УДК 372.016:004

ПОИСКОВЫЙ АЛГОРИТМ BSEARCH В СИСТЕМЕ ТЕСТИРОВАНИЯ BM TESTING

В.Э. Бойко, Д.Ю. Мелеш*, 4 курс*
Научный руководитель – ***Н.В. Силаев****, доцент*
Брестский государственный университет имени А.С. Пушкина

Система поиска (алгоритм) BSearch был разработан в 2010 году и предназначался для индексации, ранжирования и поиска информации на веб-страницах. К сожалению, из-за невозможности его использования на реальных веб-сайтах по понятным всем причинам(очень сложно добиться релевантности результатов в условиях огромного количества сайтов, страниц, веб-приложений с различными тематиками), алгоритм был переделан под более простые цели и в данный момент используется на социальном сайте topmash.com. Уже сейчас он показывает неплохие результаты как в скорости поиска, так и в релевантности выданных результатов.

Мы хотели бы реализовать ряд новых идей для повышения эффективности работы данного алгоритма. Не вдаваясь в подробности структуры и работы алгоритма, изложим некоторые ключевые особенности.

Имеется два типа известных нам способов поиска – это поиск по индексированной базе данных слов и поиск по привлекаемым к работе файлам. Основным их отличием является то, что в первом случае составляется база данных слов, по которым должен вестись поиск, что подразумевает необходимость использования дополнительного пространства на жестком диске, но зато это компенсируется более высокой скоростью работы, второй же способ не требует специальной базы данных, что дает возможность обойтись меньшими пространственными ресурсами жесткого диска.

Очевидно, что первый вариант подходит для больших проектов, где размер приложения не настолько важен, более важным является скорость работы приложения. В данной ситуации этот способ позволит добиться минимального времени отклика (время, затраченное на поиск по заданному поисковому запросу), т.е. повышает временную эффективность обсуждаемой операции.

Второй способ применим к небольшим проектам, в которых объем обрабатываемой информации не очень велик, поэтому для подобных ситуаций размер приложения критичен. В реализации нашего программного проекта мы остановили свой выбор на первом варианте, поэтому подробнее остановимся на нем.

Так как, в силу выбора, мы оказались сторонниками принципа «размер не столь важен – важна скорость работы», то в своем поисковом алгоритме мы привлекаем специальную базу данных, в которую добавляем нужные ключевые слова (ключи поиска). Основу этого множества составляют ключевые слова, комментарии к статьям и т.д., хотя возможно более полное индексирование. База данных в таком случае может содержать миллионы или даже миллиарды слов. Для того чтобы достичь минимального времени отклика, нам необходимо правильно хранить данные и использовать правильные SQL-запросы. Один из главных способов добиться хороших результатов – это индексирование базы данных, его описание выходит за рамки данной статьи. Заметим, что даже написание запросов, которые не будут долго выполняться, также является одним из способов сделать поиск удобным.

Кратко остановимся на проблеме релевантности результатов. Понятно, что добиться стопроцентной релевантности невозможно, поэтому мы старались сделать поиск максимально полезным для пользователя. На практике может сложиться следующая ситуация: относительно каждого интересующего пользователя материала мы позволяем ему ввести некоторые альтернативные слова, которые реально могут отсутствовать в материале поиска, но которые пользователь все-таки ввел в поисковом запросе. Такой подход позволяет нередко избегать случаев, в которых поисковая система не найдет ничего по сделанному запросу, даже когда нужная пользователю информация находится на сайте.

Алгоритм BSearch не чувствителен к окончаниям и приставкам слов, т.е. если пользователь в поисковом запросе указал слово «дорога», то ему, возможно, будут представлены результаты и со словом «придорожный».

Современные поисковые алгоритмы обладают возможностью проверки орфографии. Оговоримся, что BSearch пока такого делать «не умеет», но в скором времени мы планируем добавить туда и эту возможность. Тут нам видятся два альтернативных метода: первый – за счет использования словаря, второй – за счет использования в качестве эталона полученной выборки. На настоящий момент нам более интересен второй вариант, так как он должен работать быстрее. Он основывается на том, что в базе данных все слова написаны без орфографических ошибок, а значит, если в ней часто встречается слово, похожее на введенное в поисковом запросе и отличающееся всего на одну букву, то можно предположить, что пользователь допустил ошибку при составлении запроса и уведомить его об этом, либо сразу осуществлять поиск по слову, в котором нет ошибки.

На момент написания статьи рассмотренный алгоритм работает на сайте topmash.com и позволяет осуществлять поиск за долю секунды более чем по 500 статьям. Нам кажется, что это хорошие результаты, поэтому мы планируем в дальнейшем сделать его еще более релевантным и более быстрым.

В заключение отметим, что данный алгоритм мы планируем внедрить в нашу систему тестирования задач по программированию BM Testing, которая на данный момент используется в Брестском государственном университете имени А.С. Пушкина. Поиск должен будет вестись, в основном, по разделам теории, для того, чтобы студент мог с легкостью отыскать нужный ему материал. В данном случае имеется в виду, что теоретический материал формирует преподаватель, составляющий определенный раздел тестирования, поэтому дополнительно предоставляется возможность размещать на сайте тестирования как свои теоретические разработки к практическим занятиям, так и электронные пособия, выдержки из книг, примеры решенных задач.