

ПРИМЕНЕНИЕ «АЛГОРИТМА ШИНГЛОВ» ПРИ ОБРАБОТКЕ ЗАДАНИЙ НА КУРСОВОЕ ПРОЕКТИРОВАНИЕ

Сотников И.М.

ООО Научно-техническое предприятие «Криптософт», Пенза, Россия (440026, г. Пенза, ул. Лермонтова, 3), email: mails@cryptosoft.ru

С приходом интернета в широкие массы появилась проблема присвоения авторства чужого достижения в науке или какого-либо изобретения. В частности, эта проблема затрагивает студентов среднего профессионального и высшего образования при выборе и выполнении курсовой и дипломной работы. Изучив область рассматриваемой проблематики, мной в рамках диссертационной работы было выбрано направление, в результате которого решается проблема уникальности курсовой и дипломной работы у каждого студента. Выбранное направление заключается в использовании «Алгоритма шинглов» в «Автоматизированной системе формирования заданий на курсовое и дипломное проектирование». В статье рассматриваются этапы обработки текста курсовой и дипломной работ, приведение текста к нормальной форме, формирование списка последовательностей (шинглы), подсчет хэш-суммы каждой последовательности и вывод пользователю на экран результата проверки новой работы с имеющимися в базе данных учебного заведения работ предыдущих годов.

Ключевые слова: плагиат, формирование, курсовая, алгоритм шинглов.

APPLICATION OF “SHINGLE ALGORITHM” IN COURSE TASKS PROCESSING

Sotnikov I.M.

LLC Scientific and technical enterprise “Cryptosoft”, Penza, Russia (3 Lermontova str., 440026, Penza), email: mails@cryptosoft.ru

When Internet became very popular information source a problems of copyrights and text unique had appeared. In particular, this problem is met in student's studying and scientific works. To decide the problem of unique of works I investigated relevant directions. The result of my investigation became using shingle's algorithm in “Automation system of creating task for semester's and degree's works”. This system checks new work for matches with previously loaded works. In this article examined work's text processing, text form normalizing, forming list of sequences, creating hash sum of every sequences, forming checking result and output it for user.

Keywords: plagiarism, formation, exchange rate, algorithm shingles.

В научной статье рассматривается неудобство обработки документа технического задания на курсовое и дипломное проектирование, выдаваемое каждому студенту, обучающегося в высшем или средне-специальном учебном заведении. Неудобство заключается в поиске необходимой информации по большому объему документов, исправлении ошибочных документов, каталогизации по требуемым критериям и выявлении повторяющихся курсовых или дипломных работ. Для более детального изучения сути неудобства обработки этого документа был проведен устный опрос сотрудников и студентов кафедры вычислительной техники Пензенского государственного университета.

Актуальность

Привычный и прошедший несколько десятков лет процесс обработки документа технического задания должен остаться в прошлом. Поскольку в наше время существуют технологии, позволяющие сократить время процесса обработки документа технического задания, следует полностью отказаться от представления его на бумажном носителе.

На начальном этапе решения поставленной задачи следует исследовать рынок программного обеспечения, направленного на обработку документов в высших и средне-специальных учебных заведениях. В результате исследования рынка программного обеспечения выяснилось, что не существует ни одного решения поставленной задачи. На рынке программного обеспечения занимают свою нишу egr-системы (к примеру, «Галактика»), но они решают другие задачи.

На следующем этапе решения проблемы следовало разработать структуру базы данных и графический интерфейс пользователя клиентского приложения. Для работы с приложением пользователь любой операционной системы не должен был бы испытывать проблем с запуском, поэтому кроссплатформенность – это жесткое требование к разрабатываемому приложению. Системой, отвечающей за управление базой данных, была выбрана MySQL. Выбор СУБД MySQL обуславливается рядом её преимуществ:

- бесплатная СУБД, что отличает ее от других дорогостоящих продуктов;
- многопоточность – поддержка нескольких одновременных запросов;
- оптимизация связей с присоединением многих данных за один проход;
- записи фиксированной и переменной длины;
- гибкая система привилегий и паролей;
- до 16 ключей в таблице, каждый ключ может иметь до 15 полей;
- гибкая поддержка форматов чисел, строк переменной длины и меток времени;
- быстрая поддержка транзакций через механизм InnoDB.

База данных состоит из 13 таблиц, содержащих информацию о заданиях на курсовое проектирование, темах курсовых проектов, студентах, факультетах, кафедрах, дисциплинах, преподавателях, а также об аутентификационных данных пользователей, имеющих административный доступ к системе.

Построение графического интерфейса проводилось при помощи графических элементов в интегрированной среде разработки «NetBeans». Первая пробная версия приложения представляла собой прикладную программу, написанную на языке программирования Java. Преимущество данного выбора заключалось в кроссплатформенности. Но, с другой стороны, чтобы достичь требуемой кроссплатформенности, необходимо на каждой физической машине установить виртуальную java-машину или плагин для веб-браузера. Разработка в этом ключе была остановлена. Следующий и окончательный выбор платформы для разработки приложения пал на серверный язык программирования PHP 5 и язык гипертекстовой разметки HTML 5. Для построения гибкого и удобного графического интерфейса используется CSS 3. За

динамическое обновление информации в приложении отвечает технология AJAX. Выбор именно этой технологии был обусловлен следующими критериями:

- экономия трафика пользователя (вместо обновления всей страницы, загружается ее небольшая изменившаяся часть);
- снижение нагрузки на сервер. К примеру, на странице личных сообщений в приложении при выделении пользователем прочитанных писем сервер вносит изменения в БД и отправляет скрипту клиента ответ о выполнении операции без повторного создания страницы и ее передачи;
- ускорение реагирования интерфейса на команды пользователя.

Работать в системе одновременно могут несколько пользователей с разными уровнями доступа. При входе в систему производится парольная аутентификация пользователя, в ходе которой определяется, с какой функциональной ролью пользователь продолжит работу в системе. В системе предусмотрено 5 функциональных ролей работы в системе: при входе в систему производится парольная аутентификация пользователя, в ходе которой определяется, с какой функциональной ролью пользователь войдет в систему. В системе предусмотрено 5 функциональных ролей работы в системе.

Администратор ресурса. За этой функциональной ролью закреплены следующие возможности:

- управление факультетами;
- управление администраторами факультета;
- управление уровнями доступа к системе;
- управление новостями.

Администратор факультета. За этой функциональной ролью закреплены следующие возможности:

- управление кафедрами;
- управление администраторами кафедры;
- управление новостями.

Администратор кафедры. За этой функциональной ролью закреплены следующие возможности:

- управление группами;
- управление студентами;
- управление дисциплинами;
- управление преподавателями;
- управление темами курсовых работ;

- управление новостями.

Преподаватель. За этой функциональной ролью закреплены следующие возможности:

- управление закрепленными за преподавателем темами курсовых работ;
- управление новостями.

Студент. За этой функциональной ролью закреплены следующие возможности:

- управление закрепленными за студентом темами на курсовое проектирование.

Ядро системы спроектировано по принципу модульности. Модульность в системе достигается за счет разделения всех функциональных ролей пользователей на отдельную, независимую друг от друга сущность. Так, при первом запуске веб-приложения неавторизованному пользователю открывается веб-страница с двумя текстовыми полями ввода. Для успешной авторизации в системе необходимо ввести в поля ввода имеющиеся у каждого пользователя имя (т.н. логин) и уникальный пароль. По выстроенной иерархии функциональных ролей первое звено занимает администратор ресурса. Логин и уникальный пароль для администратора ресурса назначает системный администратор, проводивший установку и первичную настройку системы под конкретное учебное заведение. В случае успешной авторизации в качестве администратора ресурса, система отображает пользователю модуль управления учебным заведением, располагающийся в каталоге отдельно от остальных модулей других функциональных ролей. Авторизация в системе под остальными функциональными ролями проходит по аналогичному принципу, описанному выше.

Выгодность системы заключается в том, что установка и первичная настройка под выбранное высшее учебное заведение затрачивает минимум времени и финансирования.



Рис. 1. Структура автоматизированной системы

Плагиат с появлением Интернета превратился в серьёзную проблему. Попад в Интернет, знание становится достоянием всех, соблюдать авторское право становится всё труднее и даже невозможно. Постепенно становится сложнее идентифицировать первоначального автора.

Поиск нечетких дубликатов позволяет предположить, являются ли два объекта частично одинаковыми или нет.

Задачей не стоит определить абсолютное значение схожести объектов, а также выделения в каждом из объектов схожих частей. Нам необходимо только предположить, являются ли объекты почти дубликатами или нет.

Суть алгоритма

Итак, мы имеем 2 текста и нам нужно предположить, являются ли они почти дубликатами. Реализация алгоритма подразумевает несколько этапов:

- канонизация текстов;
- разбиение текста на шинглы;
- нахождение контрольных сумм;
- поиск одинаковых подпоследовательностей.

Канонизация текста

Канонизация текста приводит оригинальный текст к единой нормальной форме. Текст очищается от предлогов, союзов, знаков препинания и прочего ненужного «мусора», который не должен участвовать в сравнении. Также удаляются из текста прилагательные, так как они не несут смысловой нагрузки.

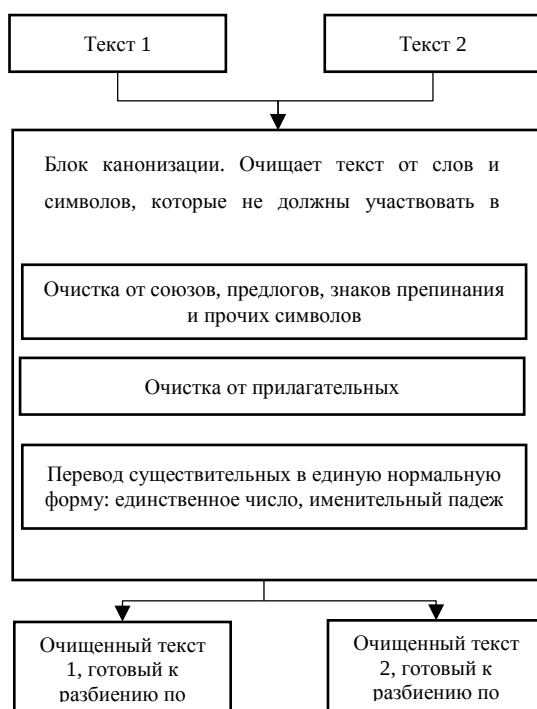


Рис. 2. Последовательность действий при канонизации текста

Разбиение на шинглы

Необходимо из сравниваемых текстов выделить подпоследовательности слов, идущих друг за другом по 10 штук (длина шингла).

Таким образом, разбивая текст на подпоследовательности, мы получим набор шинглов в количестве равном количеству слов минус длина шингла плюс один ($\text{кол-во_слов} - \text{длина_шингла} + 1$).

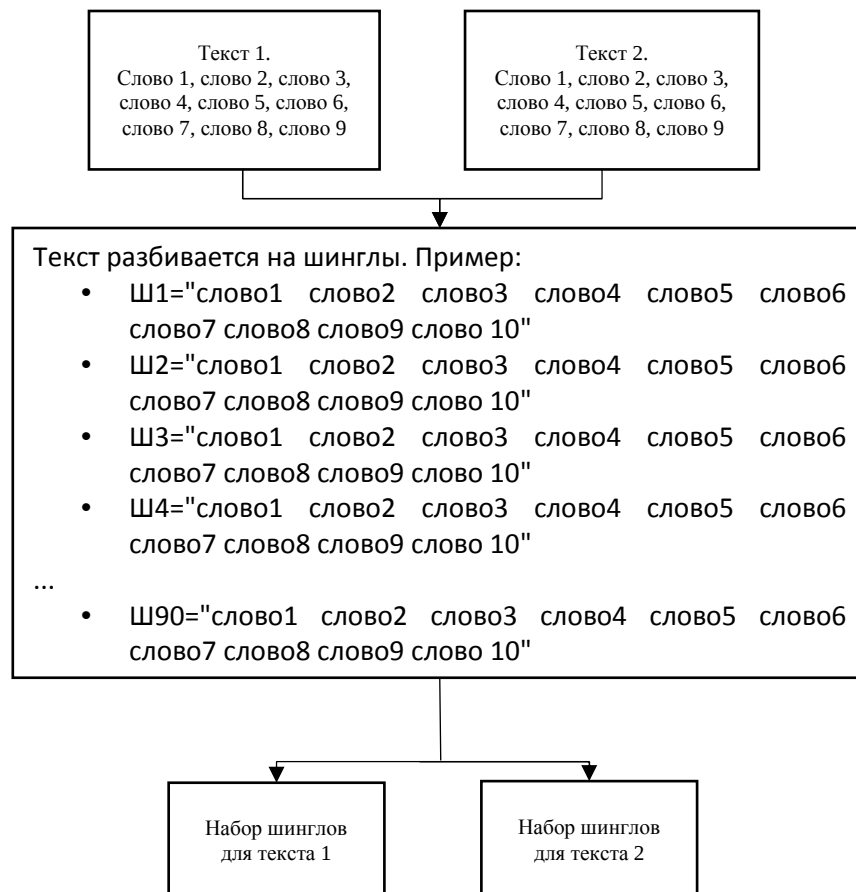


Рис. 3. Последовательность действий при разбиении на шинглы

Нахождение контрольной суммы

Теперь у каждого из текстов имеются свои наборы шинглов. Следует рассчитать контрольную сумму каждого из шинглов. Для расчета можно использовать известный алгоритм MD5.

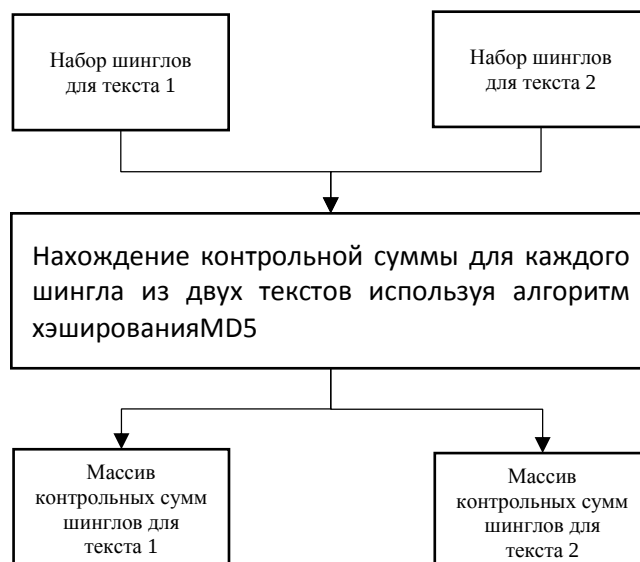


Рис. 4. Последовательность действий при нахождении контрольной суммы

Поиск одинаковых подпоследовательностей

И последний этап – сравнение.

Сравниваем между собой все элементы первого массива с соответствующими элементами второго массива, считаем отношение одинаковых значений, из этого получаем результат.



Рис. 5. Последовательность действий при сравнении элементов массива

Список литературы

1. Аргерих Л., Чой В., Коггсхол Д., Эгервари К., Сколло К. Профессиональное PHP программирование. Второе издание. – Изд-во «Символ-Плюс», 2003.
2. Дунаев В. В. Сам себе Web-дизайнер. – Изд-во «БХВ-Петербург», 2002.
3. Хендерсон К. Профессиональное руководство по Transact-SQL. – Изд-во «Питер», 2005.

4. V. Rodnenko (Виталий Родненко) Часть 1. Алгоритм шинглов для веб-документов // Интернет-ресурс URL: <http://www.codeisart.ru/part-1-shingles-algorithm-for-web-documents/>
5. V. Rodnenko (Виталий Родненко) Python: Алгоритм Шинглов — поиск нечетких дубликатов текста // Интернет-ресурс URL: <http://www.codeisart.ru/python-shingles-algorithm/>

Рецензенты:

Пашенко Д.В., д.т.н., профессор, заместитель заведующего кафедрой вычислительной техники Пензенского государственного университета, г. Пенза.

Фионова Л.Р., д.т.н., профессор, заведующая кафедрой информационного обеспечения управления и производства Пензенского государственного университета, г. Пенза.