УДК 044.031.43

Н.Р. Куклич

МЕТОДИ ТА МОДЕЛІ ІНФОРМАЦІЙНОГО ПОШУКУ, АНАЛІЗУ ТА ОБРОБКИ ТЕКСТОВИХ КОЛЕКЦІЙ

Аннотация. В данной статье рассматривается предложенное средство для борьбы с плагиатом в научных и учебных работах. Оно представляет собой Desktop-приложение реализованное на языке Java. При создании этой программы были проанализированы существующие алгоритмы поиска общих подстрок и, на основе исследований, выбраны наиболее эффективные для поставленной задачи. Подробно рассмотрены реализованные в данном программном средстве алгоритмы: Алгоритм Шинглов и Наивный Алгоритм. Проведено сравнение выбранных алгоритмов с остальными и между собой по критериям скорости, точности, эффективности при обработке больших и малых объевом данных, производительности при анализе текстов с преобладающим количеством плагиата и с преобладающим количеством уникального текста.

Ключевые слова: Плагиат, общие подстроки, Алгоритм Шинглов, Наивный Алгоритм

Постановка задачи. Детально изучить существующие алгоритмы, сравнить их по критериям скорости, точности и количества потребляемых ресурсов. Проанализировать значения рассматриваемых критериев при заданных условия поиска: анализ документа при сравнении с большой базой данных, анализ документа при сравнении с малой базой данных, анализ документа при сравнении с идентичными работами, анализ документа при сравнении с непохожими работами. Определить самые точные, быстрые и наиболее соответствующие поставленной задаче алгоритмы. Реализовать их в виде удобного инструмента анализа текста.

Цель. Разработать систему поиска плагиата в виде приложения с удобным интерфейсом и всей необходимой функциональностью, используя наиболее эффективные алгоритмы.

Введение. Как известно, плагиат это умышленное присвоение авторства чужого произведения науки или искусства, или изобрете-

ния, использование его или его части под своим именем без указания источника заимствования. Плагиат является нарушением авторского и изобретательского права и влечёт за собой ответственность. Как правило, плагиат предполагает публикацию чужого произведения под своим именем. Сегодня, когда вопросам авторского права уделяется большое внимание, дословное изложение чужого текста среди аспирантов встречается крайне редко в отличие от их младших коллег, студентов. В основном, аспиранты в своих научных работах используют перефразированные чужие работы.

В данной работе мы касаемся одной из основных проблем современного образования – плагиата. В последнее время именно плагиат составляет основную часть, так называемой, самостоятельной работы студента, тем самым заменяя собой получаемые практические навыки, способность самостоятельно мыслить и принимать решения, проводить исследования и многое другое. В связи с этим стала острая необходимость применения средств, которые сделают возможным за приемлемое время определить преподавателю, что работа студента скопирована.

Таким средством может являться программное обеспечение, работающее с документами, в которых сдаются работы студентов. Приложение должно быстро, адекватно, наглядно и, по возможности, точно определять количество скопированного текста в работе.

Для борьбы с плагиатом было изобретено множество алгоритмов. Среди них: Алгоритм Рабина — Карпа, Алгоритм Кнута — Морриса — Пратта, Алгоритм Бойера — Мура, Алгоритм Шинглов, Наивный Алгоритм и многие другие. Они используются в разных сферах, связанных со сравнением символьных последовательностей. Некоторые алгоритмы были использованы при создании популярных программ и сервисов сравнения текстов. У каждого алгоритма есть свои особенности, преимущества и недостатки.

Необходимо проанализировать известные алгоритмы анализа текста на схожесть, исследовать их на скорость и адекватность выполнения поставленной задачи и, как следствие, определить наиболее подходящий для нашей задачи.

В результате этого исследования будет разработана программа с реализацией двух отобранных алгоритма поиска плагиата. Эта про-

 ISSN 1562-9945

грамма способна оценить работу студента и сообщить преподавателю вывод о ее уникальности.

Использоваемые алгоритмы:

1. Алгоритм Шинглов. Предком данного алгоритма считается алгоритм manber1994, предложенный Уди Манбером (Udi Manber) в 1994 году. Андрей Бродер (Andrei Broder) в 1997 году дал окончательное название и довел до ума Алгоритм Шинглов. Название произошло от английского слова shingles — «черепички, чешуйки». При рассмотрении принципа его работы станет ясным, почему выбрано именно это название.

Разберем подробно данный алгоритм. Предположим, что имеется два текста, схожесть которых нам нужно проверить. Каждый из этих текстов проходит все стадии алгоритма. Реализация алгоритма подразумевает несколько этапов:

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

После завершения работы Алгоритма Шинглов наступает завершающий этап программы – это расчет процентного показателя плагиата работы студента.

Токенизация текста — распознавание токенов (слов) в тексте. Примечание: данная программа не проверяет грамматику текста, а, следовательно, считает словом любую последовательность символов отделенную от остального текста символом-разделителем. Теперь, когда в тексте выделены токены, есть возможность дополнительной обработки текста. Можно избавиться от лишних слов, знаков, и даже целых предложений по необходимости.

Приведение текста к единой канонической форме не ограничено действиями, которые описаны на данном этапе. Можно, например, каждое из существительных приводить к единственному числу, именительному падежу и т.д., для этого нужно подключать морфологические анализаторы русского языка (или других языков, где необходимы эти действия).

Следующий этап — разбиение текста на шинглы. Каждый из текстов разбивается на подпоследовательности с определенным количеством слов. Шинглы представляют собой последовательности с оди-

наковым количеством токенов, идущих внахлест. При длинне 4 первый шингл будет последовательностью с первого по четвертый слово, второй – со второго по пятое слово и т.д.

Когда тексты разбиты на шинглы можно приступать к поиску контрольных сумм. Именно по контрольным суммам шинглов будет вестись сравнение, так как сравнение чисел занимает меньшее время, чем сравнение строк. После сравнения количество совпадений делиться на общее количество сравнений и определяется коэффициент плагиата с незначительной погрешностью.

2. Наивный Алгоритм. Принцип работы данного алгоритма – поиск наибольших общих подстрок, а также нахождение наибольшей общей подстроки. Которой является подстрока двух или более строк, имеющая максимальную длину. Формально, наибольшей общей подстрокой строк $s_1, s_2, \ldots s_n$ называется строка w^*, которая удовлетворяет условию $\|w^*\| = \max(\{\|w\| | w \sqsubseteq s_i, i = 1, \ldots n\})$, операция $w \sqsubseteq s_i$ обозначает что строка w является (возможно несобственной) подстрокой строки s_i.

Решение задачи поиска наибольшей общей подстроки для двух строк s_1 и s_2, длины которых m и n соответственно, заключается в заполнении таблицы A_{ij} размером $(m+1) \times (n+1)$ по следующему правилу, принимая, что символы в строке нумеруются от единицы.

$$\begin{cases} A_{0j} = 0, & j = 0 \ldots n, \\ A_{i0} = 0, & i = 0 \ldots m, \\ A_{ij} = 0, & s_1[i] \neq s_2[j], i \neq 0, j \neq 0, \\ A_{ij} = A_{i-1j-1} + 1, & s_1[i] = s_2[j], i \neq 0, j \neq 0. \end{cases}$$

Максимальное число A_{uv} в таблице это и есть длина наибольшей общей подстроки, сама подстрока: $s_1[u - A_{uv} + 1] \ldots s_1[u]$ и $s_2[v - A_{uv} + 1] \ldots s_2[vu]$.

3. Вывод: Выбранные алгоритмы наиболее подходящие для решения поставленной задачи. Их результаты при оценке критерия скорость в разы лучше других исследуемых алгоритмов, это объясняется линейностью проводимого поиска. В отличие от других алгоритмов, где поиск общих подстрок сводится к поиску множественного шаблона в строке, Алгоритм Шинглов и Наивный Алгоритм проходят по каждой последовательности лишь однажды.

 ISSN 1562-9945

При сравнении алгоритмов между собой по критерию скорости Алгоритм Шинглов показал значительно лучшие результаты. Наивный Алгоритм опережает Алгоритм Шинглов по критерию точность, так как ошибка при оценке отношения скопированного текста к уникальному у него отсутствует. В то время как ошибка при оценке с помощью Алгоритма Шинглов всегда присутсвует и возрастае прямопропорционально длинне шингла но, остается незначительной для поставленной задачи.

Описание системы. Система представляет Java Desktop приложение, установленное на машине со свободным доступом к базе данных научных работ.

Функции программы:

- Извлечение чистого текста из файлов MS Word;
- Вывод информации о документе (кол-во символов, слов, рисунков);
- Сравнение конкретной работы с базой данных работ с помощью алгоритма шинглов;
- Сравнение конкретной работы с базой данных работ с помощью Наивного алгоритма;
- Сравнение конкретной работы с базой данных работ с помощью Алгоритма DIFF;
- Вывод рассчитанного коэффициента общего количества плагиата конкретной работы;
- Вывод рассчитанного коэффициента количества плагиата для каждой работы из базы данных;
- Вывод ссылок на документы, с которыми производилось сравнение с возможностью быстрого доступа к их содержимому;
- Выделение плагиата в работе и в источнике;
- Поиск работы с наибольшим количеством общего с нашей работой текста;

Интерфейс программы реализован в соответствии со стандартом. Он простой, не нагружен излишней настройкой опций поиска и имеет понятное название и размещение элементов.

Интерфейс содержит:

- Текстовое поле, для указания размещения анализируемого документа;

- Текстовое поле, для указания размещения базы данных документов, с которыми будет проводиться сравнение;
- Кнопку-переключатель между Алгоритмом Шинглов и Наивным Алгоритмом;
- Поле ввода параметра поиска (размер шингла для Алгоритма Шинглов и длина наименьшей последовательности для Наивного Алгоритма);
- Кнопку для запуска сравнения текстов;
- Поле вывода списка файлов, с которыми было проведено сравнение;
- Поле вывода анализируемого документа;
- Поле вывода определенной работы из базы данных;
- Кнопку для запуска функции выделения неуникального текста;
- Кнопку для запуска функции вывода в поле для работ из базы данных наиболее похожей работы.

Реализованное программное средство не требует никакого учебного пособия, так как интуитивно понятно. Сравнение происходит быстро и не требует больших затрат ресурсов. Все это облегчит его внедрение в учебный процесс.

ЛИТЕРАТУРА

1. Статья об Алгоритме Шинглов [электронный ресурс]: http://www.codeisart.ru/python-shingles-algorithm/
2. Статья о Наивном Алгоритме [электронный ресурс]: http://ru.wikipedia.org/wiki/Наибольшая_общая_подстрока
3. Steven John Metsker, Mark W. Mitchell. Building Parsers with Java. [Текст] /— В.: Издательство—Addison Wesley, 2005. — 784 с.
4. Кормен Т., Ривест Р. Алгоритмы: построение и анализ[Текст] /— 2-е изд. — М.: «Вильямс», 2006. — С. 1296.
5. Машечкин И. В., Петровский М. А. Методы анализа текстов. [Текст] // Московский государственный университет им. Ломоносова. — М.: «Анализ данных», 2004. — 364 с.
6. Круглински Д. И., Уингоу С. Дж. Программирование на Java для профессионалов. [Текст] /— М.: Издательство—торговый дом «Русская Редакция», 2004. — 861 с.
7. Эллис М. А. Справочное руководство по языку Java. [Текст] /— Москва: Мир, 1992. — 445 с.

 ISSN 1562-9945