УДК 681.31

О.И. Сафонов

Информационный поиск в Internet

Предметом нашей статьи является рассмотрение вопросов, связанных с функционированием информационно-поисковых систем (ИПС) в Internet. В статье мы намеренно будем рассматривать вопросы поиска информации в основном только в одной службе Internet, а именно WWW как наиболее массовой и собственно породившей взрыв интереса к Internet непрофессиональных пользователей. В статье сделана попытка дать рекомендации по осуществлению информационного поиска в Web.

Как наилучшим образом организовать поиск документов в Web? Наиболее простым, а часто и наиболее эффективным является свободный поиск по ссылкам. Необходимо задуматься на каком из серверов может содержаться искомая информация. Например, если мы ищем информацию об авторе учебного пособия «Методика составления обучающих программ» Н.Ф. Талызиной и из аннотации знаем, что она работает в МГУ, то можем предположить, что на сервере МГУ будут соответствующие документы. При этом надо учитывать, что, как правило, адреса Web-серверов крупных коммерческих компаний, учебных заведений формируются следующим образом:

www.<название_заведения>.<регион>.

Например, Web-сервер МГУ имеет адрес www.msu.ru. Но свободный поиск по ссылкам будет эффективным лишь в ограниченном количестве случаев. Какие же средства для поиска предусмотрены в Internet?

В настоящее время наиболее популярны два средства поиска в WWW: web-каталоги (www.yahoo.com, www.weblist.ru, www.list.ru, www.ru, www.09.open.by, www.belresource.com.by) и поисковые машины (www.altavista.com, www.lycos.com, www.rambler.ru, www.yandex.ru).

Web-каталоги представляют собой структурированные по тематическому признаку гипертекстовые ссылки на документы. Индексирование документов, что в данном случае означает отнесение их к тому или иному разделу производится вручную. Один и тот же документ может быть отнесён к разным разделам. В верхний уровень относят наиболее общие разделы (Образование, Коммерция, Развлечения и т.д.). В разделе «Образование» в свою очередь могут быть другие подразделы, например разделение по регионам или странам. Так, двигаясь по ссылкам, мы приходим к ссылке на интересующий нас документ. Ручная индексация имеет как положительные, так и отрицательные стороны. Из положительных следует отменить высокую степень информативности найденных документов, поскольку перед индексаций, как правило, осуществляется ознакомление с содержанием документа. К отрицательным следует отнести сравнительно малый охват web-документов. Поиском информации в Web-каталогах следует пользоваться только тогда, когда чётко определена информационная потребность и она достаточно широкая. Почти бесполезно пытаться найти в Web-каталоге информацию по узкоспециальной теме.

Поисковые машины или информационно-поисковые системы осуществляют индексацию в автоматическом режиме.

На сегодняшний день существует целый ряд различных информационно-поисковых систем. Хорошо известны названия таких сервисов и информационных служб, как Lycos, AltaVista, InfoSeek, Rambler, Yandex и т.д., без услуг которых сегодня практически невозможно найти что-то необходимое в тера-

байтном море информационных ресурсов Сети. Рассмотрим типовую архитектуру, реализуемую большинством существующих в настоящее время ИПС [1].

Клиент (Client) – это программа просмотра конкретного информационного ресурса. Наиболее популярны сегодня мультипротокольные программы типа Netscape Navigator или Internet Explorer. Такая программа обеспечивает просмотр документов WWW, Gopher, Wais, FTP-архивов, почтовых списков рассылки и групп новостей Usenet. В свою очередь все эти информационные ресурсы являются объектом поиска информационно-поисковой системы.

Пользовательский интерфейс (User Interface) – это не просто программа просмотра. В случае информационно-поисковой системы под этим словосочетанием понимают также способ общения пользователя с поисковым аппаратом – системой формирования запросов и просмотров результатов поиска.

Поисковая машина (Search engine) служит для трансляции запроса на информационно-поисковом языке (ИПЯ) в формальный запрос системы, поиска ссылок на информационные ресурсы Сети и выдачи результатов этого поиска пользователю.

Индекс базы данных (Data base Index) – индекс, который является основным массивом данных ИПС и служит для поиска адреса информационного ресурса. Архитектура индекса устроена таким образом, чтобы поиск происходил максимально быстро и при этом можно было бы оценить ценность каждого из найденных информационных ресурсов сети.

Запросы пользователя (Queries) сохраняются в его (пользователя) личной базе данных. На отладку каждого запроса уходит достаточно много времени, и поэтому чрезвычайно важно запоминать запросы, на которые система дает хорошие ответы.

Робот-индексировщик (Index robot) служит для сканирования Internet и поддержания базы данных индекса в актуальном состоянии. Эта программа является основным источником информации о состоянии информационных ресурсов сети.

Рассмотрим теперь назначение и принципы построения каждого из этих компонентов более подробно и определим, в чем отличие данной системы от традиционной ИПС локального типа.

В традиционных системах используется понятие поискового образа документа – ПОД. Обычно, этим термином обозначают нечто, заменяющее собой документ и использующееся при поиске вместо реального документа. Поисковый образ является результатом применения некоторой модели информационного массива документов к реальному массиву. Наиболее популярной моделью является векторная модель [2], в которой каждому документу приписывается список терминов, наиболее адекватно отражающих его смысл. Если быть более точным, то документу приписывается вектор размерности, равный числу терминов, которыми можно воспользоваться при поиске. При булевой векторной модели элемент вектора равен 1 или 0, в зависимости от наличия или отсутствия термина в ПОД. В более сложных моделях термины взвешиваются – элемент вектора равен не 1 или 0, а некоторому числу (весу), отражающему соответствие данного термина документу. Именно последняя модель стала наиболее популярной в ИПС Internet.

Существуют и другие модели описания документов: вероятностная модель информационных потоков и поиска и модель поиска в нечетких множествах. Не вдаваясь в подробности, имеет смысл обратить внимание на то, что пока только линейная модель применяется в системах Lycos, WebCrawler, AltaVista, OpenText и AliWeb. Однако ведутся исследования по применению и других моделей. Таким образом, первая задача, которую должна решить ИПС, – это приписывание списка ключевых слов документу или информационному ресурсу.

Именно эта процедура и называется индексированием. Часто, однако, индексированием называют составление файла инвертированного списка, в котором каждому термину индексирования ставится в соответствие список документов в которых он встречается. Такая процедура является только частным случаем, а точнее, техническим аспектом создания поискового аппарата ИПС. Проблема, связанная с индексированием, заключается в том, что приписывание поискового образа документу или информационному ресурсу опирается на представление о словаре, из которого эти термины выбираются, как о фиксированной совокупности терминов. В традиционных системах существовало разбиение на системы с контролируемым словарем и системы со свободным словарем. Контролируемый словарь предполагал ведение некоторой лексической базы данных, добавление терминов в которую производилось администратором системы, и все новые документы могли быть заиндексированы только теми терминами, которые были в этой базе данных. Свободный словарь пополнялся автоматически по мере появления новых документов. Однако на момент актуализации словарь также фиксировался. Актуализация предполагала полную перезагрузку базы данных. В момент этого обновления перегружались сами документы, и обновлялся словарь, а после его обновления производилась переиндексация документов. Процедура актуализации занимала достаточно много времени и доступ к системе в момент ее актуализации закрывался.

Теперь представим себе возможность такой процедуры в анархичном Internet, где ресурсы появляются и исчезают ежедневно. При создании программы Veronica для GopherSpace предполагалось, что все серверы должны быть зарегистрированы, и таким образом велся учет наличия или отсутствия ресурса. Veronica раз в месяц проверяла наличие документов Gopher и обновляла свою базу данных ПОД для документов Gopher. В World Wide Web ничего подобного нет. Для решения этой задачи используются программы сканирования сети или роботы-индексировщики. Разработка роботов – это довольно нетривиальная задача; существует опасность зацикливания робота. Робот просматривает сеть, находит новые ресурсы, приписывает им термины и помещает в базу данных индекса. Главный вопрос заключается в том, какие термины приписывать документам, откуда их брать, ведь ряд ресурсов вообще не является текстом. Сегодня роботы обычно используют для индексирования следующие источники для пополнения своих виртуальных словарей: гипертекстовые ссылки, заголовки, заглавия (H1,H2), аннотации, списки ключевых слов, полные тексты документов, а также сообщения администраторов о своих Web-страницах. Для индексирования telnet, gopher, ftp, нетекстовой информации используются главным образом URL, для новостей Usenet и почтовых списков – поля Subject и Keywords. Наибольший простор для построения ПОД дают HTML документы. Однако не следует думать, что все термины из перечисленных элементов документов попадают в их поисковые образы. Очень активно применяются списки запрещенных слов (stop-words), которые не могут быть употреблены для индексирования, общих слов (предлоги, союзы и т.п.). Таким образом, даже то, что в OpenText, например, называется полнотекстовым индексированием, реально является выбором слов из текста документа и сравнением с набором различных словарей, после которого термин попадает в ПОД, а потом и в индекс системы. Для того чтобы не раздувать словари и индексы (индекс системы Lycos уже сегодня превышает 4 Тбайт), применяется такое понятие, как вес термина. Документ обычно индексируется через 40-100 наиболее тяжелых терминов. При этом вес термина, как правило, вычисляется как отношение частотности нахождения термина в данном документе к частотности нахождения термина во всей базе докумен-

тов. Этот факт следует учитывать при организации поиска. Для поиска желательно брать термины наиболее точно характерезующие поисковую потребность и те из них, которые редко встречаются в других предметных областях.

После того как ресурсы заиндексированы и система составила массив ПОД, начинается построение поискового аппарата. Совершенно очевидно, что лобовой просмотр файла или файлов ПОД займет много времени, что абсолютно неприемлемо для интерактивной системы WWW. Для ускорения поиска строится индекс, которым в большинстве систем является набор связанных между собой файлов, ориентированных на быстрый поиск данных по запросу. Структура и состав индексов различных систем могут отличаться друг от друга и зависят от многих факторов: размер массива поисковых образов, информационно-поисковый язык, размещение различных компонентов системы и т.п. Рассмотрим структуру индекса на примере системы, для которой можно реализовывать не только примитивный булевый, но и контекстный и взвешенный поиск. Индекс рассматриваемой системы состоит из таблицы идентификаторов страниц (page-ID), таблицы ключевых слов (Keyword-ID), таблицы модификации страниц, таблицы заголовков, таблицы гипертекстовых связей, инвертированного (IL) и прямого списка (FL).

Page-ID отображает идентификаторы страниц в их URL, Keyword-ID – каждое ключевое слово в уникальный идентификатор этого слова, таблица заголовков – идентификатор страницы в заголовок страницы, таблица гипертекстовых ссылок – идентификатор страниц в гипертекстовую ссылку на эту страницу. Инвертированный список ставит в соответствие каждому ключевому слову документа список пар – идентификатор страницы, позиция слова в странице. Прямой список – это массив поисковых образов страниц. Все эти файлы так или иначе используются при поиске, но главным среди них является файл инвертированного списка. Результат поиска в данном файле – это объединение и/или пересечение списков идентификаторов страниц. Результирующий список, который преобразовывается в список заголовков, снабженных гипертекстовыми ссылками, возвращается пользователю в его программу просмотра Web. Для того чтобы быстро искать записи инвертированного списка, над ним надстраивается еще несколько файлов, например, файл буквенных пар с указанием записей инвертированного списка, начинающихся с этих пар.

Для обновления индекса используется комбинация двух подходов. Первый можно назвать коррекцией индекса на ходу с помощью таблицы модификации страниц. Суть такого решения довольно проста: старая запись индекса ссылается на новую, которая и используется при поиске. Когда число таких ссылок становится достаточным для того, чтобы ощутить это при поиске, то происходит полное обновление индекса – его перезагрузка. Эффективность поиска в каждой конкретной ИПС определяется исключительно архитектурой индекса. Как правило, способ организации этих массивов является секретом фирмы и ее гордостью.

Индекс – это только часть поискового аппарата, скрытая от пользователя. Второй частью этого аппарата является информационно-поисковый язык (ИПЯ), позволяющий сформулировать запрос к системе в простой и наглядной форме. Уже давно осталась позади романтика создания ИПЯ, как естественного языка. Если даже пользователю предлагается вводить запросы на естественном языке, то это еще не значит, что система будет осуществлять семантический разбор запроса пользователя. Проза жизни заключается в том, что обычно фраза разбивается на слова, из которых удаляются запрещенные и общие слова, иногда производится нормализация лексики, а затем все слова связываются либо логическим AND, либо OR. Таким образом, запрос типа:

– Что такое нейронная система? –
будет преобразован в:
нейронная AND система,
что будет означать примерно следующее: найди все документы, в которых слова *нейронная* и *система* встречаются одновременно. Другой подход заключается в вычислении степени близости между запросом и документом. Именно этот подход используется в Lycos. В этом случае в соответствии с векторной моделью представления документов и запросов вычисляется их мера близости. Сегодня известно около дюжины различных мер близости. Наиболее часто применяется косинус угла между поисковым образом документа и запросом пользователя. Обычно эти проценты соответствия документа запросу и выдаются в качестве справочной информации при списке найденных документов.

Наиболее развитым языком запросов из современных ИПС Internet обладает Alta Vista. Кроме обычного набора AND, OR, NOT эта система позволяет использовать еще и NEAR, позволяющий организовать контекстный поиск. Все документ в системе разбиты на поля, поэтому в запросе можно указать, в какой части документа пользователь надеется увидеть ключевое слово: ссылка, заглавие, аннотация и т.п. Можно также задавать поле ранжирования выдачи и критерий близости документов запросу.

Важным фактором является вид представления информации в программе-интерфейсе. Различают два типа интерфейсных страниц: страницы запросов и страницы результатов поиска. При составлении запроса к системе используют либо меню – ориентированный подход, либо командную строку. Первый позволяет ввести список терминов, обычно разделяемых пробелом, и выбрать тип логической связи между ними. Логическая связь распространяется на все термины. Запросы пользователя могут быть сохранены – в большинстве систем это просто фраза на ИПЯ, которую можно расширить за счет добавления новых терминов и логических операторов. Но это только один способ использования сохраненных запросов, называемый расширением или уточнением запроса. Для выполнения этой операции традиционная ИПС хранит не запрос как таковой, а результат поиска – список идентификаторов документов, который объединяется/ пересекается со списком, полученным при поиске документов по новым терминам. К сожалению, сохранение списка идентификаторов найденных документов в WWW не практикуется, что было вызвано особенностью протоколов взаимодействия программы-клиента и сервера, не поддерживающих сеансовый режим работы.

Итак, результат поиска в базе данных ИПС – это список указателей на удовлетворяющие запросу документы. Различные системы представляют этот список по-разному. В некоторых выдается только список ссылок, а в таких, как Lycos и Alta Vista дается еще и краткое описание, которое заимствуется либо из заголовков, либо из тела самого документа. Кроме этого, система сообщает, насколько найденный документ соответствует запросу. В некоторых системах это количество терминов запроса, содержащихся в ПОД, в соответствии с которым ранжируется результат поиска. Система Lycos выдает меру соответствия документа запросу, по которой производится ранжирование. При обзоре интерфейсов и средств поиска нельзя пройти мимо процедуры коррекции запросов по релевантности. Релевантность – это мера соответствия найденного системой документа потребности пользователя. Различают формальную релевантность и реальную. Первую вычисляет система, и на основании этого ранжируется выборка найденных документов. Вторая – это оценка найденных документов самим пользователем. Некоторые системы имеют для этого специальное поле, где пользователь может отметить документ как

релевантный. При следующей поисковой итерации запрос расширяется терминами этого документа, а результат снова ранжируется. Так происходит до тех пор, пока не наступит стабилизация, означающая, что ничего лучше, чем полученная выборка, от данной системы не добьешься.

Как мы выяснили, для построения логических поисковых предписаний в ИПС Internet используется булева логика. Булева логика, реализованная в ИПС Internet, в основном использует три так называемых логических, или булевых оператора – **AND**, **OR**, **NOT** и оператор близости **NEAR**. Использование этих операторов отражено в таблице.

Таблица

Общие правила построения запроса

Выражение в строке запроса	Как понимать запрос
Курс	**Должно** присутствовать слово «курс»
NOT курс ! курс	Логическое **НЕ**. **не должно** присутствовать слово «курс»
курс **AND** рубль курс **&** рубль	Логическое **И**. Должно присутствовать **и** слово «курс», **и** слово «рубль»
курс **OR** рубль курс | рубль	Логическое **ИЛИ**. Должно присутствовать **или** слово «курс», **или** слово «рубль»
курс **NEAR** рубль курс ~ рубль	Функция **ОКОЛО**. Слово «курс» и «рубль» в тексте документа должны располагаться **не далеко друг от друга**
Курс*	Необходимо найти страницы, в которых присутствуют слова, **начинающиеся на** «курс», т.е. «курс», «курса», «курсант» и т. п.
«курс рубля»	Найти страницы, в которых присутствует **словосочетание** «курс рубля»

Операция **NOT** имеет приоритет перед **AND** и **OR**,

Операция **AND** имеет приоритет перед **OR**.

Приведенные логические операторы NOT, AND, OR могут обозначаться иначе. Например, NOT, как знак «-«, AND – «+», OR – «пробел». Для каждой поисковой системы это будет оговариваться отдельно на ее справочной странице.

В запросе можно использовать круглые скобки ( ). Также, как и в арифметических операциях, с их помощью изменяют порядок действий в логических выражениях.

При использовании оператора **AND** фактом является то, что все термины будут присутствовать в документе, хотя это не означает, что они будут расположены в относительной близости и, тем более, связаны каким-либо способом – логически или концептуально. Это просто-напросто означает, что они все вместе встречаются где-то в документе, т.е. все они присутствуют, и не более того. Неверные согласования широко распространены, когда используется логическое **AND**. Логическое **AND** сосредотачивает, координирует и сужает поиск.

При использовании оператора **OR** очевидно, что некоторые документы будут также содержать оба термина, но существенно то, что хотя бы один из двух терминов должен присутствовать, чтобы в результате получить такой

документ. Логическое **OR** используется, чтобы собирать вместе, например, синонимы или альтернативные написания. Логическое **OR** расширяет и раздвигает границы поиска.

Логическое **NOT** нужно использовать только в тех случаях, когда вы абсолютно уверены в том, что хотите исключить некий термин из результата вашего поиска.

Пример:

Термин1 **NOT** *Термин2*.

Такой запрос нашел бы только те документы, в которых упомянут *Термин1*, но нигде вообще нет никаких упоминаний о *Термин2*. Поиск исключит из результата любой документ, в котором упомянут *Термин2*. Это может быть опасно, потому что в некоторых из тех документов, где упомянут *Термин2*, это сделано только вскользь, а в действительности, главным образом, речь идет о *Термин1*? Мы пропустим их совсем, а это может не отвечать нашим поисковым потребностям. Логическое **NOT** используется для исключения чего-либо. Его следует применять с осторожностью. Логическое **NOT** сужает, исключает и ограничивает поиск.

Когда два термина или фразы связаны оператором **NEAR**, поисковая система находит документы, в которых эти термины или фразы присутствуют в тексте в пределах нескольких слов друг от друга. Как правило, это означает, что они расположены в том же самом предложении, или, по крайней мере, в том же самом абзаце.

В отличие от простого логического **AND**, которое требует только чтобы объединенные им термины или фразы совместно присутствовали где-нибудь в документе, оператор **NEAR** удостоверяется, что в тексте они расположены близко друг к другу. Чаще всего в действительности это означает только вероятность появления искомых терминов или фраз в документе в одном и том же контексте, однако достаточно часто при этом обнаруживается и существующая между ними концептуальная связь. Это значительно увеличивает мощность оператора **NEAR** в сосредоточенном поиске по теме. В этом подходе имеется только две неприятности. Во-первых, вы всегда находитесь во власти языка автора, и существуют WWW-авторы, которые наслаждаются шутками. Во-вторых, иногда оператор близости работает не вполне так, как вы могли бы ожидать. Например, поисковая система может найти два термина или фразы в непосредственной близости, но на границе между двумя различными и несвязанными секциями документа. Это случается не так часто, но вполне может ввести в заблуждение неясными результатами поиска.

ЛИТЕРАТУРА

1. ***Храмцов П.*** Информационно-поисковые системы Internet // Открытые Системы, 1996, № 3. С. 19.
2. ***Солтон Дж.*** Динамические библиотечно-информационные системы. М., 1979. С. 71.

SUMMARY

The problems concerning the functioning of the information-retrieval systems (IRS) in the Internet have been considered in this article. An attempt to give some recommendations on carrying out the information retrieval in the Web has been made.

Поступила в редакцию 12.05.2001