

**ПОИСК АНАЛОГОВ С ПОМОЩЬЮ РАСПРЕДЕЛЕННЫХ ПРЕДСТАВЛЕНИЙ**

Рассматриваются модели первой стадии рассуждений по аналогии: поиск в памяти аналога некоторой ситуации. Ситуации и аналоги иерархически структурированы и могут включать отношения высших порядков, что усложняет поиск при использовании традиционных подходов. Приводятся схемы распределенного представления аналогов в виде многомерных бинарных векторов. Показано, что степень сходства ситуаций можно оценить по величине скалярного произведения представляющих их векторов. Это создает основу для моделирования процессов поиска аналогов людьми и для более эффективного поиска в базах знаний.

**Введение**

Рассуждения по аналогии – это процесс восприятия общности отношений, существующих в ситуациях (предметных областях, эпизодах), которые могут выглядеть непохожими при поверхностном рассмотрении [1-3]. Известным примером использования в науке является аналогия с солнечной системой, которую применил Резерфорд при создании планетарной модели атома. Рассуждения по аналогии являются одним из важнейших процессов в разумной деятельности людей, а их моделиро-

вание представляет собой одну из самых интересных задач искусственного интеллекта [1, 4-6].

Рассуждения по аналогии предполагают оперирование сложной структурированной информацией. В подходах [1-3, 7, 8] и других аналогии являются структурированными высказываниями, содержащими компоненты разного уровня сложности – признаки, объекты, отношения разного порядка, подструктуры и т.д. (рис. 1). Аналогии сравнивают по "поверхностному сходству", основанному на наличии общих элементов

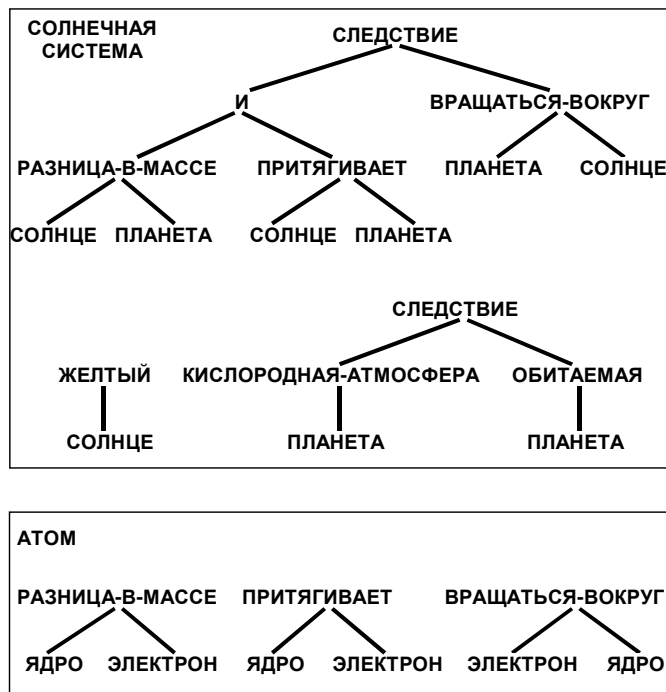


Рис. 1. Графическое представление аналогии Резерфорда "солнечная система – атом"

[1, 9], или по "семантическому сходству" этих элементов [2, 10], основанному на их некоторой априорной семантической близости (например, принадлежности к одному таксономическому классу). Однако не менее важно структурное сходство, которое определяется тем, как элементы аналогов группируются друг относительно друга. Структурное сходство основано на понятии "структурной согласованности" [3, 11] или "изоморфизма" [2, 10]. Эти обнаруженные психологами ограничения на структурное сходство, которые демонстрируют люди при сравнении эпизодов, предполагают взаимно однозначное соответствие (не более одного элемента в одном аналоге соответствует некоторому элементу в другом) и параллельную связность (соответствующие отношения должны иметь соответствующие аргументы). Насколько жесткие эти ограничения, по-прежнему является открытым вопросом [2, 7, 10].

В рассуждениях по аналогии первыми тремя стадиями обычно считают [1, 2, 9-11] поиск аналогов, установление соответствия между ними и вывод по аналогии. Поиск (извлечение из памяти, распознавание или доступ – *analogical access*) – процесс обнаружения в памяти наиболее подходящего аналога ко входному. Установление соответствия (отображение – *analogical mapping*) – процесс обнаружения соответствий между компонентами двух аналогов. Вывод по аналогии – процесс переноса знаний от одного аналога к другому. Все эти стадии требуют обработки структурированной информации, содержащейся в представлении аналогов.

Реализация и эффективность операций над структурированной информацией (таких, как оценка сходства, сравнение, нахождение соответствующих элементов, об-

ход, и др.) существенно зависит от используемой схемы ее представления. До недавнего времени представление и обработка структур осуществлялись только в рамках символьных или локальных представлений. Естественно, что именно они использованы в наиболее значимых вычислительных моделях рассуждений по аналогии – например, ARCS-ACME [9-11], см. также ссылки в [12].

Из-за ограничений используемых методов представления информации в этих моделях затруднены представление и учет семантического сходства компонентов, а применяемые алгоритмы требуют значительных затрат вычислительных ресурсов. Кроме того, имеются проблемы с объяснением возможности реализации таких моделей в нейронных сетях мозга, в то время как использование принципов и методов, на которых базируется поиск аналогов в нейробиологических структурах мозга человека, может позволить создать модели рассуждений по аналогии, приближающиеся по свойствам к рассуждениям людей.

Необходимость расширения семантического базиса представления аналогов, масштабирования подходов на случаи, когда имеется большое количество потенциальных аналогов, и повышения степени нейробиологической релевантности моделей привела к попыткам использования в моделях аналогии распределенного представления информации (например, LISA [2], Drama [8], STAR2). Однако их использование носило непоследовательный характер как из-за недостатков использованных схем для представления информации о структуре, так и из-за недостаточной проработанности подходов. В связи с этим перспективными для моделирования рассуждений по аналогии являются

появившись недавно новые типы распределенных представлений (HRR [13], BSC [6], АПНС [14]).

В данной статье рассматривается подход к представлению структурированной информации с помощью специального типа распределенных представлений – многомерных разреженных бинарных векторов. Такие схемы разрабатываются в рамках архитектуры ассоциативно-проективных нейронных сетей (АПНС). Они позволяют представить эпизоды-аналоги (или другие сложные структурированные описания) в виде "кодвекторов", величина корреляции которых, измеряемая скалярным произведением, отражает степень сходства эпизодов, оцениваемую психологами в экспериментах с людьми. Это позволяет упростить поиск аналогов и открывает возможности преодоления других недостатков традиционных подходов к моделированию рассуждений по аналогии.

**1. Традиционные модели поиска аналогов**

Большинство моделей поиска аналогов основано на *символьном* или *локальном* представлении информации [15, 16]. В них каждый новый элемент или их комбинация требуют выделения нового "узла" ("вершины" или "нейрона" в нейросетевых локальных представлениях) или ячейки памяти (в традиционных компьютерных символьных представлениях). Представление иерархических структур, каковыми являются аналоги – в общем случае, направленные ациклические графы (например, [17]), – строится с помощью семантических сетей или их символьных вариантов. Узлы более высокого иерархического уровня, соответствующие более сложным структурам, соединены с узлами более низких иерархических уровней, представляющими их подструктуры разной сложности, вплоть до неразложимых элементов нижнего уровня (рис. 2). Такие "растущие" сети локальных элементов разрабатываются рядом исследователей, например [5, 18, 19].

Для оценки сходства струк-

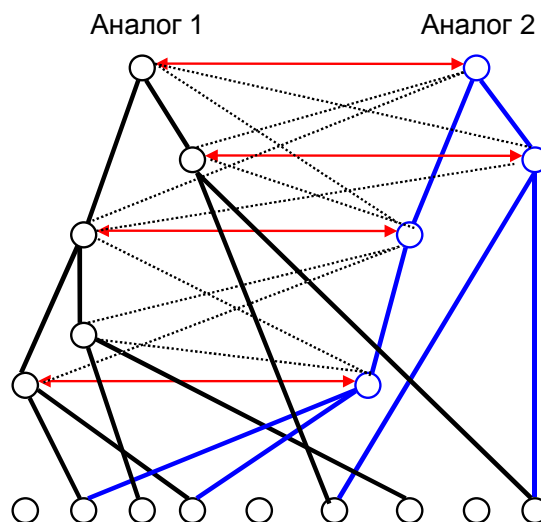


Рис. 2. Локальное представление аналогов. Аналоги являются иерархическими структурами. Определение сходства требует вычислительно сложного сопоставления их элементов. Стрелками обозначены наилучшие соответствия между элементами аналогов, а пунктирными линиями – некоторые другие возможные кандидаты на соответствия. Линиями без стрелок показаны отношения «часть-целое» между элементами раз-

турированной информации иногда привлекают подходы, использующие понятие частичного изоморфизма ([20–22] и др.). Его выявление для графов является сложной задачей. Однако изоморфизм не учитывает сходство самих элементов и не отражает особенностей оценки сходства, обнаруженных психологами в рассуждениях по аналогии у людей [2, 23].

В других существующих моделях поиска аналогов такие особенности учитывают. Используемые для этого методы включают построение и эволюцию (стабилизацию) нейроподобных сетей, удовлетворяющих ограничениям (constraint-satisfaction networks), которые находят наилучшее соответствие между элементами аналогов ([8], см. также ссылки в [12]), либо символьную обработку виртуальных сетей, представляющих эти соответствия [11], или использование "отображающих связей" между "динамическими связками" [16], представленными одновременной активацией узлов в локальной сети (см. рис. 2). Однако, несмотря на налагаемые ограничения, оценка сходства структур остается очень "дорогой" в вычислительном отношении процедурой.

Существуют различные точки зрения об относительной роли поверхностного и структурного сходства в доступе к аналогам в памяти. Хотя структурное сходство при поиске аналогов считается менее важным, чем в отображении, модели, которые принимают во внимание только поверхностное сходство, считаются неадекватными [2, 3, 9, 10].

Таким образом, модели поиска аналогов, разработанные на базе локальных и символьных представлений, в общем случае требуют сложного сравнения входного (це-

левого) аналога с (базовыми) аналогами в долговременной памяти. При этом каждое сравнение включает вычислительно дорогую стадию нахождения соответствия подструктур. Так как предполагается, что в памяти может храниться много эпизодов, процесс поиска аналогов может быть недопустимо долгим и следует использовать способы ограничения количества кандидатов на сравнение. Одним из таких способов является общая стратегия введения двухступенчатого поиска аналогов в памяти, используемая в MAC/FAC [9] и ARCS [10].

На первой стадии для выбора кандидатов используется вычислительно несложный процесс, основанный только на наличии идентичных объектов или признаков. Выбираются кандидаты, которые имеют ряд общих с входным аналогом элементарных объектов без учета их структуры и отношений.

Вторая стадия требует нахождения детального сходства с учетом структуры и выполнения сложных вычислений, чтобы оценить все аспекты сходства между входным эпизодом и кандидатами. Хотя двухступенчатая схема обнаружения наилучшего соответствия и обеспечивает экономию вычислений, эта экономия недостаточна. Кроме того, на первой стадии есть риск пропустить потенциально лучшего кандидата.

## **2. Распределенное представление и обработка структур в АПНС**

В полностью распределенных представлениях [15, 24] любой объект представлен как кодвектор, каждый элемент которого можно рассматривать как соответствующий состоянию некоторого "нейрона". Для похожих объектов естественно использовать коррелированные кодвекторы и оценивать их сходство скалярным про-

изведением. Однако ранее предполагалось, что иерархические структуры нельзя представить распределенным образом из-за "катастрофы суперпозиции" — потери информации о размещении объектов в структурах [16].

В ряде распределенных схем, таких, как АПНС [14, 16], HRR [24], BSC [25], для решения этой проблемы были предложены процедуры связывания (аналог скобок в символьных представлениях [16, 26]). В этих схемах оказалось возможным создавать распределенные представления иерархических структур с помощью кодвекторов одинаковой размерности.

На сходство кодвекторов, кодирующих эти структуры, влияют как набор компонентов, так и отношения между ними. Сходные по набору и группировке компонентов структуры дают сходные кодвекторы [16, 26]. Поэтому, в отличие от традиционных методов, нет необходимости явно определять соответствие между элементами (подструктурами) двух структур, чтобы оценить их полное сходство. Достаточно найти перекрытие их кодвекторов. Другое достоинство — это то, что сходство элементарных компонентов может быть не типа "все или ничего", а градуальным.

Кратко рассмотрим разрабатываемую нами в рамках архитектуры ассоциативно-проективных нейронных сетей (АПНС) схему распределенного представления и обработки иерархически структурированной информации. Более детальное обсуждение и сравнение с другими схемами распределенного представления структур приведено в [14, 16]. Применению этой схемы для моделирования поиска аналогий посвящен раздел 3.

**2.1. Ассоциативно-проективные нейронные сети.** В АПНС объекты (информационные

единицы) любого уровня иерархии, элементарные или составные, представлены длинными кодвекторами одинаковой размерности. Кодвекторы — это двоичные и разреженные (с малой долей единиц  $p=M/N$ ) векторы, например, с  $N=100,000$  элементов, представляющих нейроны, и  $M=1,000$  единиц, представляющих активные нейроны. Асимметричное нормированное перекрытие единиц кодвекторов может быть измерено как доля совпадающих единиц  $O(\mathbf{X},\mathbf{Y}) = |\mathbf{X}\wedge\mathbf{Y}|/|\mathbf{X}|$ , где  $|\mathbf{X}|$  — количество единиц в кодвекторе  $\mathbf{X}$ .

Векторы конструируют таким образом, чтобы величина их перекрытия отражала степень сходства объектов. непохожие объекты представляют независимыми псевдослучайными векторами, что для разреженного случая дает малое перекрытие  $p^2$ , а похожие объекты — кодвекторами с большей величиной перекрытия единиц, чем случайное.

Для иерархических структур формирование кодвекторов осуществляется следующим образом. Обычно элементы структур некоторого уровня иерархии являются совокупностями небольшого количества (под)структур — компонентов этих сложных структур с нижележащих уровней. Кодвекторы более сложных структур строятся из множества кодвекторов их компонентов. Поскольку каждый компонент может быть сложной составной (под)структурой, можно представлять структуры произвольной сложности. Чтобы сохранить информацию о группировании компонентов, в АПНС используется связывание их кодвекторов с помощью контекстно-зависимого прореживания (context-dependent thinning, CDT [16, 26]). Одна из версий CDT описана ниже (обсуждение различных версий и реализаций CDT приведено в [16]).

**2.2. Процедура контекстно-зависимого прореживания.** Возьмем побитовую дизъюнкцию  $S$  ( $S = 2, \dots, 5$ ) псевдослучайных кодвекторов  $X_s$  компонентов, которые должны быть связаны:

$$\mathbf{Z} = \bigvee_{s=1}^S \mathbf{X}_s. \quad (1)$$

Сформируем прореженный код-вектор  $\mathbf{Z}$  как

$$\langle \mathbf{Z} \rangle = \bigvee_{k=1}^K (\mathbf{Z} \wedge \tilde{\mathbf{Z}}(k)) = \mathbf{Z} \wedge \bigvee_{k=1}^K \tilde{\mathbf{Z}}(k). \quad (2)$$

Здесь  $\tilde{\mathbf{Z}}(k)$  — это  $\mathbf{Z}$  с переставленными элементами. Каждая  $k$ -я перестановка должна быть фиксированной, уникальной и независимой. Случайные перестановки были бы идеальным вариантом, однако и перестановки циклическим сдвигом со случайным количеством сдвигов достаточно удобны в приложениях. Число  $K$  дизъюнктивно наложенных векторов с переставленными элементами выбирается так, чтобы количество единиц в  $\mathbf{Z}$  было приблизительно таким, как требуется — обычно, сравнимым с плотностью кодвекторов компонентов  $\mathbf{X}_s$ .

Возможно много "конфигураций" прореживания (путем дизъюнкции различных перестановок  $\mathbf{Z}$ ). Обозначим разные конфигурации прореживания разными метками (например, 1, 2, 5, и) как верхний индекс левой угловой скобки (например,  $^1\langle \mathbf{Z} \rangle$ ,  $^2\langle \mathbf{Z} \rangle$ ,  $^5\langle \mathbf{Z} \rangle$ ,  $^i\langle \mathbf{Z} \rangle$ ).

Прореженные кодвекторы похожих наборов компонентов похожи друг на друга:  $\langle \mathbf{A} \vee \mathbf{B} \vee \mathbf{C} \rangle$  похож на  $\langle \mathbf{A} \vee \mathbf{B} \vee \mathbf{D} \rangle$ . Прореженные кодвекторы похожи на каждый из кодвекторов компонентов:  $\langle \mathbf{A} \vee \mathbf{B} \vee \mathbf{C} \rangle$  похож на  $\mathbf{A}$ . Однако представление (подмножество единиц)  $\mathbf{A}$  в  $\langle \mathbf{A} \vee \mathbf{B} \vee \mathbf{C} \rangle$  отличается от подмножества единиц  $\mathbf{A}$  в  $\langle \mathbf{A} \vee \mathbf{B} \vee \mathbf{D} \rangle$ .

### 2.3. Иерархическая память.

В АПНС, все кодвекторы всех уровней иерархии хранятся в иерархической памяти [14]: соответствующие объектам одного уровня — в одном массиве, а разных уровней — в разных массивах памяти. Например,  $\mathbf{A}$ ,  $\mathbf{B}$ ,  $\mathbf{C}$ ,  $\mathbf{D}$  и другие кодвекторы одной сложности хранятся на одном уровне, тогда как  $\langle \mathbf{A} \vee \mathbf{B} \vee \mathbf{C} \rangle$ ,  $\langle \mathbf{A} \vee \mathbf{B} \vee \mathbf{D} \rangle$  и т.д. — в массиве памяти более высокого уровня. Память на каждом уровне должна выполнить поиск самого близкого кодвектора ко входному. Заметим, что на разных иерархических уровнях используются различные конфигурации прореживания, а конфигурация прореживания, "закрепленная" за некоторым массивом памяти, постоянна.

Такое устройство памяти позволяет выполнить обход уровней иерархии представлений, используя сходство кодвекторов, следующим образом. Пусть имеется кодвектор некоторого иерархического уровня — назовем его "пробом". По пробу можно найти похожий кодвектор в массиве того же уровня иерархии (например,  $\langle \mathbf{A} \vee \mathbf{B} \vee \mathbf{C} \rangle$  по  $\langle \mathbf{A} \vee \mathbf{B} \vee \mathbf{D} \rangle$ ). Можно также найти компоненты проба ("декодировать проб") путем его подачи в массивы более низкого уровня и выбора кодвекторов компонентов, имеющих максимальное перекрытие с пробом (например,  $\mathbf{A}$ ,  $\mathbf{B}$ ,  $\mathbf{C}$  по  $\langle \mathbf{A} \vee \mathbf{B} \vee \mathbf{C} \rangle$ ). В свою очередь, они могут быть декодированы через компоненты более низкого уровня (если таковые есть) и т.д. Можно также найти более сложные кодвекторы, которые содержат проб в качестве своих компонентов, путем поиска сходных с пробом кодвекторов на верхних уровнях памяти (например,  $\langle \mathbf{A} \vee \mathbf{B} \vee \mathbf{C} \rangle$  по  $\mathbf{A}$ ). Если бы кодвекторы разной уровни сложности находились в одном массиве памяти, нельзя было бы

определить только по их сходству с пробом, найден ли кодвектор супермножества или подмножества.

Каждый из массивов может быть нейросетевой автоассоциативной памятью, например, описанного в [27, 28] типа. Однако сейчас функционирование такого массива памяти моделируется путем запоминания его кодвекторов в виде списка и выполнения полного поиска на самое близкое соответствие путем вычисления перекрытия проба со всеми кодвекторами массива.

Обсудим, как рассмотренные операции и процедуры могут использоваться для представления иерархических реляционных структур, какими являются аналогичные эпизоды.

#### 2.4. Представление иерархических реляционных структур.

Рассмотрим две схемы представления реляционных структур в АПНС [16]. Они соответствуют двум известным схемам представления предикатов в символьных представлениях: «роль – заполнитель» и «предикат – аргументы». В качестве примеров возьмем эпизоды или ситуации, которые применяются при моделировании рассуждений по аналогии (в других системах терминологии их называют суждениями, высказываниями, предикатами и др.).

**2.4.1. Схема "роль – заполнитель".** Рассмотрим иерархический эпизод "Спот укусил Джейн, заставив Джейн бежать от Спота" (Spot bit Jane, causing Jane to flee from Spot). Используя представления АПНС типа "роль – заполнитель" (role-filler), кодвектор эпизода может быть представлен как в [14]:

$$P = {}^4\langle {}^3\langle \text{cause\_antc} \vee P(\text{bite}) \rangle \vee \langle \text{cause\_cnsq} \vee P(\text{flee}) \rangle \rangle, \quad (3)$$

где

$$P(\text{bite}) = {}^2\langle {}^1\langle \text{bite\_agt} \vee \text{spot} \rangle \vee \langle \text{bite\_obj} \vee \text{jane} \rangle \rangle, \quad (4)$$

$$P(\text{flee}) = {}^2\langle {}^1\langle \text{flee\_agt} \vee \text{jane} \rangle \vee \langle \text{flee\_obj} \vee \text{spot} \rangle \rangle. \quad (5)$$

Здесь кодвекторы ролей и заполнителей связаны с помощью процедуры CDT, например  ${}^1\langle \text{bite\_agt} \vee \text{spot} \rangle$ ,  ${}^1\langle \text{bite\_obj} \vee \text{jane} \rangle$ , где **bite\_agt** соответствует "роль первого места в отношении укусил" (агент) и **bite\_obj** соответствует "роль второго места в отношении укусил" (объект).  ${}^1\langle \dots \rangle$  обозначает прореживание с использованием определенной конфигурации прореживания, помеченной "1", как описано в разд. 2.2. Кодвекторы распределены по массивам памяти различных уровней, соответствующих уровням иерархии закодированных ими компонентов (табл. 1), и метка прореживания соответствует номеру уровня. Аналогично

$$P1(\text{flee}) = {}^1\langle P\_structure(\text{jane-spot}) \vee P\_components(\text{flee,jane,spot}) \rangle, \quad (6)$$

и весь эпизод в целом будет закодирован так:

$$P1 = {}^2\langle \text{cause} \vee P1(\text{bite}) \vee P1(\text{flee}) \vee P1(\text{bite}) \rangle \gg \text{antc} \vee P1(\text{flee}) \gg \text{cnsq}. \quad (7)$$

#### 2.4.2. Позиционная схема.

Один из известных методов представления места объекта в последовательности состоит в том, чтобы отвести для каждого места отдельное подмножество ресурсов. Такой метод ведет к росту размерности кодвекторов. Чтобы сохранить ее фиксированной, предложено кодировать различные позиции объекта (циклическим) сдвигом или другой обратимой перестановкой его кодвектора.

В нашем примере это может быть реализовано как **spot**>>agent, **jane**>>object. Здесь **X** >> *n* обозначает **X**, пермутированный или сдвинутый для кодирования *n*-й позиции в последовательности. Этот тип представления порядка может быть соотнесен с известным символьным представлением "символ-аргумент-аргумент" (или "предикат-аргументы" – predicate-arguments). Кодвектор рассматриваемого эпизода, согласно этой схеме представления, будет следующим:

$$P1(bite) = {}^1\langle P\_structure(spot-jane) \vee P\_components(bite,spot,jane) \rangle, \quad (8)$$

где

$$P\_structure(spot-jane) = spot \gg agent \vee jane \gg object, \quad (9)$$

$$P\_components(bite,spot,jane) = bite \vee spot \vee jane. \quad (10)$$

Аналогично

$$P1(flee) = {}^1\langle P\_structure(jane-spot) \vee P\_components(flee,jane,spot) \rangle, \quad (11)$$

и кодвектор эпизода в целом будет выглядеть так:

$$P1 = {}^2\langle cause \vee P1(bite) \vee P1(flee) \vee P1(bite) \gg antc \vee P1(flee) \gg cnsq \rangle. \quad (12)$$

Табл. 2 показывает распределение кодвекторов по массивам памяти разных уровней иерархии. Рассмотрим, как подобного рода представления используются при моделировании процессов поиска аналогов.

Таблица 1. Пример организации памяти эпизода для схемы "роль – заполнитель"

| Уровень иерархии | Кодвекторы компонентов аналогов   | Конфигурация прореживания |
|------------------|---|---------------------------|
| Уровень#4        | <b>Probe</b> = ${}^4\langle Antc \vee Cnsq \rangle$   | ${}^4\langle \rangle$     |
| Уровень#3        | <b>Antc</b> = ${}^3\langle cause\_antc \vee Bite \rangle$<br><b>Cnsq</b> = ${}^3\langle cause\_cnsq \vee Flee \rangle$  | ${}^3\langle \rangle$     |
| Уровень#2        | <b>Bite</b> = ${}^2\langle {}^1\langle bite\_agt \vee spot \rangle \vee {}^1\langle bite\_obj \vee jane \rangle \rangle$<br><b>Flee</b> = ${}^2\langle {}^1\langle flee\_agt \vee jane \rangle \vee {}^1\langle flee\_obj \vee spot \rangle \rangle$                                      | ${}^2\langle \rangle$     |
| Уровень#1        | <b>bite_a</b> = ${}^1\langle bite\_agt \vee spot \rangle$ <b>bite_o</b> = ${}^1\langle bite\_obj \vee jane \rangle$<br><b>flee_a</b> = ${}^1\langle flee\_agt \vee jane \rangle$ <b>flee_o</b> = ${}^1\langle flee\_obj \vee spot \rangle$  | ${}^1\langle \rangle$     |
| Уровень#0        | <b>spot</b> = dog $\vee$ id_spot <b>jane</b> = human $\vee$ id_jane   |                           |
| Нижний уровень   | <b>human</b> <b>dog</b> <b>cat</b> <b>mouse</b><br><b>id_jane</b> <b>id_john</b> <b>id_fred</b> <b>id_felix</b><br><b>id_fido</b> <b>id_spot</b> <b>id_rover</b> <b>id_mort</b><br><b>bite_agt</b> <b>bite_obj</b> <b>flee_agt</b> <b>flee_obj</b><br><b>cause_antc</b> <b>cause_cnsq</b> |                           |



Таблиця 2. Пример организация памяти эпизода для "позиционной" схемы

| Уровень иерархии | Кодвекторы компонентов аналогов  | Конфигурация прореживания |                 |            |              |                |                |                |                 |                |                |                 |                |  |
|------------------|--|---------------------------|-----------------|------------|--------------|----------------|----------------|----------------|-----------------|----------------|----------------|-----------------|----------------|--|
| Уровень#2        | $P = \langle \text{cause} \vee \text{BITE} \vee \text{FLEE} \vee \text{BITE} \rangle \langle \text{cause\_antc} \vee \text{FLEE} \rangle \langle \text{cause\_cnsq} \rangle$   | $\langle \rangle^2$       |                 |            |              |                |                |                |                 |                |                |                 |                |  |
| Уровень#1        | $\text{BITE} = \langle \text{bite} \vee \text{spot} \vee \text{jane} \vee \text{spot} \rangle \langle \text{agent} \vee \text{jane} \rangle \langle \text{object} \rangle$<br>$\text{FLEE} = \langle \text{flee} \vee \text{spot} \vee \text{jane} \vee \text{jane} \rangle \langle \text{agent} \vee \text{spot} \rangle \langle \text{object} \rangle$ | $\langle \rangle^1$       |                 |            |              |                |                |                |                 |                |                |                 |                |  |
| Уровень#0        | $\text{spot} = \text{dog} \vee \text{id\_spot}$ $\text{jane} = \text{human} \vee \text{id\_jane}$  |                           |                 |            |              |                |                |                |                 |                |                |                 |                |  |
| Нижний уровень   | <table border="0"> <tr> <td><b>human</b></td> <td><b>dog</b></td> <td><b>cat</b></td> <td><b>mouse</b></td> </tr> <tr> <td><b>id_jane</b></td> <td><b>id_john</b></td> <td><b>id_fred</b></td> <td><b>id_felix</b></td> </tr> <tr> <td><b>id_fido</b></td> <td><b>id_spot</b></td> <td><b>id_rover</b></td> <td><b>id_mort</b></td> </tr> </table>       | <b>human</b>              | <b>dog</b>      | <b>cat</b> | <b>mouse</b> | <b>id_jane</b> | <b>id_john</b> | <b>id_fred</b> | <b>id_felix</b> | <b>id_fido</b> | <b>id_spot</b> | <b>id_rover</b> | <b>id_mort</b> |  |
| <b>human</b>     | <b>dog</b>   | <b>cat</b>                | <b>mouse</b>    |            |              |                |                |                |                 |                |                |                 |                |  |
| <b>id_jane</b>   | <b>id_john</b>   | <b>id_fred</b>            | <b>id_felix</b> |            |              |                |                |                |                 |                |                |                 |                |  |
| <b>id_fido</b>   | <b>id_spot</b>   | <b>id_rover</b>           | <b>id_mort</b>  |            |              |                |                |                |                 |                |                |                 |                |  |

**3. Поиск аналогов с помощью представлений АПНС**

Как указывалось выше, в распределенных представлениях, учитывающих структуру, набор компонентов и их группирование должны влиять на сходство кодвекторов так, что похожие структуры получают похожие кодвекторы. Двухступенчатое сравнение, применяемое в традиционных моделях доступа (разд. 1), в этом случае не нужно. Задача состоит в том, чтобы создать такие кодвекторы эпизодов-аналогов, величины сходства которых отражают степень сходства эпизодов, выявленную психологами в результате экспериментов с людьми.

В результате поиск наиболее близкого аналога в памяти с помощью учитывающих структуру распределенных представлений может осуществляться путем нахождения кодвектора, наиболее похожего на входной, в долговременной памяти, где хранятся кодвекторы всех эпизодов.

**3.1. Представление эпизодов-аналогов с различным типом сходства.** Чтобы проиллюстрировать поиск аналогов в АПНС, используем эпизоды из [13, 10]. Участвуют следующие персонажи

или объекты: собаки (Фидо, Спот, Ровер), люди (Джейн, Джон, Фред), кот (Феликс), мышь (Морт). Отношения: кусать, бежать, быть причиной. Назовем пробным эпизод, к которому надо найти ближайший аналог в памяти. Используем в качестве пробного эпизод P, который уже встречался ранее: "Spot bit Jane, causing Jane flee from Spot" (разд. 2.4). Эпизоды, хранящиеся в памяти, имеют те же самые отношения, что и пробный, но различные типы сходства с ним – главным образом, согласно классификации Джентнер [1] (см. также [9]).

Основными типами сходства являются структурное (сходство отношений) и поверхностное (сходство объектов). Эпизод с буквальным сходством (literal similarity, LS) имеет как структурное, так и поверхностное сходство с пробным: "Фидо укусил Джона, заставив Джона убежать от Фидо" ("Fido bit John, causing John to flee from Fido"). Эпизод с поверхностными признаками (surface features, SF) имеет поверхностное, но не структурное сходство с пробным: "Джон убежал от Фидо, заставив Фидо укусить Джона" ("John fled from Fido, causing Fido to bite John").

Эпизод перекрестного отображения (cross-mapped, CM) имеет структурное и поверхностное сходство с пробным, но типы соответствующих объектов переставлены: "Фред укусил Ровера, заставив Ровера убежать от Фреда" ("Fred bit Rover, causing Rover to flee from Fred"). Эпизод с аналогичным сходством (analogy, AN) имеет структурное, но не поверхностное сходство: "Морт укусил Феликса, заставив Феликса убежать от Морта" ("Mort fled from Felix, causing Felix to bite Mort"). Эпизоды с отношениями первого порядка (first order relations only, FOR) не имеют ни структурного, ни поверхностного сходства, а только общие предикаты: "Морт убежал от Феликса, заставив Феликса укусить Морта" ("Mort fled from Felix, causing Felix to bite Mort") (табл. 3).

Чтобы сформировать АПНС-представления эпизодов, необходимо закодировать объекты и отношения, используя кодвекторы основного уровня. Под кодвекторами основного уровня понимаем независимо сгенерированные кодвекторы самого нижнего композиционного уровня (см. табл. 1, 2). В их качестве используем случайные бинарные векторы с  $N=100,000$ ,  $M=1,000$ .

Все отношения и роли считаем непохожими и представляем кодвекторами основного уровня, например **bite\_agt**, **bite\_obj**, **flee\_agt** и т.д. Кодвекторы объектов (или экземпляров некоторого типа) сформированы как дизъюнктивная суперпозиция кодвекторов двух видов атрибутов: типа (**human**, **dog**, **cat**, **mouse**) и идентификатора или имени (**id\_john**, **id\_fido**, и т.д.). Например, объекты Джон и Морт представляем соответственно как **john** =  $\langle \text{human} \vee \text{id\_john} \rangle$ , **mort** =  $\langle \text{mouse} \vee \text{id\_mort} \rangle$ . Таким образом имитируется разная степень сходства объектов.

**3.2. Эксперименты по поиску аналогов.** Данные экспериментов [9, 29, 30] по поиску аналогов людьми свидетельствуют о том, что сходство объектов более важно, чем сходство структур. В терминах рассмотренных выше разных типов сходства аналогов это означает, что аналоги типа LS, CM, SF находятся людьми легче, чем аналоги со сходством типа AN и FOR. Среди группы эпизодов со сходством объектов эпизод LS находят легче, чем CM и SF. На основе анализа данных разных авторов в [13] приводится следующий порядок извлечения людьми аналогов из долговременной памяти:  $LS > CM \geq SF > AN \geq FOR$ .

Таблица 3. Эпизоды-аналоги с различными типами сходства с пробным эпизодом

| Тип сходства              | Эпизод |           |         |   |        |            |       |
|---------------------------|--------|-----------|---------|---|--------|------------|-------|
|                           | Спот   | укусил    | Джейн   | ⇒ | Джейн  | убежала от | Спота |
| Пробный эпизод P          | Спот   | укусил    | Джейн   | ⇒ | Джейн  | убежала от | Спота |
| Буквальное сходство LS    | Фидо   | укусил    | Джона   | ⇒ | Джон   | убежал от  | Фидо  |
| Поверхностные признаки SF | Джон   | убежал от | Фидо    | ⇒ | Фидо   | укусил     | Джона |
| Перекрестная аналогия CM  | Фред   | укусил    | Ровера  | ⇒ | Ровер  | убежал от  | Фреда |
| Аналогия AN               | Морт   | укусил    | Феликса | ⇒ | Феликс | убежал от  | Морта |

Таблиця 4. Величини сходства между кодвекторами, представляющими проб и эпизоды в памяти

| Тип сходства | Эпизод                                      | Величина сходства |             |            |
|--------------|---|-------------------|-------------|------------|
|              |   | HRRs              | АПНС        | АПНС       |
|              |   | role-filler       | role-filler | positional |
| P            | Спот укусил Джейн ⇒ Джейн убежала от Спот   | 1.00              | 1.00        | 1.00       |
| LS           | Фидо укусил Джона ⇒ Джон убежал от Фидо     | 0.71              | 0.42        | 0.40       |
| SF           | Джон убежал от Фидо ⇒ Фидо укусил Джона     | 0.47              | 0.38        | 0.24       |
| CM           | Фред укусил Ровера ⇒ Ровер убежал от Фреда  | 0.47              | 0.38        | 0.30       |
| AN           | Морт укусил Феликса ⇒ Феликс убежал от Морт | 0.42              | 0.26        | 0.14       |

Для создания кодвекторов эпизодов использовалась как схема "роль - заполнитель" (разд. 2.4.1), так и позиционная схема (разд. 2.4.1). Для схемы "роль - заполнитель" кодвекторы уровня 1 и более высоких уровней прореживались до 2M. Эта "глубина прореживания" была выбрана опытным путем, чтобы получить величины сходств между кодвекторами проба и каждого эпизода, представленные в столбце «role-filler» табл. 4.

Для позиционной схемы кодвекторы уровней 1 и 2 прореживались до 4M. Величины сходства для этой глубины прореживания даны в столбце «positional» табл. 4. Величины сходства для другого типа распределенных представлений (HRR из [13] даны в первом столбце. Результаты усреднены по 100 реализациям случайных кодвекторов нижнего уровня.

Порядок величин сходства для обоих рассмотренных типов представлений АПНС тот же, что и для схемы HRR [13]. Это также соответствует порядку экспериментальных результатов, полученному для людей [9, 29, 30] и результатам моделирования,

приводимого для MAC/FAC [9] и ARCS [10], где использовались другие эпизоды, но с тем же типом сходства.

**Выводы**

При исследовании многих предметных областей существенной является структура объектов и их описаний. Это подтверждается как данными исследований психологов, изучающих сложные когнитивные процессы, такие, как процессы рассуждения по аналогии, так и работами исследователей, занимающихся исследованием свойств разного рода сложноструктурированных объектов [31], от химических соединений [32] до военно-политических кризисов [33]. Поэтому сходство описаний таких объектов должно учитывать сходство как компонентов ("семантическое сходство"), так и отношений, существующих между ними ("структурное сходство"). Как показано в данной статье и в работах других исследователей, могут быть созданы распределенные представления структур, которые несут непосредственную информацию о наборе структурных компонентов различных иерархических

уровней и об их структурной организации.

В отличие от локальных или символьных представлений такие распределенные представления обеспечивают более адекватный учет семантического содержания, гибкость и способность справиться с зашумленной и неожиданной входной информацией. Они позволяют естественно представлять и вычислять меру градуального сходства, более надежны и нейробиологически релевантны. Такие представления позволяют строить "на лету", без предварительного обучения, кодвекторы одинаковой размерности для иерархических структур разной сложности и оценивать их сходство скалярным произведением.

АПНС являются примером архитектуры, объединяющей подходы и технологии, развитые в области искусственного интеллекта, на нейросетевой основе. Их использование может рассматриваться как комбинация символьной и образной обработки информации. Алгоритмы допускают массивно-параллельную реализацию и позволяют использовать эффективные версии распределенной ассоциативной памяти.

Поиск аналогов с помощью АПНС демонстрирует способность полностью распределенного подхода (без локальных или символьных подсистем) работать со сложноструктурированными данными. Перспективы дальнейших исследований состоят в использовании данного подхода при решении задач искусственного интеллекта и информатики, таких как представление и поиск информации в базах знаний [33, 34]. На его основе может моделироваться познавательная деятельность людей, в частности, рассуждения по аналогии [1, 4, 5].

Авторы выражают благодарность за обсуждения Л.М. Касат-

киной и анонимному рецензенту за ценные замечания и комментарии.

1. Gentner D. Structure-mapping: A theoretical framework for analogy // *Cognitive Science*. – 1983. – № 7. – P. 155 – 170.
2. Hummel J.E., Holyoak K.J. Distributed representations of structure: A theory of analogical access and mapping // *Psychological Review*. – 1997. – **104**, № 3. – P. 427 – 466.
3. Gentner, D. and Markman, A.B. Analogy-Based Reasoning // Arbib M.A. // *Handbook of brain theory and neural networks*. – Cambridge, MA: MIT Press, 1995. – P. 91-93.
4. Амосов Н.М. Моделирование мышления и психики. – Киев: Наук. думка, 1965. – 304 с.
5. Гладун В.П. Партнерство с компьютером: Человеко-машинные целеустремленные системы. – Port-Royal, 2000. – 128 с.
6. Kanerva P. Dual role of analogy in the design of a cognitive computer / Holyoak K., Gentner D., Kokinov B. // *Advances in Analogy Research: Integration of Theory and Data from the Cognitive, Computational, and Neural Sciences*. Sofia, Bulgaria: New Bulgarian University, 1998. – P. 164- 170.
7. Gentner D., Markman A.B. Structure Mapping in Analogy and Similarity // *American Psychologist*. – 1997. – **52**, № 1. – P. 45-56.
8. Eliasmith C., Thagard P. Integrating Structure and Meaning: A Distributed Model of Analogical Mapping // *Cognitive Science*. – 2001. – **2**, № 25. – P. 245-286.
9. Forbus K.D., Gentner D., Law K. MAC/FAC: A model of similarity-based retrieval. // *Ibid.* – 1995. – **19**, № 2. – P. 141-205.
10. Analog Retrieval by Constraint Satisfaction / P. Thagard, K. Holyoak, G. Nelson, D. Gochfeld // *Artificial Intelligence*. – 1990. – № 46. – P. 259-310.
11. Falkenhainer B., Forbus K., Gentner D. The Structure-Mapping Engine: Algorithm and Examples // *Artificial Intelligence*. – 1989. – № 41. – P. 1-63.
12. Rachkovskij D.A. Some approaches to analogical mapping with structure sensitive distributed representations // *J. of Experimental and Theoretical*

- Artificial Intelligence. – 2004. – **16**, № 3. – P. 125-145.
13. Plate T. Analogical Retrieval and Processing with Distributed Vector Representations // Expert Systems: The Intern. J. of Knowledge Engineering and Neural Networks. – 2000. – № 17(1). – P. 29-40.
14. Rachkovskij D.A. Representation and Processing of Structures with Binary Sparse Distributed Codes // IEEE Transactions on Knowledge and Data Engineering. – 2001. – **2**, № 13. – P. 261-276.
15. Thorpe, S. Localized Versus Distributed Representations / Arbib M. // The Handbook of Brain Theory and Neural Networks. – Cambridge, MA: MIT Press, 2003. – P. 643-646.
16. Rachkovskij D.A., Kussul E.M. Binding and Normalization of Binary Sparse Distributed Representations by Context-Dependent Thinning // Neural Computation. – 2001. – **2**, № 13. – P. 411-452.
17. Frasconi P., Gori M., Sperduti A. A general framework for adaptive processing of data structures. // IEEE Transactions on Neural Networks. – 1998. – **9**, № 5. – P. 768-786.
18. Quillian M. Semantic memory // Minsky M. Semantic information processing. – Cambridge, Mass.: MIT Press, 1968. – P. 227-270.
19. M-network as possible basis for construction of heuristic models / N. Amosov, E. Basilevsky, A. Kasatkin, L. Kasatkina, A. Luk, E. Kussul, S. Talayev // Cybernetica. – 1972. – № 3. – P. 169-186.
20. Rangarajan A., Mjolsness E. Lagrangian Relaxation Network for Graph Matching // IEEE Transactions on Neural Networks. – 1996. – № 7(6). – P. 4629-4634.
21. Jain B.J., Wysotski F. A Novel Neural Network Approach to Solve Exact and Inexact Graph Isomorphism Problems. // Proceedings ICANN/ICONIP 2003. – Istanbul, Turkey: 2003. – P. 299-306.
22. Finch A.M., Wilson R.C., Hancock E.R. Symbolic Graph Matching with the EM Algorithm // Pattern Recognition. – 1998. – **31**, – P. 1777-1790.
23. Markman A.B., Gentner D. Structure mapping in the comparison process // American Journal of Psychology. – 2000. – **113**, № 4. – P. 501-538.
24. Plate T. Holographic Reduced Representation: Distributed Representation for Cognitive Structures. – Chicago: Center for the Study of Language and Information, 2003. – 250 p.
25. Kanerva P. Binary Spatter-Coding of Ordered K-tuples // Intern. Conf. on Artificial Neural Networks ICANN'96. – Bochum, Germany: Berlin: Springer, 1996. – P. 869-873.
26. Rachkovskij D.A., Kussul E.M. Binding and Normalization of Binary Sparse Distributed Representations by Context-Dependent Thinning // Neural Computation. – 13, №2 – P. 411-452.
27. Willshaw D.J., Buneman O.P., Longuet-Higgins H.C. Non-holographic associative memory // Nature. – 1969. – № 222. – P. 960-962.
28. Frolov A.A., Rachkovskij D.A., Husek D. On Information Characteristics of Willshaw-Like Auto-Associative Memory // Neural Network World. – 2002. – №2, – P. 141-157.
29. Ross B.H. Distinguishing types of superficial similarities: Different effects on the access and use of earlier examples – 1989. – **15**, № 3. – P. 456-468.
30. Wharton C.M., Holyoak K.J., Downing P.E., Lange T.E., Wickens T.D., Melz E.R. Below the surface: Analogical similarity and retrieval competition in reminding – 1994. – **26**, № 1. – P. 64-101.
31. Вывод гипотез о составе и свойствах объектов на основе аналогии / В.П. Гладун, В.Ю. Величко, Н.Н. Киселева, Н.М. Москалькова // Искусственный интеллект – 2000, №1. – С. 44-52.
32. Величко В.Ю., Москалькова Н.М. Использование программного комплекса "Аналогия" для формирования гипотез о свойствах составных объектов // Пробл. программирования. – 2002. – №1-2. – С. 445-452.
33. The DARPA High-Performance Knowledge Bases Project / P. Cohen, R. Schrag, E. Jones, A. Pease, A. Lin, B. Starr, D. Gunning, M. Burke // AI Magazine. – 1998. – **19**, № 4. – P. 25-49.
34. Forbus K.D. Exploring Analogy in the Large // Gentner D., Holyoak K., Kokinov B. // Analogy: Perspectives from Cognitive Science. – Cambridge, MA: The MIT Press, 2000. – P. 23-58.

Получено 24.11.04

мл. науч. сотр.

**Об авторах**

*Рачковский Дмитрий Андреевич,*  
канд. техн. наук, ст. науч.  
сотр.

*Мисуно Иван Семенович,*  
вед. инж.-программист

*Слипченко Сергей Витальевич,*  
вед. инж.-программист

*Соколов Артем Михайлович,*

*Место работы авторов:*

Международный научно-учебный центр ин-  
формационных технологий и систем НАН  
Украины и МОН Украины

просп. Акад. Глушкова, 40, Киев, Ук-  
раина

Тел. 266 4119

E-mail: dar@infrm.kiev.ua  
qdamage@longbow.kiev.ua  
slipchenko\_serg@ukr.net  
sokolov@ukr.net