***О.А. Ромейко**, инженер-программист ОДО «Твинслеш»,*
***С.Ю. Михневич**, к.ф.-м.н., заместитель начальника управления высшего образования Министерства образования Республики Беларусь*

Поисковая оптимизация сайтов на основе анализа контента

В статье рассмотрен алгоритм «белой» оптимизации сайтов и его реализация. Алгоритм позволяет решить проблему оптимизации веб-приложений, не прибегая к услугам специалистов. Система, реализованная на основе данного алгоритма, в автоматизированном режиме выдает рекомендации, которые помогут оптимизировать веб-приложение.

Вместе с появлением и развитием поисковых систем в середине 1990-х появилась поисковая оптимизация[1]. Обычно, чем выше позиция сайта в результатах поиска, тем больше заинтересованных посетителей переходит на него с поисковых систем. При анализе эффективности поисковой оптимизации оценивается стоимость целевого посетителя с учетом времени вывода сайта на указанные пози-

[1] Поисковая оптимизация – комплекс мер для поднятия позиций сайта в результатах выдачи поисковых систем по определенным запросам пользователей с целью продвижения сайта.

ции и конверсии сайта [1]. Поэтому множество компаний заинтересованы в поисковой оптимизации. Так по оценке Российской ассоциации электронных коммуникаций рынок поисковой оптимизации в 2014 году составил 10,24 млрд. руб. В 2015 году прогнозируется рост рынка на 19% [2].

На заре интернета поисковая оптимизация была достаточно простой задачей. Для высокого ранга в поисковой системе достаточно было поместить ключевое слово в название и добавить мета-теги. После этого сайт уже можно было с легкостью находить на самых высоких позициях в поисковых системах.

Все это привело к манипулированию ключевыми словами, увеличению плотности ключевых слов на странице, множественному повторению ключевых слов маленькими буквами или тем же цветом, что и фон страницы. Для спаминга использовали meta-тэги, тэг title, подписи к изображениям, которые могут быть совершенно не относящимися к тематике сайта ключевыми словами.

Поисковые системы проанализировали эти манипуляции и предложили разработчикам и SEO-специалистам[2] использовать еще один принцип, который влияет на позицию сайта в результатах поисковой выдачи – количество входящих ссылок на сайт.

Случилось это примерно в то же время, когда появился Page Rank Google – алгоритм который подразумевает, что каждая ссылка на страницу, представляет собой «вес» для страницы. Поэтому чем больше ссылок на страницу, тем больше доверия к ней со стороны поисковой системы. Сконцентрировав все внимание на количестве входящих ссылок, появился еще один вид спаминга. SEO-специалисты и разработчики начали покупать ссылки или производить обмен ссылками для того, чтобы увеличить их количество. Очевидно, что страница, имеющая 1000 ссылок, более авторитетна, чем страница, имеющая 100 ссылок.

Учитывая, что ссылки можно просто купить, поисковые системы вынуждены были пересмотреть свои критерии

---

[2] SEO-специалист – специалист, выполняющий внутреннюю и внешнюю оптимизацию сайта с целью повышения позиций сайта в списке страниц, найденных поисковыми системами по конкретным запросам.

для измерения важности страниц. На этот раз внимание было сосредоточено на релевантности страницы, с которой идет ссылка, и страницы, на которую идет ссылка. Если страницы релевантны, то это считается хорошим признаком в поисковых системах. Если нет, то это плохой признак, а иногда даже потенциально вредный для результатов работы поисковых систем.

Все факторы, влияющие на положение сайта в выдаче поисковой системы, можно разделить на внешние и внутренние. К внутренней оптимизации, касающейся исключительно внутренней системы сайта, относится работа, направленная на общее повышение качества сайта, пользы, которую он приносит посетителю. Сюда можно отнести работу над структурой проекта, над облегчением восприятия контента и непосредственно над качеством этого контента. Значение общего количества таких факторов в большинстве источников колеблется в районе 200 [1].

Внешние факторы делятся на статические и динамические. Статические внешние факторы определяют релевантность сайта на основании цитируемости его внешними веб-ресурсами, а также их авторитетности вне зависимости от текста цитирования. Динамические внешние факторы определяют релевантность сайта на основании цитируемости его внешними веб-ресурсами и их авторитетности в зависимости от текста цитирования [1].

Методы оптимизации можно разделить на три класса: «белые», «серые» и «черные». К «черной» оптимизации относятся все методы, которые противоречат правилам поисковых систем. Серая оптимизация отличается от «черной» тем, что она официально не запрещена, но ее использование все равно может быть расценено как неестественное завышение популярности сайта. Изменения в мире поисковых систем дают понять, что это разделение весьма условно – любая манипуляция определенными параметрами сайта может быть расценена поисковой системой как крайне нежелательное влияние на его результаты. Так, любая попытка манипулирования поисковыми результатами прямо запрещена в лицензии на использование поисковой системы «Google». «Белые» оптимизаторы и маркетологи

пользуются рекомендациями Google по созданию «хороших» сайтов. Таким образом, продвигают сайт, не нарушая правил поисковых систем [1].

Основа поисковой оптимизации – ключевые слова. Пользователи поисковых систем находят нужный сайт, вводя в строке поиска нужное слово или словосочетание, и поисковые системы, выполняя заказ пользователя, принимаются за поиск нужных слов и предложений в проиндексированных ими сайтах. Чем более текстовой контент сайта, по мнению поисковой системы, соответствует запросу, тем выше в результатах поиска система разместит ссылку на ресурс. Здесь и кроется причина и следствие: основной объект приложения усилий специалистов по поисковой оптимизации – позиция сайта в результатах поиска по определенным ключевым словам и словосочетаниям.

В работе представлен алгоритм оптимизации, плотно работающий с ключевыми словами. Разработан прототип сервиса, который выдает рекомендации по оптимизации сайта на основе адреса и требуемых запросов. Реализация этого алгоритма максимально просто и быстро может помочь рядовым пользователям добиться повышения ранга своей страницы.

Реализуемый алгоритм считается алгоритмом «белой» оптимизации, т.е. алгоритмом, в котором не применяются запрещенные и недобросовестные методы продвижения, не нарушаются правила поисковых систем. Веб-страницы, продвигаемые при помощи алгоритма, полезны как для пользователей интернета, так и для поисковых машин, и самих SEO-специалистов.

Основным компонентом функционирования веб-приложения является работа с поисковыми системами. В данном случае в качестве ведущей информационной единицы выступает адрес сайта, иными словами, подразумевается работа с самой страничкой пользователя и текстом на ней.

В качестве источников данных о ранжировании сайтов и данных о ранке сайта выступает APIgoogle.com.

Создание архитектуры подразумевает выделение общих абстрактных компонентов (слоев) и описание функциональности каждого слоя.

Создание приложения в данном случае подразумевает создание двух компонентов: сервера данных – модуля, отвечающего за сбор и унификацию данных из источников, реализацию алгоритма в серверной части и клиентской части в виде веб-интерфейса в браузере, а также обеспечение взаимодействия между указанными модулями.

Алгоритм оптимизации построен на поиске и анализе релевантной страницы по запросу. Анализ происходит путем парсинга страниц[3]: релевантной, TOP-10 и той, которую ввел пользователь для оптимизации. Анализируется абсолютно весь контент в выбранных тегах. Последующие рекомендации выводятся путем сравнения анализированных данных с релевантной страницы, TOP-10 и страницы пользователя.

Рассмотрим более подробно алгоритм. Структура алгоритма представлена на рисунке 1.

Работа алгоритма построена следующим образом: на входе – адрес целевой страницы и запросы, по которым мы хотим подняться в ТОП поисковой системы.

Получив начальные данные, для каждого запроса находим релевантную страницу, т.е. страницу, которая наиболее точно отражает информацию о запросе, и ТОП-10 страниц по запросу.

Для целевой страницы, релевантной страницы и каждой страницы из ТОП-10 проводим своеобразную «чистку». Из кода убираем все лишнее, т.е. скрипты, комментарии, текст внутри noindex. Также нужно убрать все стоп-слова (слова и частицы, которые не индексируются поисковыми системами).

После проведения «чистки», проанализируем каждую из страниц. Нам нужно узнать следующее:

- точное вхождение запроса;
- словоформу запроса (без окончаний, с сохранением порядка слов), с указанием количества вхождения каждой словоформы;
- точное вхождение каждого слова из запроса;

---

[3] Парсинг страниц – последовательный синтаксический анализ информации, размещенной на интернет-страницах.

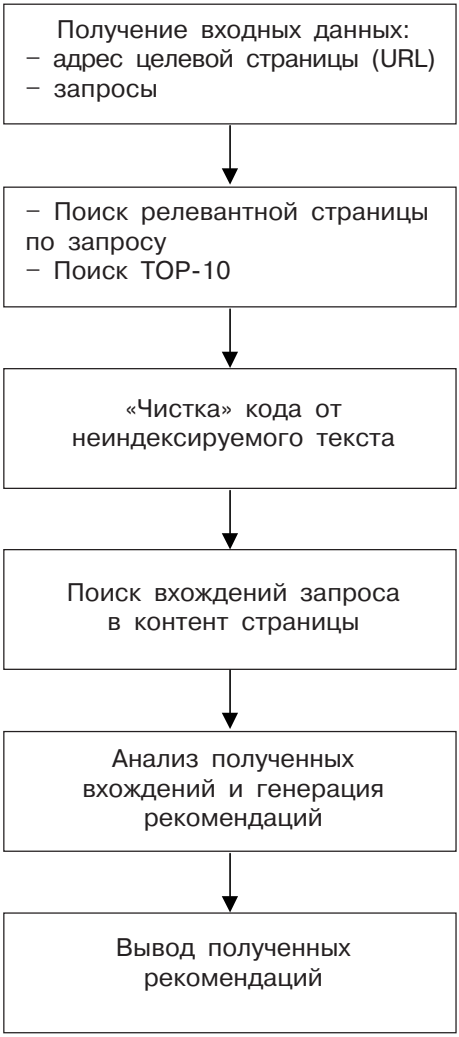

Рис. 1. Структура алгоритма оптимизации

• словоформу каждого слова запроса (без окончания), с указанием количества вхождения каждой словоформы для каждого слова;

• точное вхождение фразы с пропуском одного любого из слов (порядок слов во фразе сохраняется);

• вхождение словоформы фразы с пропуском одного любого из слов (порядок слов во фразе сохраняется), с указанием количества вхождения каждой словоформы;

• точное вхождение фразы с добавлением между словами фразы в любом месте одного случайного слова;

• вхождение словоформы фразы с добавлением между словами фразы в любом месте одного случайного слова, с указанием количества вхождения каждой словоформы;

• точное вхождение фразы с заменой одного из слов фразы на случайное слово (замена происходит в любом слове за исключением первого и последнего);

• вхождение словоформы фразы с заменой одного из слов фразы на случайное слово (замена происходит в любом слове за исключением первого и последнего), с указанием количества вхождения каждой словоформы.

Имея данные анализа страниц, можно получить рекомендации путем анализа усредненных значений релевантной страницы, данных по ТОП-10 и целевой страницы.

Пройдя все стадии работы алгоритма, мы получаем список из рекомендаций по увеличению ранга страницы в поисковой системе. Примерные выходные данные алгоритма, выглядят так:

1. You need to add»request» in <body>...</body> 37 times.

2. You need to delete «request» in <a>... 5 times.

3. You need to add»request» in <li>... 3 times.

Структура разрабатываемого приложения на базе алгоритма показана на рисунке 2. К компонентам приложения относятся:

• веб-приложение, выполняющее роль фронт-офиса для клиентов, где они могут зарегистрироваться, управлять аккаунтом и проектами;

• публичный веб-сервис, реализующий протокол общения между сервисом и клиентом;

• приложение, получающее задачи и по очереди анализирующее запросы.

Для демонстрации принципов архитектуры в целом, скорость разработки сервиса и уровень абстракции являются одними из наиболее важных факторов. Поэтому для

Рис. 2. Структура разрабатываемого приложения

демонстрации работы алгоритма использовалось самое простое приложение [3].

В общем случае на выбор среды и инструментов реализации приложения влияют также такие характеристики как функциональность, качество, скорость и простота реализации приложения.

Применение к веб принципов объектно-ориентированной разработки может оказаться весьма затруднительным. Языки скриптов, широко применяемые в веб, зачастую оптимизированы таким образом, чтобы быстро реализовывать простую функциональность, а не для модульного конструирования больших программ. Кроме того, для таких языков, как правило, не существует мощных сред разработки, привычных для языков общего назначения [4].

Однако объектно-ориентированные методы приобретают все большее значение в веб-разработке по мере того, как веб-приложения становятся все сложнее и интегрируются с традиционными серверными приложениями. Стандартные подходы и методики реализации веб-приложений используются, когда логика встраивается непосредственно в код программы. Поддерживать внутренний интерфейс таких многоуровневых систем становится сложно, тем более, если спроектированная система требует постоянных обновлений и изменений. Порой, чтобы изменить незначительную деталь

в каком-то компоненте, приходится переписывать код других компонентов, не задействованных в процессе. Многоуровневая объектно-ориентированная архитектура модель-представление-контроллер позволяет разрешить часть из этих проблем при создании крупных, структурированных систем. Удачное решение найдено в виде парадигмы проектирования модель-представление-контроллер (Model-View-Controller – MVC), которая стала центральной основой для всего графического интерфейса приложений [4].

MVC состоит из объектов трех видов: модель, представление и контроллер. Модель (Model) – это объект приложения, а представление или вид (View) – экранное представление (рисунок 3). Контроллер (Controller) описывает, как интерфейс реагирует на управляющие воздействия пользователя. До появления схемы MVC эти объекты в пользовательских интерфейсах смешивались. MVC отделяет их друг от друга, за счет чего повышается гибкость, и улучшаются возможности повторного использования [4].

MVC отделяет представление от модели, устанавливая между ними протокол взаимодействия. Такой подход позволяет присоединить к одной модели несколько видов, обеспечив тем самым различные представления.

Одним из главных преимуществ использования архитектуры MVC является возможность осуществить модуль-

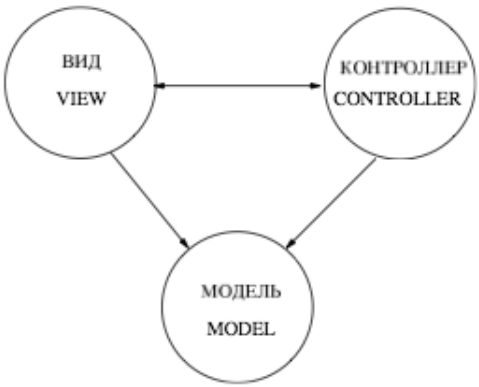

Рис. 3. Архитектура MVC-модели

ное построение программного приложения. Модульность построения обеспечивает ряд преимуществ, таких как:

- расширяемость;
- большая управляемость;
- гибкость;
- возможность повторного использования;
- возможность интеграции.

Все эти преимущества являются первостепенно важными для программных приложений в области веб, работающих в распределенных гетерогенных средах в условиях быстрого изменения состояние окружения.

Ruby on Rails – объектно-ориентированный программный каркас для создания веб-приложений, написанный на языке программирования Ruby. В его состав включены инструменты для интеграции приложения с сервером баз данных и веб-сервером. Разрабатывать приложения, использующие Ruby on Rails, можно под управлением различных операционных систем (MacOS, Linux, Windows). Ruby on Rails наиболее полно и емко реализует паттерн MVC, что не может не сказаться на качестве и скорости разработки приложения.

Ruby – современный динамический интерпретируемый объектно-ориентированный свободно распространяемый язык программирования высокого уровня. Он впитал в себя достоинства таких языков, как LISP, Perl, Python и других.

Наиболее оптимально предложенный алгоритм поисковой оптимизации может быть реализован на Ruby on Rails с использованием базы данных Postgresql (самая популярная надежная свободно распространяемая объектно-реляционная система управления базами данных с открытым кодом). С точки зрения разработчиков Ruby on Rails – это фреймворк для быстрой и эффективной веб-разработки. Он написан на интерпретируемом языке Ruby. Rails позволяет быстро приступить к разработке и сконцентрироваться на логике работы приложения, а не на решении проблем реализации. Это объектно-ориентированный фреймворк с открытым исходным кодом, распространяющийся под лицензией MIT [5].

В работе представлена разработанная простая, гибкая и современная информационная система, основанная на открытых технологиях и продуктах, в которой был реализо-

ван один из алгоритмов «белой» оптимизации. Предложены рекомендации по средам и инструментам разработки приложения в общем случае.

Использование информационной системы позволит упростить оптимизацию сайтов для рядовых пользователей, а также значительно увеличит количество объективных статистических показателей, характеризующих работу сайтов.

Литература

1. «Оптимизация сайтов». – Режим доступа: https://ru.wikipedia.org/wiki/Поисковая_оптимизация. – Дата доступа: 20.01.2015.

2. «IT и Рунет 2013: итоги года». – Режим доступа: http://www.optimism.ru/press/IT_i_Runet-2013-itogi_goda.pdf. – Дата доступа: 25.05.2015.

3. Ромейко О.А. Поисковая оптимизация сайтов на основе анализа контента // 51-я научная конференция аспирантов, магистрантов и студентов БГУИР, Минск, 13-17 апреля 2015 года.

4. Алан Найт, Нейси Дай. Объекты и Web. // Открытые системы. – 2002. – № 09.

5. Брюс Тейт. Практическое использование Rails: Часть 3. Оптимизация ActiveRecord. – Режим доступа: http://www.ibm.com/developerworks/web/library/wa-rails4. – Дата доступа: 10.02.2015.

O. Romeyko, S. Mikhnevich

Search Optimization of a Web Site Based on the Content Analysis

The paper presents the algorithm of «white» optimization of the website and its implementation. The algorithm solves the problem of optimization of web applications, without special services of specialists. On the basis of this algorithm it is realized the system, which provides recommendations for optimization of the web application in automated mode.

Статья поступила 25.05.2015