ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА

УДК 004.89

И. А. Бессмертный

МЕТОДЫ ПОИСКА ИНФОРМАЦИИ С ИСПОЛЬЗОВАНИЕМ ИНТЕЛЛЕКТУАЛЬНОГО АГЕНТА

Обсуждается возможность создания универсального интеллектуального агента для извлечения знаний из глобальной семантической сети. Рассматриваются проблемы практической реализации интеллектуального агента, в частности комбинаторная сложность. Предлагаются способы сокращения размерности задачи поиска.

Ключевые слова: *интеллектуальный агент, глобальная семантическая сеть, инженерия знаний.*

Введение. Провозглашенная W3C консорциумом (World Wide Web Consortium) цель глобальной семантической сети (ГСС, Semantic Web) заключается в создании универсальной среды для обмена данными, управления персональной информацией, интеграции корпоративных данных и глобального совместного доступа к коммерческой, научной и культурной информации. Всемирная сеть Интернет, построенная на основе протокола HTTP (Hypertext Transfer Protocol), не обеспечивает достижения этой цели в полной мере, поскольку ресурсы, представленные HTML-документами (Hypertext Markup Language), предназначены исключительно для просмотра человеком, тогда как количество документов измеряется миллиардами, что существенно превышает человеческие возможности. Используемая в HTML-документах разметка Web-страниц предназначена только для форматирования текста на экране и не содержит метаданных, позволяющих извлечь смысл документов. Концепция ГСС предполагает автоматизацию обработки информации, хранящейся на сетевых ресурсах.

В программной статье основателя Интернета Тима Бернерса Ли [1] предложено снабдить обычные Web-страницы тегами, описывающими факты, а также правилами, используя которые можно извлекать новые факты. Факты представляются триплетами $t = \{S\ P\ O\}$, где S — субъект, O — объект, P — предикат (отношение между субъектом и объектом). Триплет является частным случаем предиката в языке Prolog, поэтому в дальнейшем такие триплеты будем называть предикатами, чтобы придерживаться более привычной терминологии. Правило имеет вид „ЕСЛИ c ТО g“, где $c = (c_1$ И c_2 И … $c_n)$, $g = (g_1$ И g_2 И … $g_k)$, $c_1, c_2, \ldots, c_n$ — предикаты условия (тело правила), $g_1, g_2, \ldots, g_k$ — предикаты результата (заголовок) правила. Цель настоящей статьи — исследование принципов извлечения знаний из ГСС. Используемые методы — методы искусственного интеллекта [2].

Концептуальная модель интеллектуального агента. Для автоматического поиска знаний должны создаваться специальные программы — интеллектуальные агенты (далее — просто „агенты“), выполняющие поиск на дереве решений. Сформулируем требования к агенту, вытекающие из целей и задач ГСС [1, 3]:

— универсальность: данное требование означает применимость агента для различных предметных областей, а также языков, на которых представлены данные;

— доступность для конечного пользователя: пользователь не обязан обладать специальными знаниями, чтобы сформулировать запрос;

— автономность: отсутствие необходимости вмешательства пользователя в ход обработки запроса;

— достоверность результатов: агент должен формировать не только результат, но и оценку степени доверия к нему;

— разумное время ответа: время выдачи ответа должно быть, по крайней мере, меньше времени сохранения актуальности данных и меньше времени, затрачиваемого на традиционный поиск в Интернете.

Перечисленный набор требований является минимальным, поскольку не отражает проблем конфиденциальности данных, возможности взаимодействия многих агентов и др. Тем не менее даже в таком наборе требования противоречат друг другу. В частности, универсальность неизбежно влечет за собой сложность формулирования запроса, что требует от пользователя специальной подготовки. Требование автономности вступает в противоречие с требованием разумного времени ответа, поскольку отсутствие возможности консультации с пользователем может привести к углублению в тупиковые ветви дерева поиска. Кроме того, при автономной обработке запроса предполагается, что запрос сформулирован достаточно четко. Проще говоря, пользователь точно знает, что он хочет получить, однако такая ситуация характерна для ограниченного набора задач. Таким образом, одновременное выполнение всех перечисленных требований представляется невозможным, и при реализации интеллектуального агента должен быть достигнут компромисс между противоречивыми требованиями.

Функциональная модель интеллектуального агента может быть описана следующим образом.

1. Пользователь формулирует задание, которое преобразуется к виду, пригодному для формирования поисковых запросов.

2. Агент формирует запросы к поисковым серверам и получает список онтологий, содержащих запрашиваемые сущности или свойства.

3. С помощью онтологий задание пользователя приводится к набору предикатов условий, описывающих то, что известно, и предикатов цели, которую необходимо найти или истинность которой необходимо подтвердить.

4. Формируется запрос к поисковым серверам на получение списка ресурсов, релевантных цели.

5. Выполняется последовательная загрузка ресурсов из списка поиска и их сопоставление с предикатами цели. Если найденные факты подтверждают истинность цели, то пользователю возвращаются найденные значения переменных, и работа агента завершается.

6. Если в ходе продвижения к цели агент находит правила, в теле которых есть предикаты, отсутствующие в выборке документов, то может потребоваться новый поиск в ГСС. Таким образом, возможно углубление в цепочку правил.

Проблемы практической реализации универсального интеллектуального агента. Идеализированная модель, представленная выше, является достаточно простой для реализации. Рассмотрим свойства реальных ресурсов ГСС и задач, которые могут существенно увеличить сложность поиска решения.

Комбинаторная сложность задачи поиска в ГСС. Вышеописанная модель агента реализует классическую задачу неинформированного поиска [2] методом обратного вывода (back chain reasoning), сложность которой можно оценить следующим образом. Пусть r — среднее количество правил, релевантных запросу при каждом обращении к базе знаний,

n — среднее число предикатов в теле каждого правила, d — средняя глубина вложенности правил. Тогда дерево поиска будет содержать N вершин:

$$N = r + rn + (rn)^2 + \ldots + (rn)^d = r + \sum_{i=1}^{d} (rn)^i .$$

На рис. 1 отображена зависимость сложности (времени обхода T) дерева поиска от числа правил и количества условий в правиле для n = 6. Если, как принято в работе [2], скорость поиска считать равной 10 000 правил в секунду, то время полного обхода дерева при r=20, n=6 и d=6 составит 9,5 лет. Поясним это на примере Гражданского кодекса РФ, который насчитывает 1542 статьи по нескольку пунктов в каждой. Таким образом, база знаний на его основе будет состоять из нескольких тысяч правил, и любой запрос на юридическую тему будет к ним обращен. Поскольку сходные правила применяются для разных субъектов права, агент может извлекать правила для предпринимателей, физических или юридических лиц, относящиеся к внутреннему или международному праву. Следовательно, количество правил, релевантных запросу, может быть существенно больше единицы.

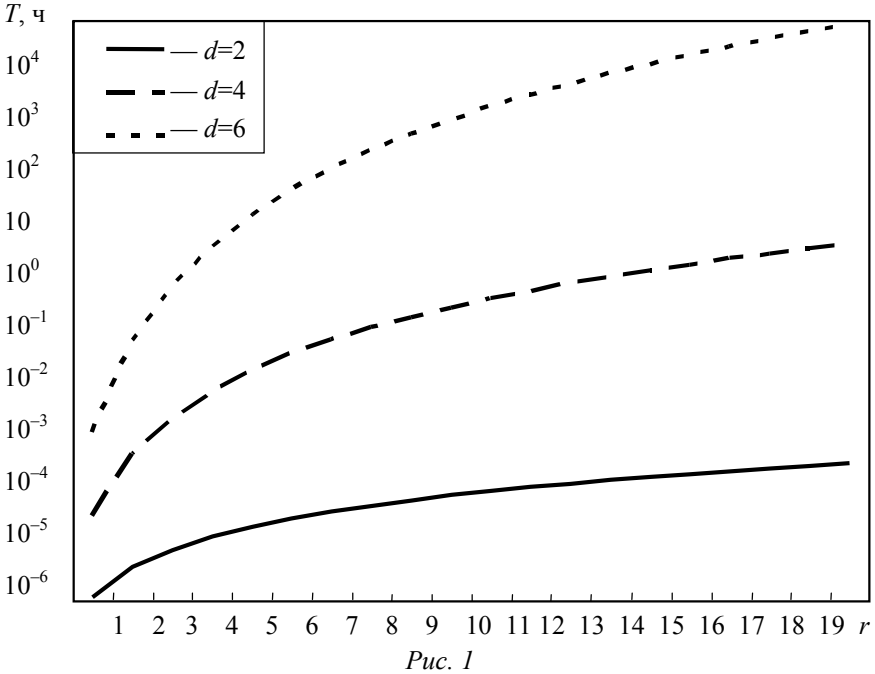

Рис. 1

Развертывание вершин дерева поиска на распределенных ресурсах требует, чтобы эти ресурсы были найдены и загружены. В случае применения алгоритма обхода дерева сначала вглубь требуется Q операций поиска, которое может быть вычислено по формуле

$$Q = 1 + rn + (rn)^2 + \ldots + (rn)^{d-1} = \sum_{i=0}^{d-1} (rn)^i .$$

Если использовать алгоритм поиска сначала в ширину, то количество операций поиска будет равно глубине d вложенности правил, но потребуется запоминать $N - (rn)^d$ (все, кроме последнего уровня) результатов поиска. Для приведенного выше примера обхода дерева при r=20, n=6 и d=6 потребуется около 25 Тбайт при условии, что одна ссылка занимает 1кбайт памяти.

Таким образом, поиск в ГСС путем простого перебора может оказаться очень длительным и даже бесконечным.

Множество онтологий для каждой предметной области. В предположении, что задание пользователя формализуется с использованием одной онтологии, соответственно и поиск осуществляется среди документов, оформленных с использованием той же онтологии. На

практике одной предметной области может соответствовать множество онтологий. Так, в частности, семантический поисковый сервер SWOOGLE (swoogle.umbc.edu) по ключевым слова *camera* и *viewfinder* возвращает 24 ссылки на одни только англоязычные онтологии, относящиеся к фотокамерам, при этом каждая из онтологий может быть использована для создания семантических документов.

Множество единиц измерения. Разные факты могут содержать одни и те же атрибуты в разных единицах измерения. Выполнение операций сравнения требует приведения всех атрибутов к одной единице измерения. Различными могут быть также и способы измерения одних и тех же атрибутов. Например, размер фоточувствительной матрицы цифровых фотокамер может выражаться диагональю в долях дюйма или длинами сторон в миллиметрах. Приведение данных к одной системе единиц измерения требует применения специальных правил, а значит, увеличения размерности задачи поиска.

Неполная информация. В ходе применения правил агент может сталкиваться с ситуацией, когда факты или значения переменных установить невозможно. Обычно это означает неполное описание проблемы пользователем и требует уточнения. Здесь существуют две проблемы. Во-первых, пользователь должен понимать, что стоит за именем переменной, значение которой он должен установить. Онтологии, представленные в настоящее время на Интернет-ресурсах, далеко не всегда снабжены подробными комментариями, позволяющими идентифицировать объекты и их атрибуты. Во-вторых, углубление в дерево поиска может приводить к чрезмерному количеству вопросов, не относящихся к проблеме.

Методы ускорения поиска. Сокращение комбинаторной сложности задач поиска обычно достигается путем использования эвристических функций, позволяющих оценивать перспективность ветвей поиска [2]. Однако выбор эвристики — процесс творческий и индивидуальный для каждой предметной области. Рассмотрим, какие другие решения могут уменьшить размерность задачи поиска.

Управление глубиной вложенности правил. График, представленный на рис. 1, демонстрирует, что уменьшение глубины дерева поиска в два раза может сократить время его обхода на 4 порядка. Сознательное ослабление логической мощности агента позволит решать несложные задачи в разумные сроки и лишь для нетривиальных задач задавать увеличенную глубину поиска: понятно, что для этого потребуется привлечение больших вычислительных мощностей. Кроме того, обычно самые простые решения, лежащие на поверхности, являются самыми эффективными.

Сужение горизонта поиска (кругозора агента). Несмотря на то, что коэффициент ветвления на дереве поиска влияет на его сложность гораздо слабее, чем глубина дерева, уменьшение количества правил может также сократить время поиска решения, причем без ухудшения качества. Например, базу знаний на основе Гражданского кодекса РФ можно разделить на 7 отдельных баз по числу его разделов, поскольку каждый из разделов содержит данные, не пересекающиеся с информацией из других разделов. В то же время нетривиальный поиск может потребовать обращения к самым разным областям знаний, что компенсируется возможностью появления неожиданных решений.

Диалоговый режим поиска. В соответствии с функциональной моделью агента предполагается, что пользователь описывает проблему, а затем агент ведет поиск решения. Однако далеко не всегда пользователь может сразу задать все исходные данные, поскольку не знает, что известно агенту. Следовательно, неизбежно уточнение цели в процессе развертывания дерева поиска. Кроме того, агент может консультироваться с пользователем перед углублением в разветвленную часть дерева, что потребует много времени на ее обход, в то время как пользователь может решить, стоит ли спускаться по этой ветви.

Обучение агента. Результаты успешного поиска могут сохраняться в виде фактов, что позволит в дальнейшем обеспечить к ним быстрый доступ. Данный метод широко используется

естественным интеллектом, начиная от таблицы умножения и заканчивая библиотеками шахматных партий лучших гроссмейстеров. Основная проблема реализации обучения — возможность получения ложных выводов и необходимость их последующего поиска и удаления в большом массиве накопленных фактов. Кроме того, требуется корректировать правила, которые приводят к созданию ложных фактов. В противном случае дерево решений станет только более разветвленным и вместо сокращения поиска будет его существенное усложнение. Обучение имеет смысл даже внутри одной операции поиска (кратковременная память). Рассмотрим в качестве примера известную логическую игру „23 спички“. Правила игры простые: два игрока по очереди берут из кучки одну, две или три спички. Проигрывает тот, кто берет последнюю спичку. На рис. 2 отображен фрагмент дерева поиска для начального состояния — 6 спичек. Число в вершине означает количество оставшихся спичек, дуги — ходы игроков. Полный обход данного фрагмента требует развертывания 28 вершин. Пунктиром на графе обозначены повторяющиеся фрагменты. Если запоминать результат первого обхода каждого из таких фрагментов, это избавит от необходимости повторного углубления и сократит число развертываемых вершин до 12 (выделены фоном), т.е. более чем в два раза. Естественно, такой прием имеет смысл только при поиске сначала в глубину. Полное дерево поиска для начального состояния (23 спички) содержит 900 140 вершин, а поиск до первого решения — 20 009 вершин; в этом случае запоминание промежуточных результатов сократит обход до 57 вершин или в 351 раз.

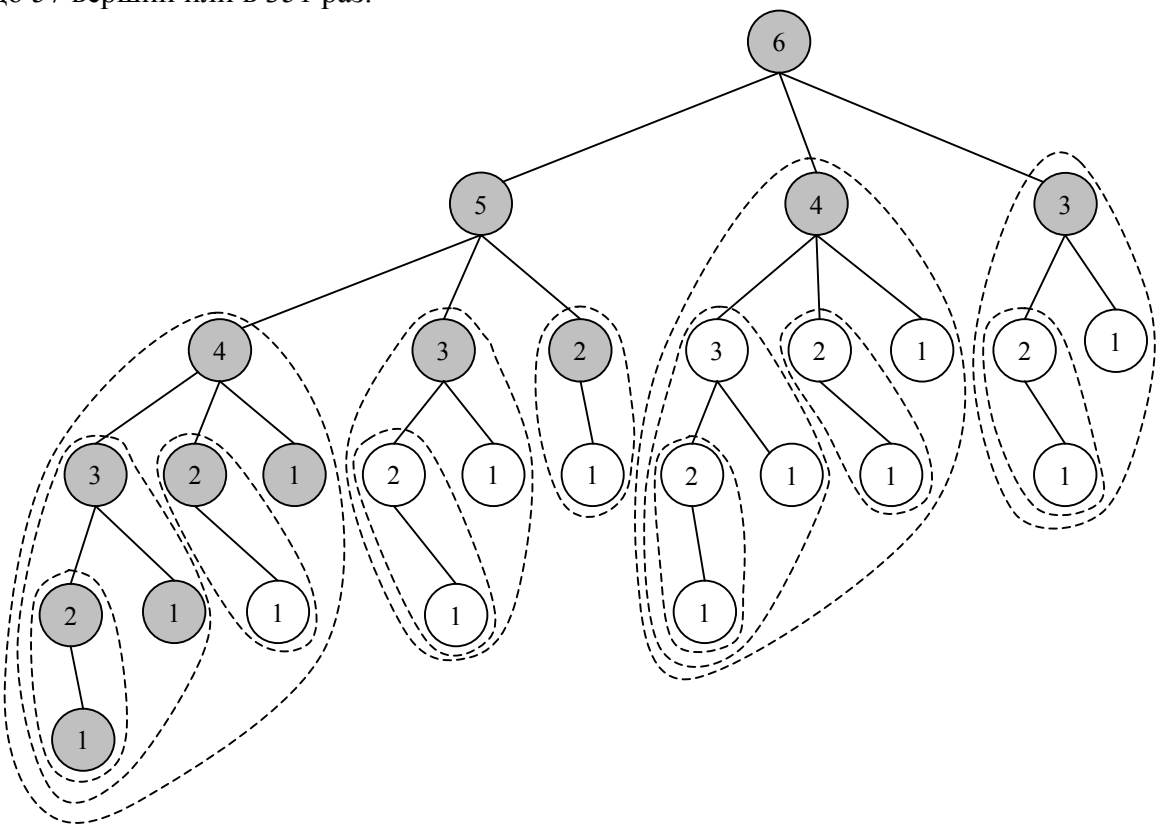

Рис. 2

Использование контекста. Пользователь может персонализировать агента созданием базы знаний о контексте. База знаний контекста также является результатом обучения агента и избавляет пользователя от необходимости постоянно отвечать на одни и те же вопросы.

Заключение. Создание универсального интеллектуального агента для глобальной семантической сети — это решение задачи поиска на дереве решений. Обширная база знаний, размещенная в ГСС, делает комбинаторную сложность этой задачи непомерно высокой, что не позволяет решить ее методом простого перебора. Вследствие недостаточного числа ресурсов ГСС, содержащих формализованные знания, отсутствует возможность проведения натур-

ных экспериментов. Поэтому для исследования методов построения баз знаний, апробации алгоритмов поиска, визуализации знаний и обучения основам семантических сетей автором настоящей статьи разработана программа, реализующая создание баз знаний и функции интеллектуального агента [4—6]. Программа написана на языке Visual Prolog 7.1 и в настоящее время проходит опытную эксплуатацию в учебном процессе на кафедре вычислительной техники Санкт-Петербургского государственного университета информационных технологий, механики и оптики.

СПИСОК ЛИТЕРАТУРЫ

1. *Berners-Lee T., Hendler J., Lassila Ora.* The semantic web // Sci. Amer. Mag. 2001. May. P. 29—37.
2. *Кальченко Д.* Интеллектуальные агенты семантического Web′а // КомпьютерПресс. 2004. № 10. С. 26—32.
3. *Рассел С., Норвиг П.* Искусственный интеллект: Современный подход: Пер. с англ. М.: Изд. дом „Вильямс“, 2006.
4. *Bessmernty I.* An intellectual agent in training systems // Proc. of 5th Intern. Symposium on Education and Information Systems, Technologies and Applications: EISTA'2007. Orlando, FL. 2007. P. 86—89.
5. *Bessmertny I., Kulagin V.*. Semantic network as a knowledge base in training systems // Proc. of 11th IACEE World Conf. on Continuing Engineering Education. Atlanta, GA. 2008. P. 95—99.
6. *Bessmertny I.* Visual prolog and semantic networks at knowledge visualization // Proc. of Visual Prolog Application & Language Conf.: VIP-ALC′08. St. Petersburg. 2008. P.107—111.

Сведения об авторе

Игорь Александрович Бессмертный — канд. техн. наук, доцент; Санкт-Петербургский государственный университет информационных технологий, механики и оптики, кафедра вычислительной техники;
E-mail: igor_bessmertny@hotmail.com

Рекомендована кафедрой вычислительной техники

Поступила в редакцию 15.04.09 г.