ИСПОЛЬЗОВАНИЕ СЕМАНТИКИ В ПОИСКОВЫХ СИСТЕМАХ

С. В. Перминов

Санкт-Петербургский институт информатики и автоматизации РАН

СПИИРАН, 14-я линия ВО, д. 39, Санкт-Петербург, 199178

<sv.perminov@gmail.com>

УДК 004.4

Перминов С. В. **Использование семантики в поисковых системах** // Труды СПИИРАН. Вып. 6. — СПб.: Наука, 2008.

Аннотация. *В статье описана система семантического поиска, использующая базу данных, а именно множество семантических классов для русского языка. Реализован прототип системы, осуществляющий сбор и анализ ресурсов из сети Интернет с последующим поиском в собранной информации. Поисковым запросом к системе является полнотекстовый ресурс. Приведен краткий обзор существующих и исследуемых методик поиска.* — Библ. 8 назв.

UDC 004.4

Perminov S. V. **Using semantics in search systems** // SPIIRAS Proceedings. Issue 6. — SPb.: Nauka, 2008.

Abstract. *Semantic search system that uses database is described. Prototype of system is implemented, it has database of Russian language semantic classes, crawls and parses resources from Internet and finally searches in crawled information. Search request is full text resource here. Short review of current search methods and methods being researched is made.* — Bibl. 8 items.

1. Введение

Во многих случаях традиционные поисковые системы Интернета (например, Google, Yandex) предоставляют релевантную запросу информацию. Другими словами, есть круг поисковых задач, в которых поиск информации по ключевым словам достаточно эффективен, что позволяет не использовать семантические методы. К таким задачам относятся:

- поиск текста песни по известной строчке из нее;
- поиск текста научной статьи по ее аннотации;
- поиск произвольного текста, если известен его некоторый фрагмент.

В то же время алгоритмы ранжирования интернет-страниц (например, PageRank [1]) позволяют найти наиболее популярные ресурсы на заданную в поисковом запросе тему или сайты организаций. Например:

- на запрос «РАН» поисковая система с высокой вероятностью вернет ответ, содержащий ссылку на сайт Российской академии наук: <http://www.ras.ru/>;
- на запрос «Открытые системы» — ссылку на сайт издательства «Открытые системы»: <http://www.osp.ru/>;
- на запрос «Новости» — ссылку на сайт государственного информационно-аналитического агентства Российской Федерации: <http://www.rian.ru/>.

Остается понять, нужны ли вообще алгоритмы семантического поиска и какие задачи они позволят решить. На мой взгляд, они нужны для решения более специализированных задач, нежели как универсальные поисковые системы. Например, когда возникает необходимость поиска неформализованного текста на естественном языке, удовлетворяющего поисковому запросу по смы-

словому содержанию, при том что в общем случае не требуется синтаксического совпадения слов в запросе и предоставляемых системой ответах. Рассмотрим некоторые из исследований в этой области.

Для того чтобы осуществить попытку смыслового поиска в существующих web-документах, необходим предварительный семантический анализ их содержимого, его основная цель — улучшить структурирование информации в документах [8]. Семантический анализ развивает идею распознавания образов, другими словами чрезвычайно сжатых представлений документов, в которых не учитывается ни их конкретное содержание, ни лексика. Можно сказать, что семантический поиск — это поиск по ключевым понятиям, а семантическое представление документа — это множество присутствующих в нем понятий или семантических категорий. Схожий взгляд на семантический анализ предложен в [4], где с этой целью для текста строится понятийная иерархия как совокупность упорядоченной последовательности слов.

Другой подход к семантическому поиску, предложенный автором в [5], заключается в использовании как предварительно формализованных семантических данных для поиска, так и множества формальных поисковых запросов. Здесь поиск рассмотрен в контексте информационных аномалий. Этот подход имеет общие черты с технологией «Интегрированная система информационных ресурсов» [6], но в отличие от нее требованием предлагаемого подхода является предопределение формальных поисковых запросов.

Описанные выше методики можно обозначить как семантический анализ неструктурированной информации, к которой на сегодняшний день можно отнести и множество документов на языке HTML. Эти исследования в основном делают попытку решения трудных для формализации задач.

2. Система семантического поиска

Рассматриваемая поисковая система может быть использована на множестве всех документов сети Интернет, но при этом имеет узкую специализацию, а именно позволяет с той или иной степенью эффективности осуществлять сравнение одних полнотекстовых документов (статей, литературных текстов, стихотворений) с другими с целью нахождения наиболее похожих по «смыслу», чем ограничивается область решаемых ею задач. Она не пригодна для решения задачи более релевантного поиска по ключевым словам. Интересные результаты данная поисковая система может предоставить только в том случае, если в качестве поискового запроса ей поставляется текст относительно большого объема (от 500 слов). Под «смыслом» здесь понимается некоторое формализованное представление текста, попытка передать обобщенную и пригодную для машинной обработки информацию о его смысловом содержании.

Основная задача любой поисковой системы состоит в приведении множества поисковых запросов ко множеству поисковых ответов. Поисковыми запросами для нее являются пользовательские или клиентские запросы (при этом в некоторых случаях клиентами могут быть другие поисковые системы), а поисковыми ответами — информационные представления web-документов (обычно это URL-адрес и фрагмент текста), в которых производился поиск. Традиционные системы синтаксического поиска (например, Google) решают упомянутую задачу при помощи различных статистических алгоритмов. Система семантического поиска также использует статистический алгоритм, однако в основе ее работы лежит не он, а база, содержащая в себе некоторое представление зна-

ний. В нашем понимании к такой базе можно отнести любое множество данных, если оно содержит в себе ту или иную лингвистическую информацию (слова, термины...) и подготовлено (формализовано) для адекватной и однозначной интерпретации вычислительной машиной. Это могут быть электронные словари, классификации, базы данных и др. Это не может быть текст на естественном языке, так как сам по себе он не может быть однозначно интерпретирован машиной. Стоит отметить, что такой текст также не всегда может быть однозначно интерпретирован человеком из-за многозначности слов. В данной реализации в качестве базы с представлениями знаний использован словарь В. А. Тузова [7], содержащий в себе информацию о соответствии между множеством слов (точнее словоформ для каждого из слов) на русском языке и множеством семантических классов.

```
(0099)  $10          Жизнь
(0100)  $100/        Жизнь /
(0101)  $100/0       Жизнь / Бытие-Небытие
(0102)  $100/1       Жизнь / Восприятие
(0103)  $100/11      Жизнь / Восприятие Зрение
(0104)  $100/111     Жизнь / Восприятие Зрение Зрячий-Слепой
(0105)  $100/112     Жизнь / Восприятие Зрение Дальнозоркий-Близорукий
(0106)  $100/113     Жизнь / Восприятие Зрение Заметный-Незаметный
(0107)  $100/114     Жизнь / Восприятие Зрение Обозримый-Необозримый
(0108)  $100/12      Жизнь / Восприятие Слух
(0109)  $100/13      Жизнь / Восприятие Обоняние
(0110)  $100/14      Жизнь / Восприятие Осязание
(0111)  $100/2       Жизнь / Питание
(0112)  $100/3       Жизнь / Дыхание
(0113)  $100/31      Жизнь / Дыхание Вдох-Выдох
(0114)  $100/4       Жизнь / Развитие
(0115)  $100/5       Жизнь / Сфера_деятельности
```

Рис. 1. Фрагмент описания семантических классов словаря В. А. Тузова.

Семантический класс представляет собой некоторую сущность или понятие, имеющие определенный смысл для человека. Возьмем, к примеру, класс «Огонь». Ему могут соответствовать слова «гореть», «поджигать», «обжечь», «тушить» и др. Одному слову может быть поставлено в соответствие несколько семантических классов. Например, слово «кинофестиваль» может быть связано с классами «Кино» и «Фестиваль».

Система семантического поиска подразделяется на две основные подсистемы: подсистему сбора и анализа информации и подсистему поиска. Обе подсистемы представляют собой программный канал, принцип работы которого соответствует принципу работы «pipe-line» в системе Unix.

2.1. Подсистема сбора и анализа информации

Входные данные $URL = \{url_1, url_2, .., url_n\}$, (1)

где URL — множество URL-адресов; $n \in N$; url — URL-адрес.

Выходные данные $db = (H, avg(H))$, (2)

где db — база данных; $H = \{hist_1, hist_2, .., hist_m\}$; H — множество гистограмм;

Труды СПИИРАН. 2008. Вып. 6. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2008. Issue 6. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

$hist$ — гистограмма семантических классов; $hist_i = \{k_{i1}, k_{i2}, .., k_{ip}\}$; $\sum_{j=1}^{p} k_{ij} = 1$;

$$avg(H) = avg\{hist_1, hist_2, .., hist_m\} = avg\begin{bmatrix} k_{11} & k_{12} & .. & k_{1p} \\ k_{21} & k_{22} & .. & k_{2p} \\ .. & .. & .. & .. \\ k_{m1} & k_{m2} & .. & k_{mp} \end{bmatrix} =$$

$$= \left\{ \sum_{i=1}^{m} k_{i1}, \sum_{i=1}^{m} k_{i2}, .., \sum_{i=1}^{m} k_{ip} \right\} = \{kavg_1, kavg_2, .., kavg_p\}$$ — функция, вычисляющая усредненную гистограмму по заданному множеству гистограмм;

$m \in N$; $0 < m < \infty$;

$p = const \in N$; p определяется числом используемых семантических классов (в текущей реализации = 1605);

Программный канал $pipe_1 =$

$$= urlsExtension \mid filterUrls \mid map(baseService) \mid filterHist \mid addAverage, \quad (3)$$

где символ «|» означает, что выходные данные функции (модуля), расположенной левее символа, передаются функции (модулю), расположенной правее символа, в качестве входных данных;

urlsExtension — производит анализ содержимого ресурсов по URL-адресам, извлекает из содержимого (HTML-документов) новые адреса и возвращает дополненное множество URL-адресов;

filterUrls — исключает из списка адресов те, которые не удовлетворяют критериям, заданным при помощи регулярных выражений;

$map(baseService)$ — применяет базовый сервис к каждому элементу из множества URL-адресов;

filterHist — исключает из множества полученных гистограмм те, которые не содержат данных; такая ситуация возникает, если в исходном тексте нет слов, которым можно сопоставить семантические классы;

addAverage — добавляет ко множеству гистограмм среднюю $avg(H)$ и возвращает базу данных db в качестве результата (см. (2));

$baseService = download \mid extractText \mid getWords \mid transform \mid getHist$;

download — загружает ресурс по URL-адресу и возвращает содержимое ресурса;

extractText — возвращает текст, извлеченный из содержимого ресурса;

getWords — извлекает множество слов из текста;

transform — извлекает множество слов из текста;

getHist — строит гистограмму *hist* классов на основе информации о множестве семантических классов, полученных на предыдущем этапе.

2.2. Подсистема поиска

Входные данные *urlreq*, (4)

где *urlreq* — URL-адрес поискового запроса.

Труды СПИИРАН. 2008. Вып. 6. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2008. Issue 6. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

Выходные данные $RESP = \{resp_1, resp_2, .., resp_m\}$, (5)

где $resp_i = (r_i, url_i)$, $i = \overline{1..m}$, $m \in N$; $resp_i$ — ответ поисковой системы;

url_i — URL-адрес из базы данных, полученной на выходе подсистемы 1;

r_i — коэффициент релевантности семантики ресурса, связанного с URL-адресом, поисковому запросу.

Программный канал $pipe_2 = baseService \mid compareWithDB$, (6)

где $compareWithDB$ — осуществляет сопоставление гистограммы поискового запроса с гистограммами из db и возвращает множество ответов $RESP$.

3. Реализация и тестирование

Система реализована с использованием традиционного императивного языка программирования (Java) и двух декларативных языков: JavaCC (язык описания грамматик) и Haskell (язык функционального программирования) [2].

В процессе тестирования системе в качестве поисковых запросов передавались URL-адреса не проанализированных ранее ресурсов, и проводилась оценка на основе метрики релевантности r ответов на запросы:

Релевантность $r_i = \left| \frac{diff_i * length(W)}{2\sum_{j=1}^{p} w_j} - 1 \right|$ (см. (5)), (7)

где $length(X)$ — функция, возвращающая количество элементов во множестве X;

$W = \{w_1, w_2, .., w_p\}$; W — множество весов;

$diff_i = \sum_{j=1}^{p} \left| w_j (kreq_j - k_{ij}) \right|$ (см. (4));

$histreq = \{kreq_1, kreq_2, .., kreq_p\}$; $histreq$ — гистограмма поискового запроса;

$w_j = \frac{a}{kavg_j} + b$; w_j — вес семантического класса (см. (2)); данное гиперболическое выражение определяет обратную зависимость частоты появления слов, соответствующих одному семантическому классу, от его веса;

$a = \frac{y_1 - y_2}{\frac{1}{x_1} - \frac{1}{x_2}}$; $b = y_1 - \frac{a}{x_1}$; $x_1 = 0.0005$; $x_2 = 0.03$; $y_1 = 1$; $y_2 = 0.05$;

x_1, x_2, y_1, y_2 — задаваемые коэффициенты.

Для решения задачи функционального тестирования прототипа в том числе использовалась база данных, заполненная информацией по двум темам: психологические проблемы и микропроцессоры. При полнотекстовом запросе из области психологии все первые ответы, возвращаемые поисковой системой, оказались адекватны поисковым запросам, т.е. имели отношение к сфере пси-

хологии. Коэффициент релевантности устойчиво снижался при уменьшении числа слов в запросе до 25, но оказался неустойчив при дальнейшем уменьшении числа слов, например: Значение коэффициента релевантности первого ответа поисковой системы для запроса, состоящего из 160, равнялось 0,8064, для запроса из 80 слов — 0,7525, из 25 — 0,7057, из 10 — 0,7444. Данное наблюдение подтвердило предположение о том, что система эффективна лишь при «длинных» запросах. Такая же ситуация наблюдалась и тогда, когда запрос был связан с микропроцессорными технологиями.

В целом в ходе тестирования было выяснено, что:

- короткие запросы, длиной менее 70-100 слов, могут обрабатываться некорректно;
- адекватные результаты появляются при значении коэффициента релевантности не менее 0,65 при достаточно большой длине запроса;
- при увеличении количества слов в текстовом запросе увеличивается не только коэффициент релевантности, но и количество адекватных результатов по данному запросу, если такие есть в базе данных;
- разность между значениями коэффициентов релевантности первого адекватного результата и первого из результатов, не имеющих отношение к предметной области, становится достаточно велика при большой длине запроса.

4. Использование онтологий и формирование специализированных поисковых систем

Так как рассмотренная поисковая система использует для своей работы формализацию текста на основе иерархии семантических классов, а иерархическое дерево является частным случаем онтологии, она вполне пригодна для использования в контексте технологий Semantic Web, а результаты ее работы могут быть скомпонованы с другими семантическими ресурсами для дальнейшего анализа и уточнения результатов поиска. Данное утверждение повторяет подход к созданию онтологий, изложенный в [3], согласно которому процесс создания онтологий состоит из следующих этапов:

1. Идентификация и описание ключевых концепций и отношений в исследуемой области;
2. Кодирование онтологии на формальном языке;
3. Интеграция одной онтологии с другими через повторное использование концепций.

Рассмотрим возможности применения данного подхода к построению поисковых систем, специализированных по рассматриваемой области и решаемым задачам. Например, при решении задачи поиска информации в текстах законов, а точнее ссылок на законы разных уровней власти, ключевые концепции будут включать в себя понятия: *Статья, Номер статьи, Название статьи, Дата публикации, Ссылка*; при поиске в Интернет-предложениях о продаже недвижимости такими концепциями будут *Предложение, Адрес, Площадь, Цена* и т.д. Таким образом, при разработке любой из таких систем, одна из задач будет состоять в преобразовании исходной информации (текстов, таблиц из сети Интернет, из XHTML и других форматов) в формат, содержащий ключевые концепции предметной области или онтологию. Здесь (для решения задач второго и третьего этапа) можно использовать форматы, разработанные консорциумом

W3C специально для применения в рамках Web, а именно RDF (Resource Description Framework — среда описания ресурсов) и OWL (Ontology Web Language — язык описания онтологий), основным преимуществом которых является их относительная зрелость инструментальная поддержка (Protégé, Jena), что позволяет использовать их как эффективное средство интеграции баз данных подобных поисковых систем. Интеграция между онтологиями здесь возможна еще и потому, что при построении различных систем могут быть использованы одни и те же понятия, однозначно интерпретируемые машиной на уровне языка RDF. Этому способствует значительное количество публично доступных и популярных словарей RDF-понятий **(**DC (Dublin Core — дублинское ядро), FOAF (Friend of a Friend — друг друга)**)**.

5. Заключение

Синтаксический метод поиска эффективен для решения ограниченного круга задач. В остальных случаях может быть полезен семантический подход к проблеме поиска информации. В настоящей статье предложен способ семантического поиска на основе использования предопределенной базы данных, а именно множества семантических классов для русского языка. Одно из интересных для исследования свойств системы — неограниченная длина поискового запроса, что позволяет осуществлять поиск и сравнение текстов по текстам. Теоретически увеличение размера текста увеличивает точность его семантического анализа и результатов поиска. Для практического использования системы необходимо провести ее объемное тестирование на основе формальных критериев оценки результатов семантического поиска. Кроме того, возможно дополнение существующей базы данных и поиск новых, ориентированных на решение более ограниченных, специализированных задач.

Литература

1. *Brin S., Page L.* The anatomy of a Large-Scale Hypertextual Web Search Engine // Computer Networks. 1998. Vol. 30. P. 107–117.
2. *Hudak P., Hudges J., Peython Jones S., Wadler P.* A history of Haskell: being lazy with class // Proceedings of the third ACM SIGPLAN conference on history of programming languages. 2007. P. 1–55.
3. *Uschold M., Gruninger M.* Ontologies: principles, methods and applications // Knowledge Engineering Review. 1996. No 11. P. 15–93.
4. *Александров В. В., Кулешов С. В., Цветков О. В.* Цифровая технология инфокоммуникации. Передача, хранение и семантический анализ текста, звука, видео. СПб.: Наука, 2008. 244 с.
5. *Перминов С. В., Афанасьев С. В.* Семантический способ поиска информационных аномалий через Web // Труды СПИИРАН. Вып. 3, т. 1. СПб.: Наука, 2006. С 279–287.
6. *Серебряков В.* А. Интегрированная система информационных ресурсов. Архитектура, реализация, приложения. М.: Вычислительный центр РАН им. А. Дородницына, 2004. 240 с.
7. *Тузов В.* А. Компьютерная семантика русского языка. СПб.: Изд-во С.-Петерб. ун-та., 2004. 400 с.
8. *Шумский С. Я.* Интернет разумный // Открытые системы. 2001. №3. С 43–46.