ИНТЕЛЛЕКТУАЛЬНЫЙ МОДУЛЬ ПОИСКА ИНФОРМАЦИИ

Р.Г. Разбежкин, В.Н. Филиппов, В.М. Гиниятуллин, М.Н. Назарова

Введение

В настоящее время проблема поиска информации в Интернете стоит очень остро. Сеть, как и всякий объект эволюции, имеет дело с организующим и дезорганизующим началами. Организующее начало проявляется в появлении в сетях различных рубрикаторов, каталогов, различных тематических сайтов и порталов, с ссылками на сайты по некоторой теме. Второе, противоположное, начало способствует хаосу в Сети: сайты повисают в пустоте из-за отсутствия на них ссылок или же ссылки вовремя не удаляются и ведут на несуществующие сайты.

Существующие на настоящий момент поисковые системы позволяют производить поиск в Сети, но результаты поиска в подавляющем большинстве случаев не удовлетворяют требованиям пользователя. Как правило, поисковые механизмы выдают в качестве результатов огромные списки найденных ресурсов, сопровождая их короткими аннотациями.

Работа существующих поисковых систем

С помощью робота, который по ссылкам переходит от одного сайта к другому, поисковая система индексирует содержимое ресурса. Некоторые поисковые механизмы индексируют все содержимое, такие называются полнотекстовыми, другие – только более значимую, по мнению роботов, информацию, например, заголовки, часто встречающиеся слова, выделенные более крупным шрифтом и тому подобное.

Когда пользователь делает запрос, поисковый механизм сравнивает слова и их формы, содержащиеся в запросе и в индексах поисковой системы. Таким образом, на этом этапе результат поиска содержит множество элементов. Чем более банален запрос, тем больше элементов будет содержать ответ и наоборот, чем более специфичен запрос, например, содержит какие-то специализированные термины или понятия, тем меньше будет элементов в результате запроса.

Следующим шагом является упорядочивание результатов по релевантности, то есть помещение более соответствующих запросу ресурсов в начало списка.

На каждом этапе работы поисковой системы есть свои трудности. На этапе сбора информации роботами, поисковый механизм сталкивается с огромным объемом информации Сети, которую он не в силах обработать всю. Многие поисковые системы позволяют владельцам или администраторам сайтов самим регистрировать информацию о своих сайтах, что положительно сказывается на быстроте синхронизации индексов поисковых машин и меняющемуся контенту сайтов.

При поиске ресурсов в индексах, удовлетворяющих введенному пользователем запросу, поисковые системы должны учитывать особенности человеческого языка, что довольно трудно. Многие поисковые системы просто сравнивают написания слов, для таких систем различные формы одного и того же слова являются различными словами. Некоторые все же учитывают формы введенных слов, например, Российский поисковой сервер Яndex (www.yandex.ru). Поисковые системы не производят семантический анализ введенных пользователем запросов и содержимого сайта, таким образом, запрос из одной предметной области может привести к выдаче результата, содержащего ресурсы из совершенно не связанной с требуемой предметной областью.

Когда дело доходит до сортировки результатов по релевантности, начинаются другие проблемы. Прежде всего, из-за того, что поисковая система незнакома с предметной областью проиндексированного документа, она не может корректно определить, насколько документ соответствует запросу. Многие поисковые документы основываются на том принципе, что наиболее значимые слова документа находятся в заголовках или первых строках документа или его абзацев, или же как либо выделены, например, жирным шрифтом или другим цветом. Эти методы далеко не всегда могут дать хорошие результаты. Многие, если не все, поисковые механизмы так же просматривают мета теги, содержащие ключевые слова и краткое описание. Использование мета тегов для определения релевантности дает лучший результат, но не всегда. Иногда авторы не создают мета теги или же наполняют их заведомо ложной информации, чтобы их ресурс занял более высокое место в списке результатов.

Когда сортировка по релевантности завершилась, необходимо каждую ссылку снабдить краткой информацией, чтобы пользователь имел представление, на какой ресурс ведет данная ссылка. Аннотация получается путем реферирования документов, то есть выделением из всего документа нескольких строк, отражающих содержание документа. Разумеется, реферирование происходит автоматически. На составление рефератов вручную не хватило бы ни времени, ни людей, так как в среднем на достаточно крупном поисковом сервере запросы делаются каждую секунду. Реферат не обязательно должен строиться из тех предложений, которые содержатся в реферируемом документе. Такие инструменты, как функция AutoSummarize в Microsoft Office, системы IBM Intelligent Text Miner, Oracle Context и Inxight Summarizer (Компонент поискового механизма AltaVista), безусловно, полезны, но их возможности ограничены выделением и выбором оригинальных фрагментов из исходных документов и соединение их в короткий текст. Подготовка же краткого изложения предполагает передачу основной мысли текста, и не обязательно теми же словами.

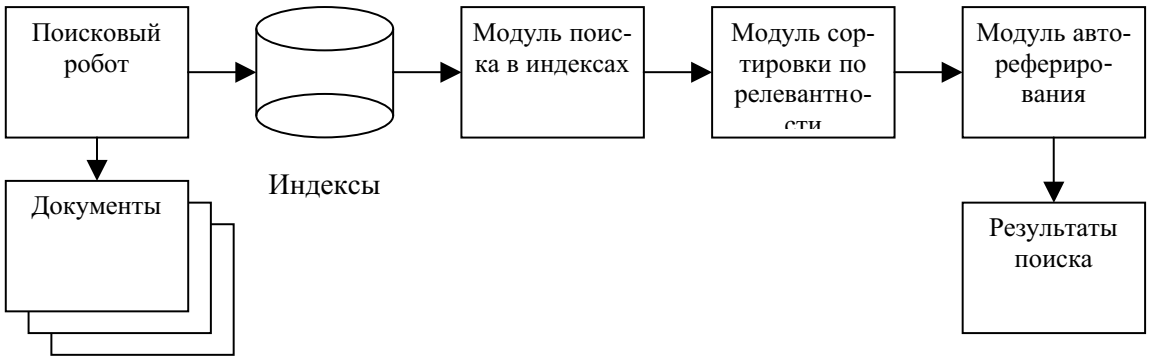

Рисунок 1 - Схема работы традиционных поисковых систем

Другой способ получить аннотацию – вырезать ее из мета тега description, в кото,ром, обычно, хранится краткое описание, если, конечно, автор документа позаботился об этом.

Принцип работы модуля сортировки результатов поиска по релевантности

В данной работе была предпринята попытка создать интеллектуальный модуль сортировки по релевантности. Модуль сортировки по релевантности на-

капливает знания о качестве поиска информации по тому или иному запросу. То есть работа этого модуля основывается на базу знаний о качестве результатов поисков более ранних запросов. Поисковый механизм может определить качество поиска по различным критериям, например, спросить пользователя, насколько удачен получился поиск или по тому, сколько времени он провел на этой странице и т.д. Определив, качественен поиск или нет, система устанавливает связи между запросами и найденными документами, причем вес связи тем больше, чем удачнее (опять же по мнению системы) поиск. Естественно, связи будут устанавливаться только между запросами и соответствующими им документами. При подобных повторных запросах результаты поиска будут выстраиваться согласно тем весам связей, которые хранятся в базе знаний, то есть, чем выше вес связи между введенным запросом и найденными документами, тем ближе к началу списка будет находиться ресурс.

Такая система сортировки документов по релевантности, так же как и нейронные сети, по принципу которых она построена, способна к самообучению, то есть чем больше будет сделано запросов, тем больше знаний получит система, и тем более релевантным будет следующий результат поиска. Разумно предположить, что в самом начале, пока система не обучена, результаты не будут отсортированы по релевантности вовсе, поэтому предложенный механизм сортировки по релевантности следует использовать совместно с другими традиционными механизмами.

Приведем пример. Пользователь вводит запрос «интеллектуальные системы» и получает в ответ 20 ссылок на различные документы. Предположим, что 18 документов из найденных 20 не соответствуют теме запроса, а оставшиеся 2 – в различной степени имеют отношение к нужной пользователю теме. Пользователь, просмотрев все найденные документы, сообщил поисковой системе, что эти два документа – это то, что ему нужно. Другой пользователь делает такой же запрос. Поисковая система, так же как и в первом случае, находит 20 документов, но благодаря накопленным за прошлый запрос знаниям, формирует список таким образом, что найденные ранее два документа оказываются в начале списка.

На рисунке 2 приведена схема работы модуля сортировки результатов поиска по релевантности.

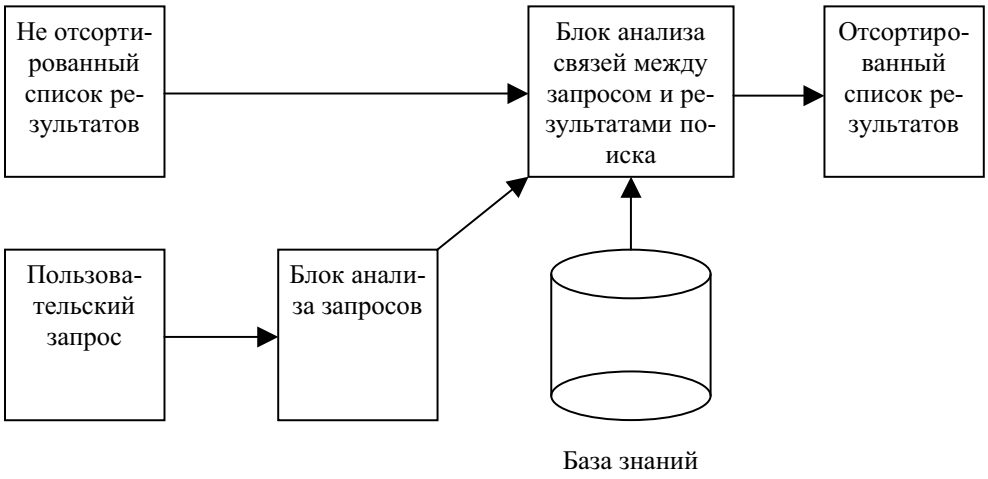

Рисунок 2 - Схема работы модуля сортировки результатов поиска по релевантности

На рисунке так же присутствует блок анализа запросов. Этот блок анализирует запрос, написанный на человеческом языке, и преобразует его в форму, удобную для обработки блоком анализа связей между запросом и не отсортированными результатами поиска, полученными на этапе поиска в индексах. Блок анализа запросов может быть как простым, так и легким. Самый простой способ преобразования – разделить запрос на отдельные слова. Более продвинутый метод, который в принципе достижим уже на настоящем этапе развития информационных технологий, устанавливает соответствие между различными формами одного и того же слова, а так же использует синонимы слов, составляющих запрос. В идеале, конечно же, модуль анализа запросов должен проводить семантический анализ запроса, устанавливать предметную область запроса и уже эти данные передавать блоку анализа связей между запросом и результатами поиска.

Экспериментальная поисковая система

Разработана простейшая поисковая система с упрощенным модулем сортировки результатов по релевантности. В качестве документов для наполнения индексов служили справочные руководства, преимущественно относящиеся к ин-

формационным технологиям. В таблице 1 приведены некоторые параметры эксперимента.

Таблица 1 – Некоторые параметры эксперимента

Формат документов	html, txt
Количество проиндексированных документов	3621
Размер индексов	46,6 Mb
Процессор	Intel Celeron 333
RAM	96 Mb
Операционная система	Windows 2000 Professional
Web сервер	Apache 1.3.6
Сервер баз данных	MySQL 3.21.29a-gamma
Язык программирования скриптов	php3 3.0.13

Операционная система была выбрана из-за удобства работы с ней, благодаря чему было потрачено мало времени на разработку экспериментальной поисковой системы. Предпочтение веб серверу apache было отдано благодаря его большой популярности, более половины сайтов в Интернете обслуживаются этими серверами. База данных mysql выбрана из-за легкости использования и настройки, высокой производительности при работе с малыми базами данных, достигнутой за счет отсутствия в ней некоторых функций, таких как транзакции и триггеры, которые не требовались в этом эксперименте, а так же распространенности среди веб разработчиков. Php3 – язык программирования, написанный специально для разработок в области веб-мастеринга, очень удобен в использовании и поэтому очень популярен среди разработчиков интернет сайтов.

Вся экспериментальная система была максимально упрощена, главной задачей эксперимента являлась проверка эффективности предложенного подхода к организации сортировки результатов поиска по релевантности.

Индексация документов проводилась специально написанным для этого модулем. Документы загружались в базу данных полностью, без предварительной обработки, благодаря чему стал возможен полнотекстный поиск. Модулю индексации передавался путь к директории с документами, которые нужно проиндек-

сировать, он (модуль) выбирал из нее файлы с расширением htm, html и txt и заносил их в базу данных. Таблица, содержащая в себе тексты документов, имеет два поля: поле url, которое является первичным ключом, и по которому осуществлялась индексация базы данных, содержит путь и имя проиндексированного файла; поле content содержит полный текст индексируемого документа. Как уже отмечалось, было проиндексировано свыше трех с половиной документов общим объемом более 46 мегабайт.

Модуль анализа запросов построен максимально просто. Он разбивает введенный пользователем запрос на отдельные слова, которые передает модулю поиска документов в индексах. Очевидно предположить, что построенный таким образом анализ запроса не неэффективен, но конкретно для этого эксперимента его вполне достаточно. Для избежания несоответствий между различными формами одного и того же слова, в запросе использовались только существительные в именительном падеже, единственном числе.

Модуль поиска документов в индексах тоже выполнен достаточно просто. Из слов, составляющих запрос пользователя, которые получены от модуля анализа запросов, составляется SQL выражение, которое производит выборку тех записей из таблицы индексов, чье содержимое поля content, то есть по сути, проиндексированный файл, содержит все введенные пользователем слова не зависимо от их расположения в документе. В терминах поисковых систем, логика запроса всегда является AND, так как в документе должны содержатся все слова из запроса. По уже упомянутым причинам, то есть скорость разработки экспериментальной системы, усложнения этого модуля не требовалось, к тому же этого не требовалось от эксперимента, так как важно было само наличие не отсортированных документов, которые мы будем сортировать написанным для этих целей модулем.

Модуль сортировки результатов поиска по релевантности получает в качестве входных параметров не отсортированный список документов и список слов, составляющих запрос. Этот модуль состоит из двух частей: первая производит выборку всех связей найденных документов со словами запроса из базы данных, вторая производит анализ этих связей и сортирует документы по убыванию

суммарного веса всех относящихся к нему связей. Отсортированный список найденных документов выводится на странице отображения результатов.

Чтобы система могла обучаться, ей необходимо «знать», удачны ли результаты поиска или нет. Предлагаемая система поиска информации включает в себя обратную связь (рис. 3).

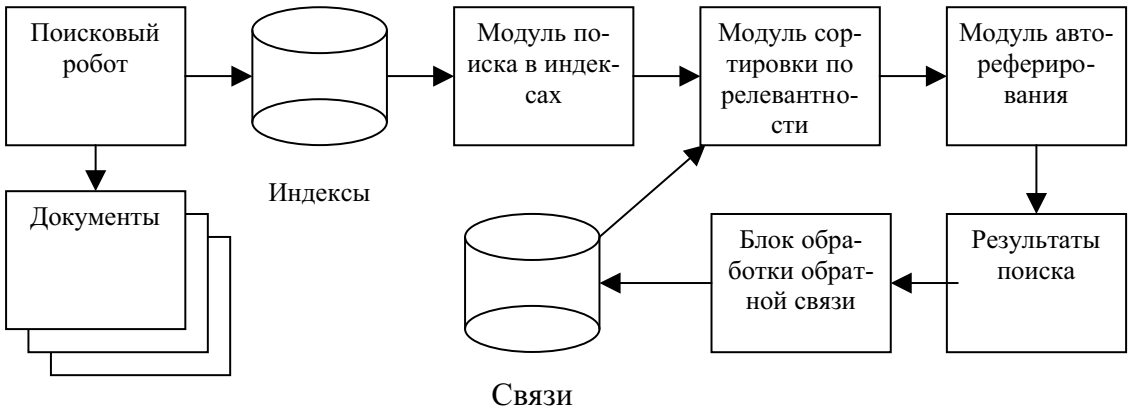

Рисунок 3 – Схема поисковой системы с обратной связью

Если пользователь удовлетворен поиском, то он сообщает поисковой системе, что этот документ соответствует запросу. Система же в свою очередь устанавливает новые или усиливает уже существующие связи между этим документом и пользовательским запросом. Этот механизм позволяет системе самообучаться в процессе работы: чем больше запросов будет выполнено, а точнее чем больше ответов о качестве запросов будет получено от пользователей, тем качественнее будут следующие результаты поиска. В разработанной экспериментальной поисковой системе уже после десятка похожих запросов список документов в результате поиска отсортирован так, что нужная информация редко когда находится после 7-8 строки результатов. А при интенсивности 50 запросов в минуту, как, например, на www.yandex.ru, связи будут устанавливаться более многообразно и прочно, а обучение будет происходить намного быстрее.

Обратная связь в экспериментальной поисковой системе реализована в виде небольших значков, расположенных рядом со ссылкой на найденный документ. Сам документ открывается в отдельном окне. Если открытый документ соответствует введенному запросу, то следует вернуться к окну со всеми результа-

тами поиска и щелкнуть по иконке обратной связи. Несколько неудобно все время переключаться между окнами, но для эксперимента это не главное, основной упор делался на воспроизведение методов поиска. На рисунке 4 приведен пример результатов поиска экспериментальной системы.

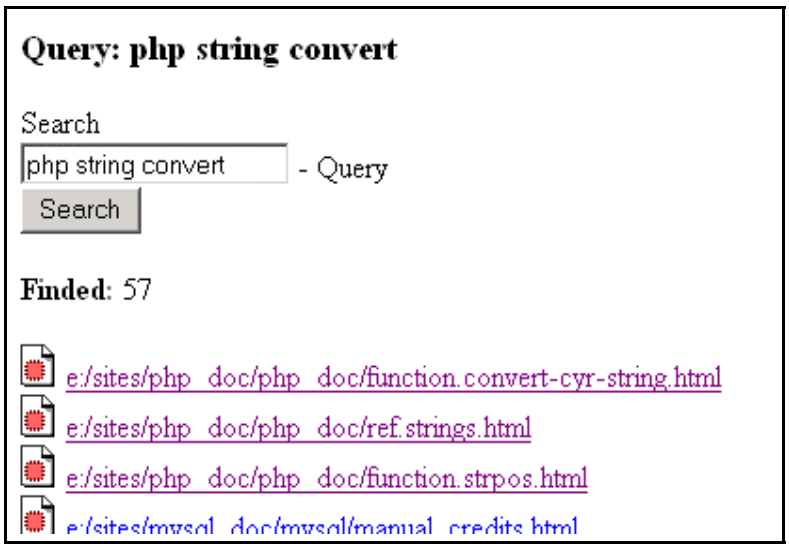

Рисунок 4 - Результаты поиска экспериментальной поисковой системы

При щелчке на иконку обратной связи [ ] запускается скрипт, который устанавливает новые или усиливает уже существующие связи между отдельными словами запроса и url документа.

При тестировании экспериментальной поисковой системы было произведено свыше 400 запросов на различные темы. В некоторых предметных областях, таких как обработка строк в php3, запросы проводились более интенсивно. Поисковая система корректно отображала список по релевантности. В верхней части, обычно, располагались документы с более конкретным содержанием, например, описание конкретной функции, ниже располагались документы более общего содержания, например, список функций определенного типа и так далее. Чистоту эксперимента нарушал тот факт, что экспериментатор уже примерно знал, где находится нужный документ, и не всегда щелкал на тот документ, который в принципе соответствует запросу, но не подходит тематике, например, строковые функции языка perl, а не php3. Была попытка сбалансировать этот недостаток

имитацией поведения человека, незнакомого с документами. Все же этот недостаток оказал несущественно роль на эксперимент.

Проведенный эксперимент показал, что используемый метод сортировки результатов поиска по релевантности достаточно эффективен. Эффективность можно улучшить путем интеллектуализации отдельных модулей, описанных выше. Предположения на счет эффективности оправдались. Принципы работы модуля сортировки по релевантности подтвердились на практике.

Поисковая система учебно-методического сайта кафедры ВТИК

В конце 2000 года был создан учебно-методический сайт кафедры ВТИК. Этот сайт содержит информацию для учебного процесса кафедры, а так же различные службы.

Сайт содержит следующие разделы:

- Новости. Сюда помещаются различные новости кафедры.
- Кафедра. Информация о кафедре вычислительной техники и инженерной кибернетики.
- Информация. Структурированный раздел, содержит различные статьи и книги.
- Методические пособия.
- Лабораторные работы. Содержит задания к различным лабораторным работам.
- Часто задаваемые вопросы.
- Объявления.

Кроме остальных сервисов, этот сайт имеет так же систему поиска. До проведения исследований модуля сортировки документов по релевантности, учебно-методический сайт кафедры ВТИК был снабжен простым, стандартным механизмом поиска, После проведения вышеописанного эксперимента, было решено усовершенствовать поисковый механизм сайта.

В таблице 2 приведена информация о среде учебно-методического сайта кафедры ВТИК.

Таблица 2 – Среда учебно-методического сайта кафедры ВТИК

Операционная система	Linux mandrake 6.0 RE
Web сервер	Apache 1.3.6
Сервер баз данных	MySQL 3.22.32
Язык программирования скриптов	php3 3.0.18

Операционная система выбрана из-за своей надежности, отказоустойчивости и высокой производительности. Остальные компоненты выбраны по тем же причинам, по которым они были выбраны для проведения тестирования экспериментальной поисковой системы.

Так как вся информация сайта хранится в базе данных и имеет строгую структуру, нет необходимости проводить индексирование документов, при поиске информации по запросу поиск нужных данных можно осуществлять непосредственно в таблицах базы данных.

Модуль поиска в индексах, который использовался в экспериментальной поисковой системе, был заменен модулем поиска в таблицах. Так как количество таблиц увеличилось, то пришлось усложнять этот модуль и писать отдельный блок для каждой таблицы. Модуль поиска в таблицах, как и использовавшийся в эксперименте модуль поиска в индексах, производил выборку тех строк таблицы, которые содержали в своих столбцах все введенные пользователем слова.

Модуль анализа запроса остался почти без изменений. Модуль, как и раньше, выделял отдельные слова из запроса. Небольшим усовершенствованием стало то, что слова длиной менее четырех символов не учитывались, это позволило снизить нагрузку на базу данных и ускорить процесс поиска без какого-либо существенного ущерба для результатов поиска.

Модуль анализа связей между адресами документов и словами запроса остался почти таким же, как и в экспериментальной системе, с той лишь разницей, что был усовершенствован с точки зрения программирования.

Модуль сортировки результатов по релевантности так же остался без изменений.

Система обратной связи была модернизирована. Теперь, если пользователь открыл найденный документ, то вес связей между словами запроса и доку-

ментом уменьшается на 1. Если же пользователь подтверждает успешность поиска, то вес связей увеличивается на 10. Это позволяет динамически менять вес связей документов, которые либо потеряли актуальность, либо не соответствуют запросу, либо ошибочно считались релевантными.

Вывод результатов так же претерпел изменения. Теперь ссылка обратной связи расположена не в списке найденных документов, а непосредственно в самом документе, при нажатии ссылка, подтверждающая успешность поиска, исчезает, предотвращая повторное увеличение веса связи.

Заключение

Разработанный метод сортировки результатов поиска хорошо себя оправдал как при работе совместно с экспериментальной поисковой системой, так и в качестве модуля поисковой системы учебно-методического сайта кафедры Вычислительная техника и инженерная кибернетика Уфимского государственного нефтяного технического университета.

Хотя все описанные компоненты поисковых систем являются простыми как с точки зрения программирования, так и в технологическом плане, все же применяемые методы совместно с разработанным модулем интеллектуального поиска информации хорошо себя оправдали.

Литература

1. Разбежкин Р.Г., Филиппов В.Н. Самообучающаяся система поиска информации //Проблемы нефти и газа: Материалы III конгресса нефтегазопромышленников России /Секция автоматизации производственных процессов /Редкол. Ю.М. Абызгильдин и др.- Уфа: Изд-во УГНТУ, 2001.- С. 165.
2. В.Н. Филиппов, Ю.О. Гаррис, Р.Г. Разбежкин. WEB- технологии в обеспечении дополнительного профессионального образования //Материалы Межотраслевой научно-практической конференции “Проблемы совершенствования дополнительного профессионального и социогуманитарного образования специалистов топливно-энергетического комплекса”. Уфа, 23-25 мая 2001 г.: Научные труды. Том 1.- Уфа: Государственное издательство научно-технической литературы “Реактив”, 2001.- С.87-88.