Table of Contents

Продвижение и раскрутка сайта в Яндексе

Раскрутка и оптимизация сайтов в Яндексе имеет ряд особенностей. Яндекс по праву считается самой популярной поисковой машиной в рунете, поэтому основной поток посетителей сайта обычно приходит после продвижения, или раскрутки сайта в Yandex.

Особенности продвижения сайта в Yandex

В начале работы с сайтом необходимо провести регистрацию сайта в поисковой системе Yandex.

Некоторые советы по регистрации сайта

Добавляйте верхнюю страницу вашего сервера - остальные Yandex найдет сам по ссылкам.

Старайтесь не регистрировать в поисковике недоконченные или пустые страницы, страницы без дальнейших ссылок. Дело в том, что страницы такого вида имеют низкий приоритет в очереди робота, следовательно, придется ждать, когда робот наконец посетит этот адрес снова и узнает, что там наконец-то появилось наполнение.

Смотрите на ответ, который выдает вам программа AddURL. Если вы ошиблись (например, в адресе, и такой страницы не существует), то AddURL сообщит вам об этом.

Создайте файл robots.txt, если вы хотите закрыть какие-то разделы от индексирования (например, из соображений секретности).

Если ваша страница была проиндексирована, а затем вы изменили ее содержание или удалили ее, не беспокойтесь - робот автоматически обойдет ее снова и обновит индекс (в случае, если страница больше не существует, она будет исключена из базы и, следовательно, из поиска).

Проверяйте, проиндексирован ли ресурс, не сразу, а через несколько дней после добавления в базу Яndex. Обычно страницы появляются в поисковой базе в течение недели после их появления или изменения. В дальнейшем робот будет самостоятельно (автоматически) находить новые и измененные документы. Причем частота обхода конкретного сервера зависит от зафиксированной роботом частоты изменения его страниц.

Положение сайта в списке результатов при поиске. Зарегистрироваться в поисковой системе - это одно дело, а добиться того, чтобы страницы вашего сайта оказывались ближе к началу списка при запросах - это другое дело. Причем второе намного сложнее первого. Вот что определяет положение сайта на первой странице поисковой системы при запросе по одному слову:

- Частотные характеристики
- Частота слова в базе
- Частота слова в документе
- Размер базы

- Размер документа
- Привилегированное положение слова в документе (например, заголовок) и/или наличие его в списке ключевых слов
- Присутствие слова в “авторитетных” ссылках на данный документ
- “Взвешенный индекс цитирования” документа
- Количество и ранг (“авторитетность”) всех страниц сайта с этим словом

Советы, которые помогут пользователям найти вашу страницу

Задавайте уникальные заголовки документов, вкратце описывающие сайт и текущий документ (но не более 20-25 слов). Слова в заголовках имеют больший вес, чем остальные.

Давайте каждому документу описание в тэге description.

Не забывайте о ключевых словах, по возможности уникальных для каждой страницы.Подробнее о тэге description и ключевых словах читаите в рубрике Тэг META.

Делайте подписи к картинкам в тэге alt.

Чем длиннее документ, тем менее заметны в нем будут слова, заданные в запросе и, следовательно, ваша страница будет ниже в результатах поиска при прочих равных. Старайтесь разбивать длинные документы на более короткие.

Яндекс работает только с текстами и не умеет распознавать графические изображения. Поэтому, если название нарисовано, стоит продублировать его в текстовом виде.

И, наконец, подумайте, по каким словам и фразам вы сами искали бы сайт вашей тематики.

Чего не следует делать при раскрутке сайта в поисковой системе Yandex

Не следует использовать спам на своих страницах. Спам - это заголовки и ключевые слова, сдобренные большим количеством слов из самых популярных запросов, большие массивы текста, “написанные” на странице цветом фона или очень мелким шрифтом, а также многие другие уловки с целью привлечения пользователя на свои страницы обманом. Их не стоит применять по двум причинам. Во-первых, это не добавляет славы создателю страниц и вызывает естественное раздражение пользователей. Во-вторых, Яндекс отслеживает такие ненормальные изменения и снижает место документа на странице результатов. Кроме этого, спам увеличивает размер документа и, следовательно, уменьшает контрастность слов в нем, что также влияет на место документа в списке найденного. В случаях злостного использования спама администрация Яндекс может исключить такие страницы и сайты из базы.

Не следует регистрировать страницы со временем перенаправления на другие страницы равным нулю также исключаются из индексирования.

Не следует использовать страницы с множеством фреймов.

Не используйте нестандартные кодировки

Не следует делать ссылки на страницы вашего сайта с помощью скриптов. Робот работает со стандартными ссылками языка HTML (href, link и frame), то есть так, как работал бы пользователь с отключенными Java и Java script.

Как именно работает поисковая система Яндекс

Задачи поисковой системы

Основная задача поисковой системы - доставлять людям информацию, то есть соединять пользователей с нужными им документами. Причем общение между пользователем и поисковой системой происходит при помощи слов поискового запроса.

Известно несколько классов алгоритмов поиска. Подавляющее большинство из них требуют предварительного индексирования (алгоритмы инвертированных файлов, суффиксных деревьев, сигнатур). В случае прямого поиска индексирование не требуется - поиск производится в лоб, путем последовательного просмотра документов. Поисковая система Яндекса использует индекс, основанный на инвертированных файлах.

Инвертированный файл - концептуально довольно простое понятие, с которым сталкивался в обыденной жизни каждый из нас. Любой индекс базы данных по ключевому полю является формой инвертированного списка. Впрочем, такие списки не обязательно должны быть реализованы на компьютере: существуют бумажные конкордансы текстов российских классиков, то есть словари, в которых в алфавитном порядке перечислены слова, употребляемые писателями, а также указана частота их употребления.

Разумеется, работа с подобным индексом гораздо эффективнее, чем без него. Гораздо проще отыскать нужное слово в конкордансе и посмотреть по ссылкам, где оно употребляется, нежели перелистывать книгу в надежде это слово отыскать.

Конечно, подробный инвертированный индекс может быть довольно большим. Для уменьшения размеров файла обычно прибегают к двум очевидным приемам. Первый заключается в минимизации объема информации, которая хранится в инвертированном файле. Проще говоря, все лишнее удаляется - остается лишь то, что действительно необходимо для подавляющего большинства запросов. Второй прием заключается в указании относительных адресов: для каждой позиции запоминается не ее абсолютный адрес, а разница адресов между текущей и предыдущей позициями. Для пущей эффективности файл упаковывается (коды Голомба и прочие не очень жесткие алгоритмы упаковки), однако эффективные алгоритмы сжатия используются редко - сказывается и отсутствие особого эффекта от сжатия, да и процессорное время, расходуемое на распаковку данных, жалко. Как правило, размер упакованного инвертированного файла составляет от 7 до 30 процентов от исходного текста.

Итак, чтобы что-то найти, поисковая система выполняет два почти независимых процесса: индексирование (получение документов, переработка, сохранение индекса) и поиск. Индекс устроен так, чтобы поиск работал максимально быстро и качественно. Находил все, что нужно, правильно ранжировал и выдавал максимум полезной информации, необходимой для процесса поиска.

Критичным с точки зрения экономики поисковых систем является, как ни странно, поиск, а не

индексирование, так как для ответа на миллионы запросов в сутки, даже прибегая к невероятным ухищрениям, не обойтись без громоздких компьютерных комплексов. Причем, главный фактор, определяющий количество участвующих в поиске серверов, - именно поисковая нагрузка. Это следует иметь в виду при попытке понять всякие странности и неприятные особенности поисковых систем.

Итак, что же происходит с документами при индексировании, а с запросами при их выполнении? Какой путь должны проделать друг к друг документы и запросы, чтобы в конечном счете нужный документ оказался в нужном списке, в том, в котором его ищут самым “нужным” запросом?

Индексирование сайта. Путь документа.

Скачивание

Индексирующую часть поисковиков принято называть роботом. Основная компонента любого робота - модуль скачивания. Так как Сеть - это огромная паутина проводов, модули скачивания лучше запускать параллельно, обычно несколько сотен на одной машине, и одновременно скачивать из разных мест сети разные документы. Скачивать документы по очереди бессмысленно.

Технически модуль скачивания может быть либо мультитредовым (Altavista Merkator), либо использовать асинхронный ввод-вывод (GoogleBot). В любом случае, разработчикам попутно приходится решать задачу многопоточного DNS-сервиса. В Яндексе реализована мультитредовая схема, скачивающие треды называются червями (worms), а их менеджер - погоняльщиком червей (wormboy).

Однако редкий сервер выдержит одновременное “поедание” тремя сотнями червей, поэтому в обязанности диспетчера может входить и слежение за тем, чтобы не перегружать чужой сервер и вообще вести себя вежливо.

Для скачивания робот использует протокол HTTP (иного просто нет, это полный синоним слова “веб”), поэтому многочисленные вопросы вебмастеров: “а что происходит с активными документами”, “а индексирует ли ваш робот Server Side Includes?” - просто-напросто не имеют смысла. Почему?

Суть HTTP-протокола в следующем. Робот передает серверу строчку: “GET /path/document” и иные полезные строки, входящие в HTTP-запрос, а в ответ получает текстовый поток, в начале которого - несколько служебных строк HTTP-заголовка, выдаваемых веб-сервером (непосредственно или с помощью вашего скрипта), а затем уже и сам документ. Это все.

Как формируется документ, из активных или пассивных частей он состоит, робот не знает и знать в принципе не может. Он имеет дело с полностью сформированным потоком, который ему возвращает ваш веб-сервер.

Скачивание может быть организовано на разных принципах: “в ширину”, по цитируемости, тематической локальности, по PageRank, но цель одна - свести до минимума сетевой трафик при максимальной полноте. Поэтому эффективное скачивание - целая наука, которой посвящены центральные доклады на лучших международных конференциях (WWW Conference, VLDB и т. п.).

Тем не менее, у всех модулей скачивания всех поисковых роботов есть общие черты. Во-первых, они подчиняются правилам для роботов, записанным в файле robots.txt, который должен лежать в корне каждого сервера. Там вебмастер может указать желательные и нежелательные области доступа тем или иным роботам (или всем сразу). Контроль поведения роботов возможен и при помощи строчки, помещаемой в документ. Тогда робот будет подчиняться тому, что там написано “по-документно”.

Однако кроме фильтров, устанавливаемых вебмастером, у роботов есть и свои собственные фильтры.

Во-первых, многие роботы опасаются индексировать так называемые динамические документы, формально относя к таковым и документы, содержащие вопросительный знак в URL. Понятно, что это всего лишь “эвристика”, предположение роботов, не более того. Ведь в руках вебмастера есть способы передавать параметры, скрывая CGI-механизм (то есть без вопросительного знака и пар имя_параметра = значение_параметра), например при помощи PATH_INFO или mod_rewrite. И наоборот, масса серверов, использующих CGI-интерфейс, годами выдают исключительно стабильное и “статичное” содержание. Заметьте, что многие роботы (например, Яндекс) на эту эвристику не обращают внимания и индексируют “динамические страницы” так же, как и “статические”.

Во-вторых, каждый робот поддерживает свой собственный список ресурсов: наказанных за спам или отфильтрованных по какой-нибудь технической причине. Об этом мы поговорим чуть позже, а пока лишь подчеркнем, что поисковики, как правило, не берут на себя функцию общественного цензора и не фильтруют “плохое” или “противозаконное” содержание. В лучшем случае они предоставляет подобную фильтрацию как специальный сервис. И здесь мы вплотную подходим к этической проблеме, слишком глубокой для обсуждения в короткой статье. Сформулирую лишь “возможный принцип”: качество поиска информации не связано с качеством самой информации. Поисковик - своего рода зеркало, отвечающее только за качество процесса отражения, но не предметов, которые в нем отражаются.

Отбрасывание повторов на сайте

За передним краем, т.е. за модулем скачивания - стоят другие модули, которые помогают уменьшать трафик, повышать покрытие и обрабатывать такие ресурсы, которые с наибольшей вероятностью “пришла пора скачать”, или же те, которые следует чаще обновлять для поддержания высокого качества поиска.

Прежде всего, это модули хранения URL и ссылок. Они позволяют не скачивать повторно один и тот же URL, обмениваться списком новых URL между разными серверами скачивания или считать полезные метрики цитируемости документов.

Далее. Модули отслеживания дубликатов решают задачу неиндексирования дубликатов, то есть позволяют избегать резкого замусоривания базы повторами. Заметьте, что для корректного сравнения нужно сначала определить кодировку документа, ведь 30 процентов серверов ее не сообщают. Этим занимается специальный модуль определения языка и кодировки, после отработки которого документу может быть приписана кодировка и язык, или же он может быть отфильтрован (еще один вид фильтра), если робот посчитает данную кодировку или язык “чужими” для себя.

Простейшая проверка на повтор содержимого состоит в вычислении контрольной суммы всех слов текста и в тесте базы данных на ее присутствие. Кстати, сразу после получения сигнала о

точном повторе поисковая машина Yandex получает команду не ходить по ссылкам от дубликата: логика такой фильтрации проста и очень популярна у всех роботов. Она построена на естественном предположении, что точно повторяющиеся документы содержат набор ссылок на точно такие же документы, какие уже получены по ссылкам оригинала.

Отдельно стоит проблема учета “слегка измененных” документов (обычно это делается по набору характерных слов или контрольных сумм), а также выявления зеркал серверов. Зеркала представляют собой специальный случай: их не надо индексировать, хотя время от времени надо проверять, не “расклеились” ли они.

Обработка содержания документа

Что значат все эти модули для конкретного документа? Что делает поисковая система с документом после скачивания?

Документ обрабатывается HTML-парсером (есть и другие форматы документов, и многие роботы их поддерживают), освобождающим документ от особенностей представления в этом формате и оставляющим только существенное для поиска: текст, заметные особенности шрифтового оформления, разбивка на абзацы, выделение ссылок и прочие полезные зоны в документе (с точки зрения возможностей поиска); для каждой ссылки запоминается, на какой URL она указывает, и т.д.

В этом этапе скрыта масса нюансов. В современных документах активно используется javascript для динамического изменения содержания, для навигации, CSS-стили для оформления и пр. Полностью интерпретировать все эти элементы слишком дорого, и то, что может позволить себе пользователь (3-5 секунд ожидания), не может позволить поисковый робот, пожирающий до ста документов в секунду. Поэтому все подобные элементы обрабатываются либо упрощенно (настолько, насколько позволяет эффективность алгоритмов обработки), либо вообще игнорируются.

Это не значит, что вебмастер может надеяться на полную гарантию того, что роботы никогда не будут понимать CSS или не ходить по ссылкам через javascript. Во-первых, роботы постоянно развиваются, во-вторых, в каждый момент времени разные роботы ведут себя по-разному. Но все же нельзя сбрасывать со счетов ограниченность поисковых роботов, как и вообще всех невизуальных агентов.

Само индексирование

И наконец, из текста выделяются слова по языково-зависимым правилам (вы не забыли, что язык роботу уже известен?) и на слова “набрасываются” алгоритмами морфологического анализа (те поисковые системы, которые это практикуют) и алгоритмами “собственно индексирования” (инвертирование текста).

Заметьте, что физически все эти этапы могут происходить в разных процессах или даже на разных компьютерах. Всё определяется логикой и функциональностью требуемых процедур и способом их оптимизации.

В результате появляется индекс. Точнее, постоянно накапливается обновляющаяся часть индекса, которая периодически сливается с большим индексом. В Яндексе это происходит два

раза в неделю.

Поисковая машина Яндекс. Путь запроса

Куда идет пользователь после ввода запроса

Итак, индекс построен. К браузеру подсел пользователь. Первым “зашедшего” на поисковый сервер пользователя встречает “умный” маршрутизатор (в случае с Яндексом это Cisco 7200), который переадресует нового пользователя на наименее загруженный веб-сервер. О загрузке веб-сервера устройство узнает через “обратную связь” одним из выбранных в конфигурации способов, например по числу одновременно выполняющихся процессов. С этого момента все запросы, приходящие с данного IP, то есть от данного пользователя, будут прозрачно переадресовываться на соответствующий веб-сервер.

Раздача запроса по серверам

Затем пользователь набирает запрос в окошке и отправляет его на поиск. В Яндексе веб-сервер служит одновременно для слияния результатов поиска от поисковых серверов и источников, в том числе и разнородных: таких как энциклопедии, рекламные объявления Директ, новостная лента, магазинные каталоги, специальная база поиска изображений и т. п. Запрос модифицируется и рассылается на поисковые серверы. Их задача - выбрать документы, удовлетворяющие поисковому запросу, и отранжировать список.

Собственно поиск и ранжирование

Этот процесс теснейшим образом связан с устройством индекса и техническими аспектами выбранной поисковой модели, то есть теми факторами, которые создатели системы считают важнейшими. Яндекс, например, “по умолчанию” ищет все словоформы даже для “несловарных” слов и при этом придает большое значение вхождению слов запроса в одно и то же или соседние предложения. Соответственно его основной индекс устроен по “леммам” и хранит номера слов и предложения для каждого слова в каждом документе.

При этом Яндекс учитывает упоминания слов в заголовках и подзаголовках документа, шрифтовые выделения. Эта информация тоже кодируется в индексе.

Использование ссылок

Особняком стоит учет ссылок на документы. Текст ссылок не только служит источником альтернативной поисковой лексики (например, позволяет находить популярные сайты даже по запросам с опечатками), но и является незаменимым ранжирующим компонентом в так называемых навигационных запросах, когда пользователю требуется перейти на популярный сайт, адрес которого он не знает.

Индекс ссылочного поиска строится отдельными процедурами с учетом пересечения ссылок между кластерами. В ссылочный индекс в Яндексе входят и ссылки из Яндекс Каталога,

который по техническим причинам индексируется чуть полнее и регулярнее, чем другие каталоги.

Введение ссылочного поиска и статической ссылочной популярности (мы называем этот фактор ВИЦ - взвешенный индекс цитирования - аналог известного PageRank) помогает поисковым системам справляться с примитивным текстовым спамом, который полностью разрушает традиционные статистические алгоритмы информационного поиска, полученные в свое время для контролируемых коллекций.

Для подавления примитивного непотистского спама (проставление взаимных ссылок с единственной целью поднять свой ранг) Яндекс использует смешанные автоматические и ручные приемы.

Слияние и группировки

Отдельная тема - ранжирование при слиянии. Для корректного ранжирования баз разного размера и с разной глобальной статистикой слов Яндекс использует оригинальную идею модификации запросов, передаваемых в поисковые источники методом проставления весов для каждого слова на основе глобальной статистики.

Наконец, важный момент - это группировки. Яндекс предоставляет широкие возможности по группированию результатов, он умеет группировать результаты по иерархическому дереву, по сайтам, регионам и пр., причем одновременно. При этом ранг группы (в частности сайта!) в Яндексе не эквивалентен рангу максимально релевантного документа; учитываются все найденные документы, хотя и очень аккуратным образом, чтобы не дать глубоко проиндексированным сайтам необоснованного преимущества.

Архитектура

В Яндексе реализована двухуровневая схема кластеров. Индекс сразу строится в кластеризованном виде, в том, в котором будет использоваться в поиске.