УДК 681.324.06 + 62:007.51

Онтологическая поисковая система Jewel для реализации интеллектуального поиска в Интернет- и интранет-сетях.

Е.В. Сизиков, Д.В. Сошников

В работе рассказывается о применении искусственного интеллекта к решению задачи поиска в Интернет- и интранет-сетях с использованием онтологий. Вводится в рассмотрение онтологический подход к аннотированию ресурсов, предлагающий внедрение в веб-страницы дополнительной структурированной информации об их содержании, которая в дальнейшем используется поисковым роботом для построения фреймовой индексной базы знаний, на основе которой производится определение релевантности найденного ресурса запросу. Также обсуждаются особенности реализации прототипа поисковой системы на основе предложенного подхода.

Искусственный интеллект – одно из интереснейших направлений научной деятельности человечества, которое ставит перед собой задачу построения вычислительного устройства, способного, подобно человеку, к самостоятельному мышлению. Методы искусственного интеллекта, как правило, используются для решения неформальных задач, постановка которых проста и понятна для человека, но для которых сложно указать алгоритм их решения.

В настоящее время характерна тенденция к использованию элементов искусственного интеллекта для решения проблемы поиска информации. За последние годы представлено большое число работ на тему интеллектуального поиска. Интерес к данной теме вполне оправдан как с научной, так и с практической точки зрения. Развитие методов интеллектуального поиска с одной стороны позволяет приблизиться к решению задачи автоматизированной структуризации больших массивов слабоструктурированных данных; кроме того, привнесение интеллектуальности в обычно рутинный процесс поиска информации в существующих системах может перевернуть индустрию поисковых систем и вывести качество обслуживания в данной области на совершенно новый уровень.

Существует множество различных подходов к использованию технологий искусственного интеллекта для решения задачи поиска. Одним из возможных путей может стать аннотирование Web страниц дополнительной

структурированной информацией об их содержании, которая в дальнейшем может быть использована поисковой машиной для определения релевантности запросу найденного Internet ресурса. В данной работе рассматривается **онтологический подход** к аннотированию веб-ресурсов.

Онтология – учение о познании (от греч. онтос – сущее, логос - понятие). В инженерии знаний под онтологией понимается детальное описание некоторой предметной области, которое используется для формального и декларативного определения ее концептуализации. Зачастую онтологией называют базу знаний специального вида, которую можно разделять, отчуждать, и самостоятельно использовать в рамках рассматриваемой предметной области [1].

Онтологические системы могут применяться для решения различных задач в сфере искусственного интеллекта, но, пожалуй, наиболее характерной сферой их применения является представление знаний в Интернет. Круг связанных с этим вопросов весьма широк и включает в себя мультиагентные системы, автоматическое извлечение знаний из текстов на естественном языке, поиск информации, интеллектуальное аннотирование, автоматическое составление авторефератов и проч.

Примером общей онтологической системы является CYC®, разрабатываемой фирмой CYCorp [2]. Проект включает в себя создание обширной онтологической системы, описывающей более чем 10^6 концептов и 10^5 аксиом. Для представления знаний фирма разработала специальный язык CYCL. Для вывода по онтологической базе знаний разработана специальная машина вывода. Основная цель этого проекта — построение обширной базы знаний обо всех общих понятиях практически во всех областях человеческой деятельности (common knowledge).

Другим примером использования онтологической системы является инициатива (KA)2 [3] (Knowledge Annotation Initiative of the Knowledge Acquisition Community). Это международный проект, целью которого является организация интеллектуального поиска в Интернет и автоматическое накопление новых знаний. В этой инициативе выделяют следующие направления:

- Аннотация web страниц интеллектуальной информацией.
- Онтологический инжиниринг.

- Организация интерфейса запросов и вывода по распределенной онтологии.

Аннотация Web страниц осуществляется за счет расширения HTML специальным тегом <ONTO>, в рамках которого можно задавать онтологии для спецификации WWW страниц. Процесс дополнения Web страниц такой онтологической информацией рассматривается в рамках направления онтологического инжиниринга. В онтологический инжиниринг входит разработка собственной онтологической системы на основе инструментария Ontolingua. На настоящий момент разработано более десятка онтологий для описания организаций, направлений исследований, рабочих процессов, личностей, продуктов производства и пр. Для поиска в рамках (KA)2 предполагается использовать подсистему Ontocrawler, а для организации запросов и вывода разработан программный продукт Ontobroker, со своим внутренним языком запросов. Существует проекты меньшего масштаба, к ним можно отнести, например, перспективную систему SHOE, разрабатываемую кафедрой информатики университета в Мериленд (Department of computer Science of Maryland University)[4].

Общим для всех систем онтологического аннотирования является то, что в качестве аннотации веб-ресурса выступает специальным образом организованная предметная онтология, которая содержит структурированные знания об аннотированном ресурсе относительно некоторой метаонтологии предметной области. Можно предложить различные способы размещения онтологической информации о ресурсе: включить онтологическое описание в HTML код через введение новых HTML тегов, либо хранить онтологическое описание ресурса в отдельном файле в каком-либо специальном представлении.

Основная задача онтологического подхода состоит в том, чтобы облегчить пользователю поиск информации в большом наборе ресурсов за счет систематизации знаний, создания единой иерархии понятий, унификации терминов и правил интерпретации. Для описания онтологий можно использовать различные языки представления знаний, применяемые, например, в экспертных системах. В данной работе предлагается использовать для составления онтологических описаний фреймово-продукционный способ представления знаний.

Как известно, **фреймы** [5] — средство описания статических знаний, удобное для описания иерархии абстрактных и конкретных понятий, близкое к объектно-ориентированному подходу [6]. Продукции, определенные над множеством фреймов и их слотов, позволяют описывать динамические знания.

В тоже время, применение фреймово-продукционных языков представления знаний в "чистом" виде недостаточно для организации эффективного онтологического поиска. Это видно, например, из следующего соображения.

Пусть всякое онтологическое описание внедрено только в описываемый этой онтологией ресурс. Мы будем считать ресурс известным, если мы в любой момент имеем доступ к его содержимому и, как следствие, к онтологическому описанию. Предположим, что нам уже известно некоторое множество онтологий, и мы хотим указать поисковой системе, какие еще онтологии мы хотели бы сделать известными. Для этого необходимо указать некоторое правило - поисковый запрос, который отделит искомые онтологии от всех остальных, имеющихся в системе. Однако, в общем случае, отсутствует возможность узнать, какие знания содержатся в онтологии до того, как она стала известной.

Таким образом, для поискового запроса не существует никаких явных связей между онтологиями в фреймово-продукционном представлении, кроме отношения наследования между их фреймами[1]. Очевидно, остается только возможность сделать запрос следующего типа: "найти все онтологии, фреймы которых унаследованы от данного известного фрейма и значения слотов которых удовлетворяет некоторому условию".

Как видно, запрос состоит из условий, накладываемых на отношение наследования и на значения слотов наследников. Но в тоже время существует опасность, что слот наследника изменил свой первоначальный смысл, так как в общем случае это уже другой фрейм, который может иметь произвольную структуру.

В данной работе предлагается модифицировать фреймовое представление знаний, явно разделив фреймы-образцы и фреймы-экземпляры, введя требование запретить изменять структуру или применять наследование к фреймам-экземплярам. В дальнейшем мы будем называть **категорией** фрейм-образец, а под **концептом** будем понимать фрейм-экземпляр. Категория во всем эквивалентна

[1] Другие связи определены продукционными правилами. Они неизвестны до тех пор, пока не стала известной сама онтология.

обычному фрейму, кроме того, что значения ее слотов воспринимаются концептами как значения по умолчанию, а концепт соответственно является точной копией своей категории с точностью до значений слотов и безусловных правил, явно присваивающих слоту его значение.

Внедрение подобного подхода позволяет существенно обогатить множество возможных поисковых запросов. Действительно, пусть существует некоторая предметная область и некоторое множество текстовых ресурсов, ее описывающих. Если сосредоточить фактические описания явлений и закономерностей - то есть категорий предметной области в нескольких онтологиях страниц, то появляется возможность искать нужную информацию во множестве страниц посредством поиска онтологий, концепты которых соответствуют требуемым условиям. Теперь условия запроса могут касаться как отношений наследования между категориями или отношений представления между категориями и их концептами, так и условий, накладываемых на значения слотов для концептов известных категорий. В сравнении с предыдущим примером имеется гарантия, что наследники не претерпели никаких метаморфоз, так как концепт нельзя дополнить новыми слотами или продукциями.

Таким образом, открывается возможность разделить поиск информации на два этапа: вначале изучается описание существующих явлений, а затем ведется поиск частных случаев изученных явлений. Это обстоятельство, при условии уникальности используемых имен, дает дополнительное преимущество, состоящее в том, что вводится принудительная унификация понятий в рамках одной предметной области, что исключает возможность двусмысленности поискового запроса.

Язык составления онтологических описаний

Для составления онтологических описаний в рамках создания онтогической поисковой системы Jewel была проведена разработка общего языка описания онтологий. В основе предлагаемого языка лежит фреймово-продукционный язык JFMDL из состава инструментария JULIA (Java Universal Library for Intelligent Applications) [7,8,9], расширенный согласно вышеописанным положениям.

Язык позволяет производить онтологические описания HTML страниц, используя понятия: категория, условное правило, безусловное правило и концепт.

Под онтологией HTML страницы (онтологией части предметной области, описываемой в странице) понимается описание некоторого ресурса, проводимое в терминах общего языка описания онтологий.

В целях повышения эффективности поиска онтологий и непротиворечивости их описания принимаются следующие соглашения:

- Каждая онтология HTML страницы предназначена для непосредственного описания той страницы, на которой она находится. Причем в теле страницы может быть определена только одна онтология.
- Каждая онтология обладает набором предопределенных свойств:
  - именем, которое совпадает с физическим местоположением HTML страницы, в теле которой содержится описание онтологии;
  - списком используемых онтологий (для описания категорий и правил создаваемой онтологии могут применяться категории и правила объявленных используемых онтологий) и их внутренних имен, ассоциированных с ними для удобства;
  - кратким словесным описанием.

Для описания онтологии используется надмножество стандарта HTML, в котором расширяется стандартный тег <SCRIPT>, а также вводятся новые теги <USE>, <CONCEPT>, <SET>, <ASSIGN>. Рассмотрим теперь подробнее теги, используемые в описании онтологий.

Приведем простой пример онтологического описания некоторой предметной области. В качестве предметной области рассмотрим справочник по моделям самолетов, представленный набором HTML страниц — по одной на каждую модель. Мы можем выделить пассажирские и транспортные самолеты. Объединим эти сведения в главной странице - aircrafts.html.

Страница aircrafts.html

```
. . .

<SCRIPT language = ONTODEF>

CATEGORY Firm
{
  SCALAR name;
  SCALAR country;
}
```

```
CONCEPT Ilushin IMPLEMENTS Firm;
SET Ilushin .name  = 'Il';
SET Ilushin .country = 'Russia';


CONCEPT Tupolev IMPLEMENTS Firm;
SET Tupolev .name  = 'Tu';
SET Tupolev.country = 'Russia';

CATEGORY Plane
{
 SCALAR name DEF 'Plane'; // Название самолета

 LIST modifications DEF []; // Список возможных модификаций
 REF firm; // Указатель на концепт, описывающий производителя
 SCALAR type; // Тип самолета (сверхзвуковой/дозвуковой)
 SCALAR speed; //Скорость самолета
}

IF Plane.speed<=1250 THEN Plane.type ='subsonic';
IF Plane.speed>1250 THEN Plane.type ='supersonic';
SET Plane.type = 'speed is unknown';

CATEGORY PassengerPlane EXTENDS Plane
{
  SCALAR passengers; // Число пассажиров
}

CATEGORY TransportPlane EXTENDS Plane
{
  SCALAR mass; // Масса полезной нагрузки
}

</SCRIPT>

. . .
```

Теперь любая страница, содержащая информацию о конкретном самолете, может быть дополнена онтологическим описанием, например, следующим образом:

Страница tu-154.html

```
. . .

<USE 'aircrafts.html' AS aircraft >

<CONCEPT  tu154  IMPLEMENTS @ aircraft ~PassengerPlane>

<ASSIGN tu154.name> Tu-154 </ASSIGN>

<SET tu154. firm = @Tupolev>
```

```
<SET tu154.speed = 900>
<SET tu154.modifications = $['Tu-154A', 'Tu-154M']>
<SET tu154.passengers = 100>

. . .
```

Таким образом, создается возможность для организации предметной онтологии, состоящей из некоторого числа онтологий HTML страниц.

Язык поисковых запросов

Для составления поисковых запросов в системе Jewel применяется специализированный язык, состоящий из следующего набора операторов:

- Оператор **SEARCH** имеет следующую форму:

  SEARCH
  USE 'адрес_1' **AS** имя_1
  . . .
  USE 'адрес_N' **AS** имя_N
  IMPORT LIBRARY имя_библиотеки_1
  . . .
  IMPORT LIBRARY имя_библиотеки_M
  WHERE "условие"

  Под условием понимается логическое выражение, определяющее искомые онтологии. В процессе поиска производится обход всех подходящих запросу онтологий[2], и к элементам каждой из них применяется указанное поисковое условие. В качестве результата возвращаются онтологии, для которых условие истинно.

  Для задания условия могут использоваться следующие предикаты:

  - **INHERITED**(имя_категории) - принимает истинное значение в текущей онтологии, если имеется категория, унаследованная непосредственно от указанной в аргументе. В противном случае предикат принимает ложное значение.
  - **EXTENDS**(имя_категории) - принимает истинное значение в текущей онтологии, если имеется категория, унаследованная (возможно не непосредственно) от указанной в аргументе. В противном случае предикат принимает ложное значение.

[2] Для повышения быстродействия все проверяемые онтологии, при помощи индекса, предварительно отбираются в кандидатное множество.

- **IMPLEMENTS**(имя_категории) - принимает истинное значение в текущей онтологии, если имеется концепт, представленный категорией, указанной в аргументе. В противном случае предикат принимает ложное значение.

Кроме предикатов в условие входят так называемые неявные выражения над концептами. Так, например, выражение (имя_категории.имя_слота > "значение") означает, что выражение будет истинно в случае, если текущая онтология имеет концепт указанной категории, и выражение для его слота истинно (для приведенного примера это означает, что значение, хранимое в слоте концепта, должно быть больше указанного).

Для проверки истинности выражения, при помощи обратного логического вывода, производится вычисление значения слота и последующее сравнение. В случае, если значение слота не вычислимо — выражение признается ложным.

Все выражения и предикаты в условии запроса могут быть связаны логическими операциями AND, OR и NOT.

- Оператор EXTRACT имеет следующие три формы:
  - EXTRACT BASE - возвращает адреса всех зарегистрированных в системе онтологий;
  - EXTRACT ROOT - возвращает адреса всех зарегистрированных в системе онтологий, которые не используют никаких других онтологий;
  - EXTRACT ONTOLOGY 'адрес' - возвращает онтологическое описание страницы, зарегистрированной по указанному адресу.

Рассмотрим более подробно процесс поиска информации в предлагаемой поисковой системе. Допустим, что имеется некоторая предметная область, для которой составлены все необходимые онтологические описания. Ставится задача найти страницу, в тексте которой описан некоторый факт. В терминах, введенных в данной работе, для описания явлений используется понятие категории, а для указания частных случаев явлений — концепты. Таким образом, требуется найти страницу, онтология которой содержит концепт некоторой неизвестной категории. Как видно, в общем случае, вначале требуется найти категорию, описывающую нужное явление. Затем требуется отыскать концепт найденной

категории, описывающий требуемый факт. Онтология, содержащая найденный концепт, будет онтологией искомой страницы. Общий алгоритм поиска для прелагаемой поисковой системы будет сводиться к следующим действиям:

- Определение корня онтологий - именно с коренных онтологий можно начать изучение структуры онтологических описаний в случае, если структура введенной в рассмотрение предметной онтологии неизвестна. Изучение онтологий найденных страниц проводится посредством просмотра с помощью команды EXTRACT ONTOLOGY.
- Изучение описаний известных явлений предметной области до тех пор, пока не будет найдена категория, концепт которой может оказаться искомым фактом. При этом поиск новых онтологий ведется преимущественно с применением предикатов типа IMPLEMENTS, INHERITED и EXTENDS к известным категориям.
- Определение отличительных особенностей искомого концепта и непосредственный поиск концепта исходя из его отличительных особенностей. Поиск онтологии, очевидно, должен вестись с использованием неявных выражений над категориями.

Приведенный алгоритм легко продемонстрировать на ранее приведенном примере. Выделение коренных онтологий командой EXTRACT ROOT даст в качестве результата адрес онтологии страницы aircrafts.html, так как она не использует в своем описании других онтологий. Страницу самолета Ту-154 легко можно найти по названию самолета:

```
SEARCH
USE ' aircrafts.html' AS aircrafts
WHERE (@aircrafts~Plane.name == 'Tu-154') AND (IMPLEMENTS(PassengerPlane))
```

Или зная, например, что искомый самолет дозвуковой и берет на борт до 100 человек:

```
SEARCH
USE ' aircrafts.html' AS aircraft
WHERE (@aircraft~Plane.type = = 'subsonic')
AND(@aircraft~PassengerPlane.passengers = = 100)
```

Последний из вышеприведенных примеров наглядно показывает элемент интеллектуальности проводимого поиска, так как информация о том, что самолет Ту-154 дозвуковой, явно нигде не указывалась, а была выведена логически по

продукционному правилу, общему для всех концептов, прямо или косвенно представляющих категорию Plane.

Легко заметить, что быстрота и качество поиска существенно зависят от качества составления онтологических описаний. Для предметной онтологии, в которой категории разбросаны по слишком большому числу онтологий страниц, поиск затруднен. Однако очевидно, что для большого объема текстов было бы неправильно сосредоточить все категории в одной онтологии, как это было сделано в вышеприведенном примере. Такой шаг может привести к нарушению смыслового разделения между понятиями и повредить точности и выразительности онтологического описания. Это, в свою очередь, негативно скажется на времени поиска, так как пользователь будет вынужден работать с большим множеством получаемых в качестве ответов страниц, аналогично тому, как это происходит при поиске по ключевым словам в классических поисковых машинах. Для более сложных онтологических систем характерно присутствие трех логических уровней:

- Первый уровень - это уровень общих абстракций. Этот слой онтологических описаний объединяет в себе все понятия предметной области и одновременно не проводит никакой конкретизации понятий.
- Второй уровень - уровень описания явлений. В этой части онтологического описания указываются конкретные явления, максимально приближенные к реальности.
- Третий уровень - предметных концептов, т.е. реализация явлений, описанных во втором уровне.

Для небольших онтологических описаний возможно сращивание первого и второго уровня, как в приведенном примере.

Как легко видеть, в таком случае поиск состоит из двух взаимосвязанных частей: поиск описаний явлений и поиск конкретных реализаций этих явлений.

Вопросы реализации

Реализация опытного прототипа системы онтологического поиска Jewel производилась на языке Java. В основу реализации был положен инструментарий JULIA для создания распределенных интеллектуальных систем на основе продукционно-фреймового представления знаний. С использованием технологии

JavaCC[13] были разработаны трансляторы с языка онтологического описания веб-ресурса во внутреннее представление JULIA, а также интерпретатор языка поисковых запросов.

В процессе работы системы онтологические описания вручную (при помощи специальных команд языка запросов) или автоматически (при помощи автономного робота) транслируются во внутреннее представление, которое затем сохраняется в виде семейства индексных файлов или в объектной базе данных. Таким образом, множество известных систем онтологий проиндексировано и сохраняется на поисковом сервере в виде множества фрейм-миров.

Для реализации пользовательского интерфейса реализованы утилита администрирования поисковой системы и предоставляющий пользователю возможность формулировать поисковые запросы Java-сервлет. В качестве продолжения работы над проектом предполагается разработка более удобного диалогового интерфейса с возможностью просмотра множества известных системе онтологий категорий и концептов.

Перспективы применения и развития

В процессе работы над системой Jewel выявились некоторые возможные пути дальнейшего совершенствования методики онтологического поиска.

Рассмотрим улучшения, которым можно подвергнуть язык запросов. Прежде всего, следует отметить, что возможности языка поисковых запросов могут быть существенно расширены за счет введения возможности поиска не только онтологий, но и их составляющих - категорий и концептов. Это позволит создавать вложенные поисковые запросы, увеличив тем самым выразительность языка.

Кроме того, весьма полезным может оказаться введение в язык возможности поиска множества возможных значений для атрибутов категорий, которые принимаются в ее концептах, поскольку частой является ситуация, когда пользователю точно неизвестны нужные значения слотов для выделения искомых концептов из всех существующих.

Немаловажные изменения можно внести в язык составления онтологических описаний. Очевидным недостатком разработанной системы является отсутствие ее способности к обучению, поэтому полезным может оказаться добавление

функций, позволяющих динамически, на этапе выполнения логического вывода, самостоятельно генерировать новые категории и концепты, а также включать их в онтологические описания. Данная функциональная особенность важна для придания системе способности адаптироваться к запросам пользователя и подстраиваться под его интерпретацию онтологического описания, исходя, например, из сопоставления множеств запросов и найденных по ним страниц. Безусловно, для придания системе возможностей самостоятельного совершенствования необходима, прежде всего, обратная связь, дающая системе материалы для анализа результатов своих ответов.

Разработка методов самообучения для поисковой машины может привести к созданию принципиально нового поколения поисковых систем, онтологические описания ресурсов которых совершенствуются в жизненном цикле такой системы без прямого участия человека.

Нельзя обойти вниманием и уже наметившуюся тенденцию к созданию программных средств автоматической генерации онтологических описаний. В основе данных разработок лежит анализ естественного языка. К сожалению, особенно революционных достижений в данной области обнаружить не удается, но наметилась объективная тенденция к росту возможностей таких систем. Прогресс в области естественно-языкового анализа в будущем неизбежно затронет и языки поисковых запросов, что будет приближать разработчиков к созданию более интеллектуальных поисковых систем.

Список литературы

1 Гаврилова Т. А., Хорошевский В. Ф. Базы знаний интеллектуальных систем. - С-Пб.: Питер, 2000. - 384 с.

2 http://www.cycorp.com (02.10.2001)

3 www.ksl.svc.stanford.edu (02.10.2001)

4 Luke S., Heflin J. *SHOE 1.0 Proposed Specificatio*n. - http://www.cs.umd.edu/projects/plus/SHOE (28.09.2001)

5 Минский М. Фреймы для представления знаний. -М.: Энергия, 1979 - 342 с.

6 Буч Г. Объектно-ориентированное проектирование с примерами применения. -М.: Мир, 1992. - 286 с.

7 Soshnikov D. Software Toolkit for Building Embedded and Distributed Knowledge-Based Systems. Proceedings of the 2nd International Workshop on Computer Science and Information Technologies, Ufa, 2000. - pp. 103 - 111.

8 Soshnikov D. Technologies for Building Intelligent web applications based on JULIA Toolkit In Proceedings of the 3rd International Workshop on Computer Science and Information Technologies, Ufa, 2001. - pp. 23-34

9 Сошников Д. В. Инструментарий JULIA для создания распределенных интеллектуальных систем на основе фреймово-продукционного представления знаний. // Труды МАИ. - 2002, № .

10 Gruber. T. Towards Principles for the Design of Ontologies used for Knowledge Sharing // International Journal of Human and Computer Studies. - 1995, №43(5/6). - pp. 907-922.

11 Stoffel K., Taylor M., Hendler J. Efficient management of very large ontologies. *Proc. of Fourteenth American Association for Artificial Intelligence Conference (AAAI-97*), Menlo Park, CA, AAAI/MIT Press.Villemin F. - 1997 - pp. 12-21.

12 Brachman R., Levesque H. The tractability of subsumption in frame-based description languages. *Proc. of the National Conference on Artificial Intelligence (AAAI-1984*), Menlo Park, CA, AAAI/MIT Press - 1984. - pp. 34–37.

13 http://www.webgain.com/java_cc (13.10.2001)

Сведения об авторах

Сизиков Евгений Владимирович, студент 6-го курса факультета прикладной математики и физики Московского государственного авиационного института (технического университета)
e-mail: sizikov@mail.ru

Сошников Дмитрий Валерьевич, старший преподаватель, аспирант кафедры вычислительной математики и программирования Московского государственного авиационного института (технического университета)
e-mail: dmitri@soshnikov.com