УДК 004.91

АРХИТЕКТУРА И АЛГОРИТМЫ СЕРВИСА ОБНАРУЖЕНИЯ ПЛАГИАТА

В.В. Дягилев, А.А. Цхай, С.В. Бутаков

Алтайский государственный
технический университет им. И.И.Ползунова
E-mail: dyagilev@mail.ru
Алтайская Академия Экономики и Права
E-mail: taa1956@mail.ru
Международная школа бизнеса Солбридж,
E-mail: butakov@solbridge.ac.kr

Описывается архитектура сервиса определения плагиата, позволяющая защищать интеллектуальную собственность авторов документов. Предлагаемая архитектура разделяет обработку документов на две части, одна из которых, использует вычислительные ресурсы локальной инфраструктуры, другая - вычислительные мощности поисковых машин Интернета.

Ключевые слова: определение плагиата, архитектура сервиса, охрана авторского права.

Введение

Стремительное развитие сети Интернет наряду с увеличивающейся компьютерной грамотностью, к сожалению, способствует проникновению плагиата в различные сферы человеческой деятельности: плагиат является острой проблемой в образовании, промышленности и научном сообществе. По данным Государственного университета – Высшая школа экономики в среднем около 50% студентов российских вузов «скачивают» рефераты и курсовые работы из сети Интернет [1]. Ещё одно исследование утверждает, что число студентов в американских средних школах, вовлеченных в различные виды плагиата, достигает 90 % [2]. Проблема плагиата существует и активно обсуждается в научном сообществе. Один из таких случаев публично

рассматривался в обществе IEEE и привёл к аннулированию ученой степени [3]. Проблема плагиата отмечена РОСФИНАДЗОРом при рассмотрении целесообразности использования бюджетных средств выделенных на НИОКР в 2009 году [1]. Не менее актуально эта проблема стоит при рассмотрении конкурсных проектов в государственные инновационные программы и фонды, а также при регистрации заявок в Роспатенте.

Проблема плагиата многогранна. Сам плагиат может варьироваться от прямого копирования текстов до плагиата идей. Само понятие «сходства» может быть формализовано различными способами [4]. В данной статье мы сконцентрируем внимание на технической стороне построения систем поиска сходства в текстах. Мы рассмотрим архитектуру систем обнаружения плагиата (СОП) и представим предложения по её улучшению, позволяющие исключить возможность присвоения авторства содержимого документов на этапе проведения работ по обнаружению плагиата.

Большинство коммерческих СОП не раскрывают деталей своей структуры и алгоритмов работы, чтобы снизить уровень уязвимости к различным способам их обмана (в теории информационной безопасности называемых атаками). Однако анализ декларируемых принципов работы СОП и детальное рассмотрение открытых СОП позволяют выявить типовую структуру для данных сервисов. Данная структура дает описание таких лидеров рынка СОП как «Антиплагиат» (РФ), Turnitin, SafeAssign (США). Типовая структура СОП отображена ниже на схеме (рис.1). Пользователь (студент или преподаватель) передает на проверку документ в СОП через информационную систему своего университета либо напрямую через web-интерфейс СОП. Затем содержимое документа преобразуется системой, с целью выделения «чистого» текста, т.е. избавления от форматирования документа присущего современным текстовым

[1] Информационная справка о результатах проверки использования министерствами, ведомствами, внебюджетными фондами и х подведомственными им организациями бюджетных средств, выделенных в 2009 году на научно-исследовательские и опытно-конструкторские работы. http://www.rosfinnadzor.ru/page/index/1236/page/7550

процессорам. На основе полученного текста строится запрос к базе данных документов СОП, результатом которого является набор документов, вероятных источников плагиата. Далее, после детального сравнения текстов документов, определяются схожие части и формируется отчёт о найденных совпадениях. В результате, пользователь получает отчёт о проведённой проверке, с указанием частей текста и источников «заимствования», если таковые имелись. При этом база данных документов СОП может содержать индексы открытых сегментов сети Интернет (как в случае с системой Turnitin), так и доступ к некоторым библиотекам с ограниченным доступом. Например, система «Антиплагиат» имеет доступ к базе данных диссертаций ВАК РФ.

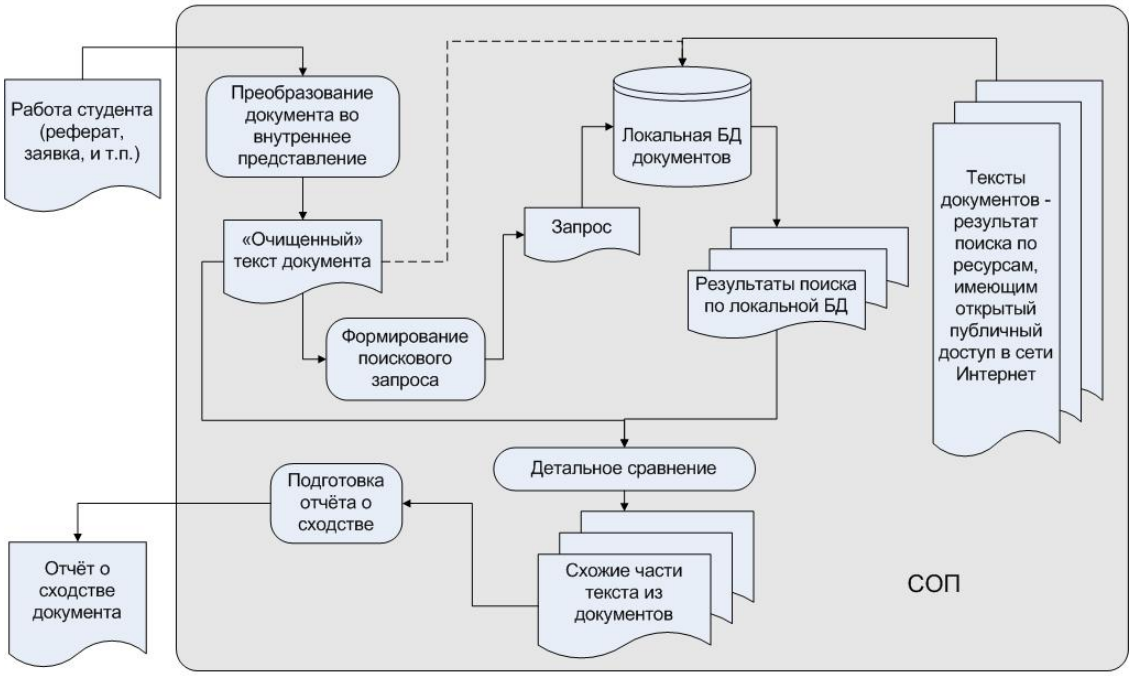

Рис. 1. Типовая архитектура СОП.

Здесь есть два важных момента, которые должны быть отмечены:

1. Содержимое проверяемого документа передается в СОП;

2. СОП, по соглашению с пользователем, сохраняет копию его документа в своей базе данных, чтобы в дальнейшем она могла служить исходным материалом для проверок.

Как поступить в случаях, когда необходимо осуществить проверку сведений, содержащихся в документе, на наличие плагиата и при этом необходимо соблюсти требования к конфиденциальности информации этих документов. Например, в случаях обработки заявок на изобретения, полезные модели, промышленные образцы или при рассмотрении конкурсных проектов в государственные инновационные программы и фонды, и т.п. Ограничение при помощи лицензирования не предупреждает техническую возможность неправомерного использования информации, передаваемой в СОП. Один из возможных способов – это использование общедоступных поисковых машин Интернета, когда часть текста помещается в кавычки и осуществляется глобальных поиск схожих документов – производится попытка найти точные совпадения текста в документах, опубликованных в открытом доступе в сети Интернет. Подобная несложная техника, может быть использована совместно с сервисом Google Alert [5]. Исследования показали, что такая техника поиска, с использованием публичных поисковых машин по критерию точного совпадения фраз, может быть очень эффективной [6], но такой поиск очень медленный, так как осуществляется вручную, и остается открытым вопрос, как определить ключевые фразы в документе по которым вести поиск.

Возможно использование бесплатных СОП, например системы Crot. Данная система имеет типовую архитектуру СОП, за тем исключением, что локальная база данных документов состоит только из внутренних ресурсов организации пользователя, а нахождение документов – потенциальных кандидатов источников плагиата, выполняется путём отправки запросов к поисковой машине Интернета. Система Crot выполняет исчерпывающий поиск, посылая запросы, сформированные «плавающим окном» [7]. Алгоритм «плавающего окна» выполняет прямой перебор фраз. Например, для Шекспировской фразы «to be, or not to be: that is the question» при длине окна $X = 4$ алгоритм сформирует 7 следующих запросов: «to be or not», «be or not to», «or not to be», «not to be that», «to be that is », «be that is the», «that is the question». Авторы

системы Crot указывают, что если значительная часть текста документа была присвоена из какого либо источника в Интернете, то нет необходимости послать все возможные запросы, а достаточно только 10% от этих числа, чтобы определить местонахождение этого источника [7]. Однако, из-за большого количества запросов, поиск «плавающим окном» значительно замедляет весь процесс обнаружения плагиата. Результаты проведённого эксперимента показали линейную зависимость времени поиска от количества слов в документе. Эксперимент был выполнен с 60 документами объёмом от 350 - 3500 слов. Эксперимент проводился на выделенном сервере с 100 Mbs интернет-каналом. Как показано на графике (рис. 2), время поиска составляло около пяти минут на каждые 1000 слов документа.

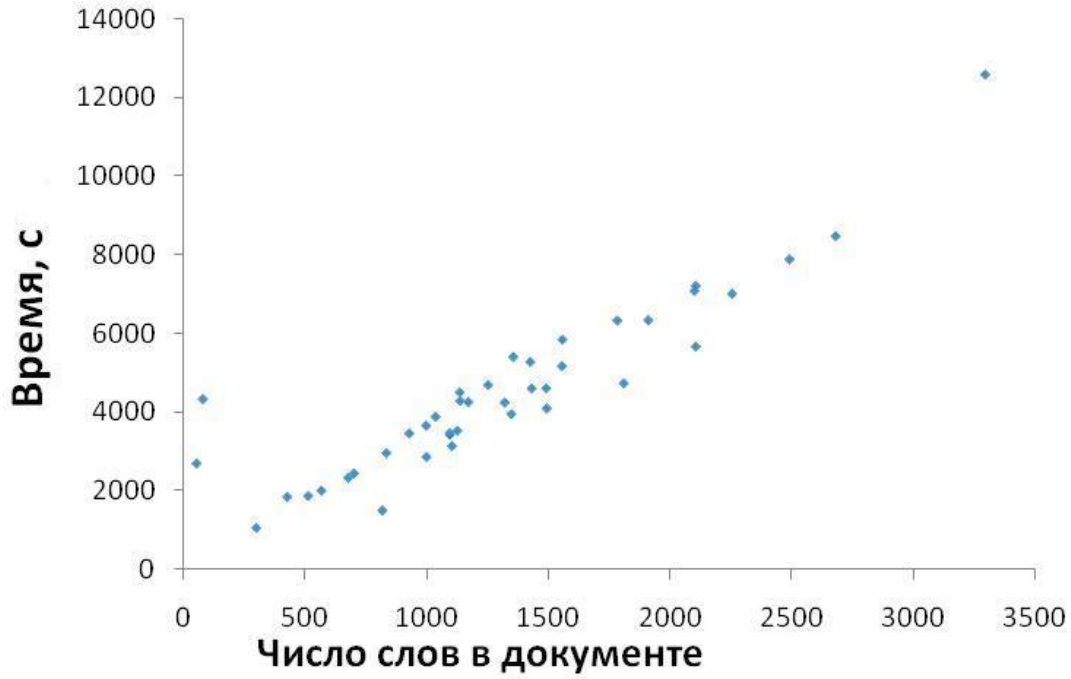

Рис. 2. Линейная зависимость между временем поиска и количеством слов в документе.

Предлагаемая архитектура сервиса определения плагиата

На схеме (рис. 3) отображена основная концепция предлагаемой архитектуры СОП. Сам сервис разделён на внутреннюю (клиентскую) часть, работающую на инфраструктуре пользователя, и на внешнюю (серверную) часть, работающую

на инфраструктуре сторонней организации. Внутренняя часть выполняет функции сервера для обращений со стороны пользователей и одновременно с этим является клиентом, выполняющим запросы к внешней части сервиса.

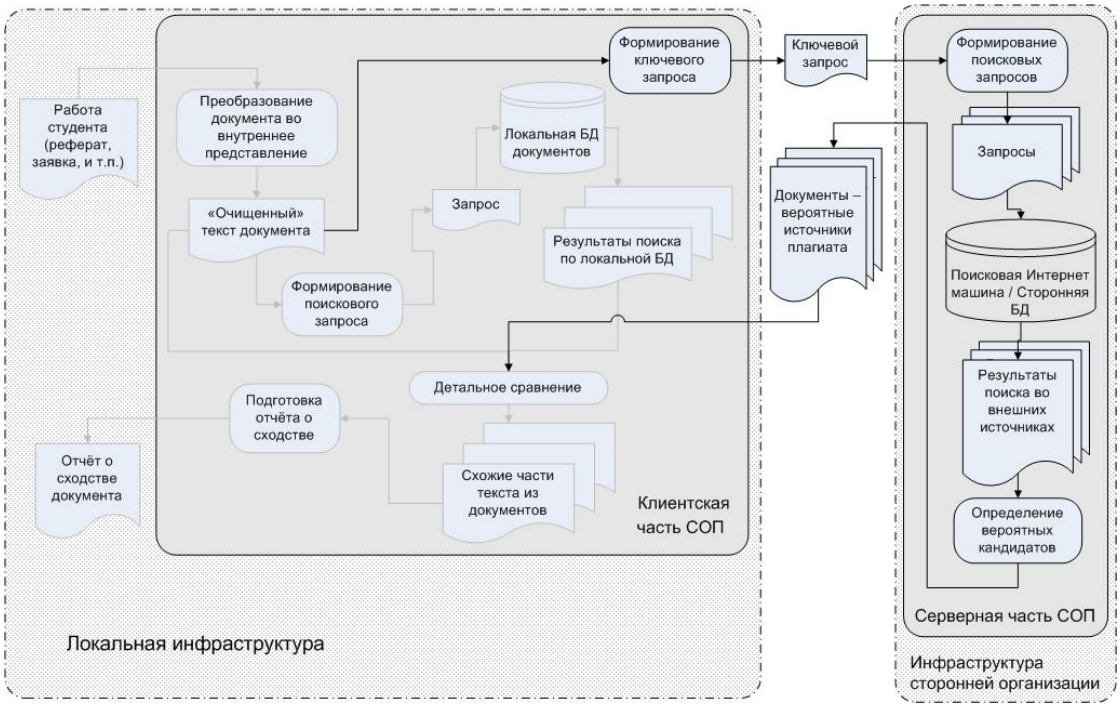

Рис. 3. Схема предлагаемой архитектуры СОП.

Предполагается, что разделённая структура сервиса будет выполнять работы в следующем порядке:

1. Клиентская часть системы, получив документ, переданный пользователем для проверки, преобразует его в «чистый» текст;
2. Клиентская часть создаёт специальный запрос, путём случайного выбора определённого количества запросов, сформированных методом «плавающего» окна;
3. Клиентская часть отправляет специальный запрос в серверную часть СОП;
4. Серверная часть, получив специальный запрос, формирует к поисковой машине Интернета запросы на нахождение документов, опубликованных в открытом доступе в сети Интернет;

5. Серверная часть получает множество ссылок на документы, как результат выполнения запросов, и выбирает те ссылки, которые встречаются чаще всего;
6. Серверная часть загружает документы, расположенные по выбранным ссылкам, и отправляет эти документы в клиентскую часть;
7. Клиентская часть производит детальное сравнение исходного текста и полученных документов;
8. Клиентская часть определяет схожие части текста и направляет отчет о сходстве исходного документа пользователю.

Большая часть работы сервиса определения плагиата происходит на оборудовании организации пользователя. Сторонняя же организация выполняет только глобальный поиск и предварительное (черновое) сравнение возможных источников плагиата из сети Интернет.

Для составления специального запроса в клиентской части СОП можно использовать алгоритм «плавающего окна», реализованный в системе Crot [7]. Данный алгоритм осуществляет полный перебор по всех фраз длины X, доступных в проверяемом документе. С учетом того, что большинство поисковых машин допускает в обрабатываемых запросах не более 10 слов, то можно ограничить $X <= 10$. Очевидно, что при длине текста Y слов, общее количество запросов $N = |Q|$, где $Q = \{q_1, q_2, ..., q_n\}$ массив запросов, определяющийся формулой $N = Y - X + 1$. С учетом того, что Y существенно больше X, можно утверждать, что подобный алгоритм сформирует число запросов близкое к числу слов в документе. Выполнение большого числа запросов к поисковой машине приводит к двум недостаткам, имеющихся у системы Crot:

- существенному времени поиска и повышенным требованиям к Интернет каналу, в случае распараллеливания выполнения данных запросов;

- из фраз, переданных поисковой машине, можно восстановить документ.

Исследования показывают, что порядок передачи поисковых фраз Q не влияет на результаты поиска [7]. Иными словами, если фразы массива Q будут перемешаны в случайном порядке, то результат поиска совпадет с результатом, полученным при последовательной передаче. Однако случайное перемешивание не решает проблемы возможного восстановления документа - случайно перемешанная мозаика может быть легко восстановлена простым перебором, так как в полном массиве соседние элементы q_i и q_{i+1} содержат $X-1$ совпадающих слов, что делает восстановление тривиальной задачей сбора мозаики фраз по пересечениям из $X-1$ слов. Однако, чтобы определить местонахождение источника «заимствования» нет необходимости послать все возможные запросы Q, а достаточно использовать только небольшой процент случайно выбранных элементов данного массива [7]. Рассмотрим насколько возможно использование этих свойств для ограничения передачи текста сторонней организации. Пусть $Q_1 \subseteq Q$ массив случайно выбранных элементов из полного массива запросов $Q : |Q_1| < |Q|$. Общее число слов в запросах, передаваемых в поисковую машину, будет равно $Y_s = |Q_1| * X$. Таким образом, если определить $|Q_1|$ из неравенства $Y_s < Y$, где Y общее число слов в документе, то можно гарантировать, что полное восстановление исходного текса на стороне поисковой машины из запросов переданных ей становится невозможным. То есть, если $|Q_1| < \frac{Y}{X}$, то исходный документ гарантированно невосстановим. В данной работе мы рекомендуем это ограничение для определения доли запросов при формировании специального запроса, направляемого в серверную часть СОП.

Очевидно, что приведенная архитектура увеличивает требования к мощности вычислительных ресурсов, используемых на стороне клиентской части СОП. Данные увеличения касаются как вычислительной мощности, так как требуются вычислительные затраты для детального сравнения документов, так и дискового пространства для хранения данных документов.

В части дискового пространства, в случае использования алгоритмов хеширования, схожих с алгоритмом Винновинг [8], для хранения хешей требуется хранить около 5% хешей, в расчете от числа символов в документе. При использовании 128- или 256-битных хешей и однобайтной кодировки текста можно говорить о том, что объем хешей будет примерно равен объему чистого текста в документе. Данное увеличение дискового пространства не представляется сколько-нибудь значимым с учетом постоянно снижающейся стоимости дисковой памяти.

В части вычислительных ресурсов хеширование одного документа не требует существенной вычислительной мощности процессора, при использовании локальных алгоритмов подобных Винновинг [8], так как затраты линейны по отношению к длине текста. Практический опыт применения алгоритмов хеширования показывает, что они предъявляют высокие требования к объему оперативной памяти. Высокие требования связаны с тем, что селекция элементов полного хеша в отпечаток документа требует чтобы полный хеш хранился в массиве в оперативной памяти во время селекции. Несмотря на то, что стоимость памяти неуклонно идет вниз, необходимо учитывать, что приложение, работающее в контейнере виртуальной машины или обычного скрипта, может быть ограничено в памяти. Одним из возможных путей обхода подобного ограничения является перенос селекции отпечатка из контекста приложения в контекст СУБД.

Рассмотрим класс локальных алгоритмов построения отпечатка документа. Пусть X - это полный хеш, рассчитанный таким образом: $X = \{x_1, x_2, ..., x_n\}$,

где $x_i = H(S_i)$;

S_i - строка длины n, выбранная из исходного текста начиная с i-го символа;

x_i – i-й хеш;

$H()$ – хеш-функция.

Очевидно, что длина полного хеша будет близка к числу символов в документе. Локальный алгоритм построения отпечатка предполагает выборку из X некоторого числа значений хеша по принципу ни одного, одно или несколько значений из окна длины W. Такая выборка позволяет существенно сократить длину отпечатка по сравнению с длиной хеша. Однако прямой перенос этих операций в СУБД будет неэффективен, так как число операций выборки из СУБД будет равно количеству окон, т.е. практически длине документа в символах. Однако если расширить окно до длины документа и ограничить выборку некоторой частью полного хеша, при этом, упорядочив хеш по значению элементов, данная операция потребует для запроса к СУБД только построение индекса и один оператор выборки. Важным вопросом при подобном переносе будет равномерность покрытия отпечатком полного хеша, т.е. хеш-функция $H()$ должна обеспечивать равномерное распределение результатов, независящее от порядка символов в строке S_i. Равномерность обеспечит отсутствие белых пятен в документе и как следствие отсутствие ложно отрицательных результатов сравнения. Теоретически хеш-функции типа MD5 подходят для формирования отпечатка. Экспериментальные данные на 500 документах из известной в Рунете библиотеке Мошкова подтверждают возможность переноса данных вычислений в СУБД.

Повышенные требования могут предъявляться к СУБД на клиентской части СОП. Фактически именно стоимость лицензии и обслуживания СУБД будет определять увлечение стоимости клиентской части СОП в предложенной архитектуре. Данное увеличение должно компенсироваться снижением стоимости работы серверной части СОП, так как пользовательские документы не хранятся на клиентской части СОП.

Заключение

В данной статье была рассмотрена новая архитектура СОП, позволяющая определить в проверяемом документе наличие присвоенного материала из текстов, опубликованных в открытом доступе в сети Интернет, при этом, гарантирующая, что сторонняя организация, осуществляющая поиск плагиата, не сможет получить «читабельное» содержимое исходного документа из передаваемой ей информации. Качество поиска документов в сети Интернет останется на прежнем уровне. Кроме того, предлагаемая архитектура увеличит производительность СОП и уменьшит нагрузку на локальную информационную инфраструктуру пользователя.

В дальнейшем мы планируем программно реализовать предложенную архитектуру СОП, а также совершенствовать её компоненты. Одно из возможных направлений исследований – это включение стилеметрии [9,10] (определения стиля текста) во внешнюю часть СОП, что позволило бы фильтровать результаты поиска на ранних стадиях, до их загрузки во внутреннюю часть СОП.

Список литературы

1. Ивойлова И. Украденные мысли: Половина студенческих рефератов и курсовых скачивается из Интернета. "Российская газета" - Федеральный выпуск №4830 от 20 января 2009 г. http://www.rg.ru/2009/01/20/referaty.html
2. Jensen, L.A., Arnett, J.J., Feldman, S.S. & Cauffman, E. (2002), It’s wrong, but everybody does it: academic dishonesty among high school students, Contemporary Educational Psychology, 27(2), 209-228
3. Kompas (2010) Saving Indonesia from Traps of Plagiarism. http://english.kompas.com/read/2010/04/28/02563687/Saving.Indonesia.from.Traps.of.Plagiarism

4. Федотов А.М., Барахнин В.Б., К вопросу о поиске документов «по аналогии» // Вестн. Новосиб. гос. ун-та. Серия: Информационные технологии. 2009. Т. 7, вып. 4. С. 5–7.
5. Carter M. (2008). How to Use Google Alerts to Detect Plagiarism. http://www.suite101.com/content/how-to-use-google-alerts-for-web-writers-a86525
6. Culwin F., & Child M. (2010) Optimizing and Automating the Choice of Search Strings when Investigating Possible Plagiarism. In Proceedings of 4th International Plagiarism Conference, Newcastle, June 2010. http://www.plagiarismadvice.org/documents/conference2010/abstracts/4IPC_0014.pdf
7. Butakov, S. and Shcherbinin, V. (2009) On the Number of Search Queries Required for Internet Plagiarism Detection. In Proceedings of the 2009 Ninth IEEE international Conference on Advanced Learning Technologies - Volume 00 (July 15 - 17, 2009). ICALT. IEEE Computer Society, Washington, DC, 482-483
8. Schleimer S., Wilkerson D., and Aiken A. (2003). Winnowing: Local Algorithms for Document Fingerprinting. Proceedings of the ACM SIGMOD International Conference on Management of Data, pages 76-85, June 2003.
9. Романов А.С. Структура программного комплекса для исследования подходов к идентификации авторства текстов // Доклады Томского гос. ун-та систем управления и радиоэлектроники" 2008. № 2 (18), часть 1, С. 106-109
10. Яцко В. А., Стариков М. С., Бутаков А. В. Автоматическое распознавание жанра и адаптивное реферирование текста // Научно-техническая информация. Серия 2, Информационные процессы и системы. 2010, №5. С.9-18.

ARCHITECTURE AND ALGORITHMS OF PLAGIARISM DETECTION SERVICE

The paper describes architecture for plagiarism detection service. The proposed novel approach fulfills the copyrights protection requirements for the documents submitted for the check up. The proposed architecture divides the checkup process into two parts. One part utilizes local resources within the organization and another one uses external engine to search documents on the Internet. The proposed division maintains the search quality and assures that copyrighted documents cannot be restored from the search requests.

Keywords: Plagiarism Detection, Service Architectures, Copyright Protection.