УДК 004.4:378
DOI 10.33764/2618-8031-2023-1-333-340

П. Ю. Бугаков[1], А. Р. Аргинбаев[1]*

Разработка программы для определения оригинальности лабораторных и курсовых работ «Defori»

[1] Сибирский государственный университет геосистем и технологий, г. Новосибирск, Российская Федерация
* e-mail: peter-bugakov@yandex.ru

Аннотация. Статья посвящена актуальной проблеме плагиата при выполнении лабораторных и курсовых работ в высших учебных заведениях. Существующие программные системы выявления заимствований работают с крупными базами текстовых документов, которые формируются большим сообществом пользователей. При этом ручная корректировка списка файлов, используемых для проверки оригинальности письменных работ, существенно ограничена. Однако результат такой проверки внутри небольшого коллектива, например, группы или потока, мог бы стать дополнительным критерием оценки качества студенческих работ. В связи с этим был разработан и протестирован прототип программного обеспечения, позволяющий проверять лабораторные и курсовые работы на предмет заимствования, используя для этого внутреннюю гибко конфигурируемую коллекцию письменных работ обучающихся университета. В дальнейшем данное программное обеспечение планируется использовать в учебном процессе на кафедре прикладной информатики и информационных систем СГУГиТ.

Ключевые слова: программное обеспечение, плагиат, антиплагиат, учебный процесс, оценка оригинальности, opencorpora, лабораторная работа, курсовая работа

P. Yu. Bugakov[1], A. R. Arginbaev[1]*

Development of the «Defori» program to determine the originality of laboratory work and term papers

[1] Siberian State University of Geosystems and Technologies, Novosibirsk, Russian Federation
* e-mail: peter-bugakov@yandex.ru

Abstract. The article is devoted to the actual problem of plagiarism in the performance of laboratory and course work in higher educational institutions. Existing software systems for detecting borrowings work with massive databases of text documents that are formed by a large community users. At the same time, manual correction of the list of files used to verify the originality of written works is significantly limited. However, the result of such a check within a small team, for example, a group or a stream, could become an additional criterion for assessing the quality of student work. In this regard, a software prototype was developed and tested, which allows checking laboratory and term papers for borrowing, using an internal flexibly configurable collection of written works of the university. In the future, this software is planned to be used in the educational process at the Department of Applied Informatics and Information Systems of SSUGT.

Keywords: software, plagiarism, anti-plagiarism, learning process, assessment of originality, opencorpora, laboratory work, term paper

Введение

Письменные работы являются одним из лучших средств оценки академических достижений студентов. По ним можно судить об объеме усвоенного

учебного материала, умении обучающихся находить, структурировать, сравнивать, анализировать информацию и, в конечном итоге, находить решение поставленной задачи [1]. В настоящее время основным информационным источником для подготовки письменных работ у студентов является интернет. С одной стороны, свободный доступ ко всему многообразию цифровых научных изданий способствует более качественному и всестороннему раскрытию изучаемой темы. С другой стороны, соблазн быстрого копирования необходимой информации подталкивает обучающихся к бездумному заимствованию чужих работ. В итоге письменные работы стали демонстрировать не степень развитости профессиональных навыков, а умение найти и скомпилировать приемлемый текст с минимальным количеством усилий. Причем плагиат охватил весь спектр студенческих письменных работ – от небольших отчетов о лабораторных работах, до серьезных курсовых работ [2]. Сложность выявления факта списывания обуславливается высокой учебной нагрузкой преподавателя современного вуза, который вынужден проверять большое количество студенческих работ в сжатые сроки. Такая ситуация делает качественный контроль оригинальности текста лабораторных и курсовых работ практически невозможным.

Сейчас учебные заведения активно используют программную систему поиска заимствований в текстовых документах «Антиплагиат». Также существует достаточно большое количество разнообразных веб-сервисов и настольных программ подобного назначения, которые пользуются популярностью среди студентов для предварительной проверки уникальности текста в процессе работы над ним. Все эти программные средства частично решают проблему плагиата в студенческих работах, однако они обладают одним существенным недостатком. Поиск заимствований осуществляется с использованием крупных поисковых вебсервисов (например, в программе AntiPlagiarism.NET) или в специализированных автоматически пополняемых базах данных научных работ (как в случае системы Антиплагиат). При этом пользователь таких систем обладает ограниченными возможностями по конфигурированию выборки работ для сравнения с анализируемым текстом, а перекрестная проверка оригинальности письменных работ студентов в рамках одной группы или потока вовсе невозможна.

В связи с этим возникает необходимость разработки программного обеспечения, позволяющего проверять лабораторные и курсовые работы на предмет заимствования, используя для этого внутреннюю коллекцию письменных работ университета.

Для достижения поставленной цели необходимо выполнить следующие задачи: составить список требований к программе; разработать алгоритм и программный прототип; выполнить его тестирование и проанализировать результаты.

Методы и материалы

Программа для определения оригинальности текста должна удовлетворять следующим функциональным требованиям: проводить анализ оригинальности текстов с различным уровнем фрагментации и строгостью сравнения;

обрабатывать файлы формата *.docx; выполнять проверку нескольких файлов; сохранять результат проверки каждого проверенного документа.

При запуске программы необходимо сформировать запрос на выполнение анализа текстовых работ. Он формируется из набора таких параметров, как минимально допустимый уровень оригинальности, пути проверяемых и сравниваемых файлов, путь для сохранения результатов анализа, список поддерживаемых форматов документов, списки анализируемых типов слов и анализируемых частей речи. В процессе обработки запроса формируются списки проверяемых и сравниваемых документов. Для каждого документа создается объект, который содержит список абзацев. Каждый абзац, в свою очередь, хранит список предложений, а каждое предложение – список слов. Для каждого слова определяется его тип (рис. 1, поле Word.TextType): латиница, кириллица, число или символ. Если тип значения не является символом, формируется форма слова (рис. 1, поле Word.Form). Если слово русскоязычное, то для него определяется нормальная форма (рис. 1, поле Word.Lemma) и вероятные части речи (рис. 1, поле Word.GrammemeTypes) с помощью словаря русского языка OpenCorpora. В зависимости от полученных результатов, для каждого слова формируется его значение (рис. 1, поле Word.Value). Если тип слова и его вероятные части речи подходят под параметры анализа, то оно помечается, как анализируемое (рис. 1, поле Word.IsAnalyzed).

Word.Text	Word.TextType	Word.Form	Word.Lemma	Word.GrammemeTypes	Word.Value	Word.IsAnalyzed
–	Symbol				–	False
Вы	Cyrillic	вы	вы	NPRO	вы	True
знаете,	Cyrillic	знаете	знаете	CONJ	знаете	True
я	Cyrillic	я	я	NPRO	я	True
сделал	Cyrillic	сделал	сделал	VERB	сделал	True
для	Cyrillic	для	для	GRND, PREP	для	True
их	Cyrillic	их	их	ADJF	их	True
воспитания	Cyrillic	воспитания	воспитание	NOUN	воспитание	True
все,	Cyrillic	все	весь	ADJF, NOUN	весь	True
что	Cyrillic	что	что	CONJ, NPRO, ADVB, PRCL	что	True
может	Cyrillic	может	могу	VERB	могу	True
отец,	Cyrillic	отец	отец	NOUN	отец	True
и	Cyrillic	и	и	CONJ, INTJ, PRCL	и	True
оба	Cyrillic	оба	оба	NUMR	оба	True
вышли	Cyrillic	вышли	выслал	VERB	выслал	True
des	Latin	des			des	False
imbeciles.	Latin	imbeciles			imbeciles	False

Рис. 1. Информация о словах предложения

Если предложение имеет хоть одно анализируемое слово, то для него генерируется хеш (рис. 2, поле Sentence.Hash) с помощью хеш-функции SHA256. При этом входным значением является строка, образованная списком значений анализируемых слов, отсортированных по возрастанию. Предложение с хешем помечается как анализируемое (рис. 2, поле Sentence.IsAnalyzed) (рис. 2).

Sentence.IsAnalyzed	Sentence.Hash
True	9e59fcd8a4fc2d10e3275ae5b14ffd7744e03e8189a00205361e8f7965cdb9a1
True	a8fec88cbaf87f681922d0ab89cd8e6f0b50c3761690fa60fedc13a1ca2481b6
False	

Рис. 2. Информация о предложениях абзаца

Если абзац имеет хоть одно анализируемое предложение, то для него также генерируется хеш (рис. 3, поле Paragraph.Hash). Входным значением является строка, образованная списком хешей, анализируемых предложений, отсортированных по возрастанию. Абзац с хешем помечается как анализируемый (рис. 3, поле Paragraph.IsAnalyzed).

Paragraph.IsAnalyzed	Paragraph.Hash
True	83cf74543d37bdfde7962375770929c11c266723a567fb21bccd40306a5698f4
True	b75f9524752af39f73cdb78be8d254a421115d9e0ba3471aef6b22ad895416ce
False	

Рис. 3. Информация об абзацах текста

После структурного анализа документов и формирования информационных объектов начинается двухэтапный поиск совпадений.

На первом этапе происходит поиск точных совпадений. Сначала выполняются пересечения множеств хэшей абзацев проверяемых и сравниваемых документов [3]. В случае совпадения хэша проверяемого абзаца в список источников абзаца добавляется название сравниваемого документа. Затем выполняются пересечения множеств хэшей предложений, проверяемых и сравниваемых документов. В случае совпадения хэша проверяемого предложения в список его первоисточников добавляется название документа, с которым происходило сравнение.

На втором этапе происходит поиск частичных совпадений. Каждое предложение, не имеющие совпадений на прошлом этапе, разбивается на фрагменты, называемые шинглами (от английского «shingle»). Они формируются из сортированных по возрастанию анализируемых слов предложения. Каждый шингл состоит из двух слов. Выборка слов происходит внахлест, а не встык, то есть каждое слово, кроме первого и последнего, попадает в два соседних шингла. Проверка проходит путем пересечения объединенного множества шинглов, проверяемых и сравниваемых документов. В случае совпадения шингла слова и его образующие помечаются как совпавшие. Предложениям, в которых соотношение совпавших слов к анализируемым превышает коэффициент 0,75, в список источников добавляется название сравниваемого документа.

Результаты анализа сохраняются в виде документов в формате .docx. В начале отображается краткая сводка, содержащая информацию о количестве

совпавших предложений, оригинальности документа и десяти самых объемных источниках заимствований (рис. 4). Процент оригинальности определяется через отношение количества слов из совпавших фрагментов документа к их общему количеству.

Точно совпавшие предложения: 346

Частично совпавшие предложения: 55

Оригинальные предложения: 282

Оригинальность документа: 80,25%

Источники заимствований:

1) 2.docx: 27,53%

2) 3.docx: 23,99%

3) 1.docx: 19,61%

4) 5.docx: 15,07%

5) 4.docx: 13,8%

Рис. 4. Краткая сводка результатов проверки

Затем текст проверяемого документа, окрашенного следующими цветами:

– черный – фрагмент не имеет списка источников;
– оранжевый – фрагмент имеет частично совпавшие источники;
– красный – фрагмент имеет точно совпавшие источники.

Анализируемые слова окрашены ярче, не анализируемые – светлее. В конце абзацев и предложений отображается список источников, окрашенный темнее (рис. 5). Если абзац и предложение имеют общий источник, то в списке источников предложения он не отображается (рис. 6).

Проверяемый документ

Она запнется, неловко замолчит. [Частично совпавшее предложение: 1) Comparable.docx.] В прочитанном она почувствует огромность жизни, жар ее и свет, но как передаст это словами? [Частично совпавшее предложение: 1) Comparable.docx.] Не сможет она выразить свои чувства – и представиться себе матросом, что окажется темной ночью на чужом корабле и никак не разберется ощупью в незнакомом такелаже. [Частично совпавшее предложение: 1) Comparable.docx.]

Сравниваемый документ

Он запнулся, неловко замолчал. В прочитанном он почувствовал огромность жизни, жар ее и свет, но как передать это словами? Не смог он выразить свои чувства – и представился себе матросом, что оказался темной ночью на чужом корабле и никак не разберется ощупью в незнакомом такелаже.

Рис. 5. Пример результата поиска частичных совпадений в предложениях

Проверяемый документ

Точное совпадение абзаца:

Тёмный еловый лес стоял, нахмурившись, по обоим берегам скованной льдом реки. Глубокое безмолвие царило вокруг. Это была глушь — дикая, оледеневшая до самого сердца Северная глушь. [Точно совпавший абзац: 1) Comparable.docx.]

Точное совпадение предложений:

И всё же что-то живое двигалось в ней и бросало ей вызов. [Точно совпавшее предложение: 1) Comparable.docx.] По замёрзшей реке пробиралась упряжка ездовых собак. [Точно совпавшее предложение: 1) Comparable.docx.]

Сравниваемый документ

Тёмный еловый лес стоял, нахмурившись, по обоим берегам скованной льдом реки. Глубокое безмолвие царило вокруг. Это была глушь — дикая, оледеневшая до самого сердца Северная глушь.

И всё же что-то живое двигалось в ней и бросало ей вызов. По замёрзшей реке пробиралась упряжка ездовых собак. Взъерошенная шерсть их заиндевела на морозе, дыхание застывало в воздухе и кристаллами оседало на шкуре.

Рис. 6. Абзацы и предложения с источниками, полученными в результате поиска точных совпадений

Результаты

Для апробации алгоритма был создан прототип программы «Defori» с текстовым пользовательским интерфейсом. Для написания кода использовался язык программирования C#. В качестве тестовых данных были взяты 60 курсовых работ, более 100 страниц в каждой. Анализировались только русскоязычные слова и информативные части речи (исключены местоимения, предикативы, предлоги, союзы, частицы и междометия). Тестирование проводилось на аппаратной платформе со следующими характеристиками:

– процессор AMD Ryzen 7 2700X с тактовой частотой 4 ГГц, 8 ядер, 16 потоков;

– оперативная память 16 Гб DDR4, с тактовой частотой 3200 МГц, в двухканальном режиме;

– SSD M.2 накопитель.

При использовании такой конфигурации обработка запроса заняла 30 секунд (рис. 7), максимальная загрузка процессора достигала 60 %, объем занятой оперативной памяти – 7 Гб, время, затраченное на проверку приведено в табл. 1.

Таблица 1

Этап анализа файла	Затраченное время, сек
Открытие и подготовка файлов для сравнения	20,94
Поиск полных совпадений	1,06
Поиск частичных совпадений	0,61
Сохранение результатов поиска	6,71
Общее время выполнения анализа	29,32

Обсуждение

Широко используемая в высших образовательных учреждениях система Антиплагиат.ВУЗ позволяет выполнять проверку курсовых и лабораторных работ, однако база текстовых документов в ней формируется коллективно – всеми уполномоченными пользователями в вузе. Они могут самостоятельно добавлять новые или удалять ранее добавленные ими файлы, но не могут управлять файлами других пользователей. Таким образом, ручное формирование списка файлов, используемых для проверки оригинальности письменных работ, становится существенно ограниченным. Такая проверка внутри небольшого коллектива, например, группы или потока могла бы стать дополнительным критерием оценки качества студенческих работ.

Разработанный алгоритм и созданный на его основе программный прототип устраняет этот недостаток, позволяет пользователю самостоятельно выбирать наборы файлов для сравнения, указывать глубину проводимого анализа и получать подробные отчеты о проведенной проверке. Результаты проверки курсовой работы на наличие плагиата среди работ одной студенческой группы, а также процент оригинальности, вычисленный в системе Антиплагиат.ВУЗ показан на рис. 7.

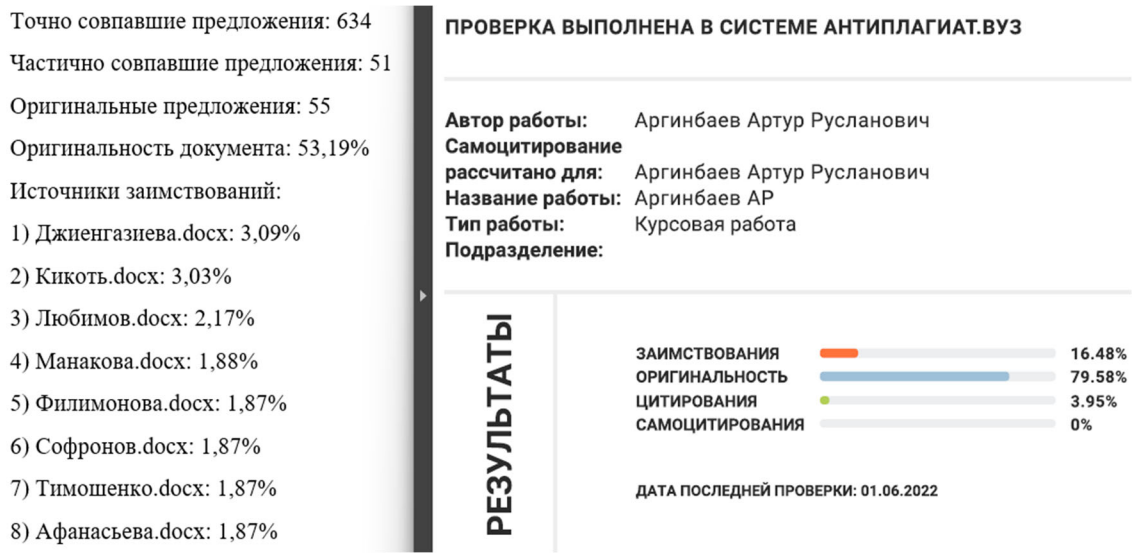

Рис. 7. Результат анализа курсовой работы в программе «Defori» и Антиплагиат.ВУЗ

Заключение

Разработанный программный прототип позволяет [4]:

– гибко настраивать параметры анализа;

– определять точные и частичные совпадения;

– выявлять плагиат после перестановки слов, фраз и предложений, смены формы слов, при незначительном добавлении новых слов в исходное предложение;

– игнорировать изменения времен, падежей, и других грамматических категорий слова.

Разработанное на основе алгоритма программное обеспечение в будущем планируется использовать в учебном процессе на кафедре прикладной информатики и информационных систем СГУГиТ для проверки оригинальности отчетов о лабораторных и курсовых работах.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Никитов А.В., Орчаков О.А, Чехович Ю.В. Плагиат в работах студентов и аспирантов: проблема и методы противодействия // Университетское управление: практика и анализ. – Екатеринбург, 2012. – С. 61–68.

2. Кацко С.Ю., Кокорина И.П. Проверка ВКР: корректные заимствования, плагиат и оригинальность текста // Актуальные вопросы образования. – 2021. – № 1. – С. 142–145.

3. Петровский А. Б. Теория измеримых множеств и мультимножеств. – Москва : Наука, 2018. – 244 с.

4. Свидетельство о государственной регистрации программы для ЭВМ 2022682095 Российская Федерация. Программа для анализа оригинальности лабораторных и курсовых работ / П. Ю. Бугаков, А. Р. Аргинбаев ; заявитель и правообладатель Федеральное государственное бюджетное образовательное учреждение высшего образования «Сибирский государственный университет геосистем и технологий». – № 2022681876 ; заявл. 18.11.2022 ; опубл. 18.11.2022