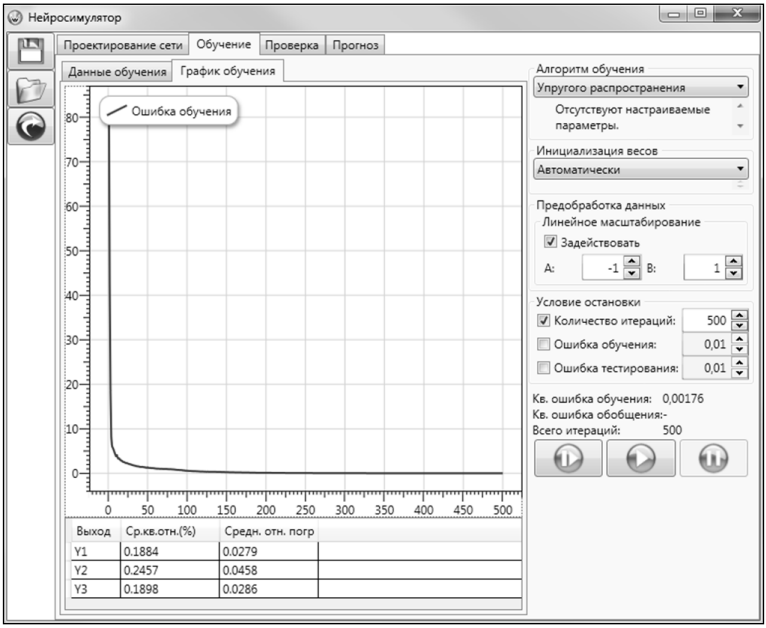

Рис. 34. Процесс обучения нейронной сети (Нейросимулятор)

ственно выше даже после нескольких тысяч эпох обучения. Читатель может самостоятельно провести эксперимент с другими конфигурациями нейронной сети.

Значительного упрощения сети можно достичь, если использовать аналоговые входы. Например, входы X2 и X3 можно объединить, подавая на вход X2 значения 0 — если субъект не является должностным лицом, 1 — если является, 2 — если его должность соответствует X3. На выходе тоже можно использовать 1 нейрон, ожидая от него, например, значения Y=0 в случае, когда деяние не подпадает под действие рассматриваемой статьи, Y=1 в случае квалификации по ст. 286 — 1, Y=2 для ст. 286 — 2 и Y=3 для ст. 286 — 3. Реализовать такую сеть (семь входных нейронов, один выходной и два на скрытом слое) с аналоговыми входами и выходами предлагается самостоятельно.

§ 4. Учебные задания

1. Подберите два различных набора параметров нейрона (весовые коэффициенты W_1, W_2 и порог Θ) для моделирования логического элемента И. Задание выполнить в рабочей тетради.

2. Подберите два различных набора параметров нейрона (весовые коэффициенты W_1, W_2 и порог Θ) для моделирования нейроном логического элемента ИЛИ. Задание выполнить в рабочей тетради.

3. Используя программу «Нейросимулятор», создайте нейронную сеть с двумя входами, одним выходом и одним скрытым слоем из трех нейронов. Обучающее множество — таблица умножения (например, на 2 и на 3), в которой намеренно пропущены строки (например, 2×5 и 3×7). Алгоритм обучения — обратное распространение ошибки, скорость обучения 0.08. Изменяя количество эпох обучения (300, 600 или более) и повторяя при необходимости процесс обучения заново, добейтесь значения максимальной ошибки не более 1%. По окончании обучения сети проверьте ее «способность» к обобщению и предсказанию, для чего на вкладке «Проверка» введите в качестве примеров строки, ранее исключенные из обучающего множества (т. е. 2×5 и 3×7). Какова ошибка предсказания? Каким способом можно ее уменьшить?

4. Повторите задание 5, увеличив число нейронов скрытого слоя до 7. Как изменилась ошибка предсказания?

5. Выполните распознавание правовой ситуации по ст. 286 УК РФ, как описано в примере 2. Ответьте на следующие вопросы:

5.1. Как квалифицируются действия должностного лица, умышленно совершившего действия, выходящие за пределы полномочий и повлекшие существенное нарушение прав и интересов граждан?

5.2. Как изменится квалификация умышленных и выходящих за пределы полномочий действий главы местного самоуправления, повлекших существенное нарушение прав и интересов граждан, если помимо применения насилия эти действия повлекли еще и тяжкие последствия?

5.3. Что изменится в квалификации (см. задание 5 b), если наличие умысла не доказано?

5.4. Как квалифицируются сопровождающиеся насилием и угрозой применения оружия и выходящие за пределы полномочий действия главы местного самоуправления, повлекшие тяжкие последствия, если нарушения прав и интересов граждан признаны несущественными? Какой вывод о формировании факторных признаков следует из этого примера?

6. Для оценки кредитоспособности банком выделено 14 характеристик, которые приведены в таблицах 2 и 3. Оцените финансовое положение и личные качества заемщиков для принятия решения о выдаче кредита (см. ниже таблицу 3 «Тестовая выборка»). Оценку проведите с использованием персептрона, первый слой которого содержит 14 нейронов (по числу выделенных характеристик), второй — 6, третий — 3, 1 нейрон

на выходе. Таблицу 2 с приведенной ниже обучающей выборкой подготовьте в *Excel* и импортируйте в «Нейросимулятор».

Расшифровка характеристик: пол (0 — мужской, 1 — женский); возраст; образование (0 — неполное среднее, 1 — среднее/среднее специальное, 2 — неполное высшее, 3 — высшее, 4 — ученая степень); семейное положение (0 — холост/не замужем, 1 — женат/замужем, 2 — вдовец/вдова, 3 — разведен(а), 4 — гражданский брак); количество иждивенцев на обеспечении заемщика; жилищные условия (0 — муниципальная квартира, 1 — кооперативная квартира, 2 — служебная квартира, 3 — собственный дом/квартира, 4 — съемная квартира, 5 — проживаю у родителей); наличие автомобиля в собственности (0 — нет, 1 — да); адрес постоянной регистрации (0 — город Омск, 1 — Омская область, 2 — другой субъект Российской Федерации); вид занятости заемщика (0 — рабочий, 1 — государственный служащий, 2 — коммерческий сотрудник, 3 — предприниматель, 4 — пенсионер); стаж — стаж работы на ныне занимаемой должности в годах; доход — среднемесячный доход в тыс. руб.; сумма кредита на товар в тыс. руб.; дата рассмотрения заявки для получения кредита; количество платежей погашения кредита. На выходе один параметр: решение (0 — отказать и 1 — открыть счет для получения кредита).

7. Решите задачу 6, используя нейронную сеть платформы Deductor.

Таблица 2. **Обучающая выборка**

Ф.И.О.	Пол	Возраст	Образование	Сем. положение	Иждивенцы	Жилье	Автомобиль	Адрес	Занятость	Стаж	Доход	Кредит	Дата	Платежи	Решение
Пономарев С. В.	0	21	0	0	0	4	0	1	0	1	5,5	10	25	6	0
Мартынова Д. Н.	1	34	3	3	1	2	0	0	1	10	6	9,2	24	12	1
Батуева А. А.	1	22	2	0	1	4	0	1	0	1,3	6,3	10,2	25	12	0
Фирсова Е. Н.	1	44	1	1	2	3	0	1	2	10,2	14	12,4	26	6	1
Копытев И. О.	0	36	1	1	3	3	0	1	0	5	9,1	12,4	3	4	1
Боталов Е.П.	0	28	2	0	0	3	1	0	1	5,4	13	13,5	27	6	1
Щербаков А. Д.	0	26	2	0	0	5	1	1	1	4,8	9,5	11,2	27	6	1
Калинина Л. И.	1	56	1	1	0	3	0	0	4	0	4,3	7,6	28	11	1
Копытов Г.К.	0	27	1	4	0	4	0	2	2	1,5	10	15,1	29	12	0
Петрова А. Г.	1	51	4	3	0	3	1	0	4	0	6,3	8,5	1	6	1
Архипов Д. П.	0	24	3	0	0	2	0	1	2	1,2	10	11,6	29	11	1
Костарева С. П.	1	30	2	1	1	2	0	1	1	3	7	7,8	30	12	0
Петухов О. В.	0	28	1	1	1	5	0	0	2	4,3	11	7,9	1	4	1
Захаров В. С.	0	54	3	2	0	3	1	1	1	28	8,6	9	2	12	1
Кирьянов А. А.	0	34	1	1	3	0	0	0	0	4,6	8,9	12,3	3	4	0

Таблица 3. **Тестовая выборка**

Ф.И.О.	Пол	Возраст	Образование	Сем.положение	Иждивенцы	Жилье	Автомобиль	Адрес	Занятость	Стаж	Доход	Кредит	Дата	Платежи	Решение
Пыстогов В. В.	0	23	1	0	0	3	0	1	0	0,5	8	5,31	8	4	
Огородников В. В.	0	39	1	1	0	3	0	0	2	0,3	12	8,29	1	10	
Кондратьева С. В.	1	31	1	1	2	1	0	1	0	1	8	5,09	31	10	
Коростина Н. А.	1	22	3	0	0	5	0	0	0	0,5	8,5	7,11	6	4	
Кожурина Н. Д.	1	28	2	0	0	3	0	0	2	1,4	10	7,09	8	4	
Локтев О. Ю.	0	27	3	0	0	5	1	0	1	8,6	12	4,11	1	4	
Гарипов Р. В.	0	42	1	1	2	3	1	0	1	7,2	12	8,02	21	6	
Гильманова Ю. М.	1	19	1	0	0	5	0	1	0	0,4	5	10,51	1	6	
Вирко И. Г.	1	24	0	0	0	5	0	0	0	0,1	8	6,88	25	12	
Быков Е. В.	0	23	1	0	0	0	1	0	0	0,6	15	7,75	6	4	
Боголюбов А. Н.	0	33	1	1	1	3	0	0	2	6,2	20	8,57	6	7	
Бабиков А. В.	0	24	1	1	1	5	0	0	2	2,3	18	13,71	25	4	

ГЛАВА IV. ВИЗУАЛИЗАЦИЯ ДАННЫХ

Нейронная сеть человека прежде всего ориентирована на распознавание образов. Человек мыслит образами, и поэтому визуальное представление информации для него наиболее удобно и естественно.

Визуализация — это инструментарий, который позволяет увидеть конечный результат вычислений, организовать управление вычислительным процессом и даже вернуться назад к исходным данным, чтобы определить наиболее рациональное направление дальнейшего движения. В результате использования визуализации создается графический образ данных. К способам визуального или графического представления данных относят графики, диаграммы, таблицы, отчеты, списки, структурные схемы, карты и т. д. Визуализация традиционно рассматривалась как вспомогательное средство при анализе данных, однако сейчас все больше исследований говорят о ее самостоятельной роли.

В этом разделе мы не будем рассматривать такие привычные нам способы визуализации количественных зависимостей, как графики, диаграммы и таблицы, а сосредоточим внимание на сущностях и связях — как концептуальных, так и социальных — между сущностями.

§ 1. Графическое описание процедур и отношений

Существуют различные способы графического представления алгоритмов и процедур. Среди последних упомянем, в частности, стандарт описания бизнес-процессов «IDEF3» и отечественную разработку — язык «ДРАКОН» (Дружелюбный Русский Алгоритмический язык, Который Обеспечивает Наглядность)[1]. Интересно, что обе эти методологии описания проектов первоначально были разработаны для нужд аэрокосмической отрасли: «IDEF» — для ВВС США, «ДРАКОН» — для советского проекта «Буран». Масштабные проекты выявили проблему взаимопо-

[1] *Паронджанов В.* Как улучшить работу ума. Алгоритмы без программистов — это очень просто! М., 2001. 360 с.

нимания между заказчиками, аналитиками, исполнителями и другими участниками, и цель обеих разработок одна и та же: сделать более простыми и наглядными описания сложных задач, облегчив тем самым их понимание и поиск пути к решению. Так, «ДРАКОН», по определению его разработчика, — это «общедоступный визуальный язык, предназначенный для описания структуры деятельности, для систематизации, структуризации, наглядного представления и формализации императивных знаний, а также для проектирования, программирования, моделирования и обучения. Это универсальный межотраслевой язык делового мира, служащий для описания научно-технических, медицинских, биологических, экономических, социальных, учебных и иных задач. «ДРАКОН» позволяет упорядочить и представить решение любой, сколь угодно сложной императивной (процедурной, деятельностной, технологической, рецептурной, алгоритмической) проблемы в виде наглядных чертежей, выполненных по принципу "взглянул — и сразу понял!"». Таким образом, «ДРАКОН» вполне применим для описания инструкций, юридических процедур и регламентов.

Для специалистов, занимающихся следственной деятельностью, более естественной моделью представления данных является граф (схема). Значительная часть аналитической работы в этом случае заключается в выявлении связей между объектами. Соответственно, основными понятиями модели являются «объект» и «связь». При этом традиционные средства представления информации в виде экранных форм и таблиц являются малопригодными. На первый план выходят *визуальные средства анализа* и такие графические представления данных, как диаграммы связей, последовательности событий и транзакций.

Модель данных «объект — связь» предоставляет специалисту следующие аналитические режимы обработки информации:

— контекстный анализ фактов;

— поиск цепочек связей;

— поиск похожих происшествий;

— ситуативный анализ.

На практике эти режимы обработки информации применяются при анализе ситуаций, связанных со страховыми случаями, правонарушениями и т. п. Например, аналитик страховой компании интересуется поиском группы водителей, одновременно вовлеченных в серию дорожно-транспортных происшествий (далее — ДТП), но в разных ролях, где группа лиц в различных ДТП поочередно выступает то в роли потерпевшего, то в роли виновного, то в роли свидетеля.

Контекстный анализ объектов — это поиск в массиве фактографической информации всех связей указанного объекта, а также связей, относящихся к нему объектов, с возможностью получения документов, содержащих описание обнаруженных объектов. Этот режим позволяет аналитику выявить ключевые объекты анализа, скрытые и косвенные связи выбранного объекта или группы объектов.

Поиск цепочек связей позволяет аналитику обнаруживать прямые и опосредованные связи заданной глубины между объектами и группами объектов. Данный режим дает возможность автоматически выполнять проверку предположений о том, что объекты имеют связь.

Поиск похожих происшествий выполняется в соответствии с заранее подготовленными и введенными в систему информационными шаблонами. Данный режим позволяет установить автоматический отбор из вновь поступающей информации всех происшествий, соответствующих этим шаблонам.

Ситуативный анализ объектов — это поиск в массиве фактографической информации зависимостей между объектами и группами объектов. Ситуативный анализ позволяет выявлять в массиве фактографической информации неявные закономерности, получая, таким образом, качественно новые знания.

§ 2. Семантические сети и онтологии

Термин «семантическая сеть» обозначает семейство представлений, основанных на графических схемах с узлами, соединенными дугами. Узлы графа соответствуют фактам или понятиям, а дуги — отношениям или ассоциациям между понятиями. Семантические сети были разработаны для анализа естественных языков и построения психологических моделей человеческой памяти. Было доказано, что люди не только ассоциируют свои понятия, но и иерархически организуют их. В результате была построена модель хранения и обработки информации человеком, использующая семантическую сеть.

Узлы в семантической сети соответствуют объектам, концепциям, понятиям, событиям. Дуги, указывающие отношения между узлами, могут определяться по-разному в зависимости от метода представления знаний. Поскольку смысловым центром предложения являются глаголы, то часто сеть строят именно от глаголов, в этом случае основными отношениями являются отношения *агент* (тот, кто делает), *реципиент* (тот, над кем производится действие), *объект* (то, над чем производится действие). Именно такой подход к представлению знаний, когда смысловые узлы соответ-

ствуют действиям, а также объектам или их признакам, а отношения — их семантическим ролям, представляется наиболее удобным для описания ситуаций в сфере права. Поскольку юридической оценке подвергаются действия (деяния) субъекта, то именно они и должны являться основными семантическими узлами сети, моделирующей правоотношения. Таким образом, моделируемые ситуации и сценарии можно рассматривать как совокупность событий, имеющих определенные атрибуты (время, место, причина и др.) и связанных между собой и со своими атрибутами одноименными причинно-следственными или иными отношениями. Графически такие сценарии отображаются ориентированными графами «с центром в глаголе», в которых отношениям соответствуют дуги, соединяющие аргументы — вершины графа. Атрибуты событий, в свою очередь, также могут характеризоваться определенными признаками. Так, субъект действия (*агент*) характеризуется личными данными, в том числе о поле, возрасте, состоянии здоровья, наличии судимости и т. п.[2]

В компьютерной лингвистике более или менее устоялся набор общеязыковых семантических отношений, разработанный для целей автоматического перевода текста. В таблице 4 в целях иллюстрации их применения эти отношения представлены вместе со своими (возможными) аргументами в виде двухместных предикатов.

Таблица 4. **Примеры семантических отношений**

АДРЕС (соучастник, передать) КОНЕЧНЫЙ_ПУНКТ (Москва, уехать) СПОСОБ (взлом, проникнуть) ИНСТРУМЕНТ (ключ, открывать) СРЕДСТВО (разоблачение, угрожать) ПОСРЕД. (показания, подтвердить) ИДЕНТ. (№ 20, дом) ИМЯ (Иванов, гражданин) СУБЪЕКТ (Петров, собственность) ПАРАМЕТР (возраст, потерпевший) СТЕПЕНЬ (частично, потерять трудоспособность)	АГЕНТ (посредник, продать) АВТОР (Правительство, постановление) ОБЪЕКТ (здоровье, ущерб) РЕЦИПИЕНТ (преступник, арест) СВОЙСТВО (тайно, изъятие) ОЦЕНКА (существенный, ущерб) СОДЕРЖ. (событие, рассказать) ЦЕЛЬ (самооборона, ударить) НАЗНАЧЕНИЕ (проживание, помещение) МАСШТАБ (Россия, банк) РЕЗУЛЬТАТ (сбыт, организовать) МОДАЛЬНОСТЬ (должен, предвидеть)	МАТЕРИАЛ (металл, дверь) ОГРАНИЧЕНИЕ (возраст, выделение) ПЕРИОД (6 месяцев, кредит) ПРИНАДЛ. (квартира, Петров) ПРИЧИНА (алкоголь, опьянение) ВРЕМЯ (вчера, произойти) ЗНАЧЕНИЕ (5000 рублей, ущерб) ИСХ_ПУНКТ (квартира, изъять) КОНТРАГЕНТ (посредник, купить) КОЛИЧЕСТВО (два, судимость) МЕСТО (Россия, проживать) ЧАСТЬ (лезвие, бритва)

[2] *Гайдамакин А. А.* О формальном описании семантических связей в статьях Уголовного кодекса. С. 99–104.

При таком подходе, например, фразу «Андрей сообщил начальнику, что он передал книгу в издательство» можно представить сетью, изображенной на рисунке 35. Здесь пунктиром выделена семантическая подсеть «Андрей передал книгу в издательство». Сети, вершины которых обладают внутренней структурой, называют иерархическими, их особенность заключается в том, что они позволяют устанавливать отношения не только между простыми вершинами, но и между подсетями. Понятие подсети аналогично понятию скобок в математической записи.

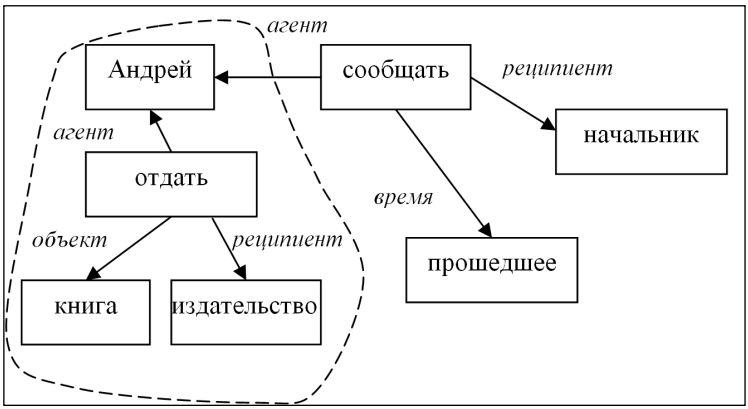

Рис. 35. Семантическая сеть с подсетью

К сожалению, современный уровень развития компьютерной лингвистики пока не позволяет полностью автоматизировать процесс синтеза таких моделей в процессе перевода текста с естественного языка. Для иллюстрации сказанного приведем результат автоматического анализа и визуализации семантических связей в известном стихотворении: «Это старушка, седая и строгая, которая кормит корову безрогую, лягнувшую старого пса без хвоста, который за шиворот треплет кота, который пугает и ловит синицу, которая часто ворует пшеницу, которая в темном чулане хранится в доме, который построил Джек» (рис. 36).

Для построения этой сети мы использовали демонстрационный стенд *EP.TestDesk* фирмы «Pullenti». Как видим, программа безошибочно выделила из текста сущности (объекты). Что касается семантических связей между объектами и их атрибутами, то для того, чтобы система верно их распознала, пришлось привести исходный текст к достаточно странному виду: «Седая старушка кормит безрогую корову, которая лягнула старого пса без хвоста, который треплет кота за шиворот, а кот пугает

синицу и ловит синицу, которая часто ворует пшеницу, а пшеница хранится в темном чулане в доме, Джек построил дом». В противном случае налицо недоразумения (без хвоста оказывается корова; кто ловит синицу и где находится чулан — неизвестно; Джек строит кота; старушка что-то строгает и т. п.).

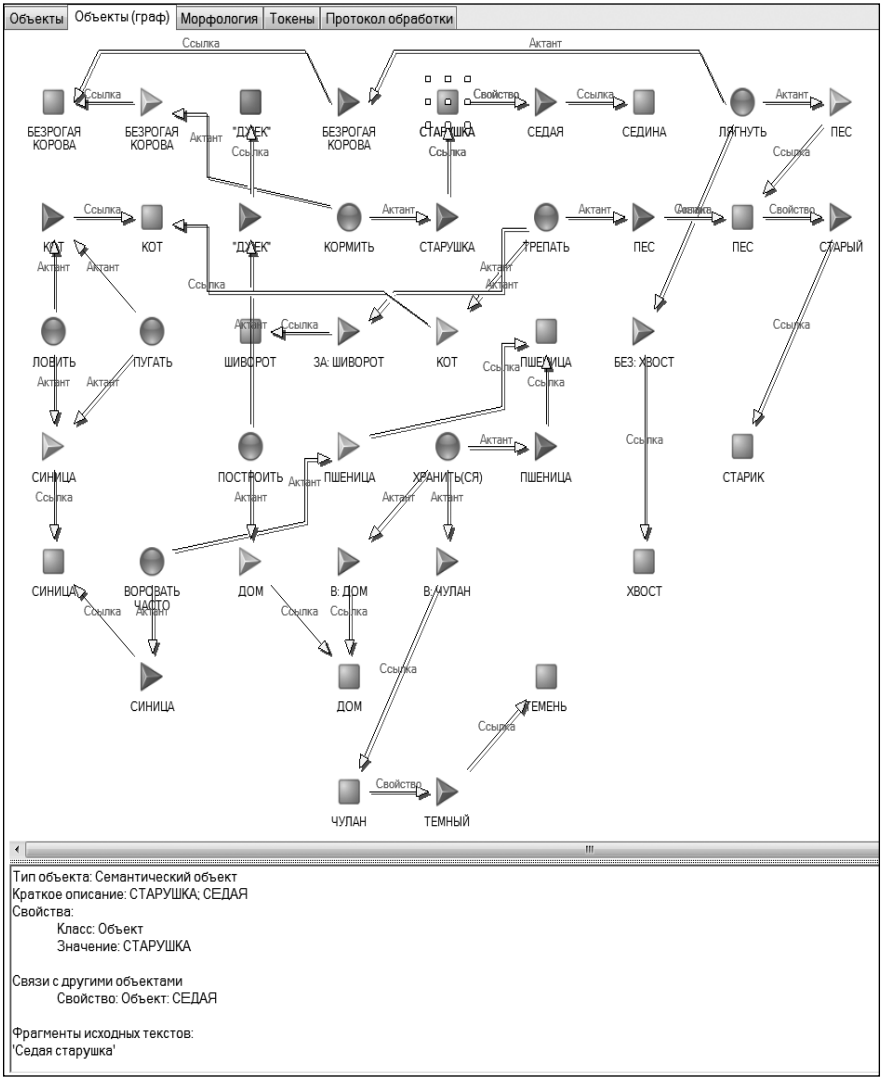

Рис. 36. Результат семантического анализа (*EP.TestDesk* фирмы «Pullenti»)

Ошибки, которые допускают современные системы при анализе текста, связаны прежде всего с контекстным характером нашей речи вообще и понятий в частности. Каждый термин обретает свой смысл только в некотором окружении других терминов, понятия являются взаимноопределяющимися. И это понятно — ведь для описания бесконечного разнообразия мира мы пользуемся ограниченным набором терминов, что с необходимостью влечет их многозначность. Поэтому для адекватного понимания текста системе нужен не только сам текст, но и достаточные знания об окружающем мире, т. е. о том, что в тексте никак явно не выражено. Обычно выделяют три взаимодействующих и взаимопроникающих слоя знаний: знание предметной области, знание сценариев и знание переднего плана. При этом первые два уровня относятся к фоновому знанию, т. е. описывающие их предикаты являются «ненаблюдаемыми» в формулировке конкретной решаемой задачи, а *знание переднего плана* представляет собой общую теорию, которая связывает наблюдаемые предикаты с фоновыми.

Фоновые знания представляют, по сути, концептуальную модель мира, включающую описания базовых понятий, организованных в родовидовые «деревья», и связи между ними. Важным элементом представления фоновых знаний в вычислительных системах являются формальные онтологии, которые включают машинно-интерпретируемые формулировки основных понятий предметной области и отношений между ними, в том числе понятия классов (понятий), свойств каждого понятия (его слотов, атрибутов, ролей), а также ограничения, наложенные на слоты. Вместе с набором индивидуальных экземпляров классов онтология образует базу знаний. Во многих дисциплинах сейчас разрабатываются стандартные онтологии, определяющие общий для данной предметной области словарь для совместного использования и аннотирования информации; они переходят из мира лабораторий по ИИ на рабочие столы экспертов. Стандартизация онтологий дает возможность:

— обеспечить общее понимание структуры информации людьми или программными агентами;

— повторно использовать знания;

— сделать явными принятые допущения;

— отделить фоновые знания от оперативных;

— производить анализ знаний.

Разработка онтологии включает в себя: определение классов; расположение классов в таксономическую иерархию (подкласс — надкласс);

определение слотов и описание допустимых для них значений, а также заполнение значений слотов экземпляров.

Основные усилия при разработке юридической онтологии затрачиваются на моделирование общих знаний о мире, поскольку право не имеет собственной онтологической основы. Собственно юридические (а не социальные, политические, управленческие и др.) вопросы связаны главным образом с осмыслением, обоснованием права (убеждений) и правовыми системами, другими словами, они имеют не онтологический, а эпистемологический характер. В связи с этим юридические онтологии на практике сочетают в себе как онтологические, так и эпистемологические сущности.

Одним из первых и основных языков для описания связей является RDF, выражения которого представляют собой триплеты вида Субъект – Предикат – Объект (Ресурс – Свойство – Значение). Каждый триплет представляет собой утверждение о связи между понятиями, обозначенными как узлы, которые эта связь соединяет. Таким образом, RDF-граф образует семантическую сеть. Язык основан на XML-синтаксисе и используется для представления информации в Word Wide Web. Сам по себе RDF не предоставляет механизмов для описания свойств и отношений между этими свойствами и другими ресурсами, однако такие механизмы предоставляются другими средствами, разработанными на основе RDF, в частности, языком Web-онтологий OWL, ориентированным на использование в проекте Semantic Web и являющимся общепризнанным стандартом для представления онтологий.

Для моделирования формальных онтологий широко используются фреймы и семантические сети. Механизм наследования, органически встроенный в эти способы представления знаний, позволяет эффективно описывать таксономии. Среди инструментальных средств, поддерживающих создание онтологий, наибольшей популярностью пользуется Protege-2000 (*http://protege.stanford.edu*). В литературе и сети Интернет существуют библиотеки готовых онтологий, но в большей части англоязычные.

Использование онтологий, а также информационно-поисковых тезаурусов позволило значительно повысить качество «понимания» текста компьютерными системами, но, как видно из примера с «домом, который построил Джек», до идеала еще далеко. Однако это не должно являться преградой для использования таких систем в юриспруденции. Да, мы не можем обеспечить однозначность «механического» перевода юридического текста с естественного языка на формальный, но обратная задача решается вполне успешно, по крайней мере, в отношении смысла. Имея готовую семантическую модель, можно автоматически синтезировать со-

ответствующие высказывания на естественном языке (предполагается, что модель и высказывание имеют общую лексическую основу). Наличие правильного и полного семантического описания в виде сети гарантирует адекватное восстановление смысла по этому описанию; можно говорить о содержательно-правовой эквивалентности. А поскольку текст законодательства представляет собой повторно используемое знание (как и онтологии), то такие семантические модели, составляя смысловой каркас правового акта, даже могли бы быть признаны первичными по отношению к тексту на естественном языке, оставляя последнему дополнительные, разъясняющие функции [3]. Проблемы перевода естественного языка на формальный, удобный для машинной обработки, решались бы при этом автоматически.

§ 3. Проект «Law Studio»

Проект «*Law Studio*» интересен тем, что использует для автоматизированного моделирования в сфере юриспруденции идеи, близкие к изложенным в предыдущем абзаце, в том смысле, что первичной в нем является именно модель отношений. Эта система призвана облегчить труд юриста, автоматически рассчитать распределение прав, оценить законность тех или иных действий, правовые последствия и юридические риски. Программа обеспечивает:

— построение моделей различных правовых ситуаций с учетом динамики развития во времени;

— анализ пользовательских моделей правовых ситуаций на предмет соответствия нормам законодательства;

— истребование у пользователя уточнения существенных параметров для принятия решения о законности или незаконности тех или иных действий;

— выдачу рекомендаций по каждой ситуации.

Модель правовых отношений в графическом виде описывает статические ситуации (распределение прав и обязанностей субъектов на определенный момент времени) и конструирует их развитие на основе сделок, транзакций и других событий. Этот подход позволяет стандартизировать правила графического отображения правовых ситуаций. В настоящее время таких стандартов нет и каждый юрист рисует ситуацию по-своему, хотя во многих других областях, требующих взаимопонимания для ко-

[3] *Гайдамакин А. А.* Прозрачность закона и информационно-коммуникационные технологии // Научный вестник Омской академии МВД России. 2011. № 2(41). С. 7.

ординации действий многих людей, используются специальные графические языки представления знаний (например, «IDEF3» или «ДРАКОН»). В системе «Law Studio» предлагается единый набор символов и правил их связывания для отображения сущностей правовой модели (правовая нотация). Правовая модель — удобный способ хранения информации о корпоративной структуре, включая систему привязки различных документов к элементам модели. В дальнейшем эти документы хранятся в составе проекта.

Главной особенностью платформы «*Law Studio*» является возможность на основе упомянутой модели автоматизировать расчет последствий сделок, транзакций и прочих событий. Пользователь системы (юрист) описывает ситуацию клиента в графическом интерфейсе «Law Studio» (субъекты и объекты права, взаимное владение, права и обязательства, сложившиеся на данный момент), а также создает сделки — генераторы событий, определяющие возникновение обязательств, и транзакции, направленные на исполнение обязательств. Система позволяет строить набор взаимосвязанных ситуаций для разных моментов времени, используя регламенты сделок и транзакции, а также прогнозировать качественные и количественные характеристики обязательств при создании новых ситуаций. Возможен анализ сложных ситуаций со множеством прав и сделок, развивающихся по различным сценариям во времени, при этом пользователь может выбирать наиболее удовлетворяющую его ветку развития сценария (лучшую альтернативу). Смоделировав сделку, можно сгенерировать документ по стандартному шаблону.

Пусть, например, некий несовершеннолетний John Doe планирует приобрести у Andrey Bogatov долю в ООО «Tantal» (владение на схеме представлено отношением Ownership rights), которая, в свою очередь, владеет долей в ООО «Xenon». Описание этой ситуации показано на рисунке 37. В нижней части окна представлено описание моделируемых объектов.

При просмотре ссылки на законодательство в отношении John Doe получим экран, представленный на рисунке 38. Как видим, система обращает внимание пользователя на необходимость получить согласие законного представителя на заключение сделки несовершеннолетним. В связи с этим пользователю следует ввести в модель сделки нового участника отношений — J. Doe Senior.

Другая выявленная системой проблема связана со ст. 7 Федерального закона от 8 февраля 1998 г. № 14-ФЗ «Об обществах с ограниченной ответственностью», согласно которой ООО (английская аббревиатура —

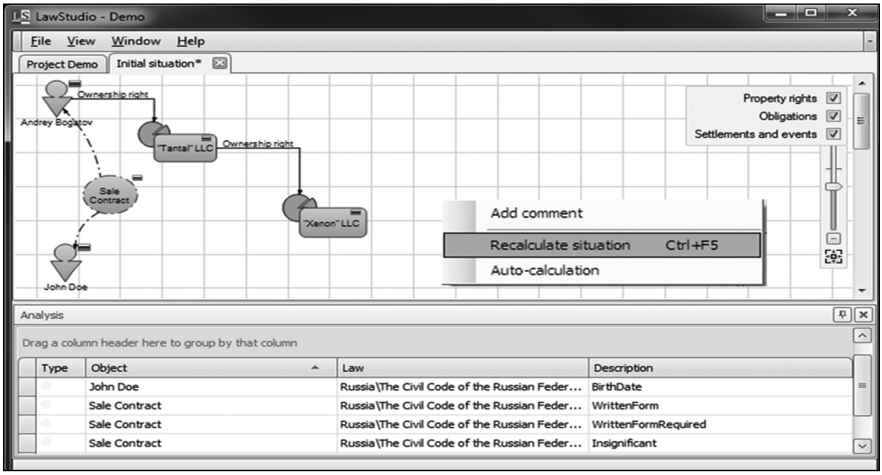

Рис. 37. Графическое представление правовой ситуации

LLC) не может иметь в качестве единственного участника другое хозяйственное общество, состоящее из одного лица (рис. 39).

Для того чтобы не возникало противоречий с законом, необходимо добавить к правовой модели второго собственника, что и сделано на рисунке 40. Здесь введено открытое акционерное общество (OJSC) «Circum» в качестве совладельца ООО (LLC) «Xenon».

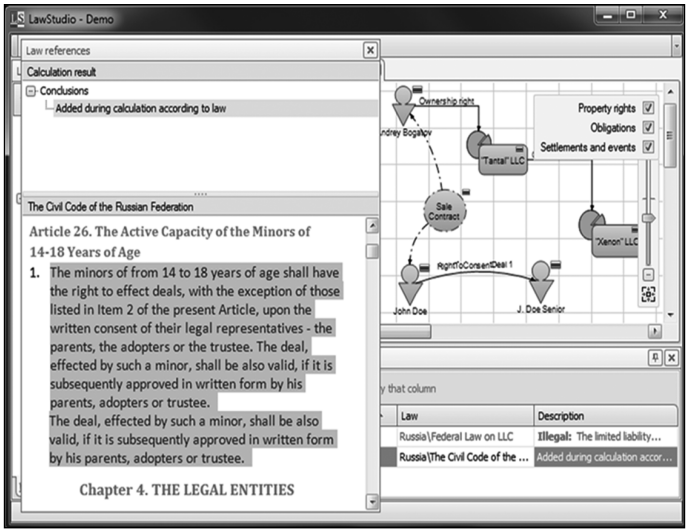

Рис. 38. Результаты анализа правовой ситуации по рис. 37 (замечания по ст. 26 Гражданского кодекса РФ)

Программа доступна в сети по адресу http://app.law-studio.ru:8080/installer. Ее возможности могут быть расширены при использовании дополнительных платных и бесплатных модулей.

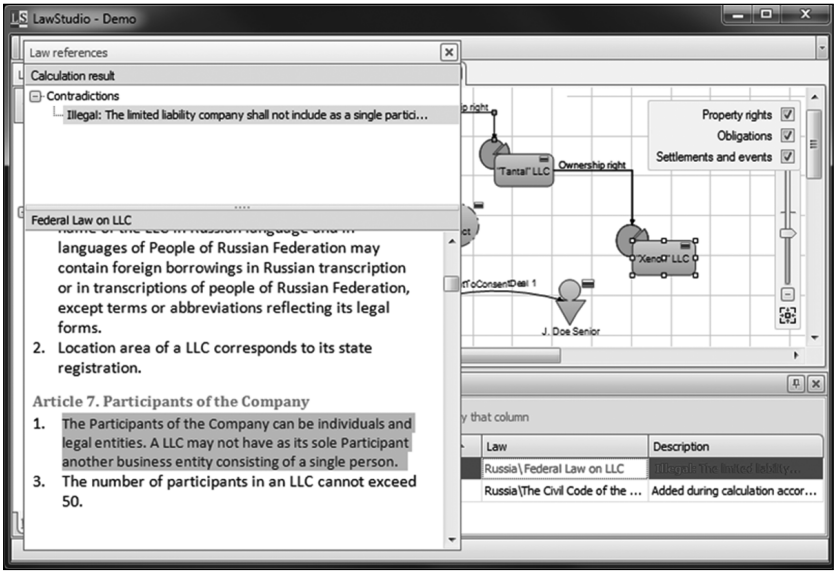

Рис. 39. Результаты анализа правовой ситуации по рис. 37 (замечания по ст. 7 Федерального закона «Об обществах с ограниченной ответственностью»)

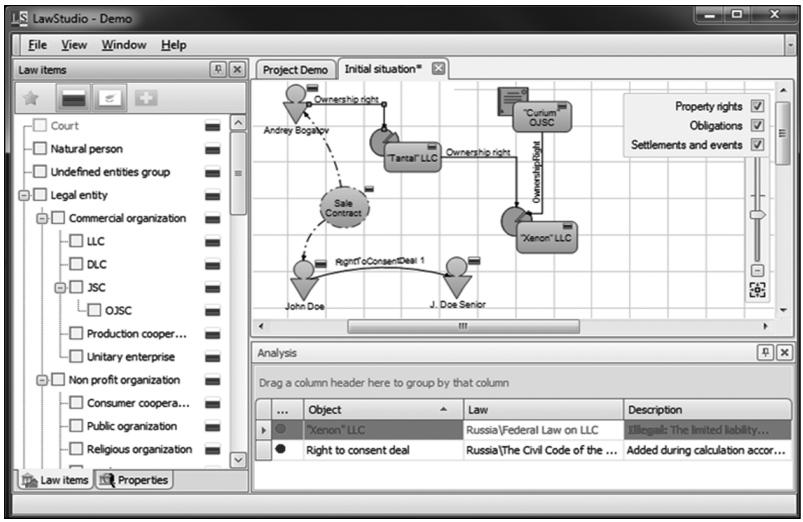

Рис. 40. Графическое представление скорректированной правовой ситуации

§ 4. Сети социальных взаимодействий

Сеть социальных взаимодействий, или социальная сеть — это организованное множество людей, состоящее из двух типов элементов: люди и связи между ними. В реальной жизни сети складываются естественным образом. В качестве действующих лиц (социальных акторов) могут выступать не только индивиды, но и социальные группы, организации, города и страны. Под связями понимаются не только коммуникационные взаимодействия между акторами, но и связи по обмену различными ресурсами, взаимодействия, связанные с совместной деятельностью, включая конфликтные отношения. Ключевым является описание характеристик, выражающих плотность, интенсивность и пространственную координацию социальных связей, что дает возможность выделять структурные единицы исследования в системе социальных отношений: «узлы», «блоки», «клики», «кусты».

Отдельным направлением исследования является визуализация (графическое отображение) социальной сети. Возможность увидеть сеть позволяет сделать важные выводы о характере взаимодействия акторов, даже не прибегая к другим методам анализа.

На рисунке 41 каждый узел обозначает человека, а каждая линия — связь между двумя людьми. Связи могут быть направленными (например, в схеме оповещения — если один человек звонит другому, то тот ему уже не звонит) или ненаправленными (если речь идет о дружеских отношениях). Количество связей у разных элементов сети различное: у кого-то больше, у кого-то меньше. Свойства связей (веса ребер) тоже могут быть различными: название связи, ее тип или значимость. Визуально значимость связи может отображаться цветом или толщиной линии ребра.

В программных средствах для визуализации сетей элементы с большим количеством связей обычно помещаются в центре, а с меньшим — ближе к краям схемы. Это помогает увидеть место каждого человека в системе: если у кого-то связей становится больше, то увеличивается и уровень его встроенности в социальную сеть.

Методы сетевого анализа позволяют исследовать такие свойства сети, как ее способность передавать информацию от одного участника к другому, способность сети сохранять свою целостность в случае удаления одного или более ее участников. В ходе сетевого анализа выявляется и положение участников внутри сети, что необходимо для понимания их роли в ней.

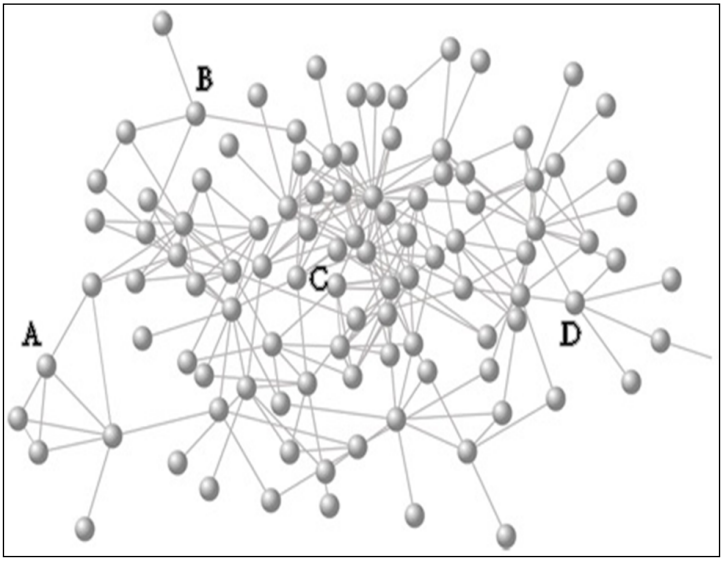

Рис. 41. Пример сети социальных взаимодействий (узлы – люди, дуги – связи)

Такой анализ полезен, например, при изучении управления внутренней жизнью организации. Он выявляет информационные связи между участниками сети и позволяет понять, откуда, куда и как идет информация, например, обнаружить участки, где потоки информации замедляются или вовсе останавливаются. С помощью сетевого анализа можно найти и устранить фрагментацию и перегрузку социальной сети организации, а также предотвратить ее изолированность. Сетевой анализ может также дать ответы на вопросы, позволяющие повысить гибкость сети: «Что будет, если какой-либо участник сети исчезнет из нее? Как избежать в таком случае ухудшения состояния сети и ее распада?». Такой анализ обычно производится на основе данных специально подготовленных *опросников* для участников сети, который может включать следующие вопросы: «Как часто вы обращаетесь за информацией к перечисленным людям?», «Согласны ли вы с тем, что перечисленные лица обладают знаниями и навыками в данной области?», «Как повлияло бы на эффективность вашей работы более частое общение с этими людьми?» и т. д. По результатам обработки данных опроса происходит *построение сети*. После этого разрабатываются *методы расчета индикаторов*, описывающие структурные характеристики сети. И наконец, производится *анализ результатов*.

По месту в структуре сети, помимо обычных участников, выделяют еще три типа акторов: «связные», «брокеры» и «пограничники».

«Связные» — это участники, у которых больше всего прямых связей. Это либо перегруженные работой лидеры, количество связей у которых настолько велико, что они уже не в состоянии обработать поток информации (в таком случае предлагается «разгрузить» социальную сеть и переложить часть обязанностей на других участников команды), либо так называемые «восходящие звезды» — участники, которые много работают и, соответственно, имеют много обязанностей перед другими людьми. Третья категория «связных» — политические игроки, которые специально замыкают на себя как можно больше прямых связей. Они действуют в собственных целях и чаще всего приносят вред, устранить который можно созданием связей между другими участниками сети или освобождением данного актора от его обязанностей.

«Брокеры» — это акторы, связывающие людей из различных подгрупп (офисов, отделов), которые между собой иначе никак не пересекаются. «Брокеры» пользуются авторитетом у коллег, поскольку связывают различные части сети между собой. Они быстрее других участников сети распространяют информацию в сети.

«Пограничники» — это акторы, находящиеся на окраинах сети. На эту роль чаще всего претендуют люди, которые не хотят или не могут устанавливать больше связей, чем у них есть. Из-за них работа идет менее продуктивно, поэтому при оптимизации структуры нужно либо устранять их из социальной сети, либо принимать меры по увеличению у них количества прямых связей. Другая группа — «новички», которые только начинают вливаться в коллектив. К «пограничникам» часто можно отнести и уникальных экспертов, которые не желают заводить новые связи и делиться своим знанием, поскольку им это не нужно и чаще всего невыгодно.

Программное обеспечение (далее — ПО) для визуализации сетей весьма разнообразно. Мы рассмотрим две программы, находящиеся в свободном доступе.

«***NetDraw***» является **бесплатным приложением**, которое можно использовать, если исходные данные хранятся в текстовом файле (подготовить его можно в Блокноте или экспортировать из Excel). **Для ввода** данных используется так называемый DL-протокол, который в себя включает несколько форматов: nodelist (список узлов), edgelist (список граней) и fullmatrix (матрица).

Формат *Nodelist*. Текстовый файл выглядит следующим образом:

```
dl
n=59
format = nodelist
data:
1 7 8 2
3 19 21 49 6
2 6
...
```

Здесь описан фрагмент некоторой сетевой структуры. В начале текста *dl* указывает на тип протокола; *n = 59* говорит программе, что ожидается 59 узлов (nodes); *format = nodelist* означает, что ожидается nodelist-формат; *data* показывает начало собственно данных в записи.

Первая строка *(1 7 8 2)* говорит о том, что персона *1* имеет связи с тремя людьми, которые обозначены как *7, 8* и *2*. Вторая строка *(3 19 21 49 6)* сообщает, что персона *3* имеет связи с четырьмя людьми, которые обозначены как *19, 21, 49* и *6*. Вместо цифровых обозначений можно, конечно, указать и буквенные, например, имена. К сожалению, программа «*NetDraw*» поддерживает только латинские символы, поэтому для русского текста придется проводить транслитерацию.

Формат *Edgelist* удобен, когда имеется информация о парах. Рассмотрим конкретный пример.

```
dl
n=204
format = edgelist
labels embedded
Alekseev Kurgan
Alkomyan Chita
Alushkin Altajsk.kr.
Bagraj Arhangel'sk
Balandin Irkutsk
Baranenko Moskva
...
```

Здесь n = 204 — это двести четыре узла, первая строка (*Alekseev Kurgan*) говорит о том, что *Alekseev* и *Kurgan* имеют связь, и т. д.

На рисунке 42 показана сеть, соответствующая данному примеру.

Здесь представлены связи между выпускниками Омской академии МВД России и региональными УМВД, направившими их на обучение. Центрами образовавшихся «звездочек» являются города или УМВД. Такая диаграмма позволяет, например, без всякого дополнительного анализа выявить наиболее многочисленные группы «земляков» (в данном

случае это курсанты из Омска, Иркутска, Тывы, Москвы, Туркмении, Хабаровска, Кемерово и др.)

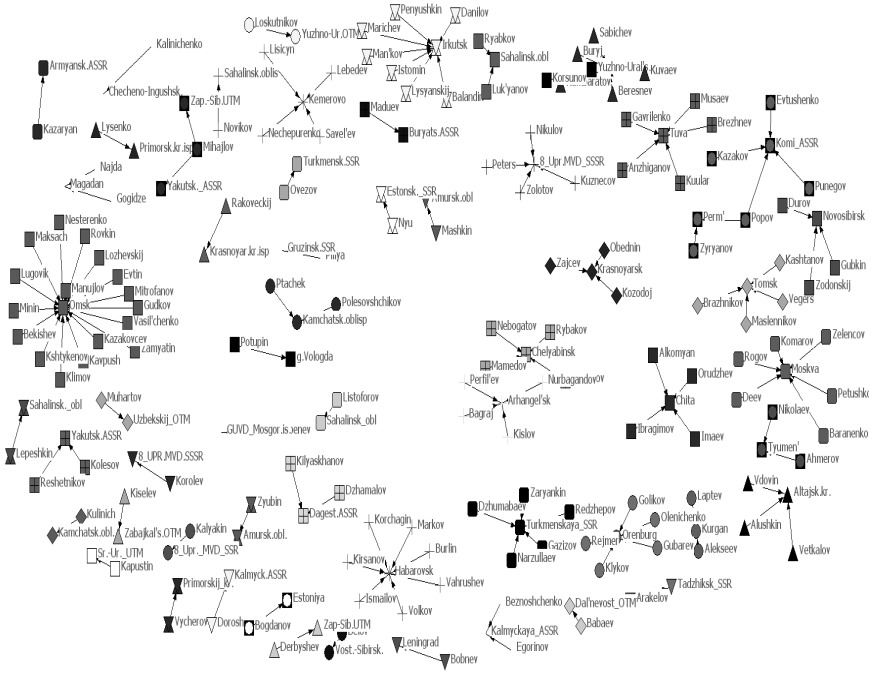

Рис. 42. Визуализация связей между выпускниками и комплектующими органами («NetDraw»)

Формат *FullMatrix*. Этот формат удобен для небольших наборов данных, которые содержат множество связей. Для подготовки данных можно использовать любой текстовый редактор или Excel (файл сохраняется как текстовый). Данные должны быть представлены следующим образом:

```
dl
n = 5
format = fullmatrix
data:
0 0 1 0 0
0 0 0 1 1
1 0 0 1 0
0 1 1 0 0
0 1 0 0 0
```

Строка *format = fullmatrix* указывает на тип формата. После указателя *data* следуют данные в виде таблицы типа «персона-персона». Поскольку у нас пять узлов (n = 5), матрица должна иметь пять строк и пять столбцов. Первая строка (00100) показывает, что персона *1* имеет связь с персоной *3*. Вторая строка говорит о том, что персона *2* имеет связи с персонами *4*, *5* и т. д.

Нецифровые метки могут быть вставлены при внесении указателя *labels embedded* способом, указанным ниже.

```
dl
n = 5
labels embedded
format = fullmatrix
data:
Bill Jan Jim Sue Zoe
Bill      0 0 1 0 0
Jan       0 0 0 1 1
Jim       1 0 0 1 0
Sue       0 1 1 0 0
Zoe       0 1 0 0 0
```

Программа «*NetDraw*» позволяет просматривать атрибуты узлов, переключаться между ними, визуализировать только определенные группы узлов, производить фильтрацию с учетом силы связей (визуально она отображается толщиной линий), изменять размеры и цвета узлов, линий, меток, фона и т. д.

Другим (и весьма мощным) бесплатным средством визуализации различных данных, который подойдет как новичку, так и опытному пользователю, является программа **«Gephi»**. Эта разработка имеет русскую локализацию. Рассмотрим построение графа на простом примере.

Пусть связи между лицами (вершинами, узлами графа) заданы матрицей, подобной *fullmatrix* в «*NetDraw*». В программе «Gephi» она называется матрицей смежности и, подготовленная в Excel, имеет для нашего примера следующий вид:

	A	B	C	D	E	F	G
1		Иванов	Петрова	Сидоров	Михайлов	Левочкин	Сергеев
2	Иванов		1				
3	Петрова	1		1		1	
4	Сидоров		1		1		1
5	Михайлова			1			
6	Левочкин		1				
7	Сергеев			1			

Рис. 43. Матрица смежности

Далее порядок действий следующий:

1) сохраняем данные в файл формата *CSV* (текстовый файл с символом «;» в качестве разделителя). Назовем его *друзья.csv*;

2) открываем этот файл в «*Gephi*», выбрав на стартовой станице опцию «Открыть файл с графом»;

3) в появившемся окне Отчета об импорте (*Import Report*) выбираем тип графа Неориентированное (*Undirected*). Поскольку матрица смежности у нас симметричная, сделать это необходимо, иначе связи будут продублированы. Нажав «ОК», получим изображение графа. Его можно редактировать, перемещая узлы, меняя цвет и размер. Кнопкой «T» внизу окна просмотра можно вывести на экран названия вершин. В итоге получится примерно такая картинка, как показано на рисунке 44;

4) для изменения внешнего вида графов используем область Укладка, или *Layout* (левее окна с графом). Выбор эффектов здесь разнообразнее, чем в «*NetDraw*»; есть, например, режим с эффектом притяжения между соединенными вершинами и отталкиванием между несоединенными. После выбора эффекта из меню со списком следует нажать кнопку «Пуск» (*Run*);

5) существует возможность изменить размер вершин в соответствии с их важностью. Так, в нашем примере у Петровой три связи, а у Иванова всего одна. Говорят, что у этих вершин разная степень (*Degree*). То есть степень вершины — это количество связанных с ней ребер.

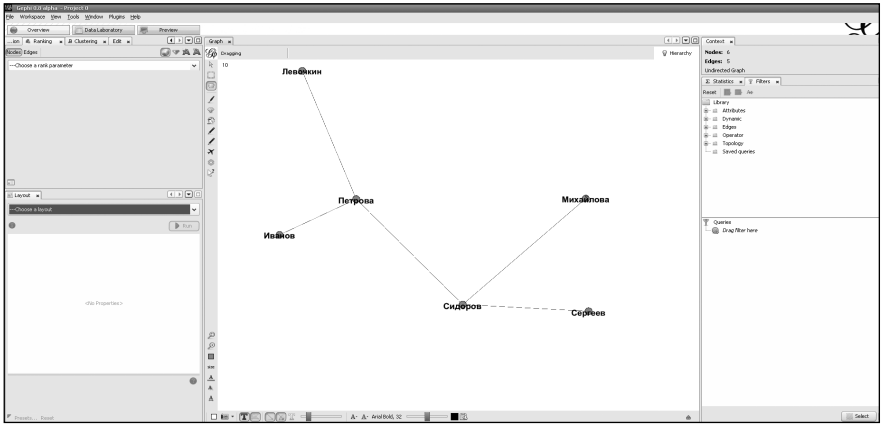

Рис. 44. Визуализация графа по данным рисунка 43

Чтобы вычислить среднюю степень графа, в правом окне перейдите на вкладку Статистики (*Statistics*) и кликните на кнопку «Пуск» (*Run*) в строке Средняя степень (*Average Degree*). Появится окно *DegreeReport*, которое в нашем примере укажет, что средняя степень графа равна 1,667 при четырех вершинах со степенью 1 и двух — со степенью 3 (действительно, (4*1+2*3)/6=1,666). Теперь это окно можно закрыть. Далее переместитесь в область *Appearance* в левой верхней части окна, выберите раздел вершин (*Nodes*), закладку *Attribute* и значок, который отображает изменение размера. В выпадающем меню выберите степень (*Degree*) и установите минимальный и максимальный размер вершин. Нажмите кнопку «Применить» (*Apply*). «*Gephi*» изменит размер вершин согласно степени их важности. В итоге получим примерно следующее (рис. 45):

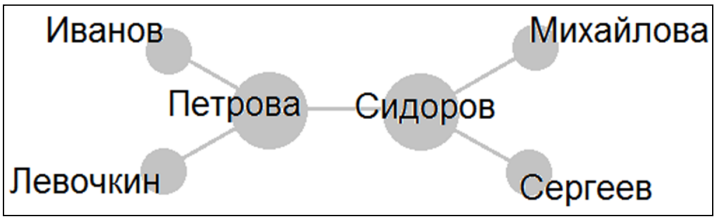

Рис. 45. Ранжирование вершин по степени их важности

«Gephi» может также импортировать файлы *Excel* и файлы *CSV* с данными в формате списка граней (*Edgelist*). Чтобы импортировать такие данные, придется подготовить два файла — один с узлами и их признаками, а другой со списком и признаками граней. Файл *CSV*, содержащий узлы, должен включать колонку *ID* с уникальными идентификаторами узлов, а также любые другие желательные признаки узла (рис. 46).

ID	Visits	SiteType	Rank
n0	1820	Politics	33
n1	2089	Politics	39
n2	1834	Media	28

Рис. 46. Структура CSV-файла с описанием узлов

Файл *CSV* со списком граней должен включать колонки «*Sourse*» и «*Target*», содержащие идентификаторы начального и конечного узла для каждой грани, а также любые другие признаки граней, которые вы считаете нужными.

«*Gephi*» может распознать еще две колонки, если включить их в таблицы «Type» с указанием типа каждой грани (*Ненаправленный* или *Направленный*) и «Weight», содержащий вес граней (рис. 47).

Source	Target	Weight	Type
n0	n1	70	Directed
n0	n109	73	Directed
n0	n110	53	Directed

Рис. 47. Структура CSV-файла с описанием граней

Чтобы импортировать файлы, сначала нужно создать новый проект *File/New Project*. Далее в Лаборатории данных нажмите кнопку «*Import Spreadsheet*». В появившемся окне выберите свой файл и убедитесь, что в выпадающем меню *As table* выбран правильный тип таблицы (т. е. *Nodes Table* для файла с узлами или *Edges Table* для файла с гранями). Импортируйте оба файла, начиная с таблицы узлов (рис. 48).

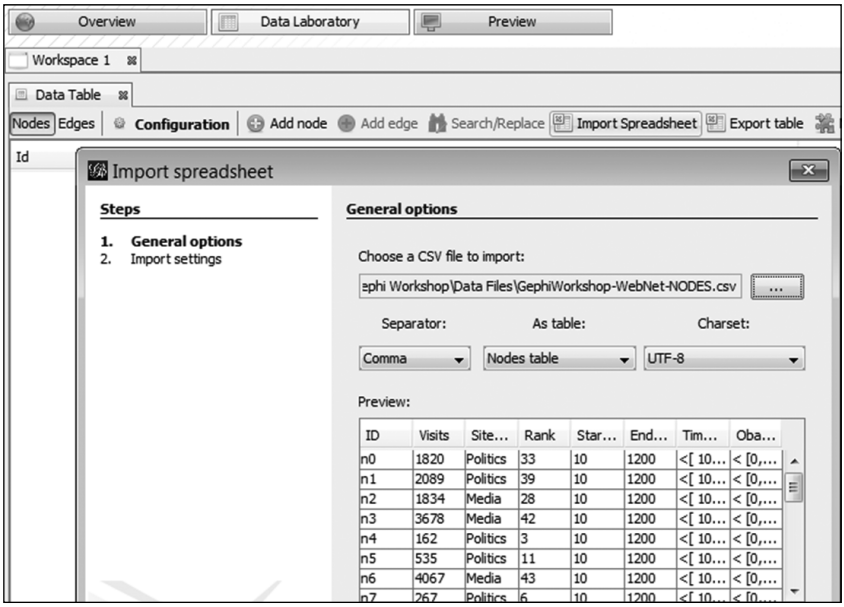

Рис. 48. Импорт данных в «Gephi» из CSV-файлов

Щелкните «*Next*» и удостоверьтесь, что каждому столбцу ваших данных назначен правильный формат. Так, текстовые метки следует импортировать как «*String*», числовые категории как «*Integer*», числовые признаки как «Float», переменные временно́го интервала для динамических сетей как «*IntervalSet*» и т. д. Важно выбрать правильный тип переменных, потому что «*Gephi*» ассоциирует типы переменных с определенными действиями. Например, меню «*Appearance*» не позволит вам увязать размер узла в вашем графе со значением некоторого его атрибута, если этот атрибут имеет тип «*String*», так как для этих целей атрибут должен иметь тип «*Float*».

Рассмотренный выше простой пример, конечно, лишь первый шаг в знакомстве с «*Gephi*». Чтобы у вас не сложилось неверного впечатления о выразительных средствах и аналитических возможностях этой программы, в качестве примера реального использования этого инструмента в аналитической деятельности приведем граф, иллюстрирующий объемы импорта продовольствия в один из российских регионов[4]. Исходные данные содержались по каждому экспортеру в таблице вида (для иллюстрации представлены лишь три строки из 18, а данные приведены по общему объему без деления по странам) (рис. 49).

Статья импорта	2012		2013		2014		2015	
	Вес, т	Ст-ть, тыс. долл.	Вес, т	Ст-ть, тыс. долл.	Вес, т	Ст-ть, тыс. долл.	Вес, т	Ст-ть, тыс. долл.
Мясо	4338,6	7995,1	5184,2	9922	4242,2	9031,4	3094,7	6385,8
Рыба	122,8	565,4	131,9	927,9	195,1	1621,4	27,6	44,8
Молоко, яйца, мед	40	155,9	40	155,9	1440,8	6630,3	1493,2	3456,7

Рис. 49. Импорт продовольствия в регион (фрагмент исходных данных)

Результат работы программы представлен на рис. 50.

Размер узлов здесь зависит от суммарной стоимости импортированных продуктов по определенной группе, году или стране, размер ребер — от стоимости продукции, которая была закуплена по определенной категории товаров в зависимости от года и страны-импортера. Программа автоматически выставляет пропорциональные размеры узлов и присваивает единый цвет элементам — узлам и дугам — одного признака (в данном случае — определенному виду продовольствия). Помимо собственно изображения *«Gephi»* позволяет обрабатывать данные графов, например, осуществлять их фильтрацию и сортировку. Это

[4] *Питиляк Д. А., Рожкова А. О.* Средства визуализации данных Gephi и Google в экономических исследованиях // Молодой ученый. 2016. № 12. С. 1410.

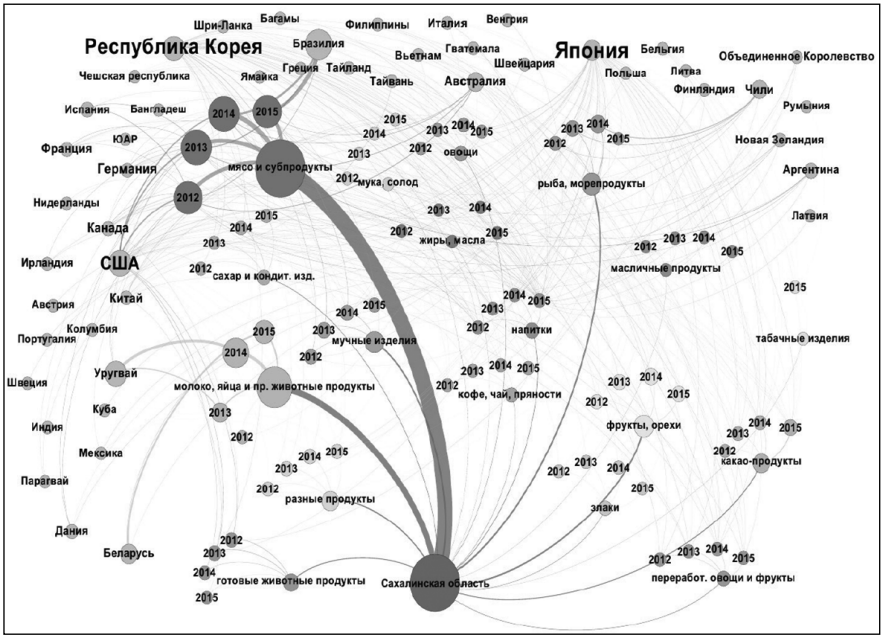

Рис. 50. Пример визуализации в программе «Gephi»

сочетание выразительных средств дает возможность одновременно осуществлять детальный динамический (по годам) и структурный анализ (в данном примере — по видам продовольствия и импортерам) больших массивов данных, что затруднительно сделать с помощью обычных видов графиков или таблиц.

§ 5. Сеть данных криминалистического анализа

Для криминалистического исследования интерес представляют не только связи между лицами. Установление связей между объектами тоже может дать важный материал для доказательства вины или невиновности. Так, объектами компьютерно-технической экспертизы могут быть самые различные цифровые устройства — от ноутбуков, мобильных телефонов и планшетов до автомобильных навигаторов.

Криминалистический анализ отдельно взятого устройства позволяет получить определенную информацию о его владельце: поисковые запросы, часто посещаемые страницы в Интернете, SMS, удаленные файлы, фотографии, а также активность в социальных сетях и программах мгновенного обмена сообщениями. Эта информация характеризу-

ет сферу интересов пользователя устройства и круг его общения; лицо, наделенное совокупностью этих данных, с точки зрения криминалистического анализа можно рассматривать как атомарный объект (сущность). Собранная информация может помочь установить, например, владел ли подозреваемый определенными сведениями о некотором событии, находился ли в момент совершения преступления поблизости, контактировал ли с жертвой или другими подозреваемыми. Немалое значение имеет и установление взаимосвязей между полученными уликами и материалами ранее расследованных уголовных дел, а также поиск похожих данных на различных, в том числе и исследованных ранее, устройствах.

Известно, что сбор и обработка такой информации «вручную» представляет собой сложную задачу. Не исключено, что даже один жесткий диск содержит тысячи электронных писем и миллионы сообщений. Кроме того, человек может иметь несколько почтовых адресов, номеров телефонов и учетных записей, и не всегда есть возможность сразу обозначить список всех собеседников и взаимодействий. Поэтому придется искать похожие учетные записи, возможно, созданные одним человеком, чтобы потом объединить их в одну сущность.

Задача поиска и анализа артефактов, оставшихся после работы пользователя на цифровых устройствах, решается с помощью специального ПО. Артефакты могут представлять собой как файлы (текстовые документы, файлы баз данных браузера со списком посещенных страниц в сети Интернет), так и данные оперативной памяти компьютера (сведения, оставшиеся после посещения сайтов: личные сообщения, письма почтовых сервисов и т. п.). Данные могут быть найдены в неиспользуемых участках кластеров жесткого диска, где сохранились удаленные файлы или куда специально могли быть скрыты временны́е данные программ, представляющие интерес для экспертов. Используемое криминалистами ПО также гарантирует, что во время анализа устройства данные не были изменены. Примером такого специального ПО является *Belkasoft Evidence Center* — продукт российской компании «Белкасофт». Программа облегчает исследователю задачи поиска, анализа и хранения цифровых улик, находящихся в оперативной памяти и на жестких дисках компьютеров, в том числе найденных в чатах интернет-пейджеров, историях браузеров, почтовых ящиках, изображениях, видео, социальных сетях и многопользовательских онлайн-играх. Например, изображения анализируются на наличие порнографии, лиц и отсканированного текста, при этом используются

нейросетевые алгоритмы. Для некоторых форматов фотографий возможна привязка по данным GPS. **Поиск события возможен по ключевому** слову или фразе, регулярному выражению (шаблону) или, при большом количестве искомых слов, словарному файлу. Доступен анализ дампов оперативной памяти, исследование файлов гибернации и подкачки, а также карвинг — последовательный побайтовый поиск различных артефактов в обход файловой системы. Результаты сохраняются в базе данных.

Приведем несколько окон этой программы, позволяющих оценить ее поисковые возможности (рис. 51).

Артефакты, обнаруженные такого рода ПО, могут иметь самостоятельное значение в качестве доказательства. Однако при исследовании социальных связей они чаще являются исходным материалом для последующего анализа, в ходе которого прежде всего необходимо выделить сущности и связи между ними. Обработанные данные желательно представить в виде графа связей, чтобы эксперт быстрее вошел в курс дела и эффективно использовал их; разрозненные данные становятся практической информацией, продвигающей расследование.

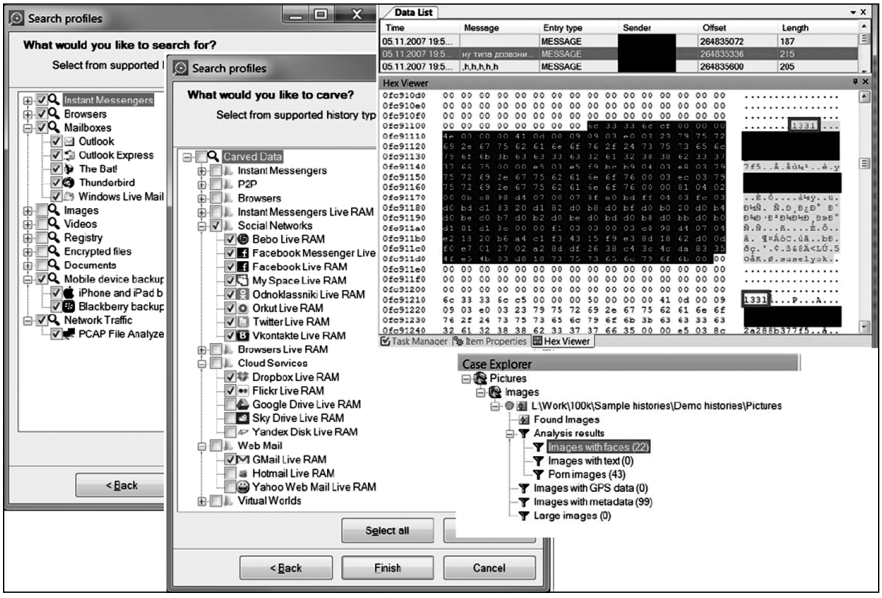

Рис. 51. Поиск профилей и карвинг в программе «Belkasoft Evidence Center»

Рассмотрим общие принципы анализа.

Ключевую роль для автоматизации выделения сущности играет контакт — информация, характеризующая человека или группу лиц. Источники этих данных разнообразны: звонки, голосовые сообщения, электронные письма, короткие текстовые сообщения, мгновенные сообщения, адресная книга мобильного телефона, информация об анализируемом устройстве (из электронного письма, например, можно извлечь контакты отправителя, получателей, получателей копии письма и скрытых получателей копии письма). Каждый контакт может содержать название учетной записи, псевдоним, адрес электронной почты, телефонные номера, имя, фамилию, название компании и т. д. Понятие сущности объединяет контакты, принадлежащие одним и тем же лицам; контакты могут быть объединены в одну сущность, если несколько их характеристик совпадают. Работа с сущностями позволяет анализировать не отдельные сообщения и звонки, а социальные взаимодействия между людьми в целом.

Взаимодействие — это факт передачи информации между контактами и сущностями. Связь между двумя сущностями означает наличие хотя бы одного факта взаимодействия между ними.

Каждая связь имеет вес, значение которого зависит от типа, количества и времени взаимодействий. Так, звонок — более значимый тип взаимодействия, чем электронное письмо: использование телефона предполагает непосредственное индивидуальное общение и бóльшую вовлеченность в диалог, в то время как письмо может иметь несколько адресатов, а ответ на него, возможно, придет через несколько часов или даже дней. Количество взаимодействий также указывает на степень близости между людьми.

Следует заметить, что кроме контактов, принадлежащих реальным людям, существуют также электронные адреса компаний, рассылающих рекламные предложения или новости, спам-боты и т. п. Они, как правило, отправляют множество писем или сообщений за короткий временной промежуток, хотя связь с ними не имеет особой ценности. Поэтому для оценки значимости нужно также учитывать время взаимодействий между сущностями.

В качестве примера приведем результаты визуализации данных программы мгновенного обмена сообщениями *Skype*, полученных от трех разных людей (рис. 52). Программой «*Belkasoft Evidence Center*» было найдено 49 713 различных артефактов, в том числе 471 контакт. Визуализация проведена с использованием библиотеки *GoDiagram*,

предназначенной для создания двумерных графов. Сущности источников данных имеют на графе наибольшее количество связей[5].

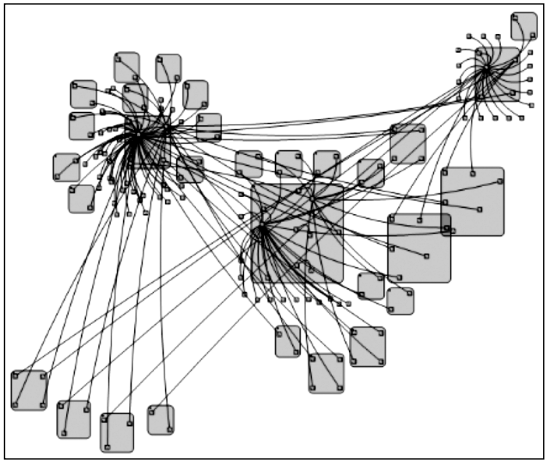

Рис. 52. Визуализация обмена *Skype*-сообщениями по данным трех источников

На основе полученного графа эксперт может сделать выводы о структуре взаимодействий, о характерном поведении некоторых сущностей. Так, из графа видно, что некоторые сущности общались с каждым из хозяев источников данных.

Рассматривая граф, можно выявить наиболее существенные связи — те, которые длятся достаточно долго и содержат большое количество сообщений, писем, звонков и прочих видов взаимодействия. В группу сущностей, которые общаются друг с другом значительно больше, чем с остальными, могут входить, например, люди с общими интересами. Такие группы тесно связанных вершин в графе называют сообществами, а задачу разбивки вершин графа на группы — выделением сообществ.

Основой для достоверного автоматического выделения сообществ считается качественный подсчет весов. Как для определения весов связей, так и для выделения сообществ (кластеризации) на их основе в области ИИ разработаны десятки алгоритмов, различающихся по скорости работы и по качеству разбиения сущностей.

[5] *Чугаева Т. В.* Поиск связей между сущностями в криминалистическом анализе источников данных. СПб., 2016. 26 с.

Среди наиболее известных программных средств для выделения и визуализации сообществ упомянем программный комплекс «Мобильный криминалист» (*Oxygen Forensics*), *Incident Response* (*Nuix*) и *I2 Analyst's Notebook* (*IBM*). Последний программный продукт является более мощным и универсальным решением, однако доступен лишь крупным компаниям ввиду сложности в использовании и высокой стоимости. В настоящее время ожидается интеграция средств интеллектуального анализа и визуализации в новые версии «*Belkasoft Evidence Center*».

ГЛАВА V. АНАЛИЗ БОЛЬШИХ ДАННЫХ

Единого определения понятию больших данных (*Big Data*) не существует. В целом оно означает применение методов ИИ к большому (сотни гигабайт) объему разнообразных, в том числе неструктурированных, оцифрованных данных, находящихся главным образом в открытом доступе, в целях получения новых знаний об интересующих аналитика взаимосвязях и тенденциях. С большими данными современный человек сталкивается ежедневно. Онлайн-магазины исходя из сведений о покупках клиентов делают прогнозы по их предпочтениям, на основе которых клиентам рассылаются сообщения о соответствующих акциях и скидках. Спам-фильтры разрабатываются с учетом автоматической адаптации к изменению типов нежелательных электронных писем. Функция автозамены в смартфонах отслеживает действия пользователя и добавляет новые вводимые слова в свой орфографический словарь. Существуют сайты, способные построить семейное древо и досье на любого человека, основываясь на данных, находящихся в открытом доступе (от рукописных записей в книгах учета до ДНК-анализа). Сайты знакомств подбирают пары на основе корреляции многочисленных атрибутов с теми, кто ранее составил удачные пары, причем предположения основываются не просто на банальном нахождении соответствий в указанных пользователями свойствах и пристрастиях, выявляются более тонкие взаимосвязи. Так, выяснилось, что относительная площадь лица на фотографии в профиле может влиять на вероятность контакта между определенными людьми. А люди с определенными гастрономическими пристрастиями могут обладать разной совместимостью: два вегетарианца с вероятностью 44% найдут общий язык, а два любителя гамбургеров с вероятностью 42% никаких отношений не заведут. Мир стоит на пороге больших данных, и этот подход меняет наше представление о мире[1].

[1] *Латышева А. М.* Big data. Актуальность и перспективы использования // Молодежный научно-технический вестник. URL: http://sntbul.bmstu.ru/doc/724143.html (дата обращения: 23.03.2019).

Большие данные предназначены прежде всего для прогнозирования. Эти системы работают эффективно благодаря поступлению большого количества данных, на основе которых можно строить прогнозы, например, о том, что данное электронное письмо является спамом, что траектория и скорость движения человека, переходящего дорогу в неположенном месте, говорят, что он успеет перейти дорогу и автомобилю надо лишь немного снизить скорость, и т. п. Системы проектируются таким образом, чтобы самообучаться по мере поступления новых данных. В качестве определяющих характеристик для больших данных отмечают так называемые «три V»: объем (*volume*), скорость (*velocity*) и многообразие (*variety*) в смысле одновременной обработки различных типов структурированных и полуструктурированных данных.

Существует много методик анализа больших массивов данных, инструментарий которых заимствован из статистики и информатики. Вот некоторые из них:

— методы *Data Mining* — обучение ассоциативным правилам, классификация, кластерный анализ, регрессионный анализ;

— смешение и интеграция данных. Это набор техник, позволяющих интегрировать разнородные данные для глубинного анализа. К таким техникам относятся цифровая обработка сигналов и обработка естественного языка;

— машинное обучение с учителем и без учителя;

— нейронные сети, сетевой анализ, методы оптимизации, генетические алгоритмы;

— распознавание образов;

— прогнозная аналитика;

— имитационное моделирование;

— пространственный анализ, в том числе использование в данных топологической, геометрической и географической информации;

— статистический анализ;

— визуализация аналитических данных.

§ 1. Понятие и методы «добычи знаний»

Внедрение информационных технологий во все сферы общественной жизни привело к регистрации большого количества самых разных фактов нашей жизни в различных базах данных. Среди них: телефонные звонки и SMS-сообщения, данные фото и видеорегистрации, различного рода происшествия, финансовые транзакции и т. д. Соответствующие базы данных содержат много информации, которая может быть очень по-

лезной для решения различных прикладных задач. Долгое время основным инструментом анализа данных была математическая статистика, а также средства оперативной аналитической обработки данных (*online analytical processing — OLAP*), которые, однако, далеко не всегда позволяют успешно решать такие задачи. Действительно, часто необходимая в конкретной ситуации информация находится в огромном массиве «ненужных» данных. В ряде случаев она может содержаться в неявном виде, например, связи между некоторыми показателями, временны́е и пространственные зависимости и т. д. В этих случаях извлечь из базы данных нужную информацию часто оказывается невозможным без применения современных методов интеллектуального анализа данных, именуемых *Data Mining* (добыча данных), или *DM*.

Приведем общую классификацию задач *Data Mining*:

— классификация;

— кластеризация;

— прогнозирование;

— ассоциация;

— визуализация.

В результате решения задачи *классификации* обнаруживаются атрибуты, которые характеризуют группы объектов исследуемого набора данных — классы. На основании этих атрибутов новый объект относится к тому или иному классу. Для решения задачи классификации, наряду с другими методами, используются нейронные сети и «деревья» решений.

Кластеризация — это логическое продолжение идеи классификации, но она является более сложной, поскольку в этом случае классы объектов изначально не предопределены. Результатом кластеризации является разбивка объектов на группы. Основные проблемы, возникающие при решении задачи кластеризации, связаны с тем, что оптимальное количество кластеров в общем случае неизвестно, а выбор меры похожести или близости свойств объектов между собой, как и критерия качества кластеризации, часто носит весьма субъективный характер. Поэтому обычно считается, что кластер является массивом векторов (объектов), расстояние между которыми внутри кластера всегда меньше, чем до любого вектора другого кластера. То есть кластер — это область векторного пространства, содержащая группу близко расположенных векторов. Одним из популярных инструментов кластеризации является ИНС Кохонена.

В результате решения задачи *прогнозирования* на основе особенностей данных о прошлом состоянии некоторой системы оцениваются пропущенные или же будущие значения целевых численных показателей.

Для решения таких задач широко применяются методы математической статистики, нейронные сети и др.

В ходе решения задачи поиска *ассоциативных правил* устанавливаются закономерности между связанными событиями в наборе данных. Отличие ассоциации от предыдущих задач *Data Mining* в том, что поиск закономерностей осуществляется не на основе свойств анализируемого объекта, а между несколькими событиями, которые происходят одновременно.

В результате *визуализации* создается графический образ анализируемых данных. Для решения задачи визуализации используются графические методы, показывающие наличие закономерностей в данных.

Методы *Data Mining* все более широко применяются в различных отраслях знаний и прикладных сферах, в том числе:

— в социальных сетях;

— в системах электронной коммерции;

— в банковском деле;

— в средствах массовой информации;

— при анализе массива нормативных правовых актов (далее — НПА);

— при анализе обращений граждан;

— в договорах, исках, решениях, протоколах о происшествиях.

DM — это междисциплинарная методология, т. е. совокупность методов, технологий и алгоритмов интеллектуального анализа данных, используемых в целях обнаружения скрытой (неочевидной) и нетривиальной информации, полезной для принятия решений. Программные продукты *DM* входят в состав систем *Business Intelligence*, включающих в себя средства построения хранилищ данных; системы оперативной аналитической обработки (*OLAP*); информационно-аналитические системы; средства интеллектуального анализа данных (собственно *Data Mining*) и инструменты для выполнения запросов и построения отчетов.

Сегодня методология *DM* рассматривается как самостоятельное научное направление. Она базируется, помимо математической статистики и *OLAP*, на подходах и методах машинного обучения, ИИ, проектирования и управления базами данных, а также в других смежных областях *ИТ*. Главная ее ценность — это практическая направленность, путь от «сырых» данных к конкретному знанию, от постановки задачи к готовому приложению, при поддержке которого принимают решения[2].

---

[2] *Нефедов С. Н., Пархименко В. А., Татур М. М.* Применение методов интеллектуального анализа данных в криминалистике и судебной экспертизе // Вопросы криминологии, криминалистики и судебной экспертизы. 2017. № 2. С. 60.

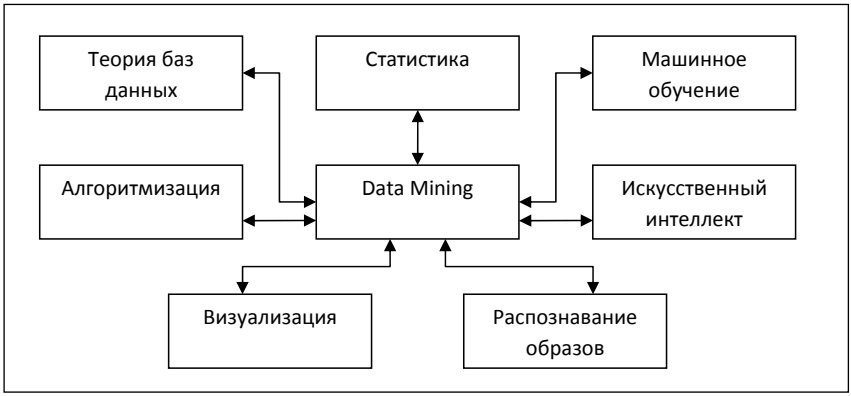

Рис. 53. Компоненты методологии *Data Mining*

Технологии *Big Data* позволяют анализировать данные в их первоначальном состоянии, без дополнительного структурирования. Обратим внимание на то, что среди перечисленных выше областей применения три первых (социальные сети, банковское дело и электронная коммерция) используют информацию, структурированную изначально, так как ее источниками обычно являются электронные формы. Другие же, в том числе и юридические применения, в качестве источников используют неструктурированные или слабоструктурированные текстовые документы.

Процесс интеллектуального анализа данных состоит из следующих шагов:

1) по определенному алгоритму создается модель добычи данных, настроенная на некоторую их входную выборку;

2) с помощью некоторых обучающих данных (в которых известны как исходные атрибуты, так и те, которые необходимо предсказывать в будущем) производится обучение созданной модели;

3) после обучения на вход модели добычи данных подаются предсказываемые данные, т. е. такие, в которых неизвестны интересующие нас (и потому являющиеся объектами прогнозирования) атрибуты. В результате работы алгоритма эти неизвестные атрибуты будут с определенной вероятностью предсказаны.

Среди *алгоритмов интеллектуального анализа данных* можно выделить алгоритмы: кластеризации, взаимосвязей, дерева принятия решений, линейной и логистической регрессии, Байеса, нейронных сетей, кластеризации последовательностей, временны́х рядов[3]. Наиболее пол-

[3] *Чернышова Г. Ю.* Интеллектуальный анализ данных : учеб. пособие. Саратов, 2012. 92 с.

ным учебным курсом по добыче данных представляется работа В. Дюка и А. Самойленко[4].

Если условно обозначить основные направления, по которым юристы пересекаются с новой технологией, можно выделить три направления:

1 — большие данные, которые юристы используют в работе;

2 — методология больших данных, призванная заменить юристов;

3 — большие данные как явление, с которыми юристам приходится сталкиваться.

Первое направление, в рамках которого юристы работают с анализом *Big Data*, — использование массивов данных для автоматизации работы в целях поиска разного рода несоответствий или, наоборот, возможностей в законах. В США, например, успешно работает система прогнозирования вероятности прохождения законов через конгресс. Анализ данных существенно помогает в судебной практике. Так, он дает возможность анализировать судебные споры, получать краткое и содержательное заключение на основании анализа сотен судебных актов, экономя время и деньги, создавая преимущество перед оппонентами. Можно просчитать, какие аргументы работают для большинства, а что имеет значение для конкретного судьи. Также благодаря анализу массивов данных юридической фирме проще понять, стоит ли вообще браться за дело: если раньше подобное исследование вопроса требовало недель, то сегодня благодаря технологиям оно займет 20 мин. Разумеется, аналитические системы должны иметь доступ к массиву судебных данных в электронном виде.

Другое направление применения анализа больших данных — автоматизация работы юриста. Известен пример с автоматизацией поиска незаконных штрафов за парковку в Нью-Йорке, когда всего лишь один аналитик, сидящий за компьютером, нашел с помощью городских открытых данных тысячи нелегально выписанных штрафов.

Заметим, что использование больших данных в прогностических целях создает для юристов некоторые проблемы. Появление новых возможностей предвидения развития ситуации — это, безусловно, хорошо. Однако столь мощный инструмент в руках юриста закономерно приводит к тому, что от него ожидают более эффективного и предсказуемого результата, таким образом возрастает и его ответственность за свои решения. С другой стороны, какой именно должна быть эта ответственность, если ожидания все же не оправдаются? Риск ошибки уменьшается за счет

[4] *Дюк В., Самойленко А.* Data Mining : учебный курс. СПб., 2001. 368 с.

технологий, но все же прогнозы — вещь неблагодарная. При широком внедрении технологий в право и управление вряд ли будут приниматься во внимание популярные сегодня оправдания вроде «это не я, это программа». В США, например, юрист обязан быть в курсе перемен в профессии, в том числе связанных с рисками от использования технологий. И если юрист недостаточно информирован о рисках, это вполне можно расценивать как недобросовестность.

§ 2. Основные направления применения технологии DM в деятельности полиции

Технология *DM* предоставляет новые возможности по выявлению и анализу скрытых закономерностей в данных, касающихся фактов преступлений и противоправных действий. Это автоматизированное выявление общих трендов и закономерностей в сфере преступности; прогнозирование преступлений; объяснение преступных явлений и поведения преступников; автоматизация отдельных этапов аналитической работы криминалистов и замена дорогостоящего труда экспертов; полноценное использование огромного массива накопленной информации о преступлениях; автоматическое выявление незаконной активности в сети Интернет и др. Таким образом, значительно облегчается принятие решений, направленных на предотвращение или раскрытие преступлений.

Остановимся подробнее на идентификации преступника. Известно, что совершающий многократные насильственные преступления часто оставляет характерный элемент преступного поведения, называемый «визитной карточкой», или сигнатурой. Это его уникальное поведение неосознанно. Совершая преступление, серийный преступник не всегда способен удовлетворить свои потребности, и это неудовлетворение он вынужден компенсировать, заходя за грань обычных действий, исполняя свой ритуал. Например, сигнатура насильника обнаруживается в актах доминирования, манипуляции или контролирования в процессе словесных, физических действий или сексуального насилия. Другой тип сигнатуры определяется чрезмерным использованием силы, условиями совершения преступления и т. д. Сигнатура является константной и неизменной чертой каждого преступника. Она никогда не меняется, но склонна эволюционировать: элементы личного ритуала могут развиваться, становятся все более изощренными. Сигнатура не всегда проявляется на местах преступлений из-за таких непредвиденных обстоятельств, как неожиданные действия жертвы или вызванная чем-то тревога преступника. Следователи не всегда бывают в состоянии распознать сигнатуру,

а методы *DM*, корректно к данным о прошлых событиях, могут помочь им в этом, способствуя ассоциации преступлений между собой и с конкретным преступником.

Так, применение кластерного анализа позволило Томасу Хэргроуву создать алгоритм поиска серийных убийц [5]. Получив однажды доступ к отчету ФБР, содержащему 16 тыс. записей об убийствах с указанием возраста, пола, расовой принадлежности жертвы и способе убийства, о полицейском участке, который вел дело, об известных обстоятельствах и об обвиняемом (если его личность установлена), Хэргроув задался целью научить компьютер вычислять серийных убийц: анализировать тенденции нераскрытых преступлений, используя общедоступную информацию, на которую никто, даже правоохранительные органы, не обращали внимания.

Хэргроуву удалось, изучая уже раскрытые дела, выделить кластеры, верно идентифицирующие преступника (атрибуты: геолокация, пол, возрастная группа, способ убийства). Затем на основании исследования нераскрытых дел (возможные неопознанные жертвы серийных убийц) к этим атрибутам были добавлены другие, характеризующие социальные кластеры жертв как наиболее легкую добычу. В результате был получен алгоритм, реализованный в некоммерческом проекте *MAP* (*Murder Accountability Project*), сделавшем данные ФБР широко доступными. База данных *MAP* (рис. 54) содержит информацию почти о 800 тыс. преступлений, начиная с 1976 г., в том числе десятках тысяч дел, по разным причинам не дошедших до ФБР [6]. Это наиболее полный и наиболее детальный из существующих список убийств, совершенных на территории США, который находится в открытом доступе. Методика Хэргроува весьма проста — это еще не *DM*, но история ее создания и использования достаточно показательна.

На рисунке 54 приведена одна из записей базы данных *MAP*, дающая представление о характере информации, предоставленной проектом. В целях демонстрации мы использовали форму *Access*, хотя оригинальная база имеет формат *Excel*. В §7 данной главы предложены учебные задания с использованием базы *MAP*. Для их решения необходимо предварительно скачать исходную базу, находящуюся в открытом доступе, либо воспользоваться ее учебной версией.

---

[5] *Колкер Т.* Bloomberg: алгоритм для поиска серийных убийц. URL: http:// theidealist.ru/algorithmformurdercases (дата обращения: 21.08.2018) ; *Бельков В. А., Алдашкина А. С.* Использование специальных программ для установления серийности при расследовании убийств // Пролог: журнал о праве. 2017. № 3. С. 22–28.

[6] URL: http:// www.murderdata.org (дата обращения: 26.04.2019).

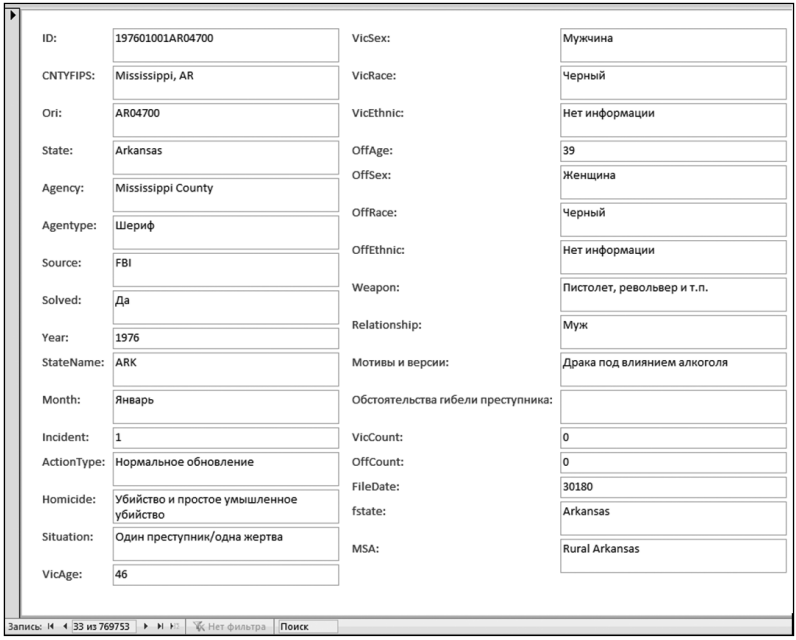

Рис. 54. Типичная запись проекта Murder Accountability Project (США)

Другой пример — математическая модель расчета вероятности преступлений, которая каждый день составляет новый маршрут для патрульных машин с указанием десяти «горячих точек», основываясь на статистике преступлений по улицам. Соответствующая программа работает с 2011 г. в г. Санта-Крус (США). Учитываются день недели, время суток, наличие/отсутствие футбольных матчей по телевидению и другие факторы. Модель составляется на базе статистики преступлений за последние несколько лет и напоминает расчет вероятности афтершоков — повторных сейсмических толчков меньшей интенсивности по сравнению с главным толчком. Как и в случае с землетрясениями, каждое преступление тоже рождает волны «афтершоков», то есть повышает вероятность новых преступлений в том же месте в будущем. Хотя один процесс происходит в земной коре, а другой в человеческом обществе, но, как ни странно, для их описания используются похожие формулы. На карте для каждого квадрата размером 150 м на 150 м указываются: вероятность совершения преступления в 24-часовой период (P); распределение этой вероятности по двум видам преступления — автомобильные ($Pveh$) и домашние ($Pres$); время начала двух самых опасных часовых интервалов (TW) (рис. 55).

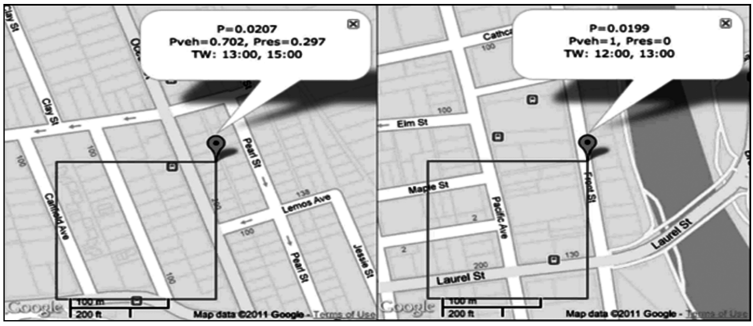

Рис. 55. Прогноз вероятности преступлений (полиция г. Санта-Круз)

Новые данные добавляются в систему каждый день. В первый месяц тестирования система позволила предотвратить несколько преступлений и привела к пяти арестам, а количество ограблений машин снизилось на 27% по сравнению с тем же месяцем предыдущего года.

Полезным для правоохранительных органов является и алгоритм, предложенный *Mirco Musolesi* (университет Бирмингема). Используя оперативные данные из сетей сотовой связи, алгоритм с высокой степенью вероятности прогнозирует перемещения каждого абонента по сигналам его телефона, истории звонков и текстовым сообщениям. Иногда алгоритм прогнозирует координаты пользователя с точностью до 20 м^2. Алгоритм работает эффективно только при условии, что одновременно отслеживается вся сеть друзей указанного пользователя. Если отслеживать только одного человека, то точность предсказания координат снижается до 1000 м^2, если же удается извлечь уточняющую информацию всего у одного друга, то точность сразу резко увеличивается. Таким образом, алгоритм способен вычислить место и время, где через 20–30 мин встретится группа потенциальных преступников. Можно вычислить конкретную улицу, квартал, иногда даже дом — место потенциального преступления. Очевидно, что туда на всякий случай нужно направить патрульную машину.

Среди программных продуктов, предназначенных специально для полиции, лидером является «*COPLINK*» британской компании «*i2 Group*». Он позволяет быстро получить доступ к информации, накопленной правоохранительными органами, и проявить в ней скрытые связи между людьми, местами, автомобилями, мобильными телефонами и т. п. Программа внедрена и эффективно работает во многих городах. Вот функции ее основных модулей:

— быстрый поиск возможных подозреваемых по всей доступной информации;

— идентификация подозрительной активности на территории, взятой под наблюдение, исходя из информации, полученной из разных источников;

— идентификация подозреваемого по фотографии или фотороботу;

— предоставление инструментария для статистической обработки информации и ее визуализации;

— выявление и визуализация географической и временно́й связи между преступлениями;

— визуализация отношений и ассоциаций между людьми, событиями, местоположениями, организациями.

В литературе часто упоминаются следующие направления использования методов *DM* в деятельности полиции[7]:

— составление профилей преступников, в том числе быстрая идентификация;

— выявление и прогнозирование территорий/объектов повышенной криминогенности;

— ассоциация преступлений между собой и с конкретным преступником, организацией, транспортным средством для выявления серийных преступлений и преступных групп, а также ассоциация ранее не раскрытых преступлений с конкретным лицом или группой лиц;

— оптимизация распределения ограниченных полицейских ресурсов/сил;

— выявление преступников, совершающих преступления с одним и тем же «почерком»;

— обнаружение зависимостей между характеристиками жертвы преступления, местом, средством и другими обстоятельствами преступления;

— выявление обмана в предоставляемых задержанным данных о себе за счет сравнения текстовых данных в различных источниках;

— ранжирование подозреваемых на основе обработки свидетельских показаний;

— обнаружение мошенничества в финансовых транзакциях, страховании, телекоммуникационном секторе и здравоохранении;

— обнаружение несанкционированного доступа к компьютерной сети посредством выявления повторяющейся последовательности действий в сетевых транзакциях;

— анализ преступных сетей для выявления связей, ролей, подгрупп в иерархии преступников;

---

[7] *Нефедов С. Н., Пархименко В. А., Татур М. М.* Указ соч. С. 60.

— автоматические извлечения структурированной информации из письменных отчетов правоохранительных органов и открытых источников;

— автоматические извлечения характеристик ПО в целях выслеживания и установления личности хакеров.

Следует отметить, что при использовании методов *Data Mining* и *Big Data* следует соблюдать осторожность. Любая модель все же создается людьми, именно они выбирают аналитический алгоритм и набор анализируемых атрибутов, и поэтому их ошибки при построении модели приводят к неверному результату анализа. Для иллюстрации того, насколько опасно некритичное применение методов *Data Mining*, специалист по анализу данных Джим Адлер загрузил в самообучающийся алгоритм сведения о нескольких десятках тысяч жителей американского штата Кентукки, привлекавшихся к уголовной ответственности с начала восьмидесятых годов прошлого века. Из опубликованных полицией записей извлекались приметы: наличие татуировок, цвет кожи, глаз и волос, пол, история нарушений закона и тяжесть правонарушений, совершенных в прошлом. Алгоритм построил «дерево решений», где каждой ветви присвоен определенный вес. Например, при выборе мужского пола результат вырастает на 0,1, а при выборе женского — уменьшается на 0,5. Вес всех выбранных ветвей суммируется. Если результат превышает вычисленное при обучении пороговое значение, то рассматриваемый человек объявляется потенциальным преступником.

При тестировании с наиболее «жесткими» настройками алгоритм верно идентифицировал 51 246 человек, осужденных за тяжкие преступления, т. е. все 100% преступников, упомянутых в выборке Адлера. При этом количество ложных срабатываний составило 2200 (около 4%). При более мягких настройках невиновные встречаются реже (152 ложных срабатывания (0,3%), но тогда ускользает и часть преступников (учтено 37 842 из 51 246).

На первый взгляд, классификатор неплохо справлялся со своей работой. Однако Адлер не уверен, что это можно считать успехом. Ведь что, в сущности, показал эксперимент? Что преступников можно определить по цвету кожи, полу и наличию татуировок? Или то, что в людях с известным цветом кожи полиция заранее подозревает преступников, а наличие татуировок оказывает влияние на отношение суда? В базе данных, по которой обучался алгоритм, нет нераскрытых преступлений. Отсутствуют в ней и оправдательные приговоры, а ведь цвет кожи определенно связан и с финансовыми возможностями, нужными для

того, чтобы нанять хорошего адвоката; это не может не влиять на результат. Таким образом, в результате переработки исторических данных получился не классификатор преступников, а экспертная система, кодирующая предрассудки полицейских из Кентукки. По мнению Адлера, это демонстрирует необходимость критично относиться к анализу данных. Некоторые результаты — вовсе не то, чем они кажутся. «Как и настоящее искусство, настоящий анализ данных порождает не меньше вопросов, чем ответов. Иногда эти вопросы неприятны, но в конечном счете это к лучшему» [8].

§ 3. Применение DM для выявления мошенничества в системах электронной коммерции

Весьма эффективно применение технологии *Data Mining* в системах электронной коммерции. Формы, которые заполняют покупатели в интернет-магазинах и участники интернет-аукционов, являются источником огромного массива хорошо структурированной информации, что способствует широкому применению как статистических методов, так и нейронных сетей.

Сформированные таким образом базы данных позволяют оценить риски от сделок, расследовать мошенничества, например, обучая нейронную сеть на основе прошлого опыта. Эксперты оценивают процент мошеннических схем на электронных торгах имуществом примерно в 15–20%. Рассмотрим в качестве примера основные формы аукционного мошенничества в сети Интернет. Каждая форма мошенничества имеет свои характерные атрибуты, которые и используются для ее идентификации.

Непоставка — продавец размещает предложение, когда фактически предложения или намерения продать не существует. Товар после покупки покупателю не поставляют.

Искажение — продавец старается обмануть покупателя относительно истинной ценности товара, предоставляя ложную информацию. Иногда, чтобы отбить желание у других участников купить конкретный лот, характеристики и качество лота в описании искусственно занижают. Например, указывается на некомплектность оборудования, необходимость ремонта недвижимости и т. п. Цель — продать лот конкретному заказчику. Но чаще ценность товара, наоборот, искусственно завышается для увеличения дохода.

---

[8] *Парамонов О.* Большие данные на службе полиции (и преступников). URL: https://www.computerra.ru/228030/crime-bigdata/ (дата обращения: 02.09.2019).

Неуплата — покупатель предлагает самую высокую цену, выигрывая аукцион, а после получения товара не перечисляет денег. Потерпевшим в этом случае является продавец.

Триангуляция включает три стороны: преступник, покупатель и торговец онлайн. Преступник покупает товары у торговца онлайн, используя чужие учетные записи и номера кредитной карточки. Далее преступник продает товары на аукционных площадках онлайн добросовестным покупателям. Позже, когда полиция изымет украденные товары как доказательства, потерпевшими оказываются и покупатель, и продавец.

Накрутка цены — добавление скрытых трат к предложению после завершения аукциона. Вместо общего тарифа за стоимость пересылки и обработку продавец отдельно взимает плату за стоимость пересылки, обработку и отгрузку. В результате покупателю приходится заплатить больше, чем ожидалось.

Товары черного рынка включают контрафактное ПО, компакт-диск музыки, видео и т. д. Товар поставляют без упаковки, гарантий или инструкций.

Многократное предложение цены — покупатель размещает несколько предложений (некоторые с высокими ценами, а некоторые с низкими), используя различные псевдонимы. Множество предложений высокой цены заставляют ее расти, отпугивая других потенциальных покупателей от того, чтобы сделать свои предложения. Затем, в последние несколько минут аукциона, тот же покупатель отзывает свои «высокие» предложения, чтобы приобрести товар на основании своего предложения самой низкой цены. Вариант: тайно сговариваются два покупателя; один претендент предлагает низкую цену, а другой высокую. Перед завершением аукциона предлагающий высокую цену уходит.

Фиктивный аукцион устраивается исключительно для получения имен и номеров кредитных карт.

Подмена — покупатель получает товары и отказывается от них, предварительно подменив оригинальные товары другими, более низкого качества.

Требование возврата — покупатель утверждает, что товар был поврежден и он избавился от него. В связи с этим он требует возврата денег.

Тарабарщина — примитивный, но эффективный способ избавиться от нежелательных покупателей имущества — текст объявления непонятен, неясно, что продается.

Таран — сговор как минимум трех организаций, выходящих на аукцион. У двух из них документы-заявки имеют намеренно организованные

несоответствия. Вначале одна компания делает минимальную ставку на понижение. А затем две другие стремительно начинают ронять цену, используя максимальный шаг. Другим участникам торговаться в таких условиях просто нет смысла — рентабельность уходит практически в ноль. Иногда остальных участников даже программно блокируют, чтобы не допустить появления новых ставок. Если же кто-то и успел внести свое предложение, то уже почти перед самым закрытием торгов первая компания делает выгодную ей ставку таким образом, чтобы оказаться на третьем месте, но с максимальной ценой. Ей не страшен большой разрыв с двумя лидерами, поскольку известно, что их дисквалифицируют ввиду несоответствия документов.

Нереальные сроки на исполнение заказа обычно свидетельствуют о сговоре заказчика с исполнителем: взяться за работу (например, снять фильм или создать компьютерную программу) могут только те участники, которые уже проделали большую ее часть. Здесь госзаказ только формально участвует в аукционе, так как предназначен конкретному исполнителю.

Конечно, это далеко не все схемы мошенников. Исследования характерных способов мошенничества позволяют выявить их сигнатуры, (идентифицирующие признаки обучающей выборки). Так работает система *FADE* (*Fraud and Abuse Detection Engine*) — система обнаружения мошенничеств и злоупотреблений, используемая в целях безопасности аукционной онлайн-площадкой eBay. Данные об истории онлайн-аукционов позволяют на основе прошлого опыта формировать репутацию их участников. Для этого предложены специальные алгоритмы, реализуемые на специализированных репутационных сервисах, от которых можно получить большой объем сведений в отношении потенциальных контрагентов и торговых площадок.

Использует технологию *DM* для выявления мошенничества и финансовая служба Министерства обороны США. Так, одна из анализируемых структур данных представляла собой совокупность полей, выгруженных из миллионов коммерческих счетов и помещенных в базу данных. Аналитики снабдили каждую запись меткой, определяющей каждый счет либо как «мошеннический», либо как «не мошеннический». Пометка «мошеннический» ставилась только на тех счетах, принадлежность которых к этой категории была установлена экспертами и доказана в судебном порядке. Однако большинство счетов не проверялось никогда — по умолчанию они были отнесены в категорию «не мошеннических».

На подготовительном этапе анализа данных специалисты службы выделили несколько сделок с доказанными случаями мошенничества и использовали исходные документы, чтобы воссоздать все остальные

сделки и поместить их в базу данных вместе с некоторым числом непомеченных сделок, которые в большинстве своем не являлись «мошенническими». При этом, если в выборке участвовало лишь несколько тысяч таких непомеченных сделок, вероятность пропустить «мошенническую» была относительно мала. Таким образом, получался комбинированный набор данных, который разделялся на 11 поднаборов методом отбора перекрестной проверкой. Наконец, вместо всего двух категорий «мошенничество» и «не мошенничество» целевая переменная могла принимать пять значений в соответствии с четырьмя возможными типами (классами) мошенничества (предварительно выделенными при участии экспертов) и одним типом, характеризующим «не мошеннические» сделки. Окончательные результаты в данном случае таковы: комплексная модель верно определила 97% известных случаев мошенничества (в контрольной выборке), и 1217 платежей были направлены экспертам для дальнейшей проверки.

§ 4. Источники информации для DM

Репутация важна не только для участников интернет-аукционов. Заключение сделок, решение кадровых вопросов, и, конечно, проведение следственных действий и оперативно-розыскных мероприятий требует сбора разносторонней и достоверной информации для формирования досье на объект. Для получения таких сведений о физических и юридических лицах в сети Интернет тоже существуют сервисы, один из них так и называется — «Репутация». Сервис предоставляет: 1) информацию об исполнительных производствах — текущих, оплаченных и закрытых, а именно сумму и объект исполнения (штрафы, задолженности, конфискация и пр.), а для закрытых производств — причину их закрытия; 2) данные о плановых проверках, в том числе о проверяющем органе, дате и цели проверки и ее результатах, включая выявленные нарушения; 3) информацию о залогах — заложенную долю учредителя, дату возникновения залога, его описание; 4) по арбитражным делам: тип дела и роль контрагента в нем, сумму иска, тип дела и документы по нему. Кроме того, в автоматизированном режиме делается прогноз исхода дела. Реализация этой последней функции свидетельствует об использовании упомянутым сервисом аналитических алгоритмов. Другая же предоставляемая сервисом информация имеет фактографический характер и при наличии соответствующих баз данных не требует применения методов ИИ. Однако сам процесс автоматизированного формирования этих баз данных (если «сырые

данные» представляют собой текстовые документы с решениями арбитражного суда) невозможен без использования ИИ.

Много полезной информации о человеке можно получить и не прибегая к помощи платных сервисов. Ниже приведены адреса сайтов, на которых можно получить необходимые сведения бесплатно.

Проверка действительности паспорта:

— services.fms.gov.ru/info — service.htm?sid=2000 (сайт ФМС);

— service.nalog.ru/inn.do (сайт ФНС).

Номер ИНН:

— service.nalog.ru/debt;

— service.nalog.ru/inn-my.do.

Задолженность по налогам:

— service.nalog.ru/debt.

Адрес прописки и домашний телефон:

— telkniga.com; www.telpoisk.com; www.nomer.org.

Розыск подозреваемых:

— www.fssprus.ru/iss/suspect_info (сайт Федеральной службы судебных приставов).

— mvd.ru/wanted (база данных МВД).

Розыск по исполнительным производствам:

— www.fssprus.ru/iss/ip_search.

Проверка по банку данных исполнительных производств:

— www.fssprus.ru/iss/ip.

Розыск автомобилей должников:

— www.fssprus.ru/iss/search_amts.

Получение информации (адрес, e-mail, телефон) по IP-адресу:

— www.dnsstuff.com (служба доменных имен).

Проверка существования фирмы:

— egrul.nalog.ru.

Поиск компании (фактический адрес, телефон):

— www.allinform.ru.

Судебные процессы в гражданских судах общей юрисдикции и в арбитражных судах:

— sudrf.ru/index.php?id=300#sp (поиск по делам и судебным актам);

— bsr.sudrf.ru/bigs/portal.html (поиск по текстам судебных решений).

Проверка на банкротство:

— ras.arbitr.ru (банк решений арбитражных судов);

— kad.arbitr.ru (картотека арбитражных дел, которые находятся в производстве).

Проверка документов об образовании:

— frdocheck.obrnadzor.gov.ru (Федеральная служба по надзору в сфере образования и науки).

Поиск ссылок по фото:

— www.imageraider.com (поиск в Google, Bing и Yandex);

— images.google.com (поиск в Google);

— yandex.ru/images/search;

— www.tineye.com.

Богатый материал для анализа предоставляют данные социальных сетей. Так, социальная сеть «ВКонтакте» хранит такую информацию:

— привязанный к странице e-mail;

— дата регистрации аккаунта;

— IP-адреса регистрации и последних восьми входов в аккаунт;

— дата последнего изменения пароля (и с какого IP-адреса это было сделано);

— дата изменения привязанного номера телефона;

— история обращений в службу поддержки: даты создания и заголовки диалогов;

— история изменения имени и фамилии;

— заявки на восстановление страницы;

— история блокировок страницы с указанием причины;

— список друзей на момент создания архива — имена и ссылки на профили;

— отправленные и полученные сообщения за всю историю аккаунта (с 2007 года) с прямыми ссылками на прикрепленные к ним картинки и текстом пересланных сообщений;

— посты и комментарии на стене пользователя, оставленные как им самим, так и другими;

— фотографии, на которых отметили пользователя (с прямыми ссылками на файлы);

— все содержимое групповых чатов, в которых состоял пользователь, в том числе сообщения других участников;

— все альбомы пользователя с фотографиями, существовавшие на момент создания архива (включая закрытый альбом «Сохраненные фото»), со ссылками на файлы, временем загрузки и подписью;

— все видеозаписи, загруженные или добавленные на страницу: название, ссылка на страницу, дата загрузки, IP-адрес, длительность, количество просмотров;

— документы, в том числе «граффити», отправленные в личных сообщениях, с датой загрузки, IP-адресом и прямой ссылкой на файл;

— группы, в которых состоит пользователь, с указанием тех, где он является администратором;

— аудиозаписи, добавленные в «Мои аудиозаписи»: названия и даты добавления;

— политические и религиозные предпочтения (те, что были указаны на странице);

— дата рождения;

— мобильный телефон;

— город.

При этом сохраняется не только актуальная, но и удаленная (стертая) переписка. Согласно законодательству социальные сети, внесенные в реестр организаторов распространения информации, должны хранить переписку шесть месяцев, а другую информацию — в течение года.

§ 5. Общие задачи обработки неструктурированной информации

Для того, чтобы подвергнуть анализу структурированную информацию, хранящуюся в базах данных, ее необходимо предварительно обработать: осуществить ввод информации по определенным правилам, разместить ее в таблицах и т. д. Иными словами, для анализа этой информации и выделения из нее новых знаний необходимо затратить дополнительные усилия, которые при этом не всегда связаны с анализом и не обязательно приводят к желаемому результату. В итоге эффективность анализа информации снижается. Кроме того, не все виды данных можно структурировать без потери полезных сведений. Так, практически невозможно преобразовать текстовые документы в табличные без потери смысла, и поэтому такие документы хранятся в базах данных без преобразований, в виде текстовых полей. В тексте скрыто огромное количество информации, однако ее неструктурированность не позволяет использовать алгоритмы *Data Mining*. Решается эта проблема при помощи метода анализа неструктурированного текста. В западной литературе такой анализ называют *Text Mining*.

С технологиями анализа неструктурированного текста мы чаще всего встречаемся, работая с программами автоматического перевода текста (например, «*PROMT*» или «*Google-переводчик*»), средствами проверки орфографии (например, в текстовом редакторе *MS Word*) или системами проверки заимствований (типа «Антиплагиат»). Это самые

очевидные, но далеко не все применения *Text Mining*. Ведь подавляющее большинство важных документов, в том числе юридических, не структурировано.

О роли, которая отводится методам интеллектуальной обработки текстов в борьбе с преступностью, свидетельствуют назначение и целевое применение сервиса управления оперативно-розыскной информации единой информационной системы централизованной обработки данных МВД России:

— поиск объектов, находящихся в розыске, посредством мониторинга и сбора информации о продаже объектов с заданными характеристиками на различных тематических интернет-аукционах, магазинах, а также появления информации об искомых объектах на различных тематических ресурсах (форумы, блоги и т. д.);

— предотвращение случаев разжигания межнациональных конфликтов путем своевременного мониторинга социальных ресурсов на предмет наличия сообщений, статей с соответствующим содержанием;

— поиск подозреваемых, находящихся в розыске, при помощи сбора и мониторинга информации из социальных сетей, построения иерархических взаимосвязей между людьми, объектами, событиями по различным критериям (родственные и служебные связи, имущество и т. д.);

— сбор информации о предполагаемом проведении несанкционированных акций, митингов, шествий с прогнозированием количества участников и возможных путей перемещения и дислокации;

— просмотр исторической справки по объектам и событиям на основании данных, собранных из текстовых документов (отчеты, справки, постановления, выписки, приказы и т. д.);

— иные варианты поиска и мониторинга ситуации: торговля людьми, детская порнография, сбыт наркотиков, терроризм и т. д.

Методы анализа в неструктурированных текстах лежат на стыке нескольких областей: *Data Mining*, обработка естественных языков, поиск информации, извлечение информации и управление знаниями.

Обнаружение знаний в тексте — это нетривиальный процесс обнаружения действительно новых, потенциально полезных и понятных шаблонов в неструктурированных текстовых данных, т. е. в документах, содержащих связанный логически текст произвольной структуры. К таким документам относятся web-страницы, электронная почта, нормативные документы и т. п. В общем случае такие документы могут быть сложными и большими по объему и включать в себя не только текст, но и графическую информацию. Документы, использующие язык расширяемой раз-

метки *XML* и другие подобные соглашения по структуре формирования текста принято называть полуструктурированными документами. Они также могут быть обработаны методами *Text Mining*.

Алгоритм анализа текстовых документов следующий:

1) поиск информации: на данном этапе определяется, какие документы должны быть подвергнуты анализу. Как правило, при их большом количестве приходится использовать автоматизированный отбор по заданным критериям;

2) предварительная обработка документов: удаление лишних слов и придание тексту более строгой формы;

3) выделение ключевых понятий (сущностей), над которыми в дальнейшем будет выполняться анализ;

4) применение методов Text Mining: извлекаются шаблоны и отношения, имеющиеся в текстах;

5) интерпретация результатов: она заключается в представлении результатов на естественном языке или в их визуализации;

6) визуализация также может быть использована как средство анализа текста. Для этого извлекаются ключевые понятия, которые и представляются в графическом виде. Это помогает пользователю быстро идентифицировать главные темы и понятия, а также определять их важность.

В настоящее время в литературе описано много прикладных задач, решаемых с помощью анализа текстовых документов. Это и такие классические для *Data Mining* задачи, как классификация и кластеризация, и характерные только для текстовых документов задачи: автоматическое аннотирование, извлечение ключевых понятий и др.

Классификация — стандартная задача из области *Data Mining*. Ее целью является определение для каждого документа одной или нескольких заранее заданных категорий, к которым он относится. Особенностью задачи классификации является предположение, что множество классифицируемых документов не содержит «мусора», т. е. каждый из документов соответствует какой-нибудь заданной категории. Частным случаем классификации является задача определения тематики документа.

Большинство методов классификации текстов так или иначе основаны на предположении, что документы, относящиеся к одной категории, содержат одинаковые признаки (слова или словосочетания), и наличие или отсутствие таких признаков в документе говорит о его принадлежности или непринадлежности к той или иной теме. Задача методов классификации состоит в том, чтобы наилучшим образом выбрать такие признаки и сформулировать правила, на основе которых

будет приниматься решение об отнесении документа к рубрике. Необходимо подчеркнуть, что основанием классификации текстовых документов являются именно эти наборы признаков, в то время как в *Data Mining* — наборы атрибутов.

Цель кластеризации документов — это автоматическое выявление групп похожих по смыслу документов из заданного набора. Отметим, что группы формируются только на основе попарной похожести описаний документов, и никакие характеристики этих групп не задаются заранее.

Автоматическое аннотирование позволяет сократить текст, сохраняя его смысл. Решение этой задачи обычно регулируется пользователем при помощи определения количества извлекаемых предложений или процентом извлекаемого текста по отношению ко всему тексту. Результат включает в себя наиболее значимые предложения в тексте.

Первичной целью извлечения ключевых понятий является идентификация фактов и отношений в тексте. В большинстве случаев такими понятиями являются имена существительные и нарицательные: имена и фамилии людей, названия организаций и др. Алгоритмы извлечения понятий могут использовать словари, чтобы идентифицировать некоторые термины и лингвистические шаблоны для определения других понятий.

Навигация по тексту позволяет пользователям перемещаться по документам относительно тем и значимых терминов. Такая возможность достигается за счет идентификации ключевых понятий и некоторых отношений между ними.

Анализ трендов позволяет выявить тренды (тенденции) в наборах документов на какой-то период времени. Тренд может быть использован, например, для обнаружения изменений интересов компании от одного сегмента рынка к другому.

Поиск ассоциаций также является одной из основных задач *Data Mining*. Для ее решения в заданном наборе документов выявляются ассоциации между ключевыми понятиями.

Существует достаточно большое количество разновидностей перечисленных задач, а также методов их решения, что еще раз подтверждает значимость анализа текстов.

Интересным примером реализации технологии *Text Mining* является создание интеллектуальных систем, способных на имеющейся базе данных судебных документов выявлять общие зависимости, предоставлять судьям для ознакомления близкие по тематике дела, рекомендовать наиболее вероятные исходы или помечать важные места, на которые судебным работникам следует обращать внимание при процессуальных

действиях. Подобная система может помочь участникам судебного процесса точнее оценивать свои позиции или выбирать лучшую стратегию поведения, а судьям — в сжатые сроки формировать подборку связанных документов, не тратя для вынесения вердикта лишнего времени на поиск во всем архиве документов. Так, в Приволжском федеральном университете разрабатывается информационная система, получившая название «Робот-юрист». Она должна позволить участникам юридического процесса правильно проводить подготовку дела, а также осуществлять планирование судебной деятельности. Эта система ориентирована на арбитражные суды, занимающиеся рассмотрением споров в сфере предпринимательства[9].

Достижение поставленных целей предполагает:

— формирование шаблонов исковых заявлений с отслеживанием их жизненного цикла;

— разметку и анализ существующей базы судебных решений, исковых заявлений (классификация заявлений и решений, извлечение сущностей и фактов);

— подбор аналогичных дел и решений;

— сопоставление исковых заявлений и судебных решений;

— распределение судебных дел между судьями с учетом их специализации и текущей загрузки;

— прогнозирование вероятного решения по предоставленным исходным данным.

Разметка существующего массива документов необходима для дальнейшего обучения сервисов системы. Разметка текстов судебных дел важна для выделения классов и подклассов сущностей, их зависимостей в целях дальнейшего построения модели машинного обучения. Размеченный текст далее используется для обучения подсистемы поиска аналогов и прогнозирования вердикта по делу.

Одной из важнейших задач информационной системы «Робот-юрист» являются поиск и предоставление аналогичных решений по идентичным судебным искам. Предназначение соответствующего сервиса, входящего в состав системы: формировать модель предметной области на основе массива документов (включая подготовительные операции —

[9] Зуев Д. С., Марченко А. А., Хасьянов А. Ф. Применение инструментов интеллектуального анализа текстов в юриспруденции. // Труды XIX Международной конференции «Аналитика и управление данными в областях с интенсивным использованием данных» (DAMDID/ RCDL'2017), Москва, Россия, 10–13 октября 2017 года. URL: http://ceur-ws.org/Vol-2022/paper35.pdf (дата обращения: 23.03.2019).

приведение к векторному виду, кластеризацию и т. п.); получать входной документ и выдавать список документов, близких к нему. Алгоритм работы сервиса можно разделить на два этапа. На подготовительном этапе обрабатываются все имеющиеся документы: вырезаются знаки пунктуации, термы приводятся к единому виду (для слов с разными окончаниями и суффиксами). Далее каждый документ приводится к векторному виду (ему в соответствие ставится числовой вектор, учитывающий частоту использования каждого слова из некоторого набора, причем количество слов в наборе определяет размерность вектора).

На основном этапе работы на вход сервису подается идентификатор документа. Производится приведение его к векторной форме, которая обрабатывается моделью, причисляется к определенному кластеру. На выходе алгоритм выдает первые *n* документов из того же кластера, что и входной документ. Процесс переобучения модели следует проводить периодически либо после существенного изменения всего корпуса документов. Обработка массива из 3250 документов занимает 5 минут.

Аналогичный алгоритм используется и в процессе определения категории и характера спора. Правильное решение этой задачи влияет на назначение судьи на соответствующий процесс; назначаемый судья должен иметь максимальный опыт рассмотрения подобных споров. Общий алгоритм обработки документа выглядит следующим образом: на вход подается идентификатор документа; из документа выделяются ключевые слова и их количество; проводятся анализ и подбор класса дела; алгоритм возвращает идентификатор класса судебного дела, который становится дополнительным свойством (атрибутом) документа. При добавлении нового класса проводятся анализ допустимых ключевых слов и повторное обучение нейронной сети.

§ 6. Некоторые программные средства

Ранее мы уже упоминали ряд программ, предоставляющих инструментарий для интеллектуальной обработки данных. В этом параграфе мы рассмотрим некоторые отечественные программные продукты, как коммерческие, так и имеющиеся в свободном доступе.

Информационно-аналитическая система «Арион» (платная программа) позволяет выполнять комплексную обработку входных данных в разных форматах из различных источников и извлекать из них полезные знания. Работает с неформализованной информацией, содержит средства визуализации и набор готового инструментария для работы

с различными типами информационных ресурсов. Предназначена для решения следующих классов информационно-аналитических задач:

— аналитическая обработка обращений граждан и организаций;

— эффективная организация расследования происшествий;

— сбор и ведение досье на объекты учета;

— оперативная обработка и мониторинг материалов СМИ;

— анализ деятельности организации;

— формирование онтологий предметных областей;

— выделение значимых материалов из больших информационных массивов;

— оперативная подготовка ответов на запросы руководства.

Не рассматривая подробно все функции информационно-аналитической системы «АРИОН», отметим, что ее ключевой возможностью является проблемный анализ текстов на естественном языке, т. е. извлечение из текстов сведений об интересующих объектах, фактах и событиях. Полученные таким образом сведения представляются в формализованной форме в виде объектов предметной области и связей между ними (семантический анализ), после чего поступают на обработку традиционными методами в зависимости от текущих задач. Для реализации этой возможности «АРИОН» содержит лингвистический процессор. Результатом работы процессора является набор объектов и связей между ними, который традиционно представляют в виде так называемой семантической сети. Каждый объект имеет набор атрибутов (например, «Имя», «Фамилия» и «Дата рождения» для объекта «Человек»), заданных в рамках описания предметной области.

Рис. 56. Выделение сущностей («Арион»)

Система эффективно реализует необходимый при расследовании происшествий контекстный анализ объектов, поиск цепочек связей, похожих происшествий и ситуативный анализ.

Важной функцией системы является ведение досье в целях оперативного отслеживания всех материалов, связанных с данным объектом. Для построения досье сначала выбирается объект учета (как правило, лицо или организация) и выполняется сбор информации, связанной с данным объектом на заданную глубину. Построение досье может выполняться как по всему объему имеющихся данных, так и с определенными ограничениями (за некоторый промежуток времени и т. п). Процедура ведения досье предполагает регулярное обновление построенного досье в режиме мониторинга. Это позволяет отслеживать динамику изменения информационной карты вокруг объекта, на который ведется досье.

С практической точки зрения важной является возможность представить отчетные материалы (досье) в течение нескольких минут по запросу руководителя. На практике при работе с досье обычно также используются такие режимы обработки информации, как построение информационных подборок и построение аналитических отчетов.

Программа **«Доктор Ватсон»** (бесплатная программа) предназначена для исследования массивов текстовой информации в целях выявления сущностей и связей между ними. В качестве источников информации используются файлы форматов: .doc, .docx, .rtf, .txt, .html, .odt, .pdf. Результатом работы является отчет об исследуемом объекте. Программа может быть полезна всем, кто работает с большими массивами текстовой информации, нуждающейся в структурировании: аналитикам, работающим с текстовыми данными, специалистам служб безопасности, конкурентной разведки, маркетинга, PR, журналистам, детективам, политтехнологам. Программа имеет полнофункциональный демонстрационный режим, имеющий единственное ограничение — результат нельзя сохранить.

Общий алгоритм использования программы следующий:

1) загрузить анализируемые тексты;

2) запустить анализ, после чего программа выведет списки сущностей и связей, которые смогла сама распознать;

3) добавить сущности и связи, которые программа не распознала, или о которых знает только пользователь;

4) указать, что именно нужно отразить на диаграмме, скорректировать при необходимости;

5) сформировать отчет, указав, какие блоки и в какой последовательности интересуют пользователя.

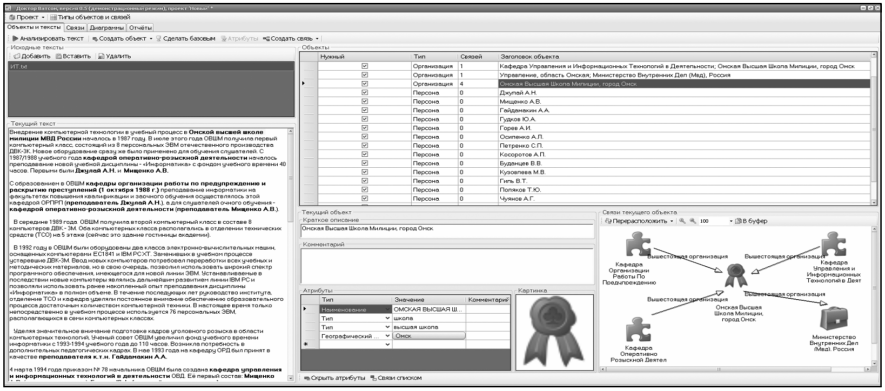

Рис. 57. Выделение сущностей («Доктор Ватсон»)

На рисунке 57 представлен экран программы, в которой открыт проект с историей кафедры информационных технологий. Как видим, в окне «Объекты» перечислены объекты, автоматически идентифицированные программой в результате анализа теста: названия организаций (подразделений) и имена преподавателей. Кроме того, для выделенного объекта (Омская высшая школа милиции) в окне «Связи текущего объекта» верно указаны связи между объектами и определен тип отношения «Вышестоящая организация».

Вообще программой предусмотрены всего два типа сущностей: персоны и организации.

Персоны характеризуются следующими атрибутами: фамилия, имя, отчество, дата рождения, дата смерти, контакт, адрес, паспорт, ИНН. Перечень атрибутов может быть расширен пользователем.

Организации по умолчанию имеют такие атрибуты: наименование, тип, номер, географический объект, эпоним, ИНН. Этот перечень также можно дополнить самостоятельно.

Если программа не распознала необходимый объект как сущность, то можно либо создать его «с нуля» (кнопка «Создать объект»), либо выделить упоминание объекта в исследуемом тексте и вызвать правой клавишей мыши контекстное меню, в котором выбрать вид сущности (персона или организация) и далее заполнить необходимые для нее атрибуты.

Все данные, введенные на персоны и организации, можно редактировать.

Кроме сущностей программа распознает пять типов связей между объектами: родственный, вышестоящая организация, учеба, владение и работа. Этого, как правило, недостаточно, поэтому предусмотрена воз-

можность создавать нужные для работы типы как объектов, так и связей (кнопкой «Типы объектов и связей» в верхней части экрана).

Если программа не распознала необходимую связь, ее можно ввести вручную (пункт «Редактировать связь» на вкладке «Связи»), либо, как и при создании объекта, выделить необходимый объект в исследуемом тексте и в контекстном меню выбрать вид связи.

Вкладка «Диаграммы» предназначена для визуализации связей между объектами. Здесь отображаются список диаграмм в проекте (их может быть много), список объектов проекта и визуализируется та диаграмма, которая выделена в списке. Объекты на диаграмме можно свободно перемещать без разрыва связей. Если нужно добавить какой-либо объект на диаграмму, то необходимо выделить его в списке левой кнопкой мыши и «перетащить» в поле диаграммы. Доступно три способа автоматического размещения: в виде графа, иерархически и прямоугольно. На рисунке 58 приведен пример размещения объектов в виде графа.

Вкладка «Отчеты» формирует отчет об объекте, который выбран в качестве базового. Для создания отчета прежде всего создается его структура в виде иерархии информационных блоков (узлов). В результате получается древовидная структура, похожая на структуру файловой системы, с тем отличием, что здесь имеется не два типа объектов (файлы и папки), а четыре. Каждый блок специализирован и содержит либо атрибуты объекта исследования, либо его связи, либо диаграммы, а также может быть контейнером для блоков первых трех типов. Имена блоков являются заголовками разделов отчета. В формируемом отчете будет отображаться только та информация, которая соответствует настройкам для выбранных блоков.

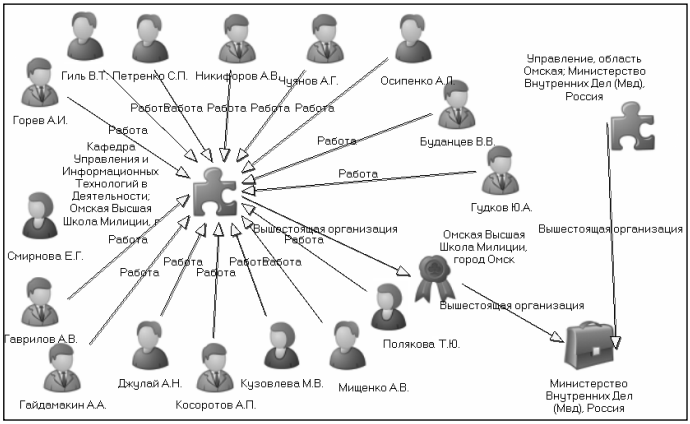

Рис. 58. Выделение сущностей («Доктор Ватсон»)

Следует отметить, что программа осуществляет не формальный, а именно интеллектуальный анализ текста. В частности, в приведенном примере из текста не только правильно были извлечены сущности, представляющие названия кафедр, учебных заведений и министерств, но и точно установлены иерархические связи между ними, хотя в тексте они явно не выражены. Были идентифицированы и имена сотрудников, причем даже их пол был определен программой (обратите внимание на иконки, которые предложила программа; в сомнительных случаях программа установила иной вид иконок). С другой стороны, отношение «Работа», представленное на рисунке 58, программе выявить не удалось, и эту связь пришлось добавлять вручную.

Система «***TextAnalyst 2.0***» (доступная для бесплатной загрузки) решает следующие задачи *Text Mining*:

— создает семантическую сеть большого текста;

— подготавливает резюме текста;

— осуществляет поиск по тексту;

— автоматически производит классификацию и кластеризацию текстов.

Навигация в текстовых массивах осуществляется на основе гиперссылок по ключевым словам семантической сети на те предложения в документе, которые содержат необходимые комбинации слов. Отдельные предложения могут иметь, в свою очередь, гиперссылки на те места в исходном тексте, где они были обнаружены. С помощью подключения пользовательских словарей (включаемых и исключаемых слов) программа позволяет исследователю сконцентрироваться на изучаемом предмете. Кластеризация текстов базируется на удалении слабых ссылок в семантической сети, что приводит к разбиванию текста на семантически однородные кластеры. В продукте реализованы технологии лингвистического анализа и нейросетей.

Получив на вход текст, «*TextAnalyst*» формирует информационную модель предметной области — семантическую сеть. Каждое понятие, которое в тексте может повторяться несколько раз, в семантической сети представляется единственным элементом. К каждому понятию присоединяется список других понятий, в сочетании с которыми оно встречается в тексте, а также список предложений, в которых это понятие употребляется. Таким образом происходит сбор информации о понятии.

Каждое понятие семантической сети характеризуется числовой оценкой — смысловым весом. Эти оценки позволяют сравнить относительный вклад различных понятий в общий смысл текста. Вес может

принимать значения от 1 до 100. Максимальное значение веса понятия, равное 100, означает, что оно является ключевым в тексте. Значение близкое к единице — в тексте мало информации относящейся к данному понятию. Связи между парами понятий тоже имеют характеристику — вес связей. Вес связей также может принимать значение от 1 до 100. В программе есть возможность настраивать вид семантической сети на экране, изменяя количество отображаемых понятий и связей, а также способ их сортировки.

На рисунке 59 приведен фрагмент семантического «дерева», построенного в программе «*TextAnalyst*» для текста послания Президента Российской Федерации Федеральному Собранию Российской Федерации 1 марта 2018 г.

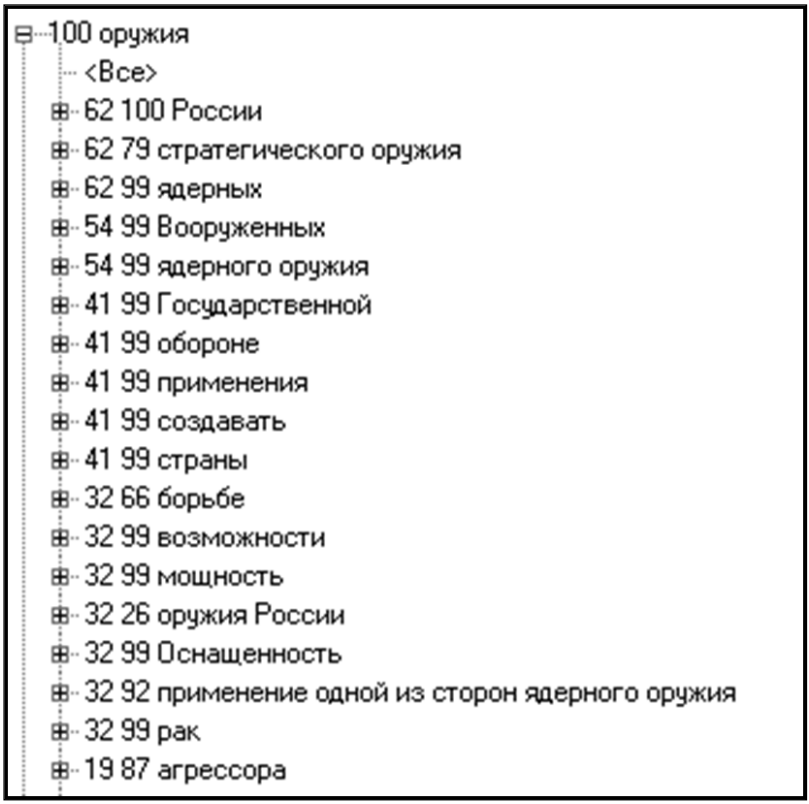

Рис. 59. Семантическое «дерево» («TextAnalyst»)

В этом семантическом «дереве» для каждого понятия приведены два числа. Первое число характеризует вес связи, второе — вес понятия.

Взглянув на этот фрагмент семантической сети, можно заметить, что основной темой выступления Президента является оружие. Так же сразу можно выделить понятия, которые связаны с оружием, имеющие с ним связи с наибольшим весом:

— Россия;

— стратегическое оружие;

— ядерное.

Функция автоматического реферирования порождает текст, состоящий из наиболее информативных предложений исходного текста, отражающих основные смысловые связи между главными понятиями семантической сети. Предложения реферата снабжены отсылками к соответствующим местам исходного текста, что позволяет быстро посмотреть и понять контекст текущего тезиса. В рассматриваемом случае объем исходного текста при реферировании сократился в восемь раз.

Кроме отдельного продукта «*TextAnalyst*» также предлагается дополнительная компонента «*TextAnalyst Lib*», которая может использоваться для построения гипертекстовых электронных книг. Существует также плагин «*TextAnalyst*» для браузера «*Microsoft Internet Explorer*».

«Система Юрист» (сервис платный, но можно получить пробный доступ на три дня) включает online-сервис https://dogovor.1jur.ru/ для проверки текстов договоров и анализа юридических рисков. Сервис позволяет распознавать тип договора, выделять существенные условия (даты, сроки, суммы, ответственность, риски, реквизиты контрагентов и т. п.). Его разработка является результатом сотрудничества компаний «Актион» («Система Юрист») и «АйТи» (проект «Правовая экспертиза»).

Сервис значительно упрощает процесс проведения правовой экспертизы и сокращает его сроки, выполняя рутинные операции и снижая затраты рабочего времени. Пользователи загружают договоры в «Систему Юрист», и на экране отображается вся информация, содержащаяся в документах. Система выделит ссылки, ошибки и риски, покажет связь пунктов договора с базой нормативных правовых актов, делая активными ссылки на все упоминаемые в договоре. Сервис распознает банковские реквизиты сторон, определит и проверит структуру договора. Все ключевые условия — сроки, денежные суммы, стороны договора, упоминания порядка расчетов, штрафных санкций и других важных для юриста положений договора — будут собраны

вместе с возможностью удобной навигации по данным. Сервис поддерживает умный поиск, позволяющий находить договор по названию, номеру и реквизитам контрагентов.

«Deductor Studio» (программа бесплатная) — это аналитическая платформа, основа для создания законченных прикладных решений в области анализа данных. На базе единой архитектуры можно пройти все этапы построения аналитической системы: от консолидации данных до построения моделей и визуализации полученных результатов. «Deductor Studio» — это рабочее место аналитика (приложение, позволяющее пройти все этапы построения прикладного решения).

Система позволяет анализировать любые табличные данные и для решения аналитических задач. В ней предусмотрена возможность использования следующих специальных технологий:

— создание хранилища данных — консолидация данных и обеспечение быстрого и понятного для аналитика доступа к ним;

— *OLAP* (многомерный анализ) — визуализация, отчетность и удобное манипулирование большими объемами данных;

— *Data Mining* (моделирование, прогнозирование, интеллектуальный анализ данных) — поиск скрытых закономерностей, выявление причинно-следственных связей, анализ рисков;

— обнаружение, извлечение знаний — построение сценариев обработки от очистки и предобработки данных до моделирования.

В «*Deductor*» используются мощные технологии анализа данных, но при этом акцент сделан на самообучающиеся методы, что позволяет строить системы, способные реагировать на изменение ситуации.

В главе IV мы пользовались инструментарием этой платформы для обучения нейронной сети (пример 1). Однако нейронные сети — это только один из многих интеллектуальных инструментов рассматриваемой программы. «*Deductor*» имеет также средства очистки данных, факторного анализа (понижение размерности пространства данных), корреляционного анализа (оценка зависимости выходных факторов от входных), выявления дубликатов и противоречий в табличных данных, средства фильтрации, инструменты логистической регрессии для вычисления рейтинга (вероятности того, что событие наступит для конкретного испытуемого), средства поиска ассоциативных правил и построения «деревьев» решений, а также инструменты кластеризации (карты Кохонена и метод k-средних).

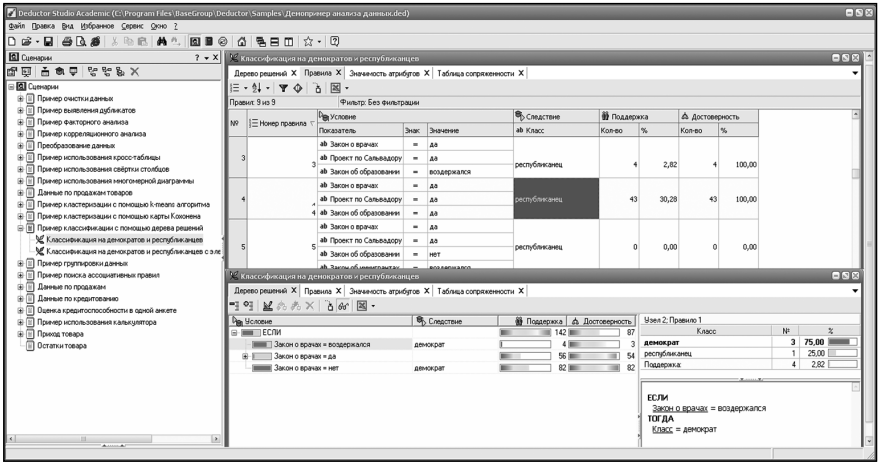

Рис. 60. «Дерево» решений («Deductor Studio»)

На рисунке 60 представлен результат работы программы при решении задачи классификации парламентариев на два класса (по партийной принадлежности) на основании таблицы данных голосования по ряду различных проектов. Верхнее окно «Правила» содержит, например, информацию о том, что согласно правилу № 4 (из девяти, автоматически сформулированных системой) положительно голосовавший за все три проекта — о врачах, об образовании и по Сальвадору — с вероятностью 30% относится к республиканцам. А нижнее окно «Дерево решений» сообщает, что отрицательно голосовавший по проекту о врачах с вероятностью 82% является демократом.

Информационно-аналитическая система **«Семантический архив»** (программа платная) предназначена для автоматизации деятельности аналитических подразделений. Она позволяет организовывать сбор текстовой информации из открытых источников (электронные СМИ, аналитические отчеты экспертов), осуществлять их автоматизированную обработку, хранение, анализ и генерацию отчетов. Как и рассмотренная выше система «Арион», «Семантический архив» предоставляет аналитикам возможность сформировать формальные досье на различные объекты мониторинга (сущности) — персоны, компании, государственные структуры, а также хранить описания их взаимоотношений и происходящих с ними событий. Часть отношений и событий могут иметь ссылки на текстовые материалы, в которых они упоминались.

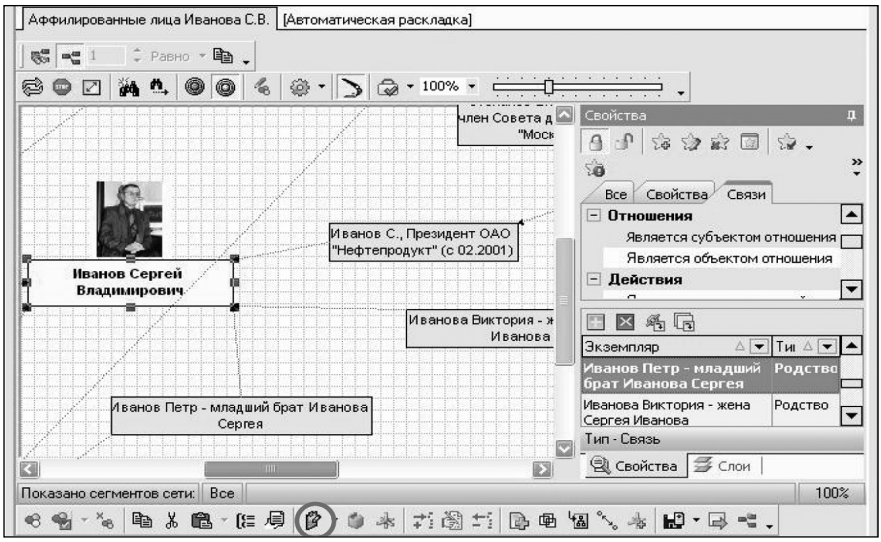

Рис. 61. Визуализация связей («Семантический архив»)

Поставляемые с системой интернет-роботы позволяют собирать новости из сети Интернет, оператор имеет возможность вставлять документы с жесткого диска и импортировать данные из внешних баз данных. В системе реализовано автоматическое выделение объектов мониторинга из текстов документов и автоматизированное (с участием оператора) выделение событий и отношений между ними. Широко представлены следующие средства визуализации:

— визуализация информации в виде таблицы документов, объектов или событий;

— визуализация параметров событий (цена акций компании, количество голосов электората) средствами бизнес-графики;

— визуализация событий и их привязка к карте;

— визуализация объектов, отношений и событий на семантической сети (сетевой вид);

— визуализация объектов, отношений и событий на семантической сети (сетевой вид).

«Логика ECM. Правовая экспертиза». Система предназначена для автоматизации процесса проведения экспертизы проектов организационно-распорядительных документов, договоров и др. Она упрощает процесс осуществления правовой экспертизы и сокращает ее сроки, помогая, например, автоматически установить:

— не содержатся ли в проверяемом документе ссылки на НПА, которые утратили силу;

— нет ли в тексте фрагментов других документов, не возникает ли избыточное дублирование нормативной документации;

— соответствуют ли оформление и структура документа установленным в организации правилам;

— нет ли ошибок в оформлении цифровой информации в договоре, соответствуют ли друг другу суммы, указанные цифрами и прописью, правильно ли рассчитан НДС и т. п.;

— выделяются ли в тексте документа упоминания структурных подразделений организации.

При обнаружении в проекте документа ссылок на внешние (российское законодательство) или внутренние (приказы, распоряжения, инструкции, регламенты, протоколы и т. п.) правовые документы система сама сформирует гипертекстовые ссылки на них. В результате пользователю будет предоставлен быстрый доступ к тексту НПА или внутреннего организационно-распорядительного документа (причем именно к той части, разделу или статье НПА, которая упомянута в тексте). Более того, выделив какой-либо фрагмент текста документа, пользователь может использовать его как полнотекстовый запрос к системе и получить перечень нормативных правовых документов, имеющих непосредственное отношение к заданной правовой теме.

Система поможет также в проведении антикоррупционной экспертизы, автоматически проверив проект НПА на наличие в нем так называемых запрещенных или нечетких комбинаций, т. е. словосочетаний типа «вправе», «в случае необходимости», «на усмотрение» и т. п.

Анализируемый документ загружается и отображается в окне *MS Office Word*, которое встраивается в клиентскую часть системы «Логика ECM. Правовая экспертиза». Замечания оформляются комментариями *MS Word* прямо в анализируемом документе. Также в нем создаются гиперссылки на внешние НПА, присутствующие в центральной базе данных. На отдельных панелях отображается структура документа, экспертные замечания и информация о «плагиатных» фрагментах. Документ может редактироваться, после чего экспертиза проводится заново.

Пользователь может выбирать правила экспертизы, а также настраивать их и задавать новые из набора, поддерживаемого системой. Правила охватывают широкий спектр потребностей, которые поддаются автоматизации: от проверки форматирования (шрифты, отступы, нумерация) до проверки семантической корректности тех или иных конструкций

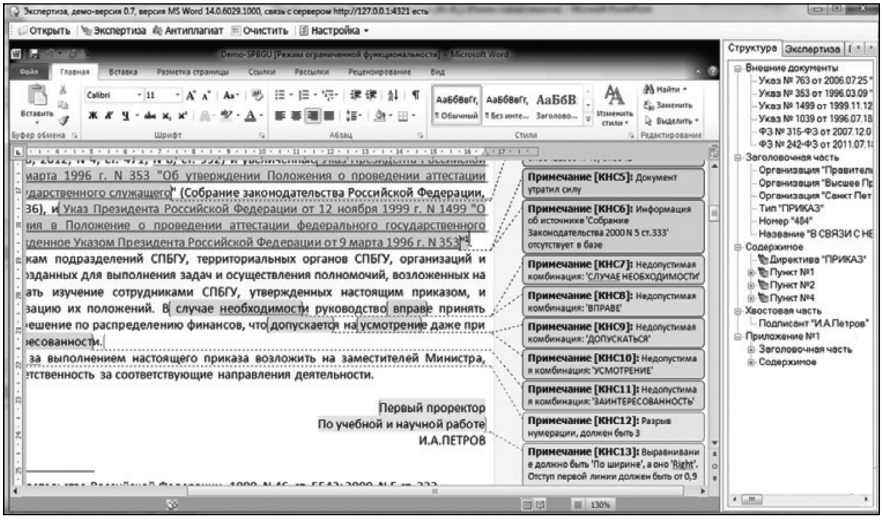

Рис. 62. Разметка документа Word («Логика ECM. Правовая экспертиза»)

и структуры документа в целом. Правила могут сохраняться в файлах и загружаться по мере необходимости.

Процесс автоматической экспертизы состоит из следующих шагов:

— из документа удаляются все комментарии и гиперссылки (результат предыдущей экспертизы);

— выделяется текст из документа;

— текст подвергается семантическому анализу для определения структуры документа и выделения некоторых семантических сущностей: ссылок на другие НПА, организации, персоны, даты и др.;

— последовательно применяются правила из набора, каждое из которых формирует свои элементы при выполнении определенных условий (к таким элементам могут относиться комментарии и гиперссылки в самом документе, а также элементы структурной панели, которая появляется справа от документа). Порядок применения правил на результат не влияет.

Система предоставляет возможность поиска похожих фрагментов анализируемого документа и документов в базе («антиплагиат»). В случае нахождения таких фрагментов список документов выводится на вкладке «Плагиат», для выбранного документа в средней панели отображается список таких фрагментов, при выборе конкретного фрагмента они подсвечиваются в анализируемом документе и в тексте документа базы.

Система «Логика ECM. Правовая экспертиза» успешно прошла тестирование в МВД России, где использовалась для сбора и агрегирования предложений по совершенствованию законодательства, регламентирующего деятельность органов внутренних дел, для правовой (в том числе антикоррупционной) экспертизы проектов НПА и поиска правовых пробелов и коллизий.

§ 7. Учебные задания

Используя данные проекта *MAP* (см. §2 главы VI, URL: www.murderdata.org) и инструментарий *Excel*, постройте график распределения по годам раскрытых и нераскрытых убийств.

1. Как изменилась раскрываемость в США за последние 40 лет? Сколько всего нераскрытых убийств зарегистрировано за тот же срок?

2. Сравните тенденции развития насильственных преступлений в разных штатах. Какой из штатов имеет существенные особенности в этом отношении?

3. Сравните долю сообщений с неполной информацией о жертвах среди мужского и женского населения.

4. Сравните долю нераскрытых преступлений среди жертв мужского и женского пола.

5. Сравните данные разных регионов о женщинах, погибших в результате удушения.

6. Найдите запись об удушении шестилетней девочки в декабре 1996 г. Раскрыто ли это дело?

7. Были ли в том же штате и в то же время другие подобные (тот же способ убийства) преступления против девочек 5–10 лет?

8. Были ли похожие нераскрытые преступления в других штатах? Сколько их и в каких штатах они произошли?

9. В отношении каких дел можно предполагать общность с делом 1996 г.?

СПИСОК РЕКОМЕНДУЕМЫХ ИСТОЧНИКОВ

1. Бондарь К. М., Юдин В. С., Рыбак А. В., Скрипко П. В. Применение современных инфокоммуникационных технологий в управлении деятельностью подразделений органов внутренних дел : учеб.-практ. пособие. М., 2018.

2. Братко И. Алгоритмы искусственного интеллекта на языке Prolog / пер. с англ. И. Братко. М., 2004.

3. Гайдамакин А. А. Новые образовательные стандарты и роль информационно-правового блока в подготовке юристов // Вестник Волгоградской академии МВД России. 2012. № 3.

4. Гайдамакин А. А. О формальном описании семантических связей в статьях Уголовного кодекса // Юристъ-правоведъ. Ростов н/Д, 2008, № 4.

5. Гайдамакин А. А. Прозрачность закона и информационно-коммуникационные технологии // Научный вестник Омской академии МВД России. 2011. № 2(41).

6. Гайдамакин А. А. Формальные модели в юридической науке и технике : монография. Омск, 2017.

7. Добровольский В. И. Как стать хорошим юристом. М., 2017.

8. Дюк В., Самойленко А. Data Mining : учебный курс. СПб., 2001.

9. Зуев Д. С., Марченко А. А., Хасьянов А. Ф. Применение инструментов интеллектуального анализа текстов в юриспруденции // Труды XIX Международной конференции «Аналитика и управление данными в областях с интенсивным использованием данных» (DAMDID/RCDL'2017), Москва, Россия, 10–13 октября 2017 года. URL: http://ceur-ws.org/Vol-2022/paper35.pdf.

10. Бельков В. А., Алдашкина А. С. Использование специальных программ для установления серийности при расследовании убийств // Пролог: журнал о праве. 2017. № 3.

11. Курносов Ю. В., Конотопов П. Ю. Аналитика: методология, технология и организация информационно-аналитической работы. М., 2004.

12. Латышева А. М. Big Data. Актуальность и перспективы использования // Молодежный научно-технический вестник. URL: http://sntbul.bmstu.ru/doc/724143.html.

13. Нефедов С. Н., Пархименко В. А., Татур М. М. Применение методов интеллектуального анализа данных в криминалистике и судебной экспертизе // Вопросы криминологии, криминалистики и судебной экспертизы. 2017. № 2.

14. Ольков С. Г. Аналитическая юриспруденция : учебник. Сургут, 2012.

15. Осипов Г. С. Методы искусственного интеллекта. М., 2011.

16. Парамонов О. Большие данные на службе полиции (и преступников). URL: http://www.computerra.ru/228030/crime-bigdata.

17. Панов Д. В. Решатель юридических проблем: скорая правовая помощь на все случаи жизни. М., 2012.

18. Пароннджанов В. Как улучшить работу ума. Алгоритмы без программистов — это очень просто! М., 2001.

19. Питиля Д. А., Рожкова А. О. Средства визуализации данных Gephi и Google в экономических исследованиях // Молодой ученый. 2016. № 12.

20. Чернышова Г. Ю. Интеллектуальный анализ данных : учеб. пособие. Саратов, 2012.

21. Чугаева Т. В. Поиск связей между сущностями в криминалистическом анализе источников данных : магистер. дис. … СПб., 2016.

ОГЛАВЛЕНИЕ

Учебное издание

Гайдамакин Андрей Андреевич

ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ В ЮРИДИЧЕСКОЙ АНАЛИТИКЕ

Редактор Е. А. Жукова
Корректор Л. И. Замулло
Технический редактор П. В. Ярославцева
ИД № 03160 от 02 ноября 2000 г.
Подписано в печать 04.12.2019. Формат 60×84/16. Бумага офсетная № 1.
Усл. печ. л. 7,6. Уч.-изд. л. 6,6. Тираж 100 экз. Заказ № 212.

Редакционно-издательский отдел
Отделение полиграфической и оперативной печати
644092, г. Омск, пр-т Комарова, д. 7