

2. Гребенников А. В. Моделирование сетевого трафика и прогнозирование с помощью модели ARIMA / А.В. Гребенников, Ю.А. Крюков, Д. В. Чернягин // Электронный журнал «Системный анализ в науке и образовании» 2011. № 1. С. 1–11.
3. Numerical Method [Electronic resource] Mode of access: <http://numericalmethod.com/blog/>. Date of access: 13.04.2013.

СИСТЕМА ПОИСКА ПЛАГИАТА

А. И. Логинов

ВВЕДЕНИЕ

В настоящее время большинство служб поиска плагиата ищут его исключительно в обычном художественном тексте. В то же время исходные тексты программ во многом остаются без внимания. Конечно, существуют и такие сервисы. Вот только практически все они являются либо платными, либо уже давно не поддерживаются, либо никогда не выкладывались в общий доступ.

Именно поэтому целью данной работы стало создание такой системы поиска плагиата, которая бы никогда не потеряла своей актуальности, системы, которую можно было бы использовать даже в случае прекращения поддержки. Эта универсальная программа должна искать плагиат в исходных кодах на разных языках программирования с возможностью расширения их количества.

1 ЯП. РАСШИРЯЕМОСТЬ

1.1 Универсальная система

Одной из ключевых особенностей данной системы является токенизация исходного текста программ. Тем самым, задача расширяемости свелась к динамическому способу добавления токенизаторов для различных языков программирования.

1.2 Генерация парсеров

Генераторы парсеров позволяют программисту работать на более высоком уровне абстракции. Создание анализатора языка в этом случае включает в себя написание грамматики. Очевидно, что поддерживать несколько сотен строк грамматики намного проще, чем несколько тысяч строк парсера, написанного вручную.

1.3 ANTLR

Генератор парсеров ANTLR на сегодняшний день является наиболее популярным СС фреймворком. Для каждого полученного токена токенизированной программы можно получить также и расположение этого токена в исходном тексте (номер строки, позицию в строке и т.п.).

Для того, чтобы этот фреймворк смог сгенерировать классы парсера и лексера, необходимо предоставить ему описание LL(*)-грамматики, описанной в расширенной БНФ.

1.4 Реализованные идеи

Добавление поддержки нового языка осуществляется с применением библиотеки ANTLR. Сразу на вход программе подается грамматика в РБНФ. Затем с помощью этой библиотеки генерируются файлы лексера и парсера для этой грамматики на языке Java.

При выборе файлов для анализа, разработанная система рассматривает папки верхнего уровня как группы. Если, например, сразу выбирается несколько групп (т.е. несколько папок), то плагиат будет искаться лишь между этими группами. Если же выбрана всего одна группа, то сравниваться будут все файлы доступные в ней рекурсивно.

2 АЛГОРИТМЫ. РАСШИРЯЕМОСТЬ

В настоящее время существует несколько алгоритмов поиска плагиата в исходных текстах программ. Проанализировав ранее эти методы, выбор пал на метод отпечатков и метод смещения, ввиду их эффективности [1].

2.1 Метод отпечатков

Идея метода отпечатков заключается в выборе k -грамм с дальнейшим их хэшированием и пересечением. В данной работе использовался метод отпечатков вместе с алгоритмом просеивания [2].

Метод отпечатков изначально разрабатывался для обычных текстов, написанных на естественных языках. Ввиду того, что мы рассматриваем код программы в виде токенизированного представления, изменения затронули процесс хэширования. Если раньше в качестве k -грамма выступала последовательность из k символов, то теперь k -грамм – это последовательность из k токенов. Таким образом, по последовательности хэшей после просеивания можно восстановить исходный код с точностью до токена, а не символа, как раньше.

2.2 Метод смещений

Метод смещения представляет собой анализ размещения токенов в теле программы [3]. По сути, идет своеобразное наложение одной программы на другую на каждом смещении. Причем все совпавшие токены, а также их количество выписываются, чтобы затем можно было отобразить совпавшую часть кода.

2.3 Возможности расширения. Специальное API

Методы и алгоритмы могут устареть, на их место могут придти новые. Чтобы данная система могла использоваться и дальше, появилась необходимость внедрить возможность динамического добавления методов поиска плагиата.

Примерно оценив, что у каждого алгоритма может быть общее, было написано специальное API для этой системы, которое может быть использовано в дальнейшем.

3 ПРЕДСТАВЛЕНИЕ РЕЗУЛЬТАТОВ

3.1 Таблица сравнения парами

Поскольку в реализованной системе сравнение происходит по принципу каждый-с-каждым (учитывая разбиение по группам), то первоначальные результаты сравнения удобнее всего отобразить в виде таблицы. Такая таблица результатов отображает пары файлов, отсортированных по убыванию схожести исходных текстов. Причем она не зависит от алгоритма и языка программирования.

3.2 Визуализация

После отработки любого используемого в системе алгоритма (даже еще не реализованного), каждому исходному тексту можно поставить в соответствие набор токенов, которые совпадают с каким-то другим набором токенов другого файла. С помощью библиотеки ANTLR каждому токену ставится в соответствие номер его строки в файле, а также позиция его начала и конца в строки. Таким образом, визуализация найденного плагиата стала возможной. В каждой визуализации любого метода представлено число, которое обозначает процент совпадения рассматриваемых файлов.

4 ЭФФЕКТИВНОСТЬ РАЗРАБОТАННОЙ СИСТЕМЫ

4.1 Список неудачных попыток сокрытия плагиата от системы

Разработанная система поиска плагиата способна учитывать практически все изменения, которые могли бы внести те же студенты в свой код: всевозможные переименования, изменение констант, перемещения (классов, методов, полей), вставка неиспользуемого кода, изменение порядка условных операторов (if-else, switch-case).

4.2 Сравнение с существующими аналогами

К сожалению, сейчас осталось не так много поддерживаемых проектов, позволяющих искать плагиат в исходном коде. Из работающих и доступных систем были рассмотрены SIM и Plaggie.

Проект SIM. Эта система уже не поддерживается и не развивается, но ее исходный код можно найти в Интернете. SIM предлагает единственный способ управления им через консоль. В качестве результата работы генерируется один большой файл, по которому очень трудно понять, присутствует ли плагиат. Проведя несколько экспериментов, стало ясно, что используемый этой системой алгоритм является неэффективным, поскольку упускал очевидные заимствования.

Проект Plaggie является консольным Java приложением, с управлением через командную строку с тривиальными вариантами использования. В качестве результатов генерируется множество html страниц с результатами работы. К сожалению, у этой системы слишком много ложных срабатываний: она дает очень высокий процент похожести исходным текстам, которые таковыми точно не являются.

ЗАКЛЮЧЕНИЕ

Можно однозначно сделать выводы, что реализованная система поиска плагиата обладает множеством плюсов. Тем не менее, система не может однозначно *доказать* факт наличия или отсутствия плагиата. Она может лишь указать степень схожести каждой пары исходных текстов.

С моей точки зрения, реализованная система является лучшей среди доступных аналогов. Используемые алгоритмы являются эффективными, а сама система спроектирована так, чтобы с течением времени она продолжала справляться со своей основной задачей – обеспечением возможности выявления плагиата.

Литература

1. *Логинов А. И.* Алгоритмы поиска плагиата в исходных текстах программ // Всероссийский конкурс НИРСА в области информатики и информационных технологий: сб. науч. работ: в 3 т. Белгород, 2012. Т. 3. С. 302–305.
2. Интернет-адрес: <http://theory.stanford.edu/~aiken/publications/papers/sigmod03.pdf>
3. Интернет-адрес: <http://www.kv.by/index2000491105.htm>.

СОЗДАНИЕ ПОЛНОФУНКЦИОНАЛЬНОГО ПРИЛОЖЕНИЯ ДЛЯ МОДЕЛИРОВАНИЯ СТРЕЛЬБЫ ИЗ МИНОМЕТА

А. С. Новиков

ПОСТАНОВКА ЗАДАЧИ

Разработка электронного тренажера «Выполнения огневой задачи № 1 М Курса подготовки артиллерии (КПА-2000)» для обучения студентов управлению огнем артиллерии. В программе должна быть реализована возможность осуществления просмотра местности, добавления на эту местность ориентиров, целей и выполнения стрельбы полупрямой и раздельной наводкой по этим целям.

ВВЕДЕНИЕ

Артиллерийские соединения являются одним из наиболее массовых родов войск СВ. Они предназначены для ведения боевых действий совместно с другими родами войск и специальными войсками. Оснащение механизированных частей и подразделений минометными комплексами, сочетающими в себе мощное вооружение, высокую скорость движения, проходимость и маневренность, повышает боевые возможности войск на поле боя. Исходя из принципов ведения современного общевойскового боя и изменений в тактике ведения боевых действий, с каждым годом повышаются требования к качеству подготовки расчетов минометов, поддержанию техники в боеготовом состоянии. Наряду с этим остро стоит вопрос об экономии моторесурсов и материальных средств. Создание программы позволило выполнить вышеуказанные требования, повысить практические навыки расчетов минометов.

МАТЕМАТИЧЕСКАЯ МОДЕЛЬ

Для описания полета снаряда использовалась упрощенная физическая модель с некоторыми допущениями.