РАЗРАБОТКА И АПРОБАЦИЯ СИСТЕМЫ ПОИСКА ДУБЛИКАТОВ В ТЕКСТАХ ПРОЕКТНОЙ ДОКУМЕНТАЦИИ

Д.И. Игнатов,
д.ф.-м. н., Государственный Университет – Высшая Школа Экономики
С.О. Кузнецов,
Государственный Университет – Высшая Школа Экономики
skuznetsov@hse.ru
В.Б. Лопатникова, *к.т.н., ООО «Кварта ВК»*
И.А. Селицкий, *ООО «Мастерхост»*

В статье рассмотрена система поиска (почти) дубликатов в текстах проектной документации. Описаны ее архитектура, математические модели и алгоритмы поиска документов-дубликатов, а также их реализация. Предложены методики подбора оптимальных параметров методов и тестирования системы. Обозначены актуальные для подобных систем исследовательские задачи.

Постановка задачи и актуальность

Выявление дублирования в текстах проектной документации особенно важно при анализе эффективности результатов выполнения научно-исследовательских и опытно-конструкторских работ (НИОКР), финансируемых за счет бюджетных средств. Автоматизация поиска дубликатов проектных документов позволяет повысить качество приемки работ за счет упрощения деятельности экспертов при анализе результатов выполнения НИОКР. Внедрение автоматизированной системы поиска дубликатов обеспечит формирование фонда уникальных разработок, полученных в ходе выполнения НИОКР. Исключение дублирующихся разработок является неотъемлемой частью патентных исследований.

В настоящее время широкое распространение получила система «Антиплагиат» [Antiplagiat, 2008], предназначенная для обнаружения результатов плагиата в курсовых, реферативных и дипломных работах студентов. На сайте разработчика [Forecsys, 2008], компании «Форексис», в 2007 году сообщалось также о создании пакета «Антиплагиат.ВАК», предназначенного для выявления авторства и фактов плагиата в диссертационных работах. Потребность решать похожие задачи с помощью такого рода поисковых систем возникает и в других областях, связанных с научной и проектной деятельностью.

Цель нашего проекта состоит в разработке методологии и программного инструментария для анализа дублирования в текстах проектной документации НИОКР.

Основная задача проекта – отслеживать недобросовестное копирование документов других авторов, в том числе и своих собственных текстов.

Уточним, что в данной работе под дублированием мы будем понимать значительное совпадение фрагментов текстов.

Для достижения поставленной цели нами были отобраны представительные методы, наиболее подходящие для решения проблемы сравнения слабоструктурированных текстов разной тематической направленности. Был создан программный комплекс, основой которого является библиотека настраиваемых алгоритмов, реализующих выбранные методы анализа дублирования. Затем были проведены

эксперименты по подбору параметров алгоритмов, позволяющих с наименьшей потерей точности отнести анализируемый документ к классам «уникальный» или «дубликат». В результате нами предложена технология оптимизации анализа текстов, позволяющая исключить из поискового пространства те документы, которые заведомо не могут быть дубликатами. По сравнению с системой «Антиплагиат», использующей по существу синтаксические методы, разработанный программный продукт обладает рядом ценных для аналитика преимуществ. Во-первых, в системе используются известные модели и методы поиска дубликатов (см. раздел 3), поведение которых изучено для различных типов и коллекций документов, а их достоинства и недостатки хорошо известны. Аналитик (пользователь системы) может самостоятельно выбирать один или несколько методов для поиска сходных документов, изменять параметры, установленные для каждого из них по умолчанию. Тем самым снимается эффект «черного ящика» при использовании системы. Во-вторых, аналитику предоставляется возможность указать уровень текстуального сходства документов и даже выбрать вид агрегированной меры сходства (в случае если используется несколько методов одновременно). Это позволяет учесть специфику методов и изменить вес того или иного из них для достижения более точных результатов В работе предложена авторская методика автоматической калибровки методов по полноте и точности поиска, которая также представлена в системе в виде отдельного модуля.

Описание системы

Система представляет собой комплекс программных средств, предназначенных для анализа дублирования текстов проектной документации. Система также реализует поддержку жизненного цикла НИОКР за счет контроля этапов подготовки и проведения работ. В состав Системы входят следующие функциональные модули: модуль нормативной базы, библиотека НИОКР, модуль справочников и классификаторов, аналитический модуль, модуль выполнения отчетов.

Модуль нормативной базы реализует функции ввода и хранения нормативно-методических документов, регламентирующих порядок выполнения НИОКР.

Библиотека НИОКР реализует функции ввода и хранения данных о НИОКР в виде карточек НИОКР и текстов проектной документации как результата выполнения НИОКР.

Модуль справочников и классификаторов предназначен для ведения справочников юридических лиц, являющихся заказчиками и исполнителями НИОКР, а также справочников видов документов. В Системе доступен также справочник ГРНТИ.

Аналитический модуль реализует проверку текстов проектных документов НИОКР в форматах TXT, DOC, RTF, PDFна повторяемость в рамках заданной коллекции документов. Модуль позволяет задавать различные параметры анализа для каждой коллекции документов.

Модуль выполнения отчетов предназначен для просмотра информации о состоянии НИОКР в различных разрезах (тематика, состояние выполнения, исполнители, наличие документов-дубликатов и др.).

Система реализована по принципу клиент-серверной архитектуры, с инкапсуляцией ядра системы на уровне СУБД с доступом через Web-браузер. База данных и ядро системы реализуются на основе СУБД Microsoft SQL Server 2005, а дополнительные библиотеки – на языке C# и работают в среде .NET Framework CLR.

Архитектурно база данных Системы построена на основе объектно-реляционной модели и делится на три логических фрагмента:

- ✧ мета-данные, описывающие информационные объекты, их взаимосвязи и объекты визуальных представлений данных (экранные и печатные формы);
- ✧ внесённые данные о НИОКР;
- ✧ программное ядро, состоящее из хранимых процедур и функций на языке T-SQL и дополнительных библиотек на языке C#.

Web-клиент Системы реализован в виде ASP.NET приложения, работающего под управлением Microsoft Internet Information Services.

Методы поиска дубликатов

При нахождении множеств (почти) дубликатов документов основными являются следующие этапы:

1. Представление документов множеством признаков;

2. Составление образа документа путем выбора подмножества признаков;

3. определение отношения сходства на образах документов.

Затем, в зависимости от конкретной задачи, могут проводиться и комбинироваться следующие этапы:

4. Вычисление кластеров сходных документов;

5. Слияние кластеров сходных документов из различных коллекций;

6. Принятие решений о дублировании и компиляции.

На первом этапе, после снятия разметки (например, HTML), документы, как линейные последовательности слов (символов), преобразуются во множества слов, возможно с указанием кратности вхождения. Здесь двумя основными типами методов, определяющими весь возможный спектр смешанных стратегий, являются:

1. Синтаксические методы (в которых осуществляется выбор последовательностей символов, слов, или предложений);

2. Лексические (семантические) методы (в которых происходит выбор представительных языковых единиц).

Основным синтаксическим методом является шинглирование, когда документ, очищенный от разметки, пробелов и знаков препинания, сперва представляется набором всех подцепочек последовательных слов (символов) определенной длины. Такие цепочки, выбираемые с определенным сдвигом по линейной структуре текста, называют шинглами (от англ. shingle – черепица, чешуйка). Каждой цепочке сопоставляется хеш-код, при выборе которого обеспечиваются следующие важные свойства: равенство цепочек гарантирует равенство кодов (т.е. кодирование есть хеш-функция), а равенство кодов говорит о высоком сходстве цепочек. Наиболее распространенными являются хэш-коды SHA1 [NIST, 1995] и Rabin [Broder, 1997]. Необходимым условием является миниальное число коллизий для хеш-функций. Из множества хеш-кодов цепочек, в соответствии с некоторой схемой рандомизации, выбирается подмножество, которое и служит т.н. «отпечатком» (образом) документа. Данный метод используется во многих системах определения сходства документов, а также в таких поисковых системах как Google и AltaVista.

Среди способов выбора подцепочек используются следующие методы выбора подцепочек: фиксированный, или логарифмического от длины текста, выбор каждой k-й цепочки и т.д.

В методах лексического типа реализуется отбор множества представительных слов исходя из показателей значимости этих слов. В множество значимых слов не включаются слова из заранее фиксированного списка стоп-слов. Список стоп-слов для каждого языка является стандартным и включает в себя предлоги, артикли, вводные слова и т.п. Показателями значимости служат частотные характеристики: для дальнейшего анализа отбираются слова, чьи частоты лежат в некотором интервале, так как высокочастотные слова могут быть неинформативными, а низкочастотные — опечатками или случайными словами.

В лексических методах, таких как, в известном методе I-Match [Chowdhury et al., 2002], используют большой текстовый корпус для порождения лексикона, то есть набора представительных слов. Документ представляется множеством тех его слов, которые входят в лексикон. При порождении лексикона отбрасываются самые низкочастотные и самые высокочастотные слова. I-Match порождает сигнатуру документа (множество слов-термов), а по ней — хэш-код документа, причем два документа получают один хэш-код с вероятностью равной их мере сходства (по метрике косинуса). I-Match порой неустойчив к изменениям текста [Ko?czetal., 2004], например, к рандомизации по существу одних и тех же спамерских сообщений. Для устранения этого недостатка, помимо стандартной сигнатуры, создается еще K сигнатур, каждая из которых получается случайным удалением некоторой доли всех термов из исходной сигнатуры (таким образом, все новые сигнатуры являются подмножествами исходной). Два документа можно считать очень сходными, если их наборы из K+1 сигнатуры имеют большое пересечение хотя бы по одной из сигнатур. Такой подход сходен с подходом на основе супершинглов (конкатенации шинглов), когда сходство документов определяется как совпадение хотя бы одного супершингла [Ko*l*cz et al., 2004].

В лексическом методе [Ilyinsky et al., 2002] большое внимание уделяется построению словаря — набора дескриптивных слов, который должен быть небольшим, но хорошо покрывать коллекцию, а присутствие каждого из слов в образе документа устойчиво по отношению к малым изменениям документов. Проблема автоматического порождения адекватного словаря для анализа сходства документов по определенной теме связана с составлением представительной коллекции документов по данной теме – корпуса текстов, что затрудняет применение лексических методов без постоянной поддержки такого корпуса.

На первом этапе можно учесть структуру текста и его шаблонов, представляя исходный документа в виде кортежа разделов определенных шаблонами и производя шинглирование внутри компонент кортежа.

На втором этапе из документа, представленного множеством синтаксических или лексических признаков, выбирается подмножество признаков, образующее краткое описание (образ) документа. В синтаксических методах такого рода отбор чаще всего осуществляется с помощью схем рандомизации [Broder, 1997, Broder et al.,1997, 1998], в лексических методах — с помощью методов выбора

существенных слов, например, на основе заранее созданных словарей[Chowdhury et al., 2002], или на основе какой-либо меры существенности слова для текста [Ilyinsky et al., 2002].

Техника отбора подстрок, начиная с некоторого определенного места в документе, также позволяет учесть структуру документа с использованием так называемых «якорей» – особых мест документа: начала абзаца, раздела, ключевого слова и т.п. [Hoad et al., 2003]. Являясь одной из самых эффективных, эта техника может потребовать много времени для тонкой ручной настройки по каждой коллекции документов.

На третьем этапе определяется отношение сходства на документах. Для этого используется определенная числовая мера сходства, сопоставляющая двум документам число на отрезке [0, 1], которое характеризует сходство, и некоторый параметр – порог, превышение которого свидетельствует о большом сходстве документов или о том, что документы являются (почти) дубликатами друг друга [Broder, 1997, Broder et al.,1997, 1998].

На четвертом этапе, на основе отношения сходства документы могут объединяться в кластеры (почти)дубликатов. Определение кластера также может варьироваться. Самый частый используемый на практике подход [Broder, 1997]: если документам сопоставить граф, вершины которого соответствуют самим документам, а ребра – отношению «быть (почти) дубликатом», то кластером объявляется компонента связности такого графа. Достоинством такого определения является эффективность вычислений: компоненту связности можно вычислить за линейное время от числа ребер. Недостаток такого подхода: отношение «быть (почти) дубликатом» не является транзитивным, поэтому в кластер сходных документов могут попасть абсолютно разные документы. Противоположным – «самым сильным» – определением кластера, исходя из отношения «быть (почти) дубликатом», является его определение через клики графа (максимальные по вложению полные подграфы) коллекции документов. При этом каждый документ из кластера должен быть сходным со всеми другими документами того же кластера. Такое определение кластера более адекватно передает представление о групповом сходстве, но, может встретить трудности при масштабировании системы поиска документов-дубликатов в силу того, что поиск клик в графе – классическая труднорешаемая задача.

Указанные два определения кластера задают спектр промежуточных формулировок, в которых можно находить необходимый баланс между точностью и полнотой определения кластеров, с одной стороны, и сложностью вычисления кластеров с другой стороны.

Другие методы определения кластеров основаны на вариациях стандартных методов кластерного анализа, например, когда при отнесении очередного объекта к кластеру используется расстояние до центров масс кластеров. Такого рода методы существенно зависят от последовательности поступления объектов, образующих кластеры. Это означает, что документы, попавшие в коллекцию раньше, сильнее определяют структуру кластера, чем документы, поступившие позднее.

Недостатки методов кластеризации, основанных на мерах сходства между документами, является частая возможность объединения в один кластер документов лишь попарно сходных друг с другом, но не имеющих общекластерного сходства. Альтернативой такому подходу служат методы, в которых кластер сходных документов определяется как множество документов, у которых число общих элементов описания превышает определенный порог. Такие методы основаны на би-кластеризации [Mirkin et al., 1995] и решетках формальных понятий [Ganter et al., 1999].

На пятом этапе работы системы возможен учет работы с распределенными коллекциями документов. Здесь возможны две противоположные стратегии (задающий спектр промежуточных между ними): рассмотрение дублирования по отношению к каждой коллекции по отдельности и рассмотрение дублирования по представителям кластеров из разных коллекций, например, в работе [Yang et al., 2006] предлагается использовать документ источник в качестве представителя кластера.

Шестой этап работы системы предусматривает принятие решений о дублировании и компиляции. Здесь необходимо создание удобного интерфейса, позволяющего ЛПР просматривать документы, чье сходство было установлено автоматически и принятие окончательного решения о дублировании или плагиате, в том числе возникшего путем компилирования нескольких документов из разных коллекций. Так же на этом этапе определяется документ-источник для каждого кластера.

Реализация поиска дубликатов в системе

Для реализации в Системе отобраны следующие модификации методов:

- ✧ модификация методаI-Match со сравнением сходства по метрике косинусов;
- ✧ модификация методаI-Match со сравнением сходства по метрике TF-IDF;

- ✧ DSC (полнотекстовый поиск);
- ✧ DSC (первый в sперестановках);
- ✧ DSC-SS.

Пользователями настраиваются следующие параметры методов:

- ✧ для I-Match: нижний частотный порог слов α, попадающих в словарь (редкие слова отбрасываем);
- ✧ для I-Match: верхний частотный порог слов β, попадающих в словарь (частые слова отбрасываем);
- ✧ для DSC, DSC-SS – сдвиг (расстояние между шинглами);
- ✧ для DSC, DSC-SS– размер шингла (число слов в одном шингле);
- ✧ для DSC-SS – размер супершингла;
- ✧ пороговое значение, при превышении которого документ становится кандидатом в дубликаты.

Приведем краткую схему алгоритма анализа документа на дублирование перечисленными выше методами (*рис. 1*).

Далее приведем сценарий, согласно которому пользователь выполняет анализ проектных документов на дублирование в Системе.

Проведение анализа документов в Системе

Система анализирует документы, представленные в форматах TXT, DOC, RTF иPDF. В процессе проверки выполняются следующие действия.

ШАГ 1. Пользователь загружает файлы проектной документации НИОКР в Систему. Документам, впервые поступившим в систему, присваивается статус «Не проверен».

ШАГ 2а. Система выполняет автоматическую проверку всех непроверенных документов с коллекцией документов, уже хранящихся в системе и признанных уникальными. Автоматическая проверка организована в виде назначенного задания ОС Windows и запускается в установленное время (например, ночью).

ШАГ 2б. Пользователь выполняет проверку всех непроверенных документов в ручном режиме.

ШАГ 3. По результатам автоматической проверки документы, получившие значение сходства выше установленного порогового значения, считаются «подозрительными», система присваивает им статус «кандидат в дубликаты». Документы со значением сходства, не превышающим установленный порог, признаются оригинальными, им присваивается статус «уникальный». Для каждого документа,

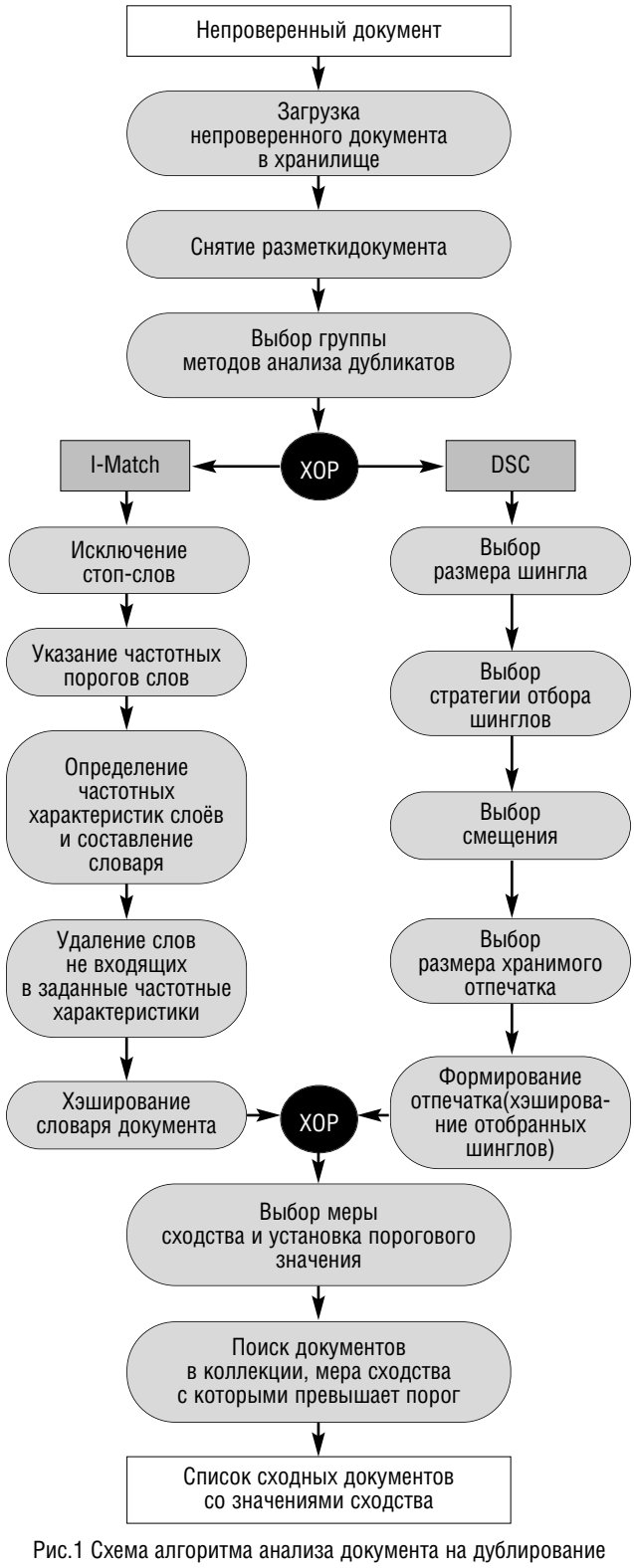

Рис.1 Схема алгоритма анализа документа на дублирование

признанного кандидатом в дубликаты, указывается источник сходства (или несколько источников – документов из уникальной коллекции).

ШАГ 4. Пользователь принимает решение – является ли документ уникальным или дубликатом – после просмотра двух документов с выделенными фрагментами совпадающих текстов. Документам,

признанным аналитиком дубликатами, Система присваивает статус «дубликат».

ШАГ 5. В случае необходимости пользователь может изменить параметры анализа и провести повторный анализ и в зависимости от результатов принять решение о статусе документа.

ШАГ 6. Пользователь может выполнить отчет о проверенных документах за определенный период.

Подбор параметров и тестирование

Авторами была разработана и апробирована методика подбора параметров работы алгоритмов поиска дубликатов. Были выбраны следующие способы модификации документов, *см. табл. 1.* Суть тестирования заключается в проверке способности методов выявлять (почти) дубликаты полученные из исходных указанными способами. Кроме того авторы произвели подбор наилучших параметров методов для различных типов документов. Для подбора параметров методов, реализованных в Системе, и проведения тестирования авторами была разработана специальная утилита – «Генератор тестов». Программа «Генератор тестов» позволяет автоматизировать создание тестовых изменённых документов и расчёт параметров сходства и показателей работы методов.

Таблица 1

Способы генерации тестовых данных

№	Название метода	Параметры метода
1	Перестановка параграфов	Доля переставляемых параграфов
2	Удаление параграфов	Доля удаляемых параграфов
3	Добавление параграфов	Доля добавляемых параграфов
4	Замена слов	Доля заменяемых слов
5	Добавление повторяющихся абзацев	Количество абзацев и количество повторений каждого
6	Замена букв	Множество пар букв: (<исходная буква>, <новая буква>)

В табл. 1 приведены способы создания тестовых данных на основе исходных документов. Для оценки качества нахождения документов-дубликатов используются стандартные для информационного поиска меры: полнота, точность и *F*-мера. В ходе проведения тестирования выяснилось, что даже при сохранении малой доли (до 10%) исходного документа в тестовой коллекции удается адекватно выявлять такие документы как дубли при малом пороге сходства. Относительное число «ложных дубликатов» в худшем случае оказывается невелико (≤30%).

Таблица 2

Оптимальные параметры тестируемых методов

Метод анализа	Интервал для частотных порогов	Размер шингла, слов	Смещение, слов	Размер супершингла
I-Match(cos)	(0.35, 0.85)	–	–	–
I-Match (TF-IDF)	(0.35, 0.85)	–	–	–
DSC (Fulltext)	–	10	1	–
DSC	–	10	1	–
DSC-SS	–	10	1	5

Для метода I-match с помощью алгоритма оптимизации Хука-Дживса были найдены верхняя и нижняя частотные границы для построения словаря по исходной коллекции из 13 документов. Для методов группы DSC проводился подбор оптимальных значений параметров – размер шингла, размер супершингла, величина сдвига для слов русского языка и размер образа документа (для неполнотекстовых методов размера образа в 100–150 шинглов вполне достаточно).

Следующая проблема, которую приходилось решать – это способ агрегирования значений сходства, полученных всеми реализованными в Системе методами.

Для проведения соответствующих экспериментов в генератор тестовых документов загружались два документа одного из рассматриваемых типов, далее номера файлов документов №0 и №10 соответственно. Документы №0 и №10 являются документами тестовой коллекции. На их основе с использованием способов из *табл. 1* были сгенерированы документы, которые затем сравнивались с документами данной коллекции. В нашем случае было порождено 9 документов с номерами №1,..., №9. Генерация производилась следующим образом: сначала удалением 10%, 20%, ... , 90% абзацев из файла №0, а потом добавлением 10% к 90%, 20% к 80%, 30 к 70%, 40% 60%, ..., 90% к 10% частей файла №10 к оставшейся части файла №0. Все изменения, указанные в *табл. 1*, производились случайным образом, например, добавление случайных абзацев в случайное место, замена случайных слов и т.д.

При этом истинное значение сходства двух документов вычислялось по мере Жаккара:

$$Sim(A,B) = \frac{|A \cap B|}{|A \cup B|},$$

где A и B – представление документов в виде цепочек слов, а не в виде множеств.

Далее использовался подход из области машинного обучения, который называется бустингом (boosting).Предполагается, что мы оцениваем значение некоторой величины, в данном случае сходства. При этом мы имеем оценку сходства для каждого из методов I-Match, DSC, DSC-SS и DSC-Fulltext для наблюдений с 1 по 20. Используя парную регрессию со свободным членом как линейный классификатор, мы строим линейную модель для каждого из методов:

$$y = c_{I-match} + \alpha x_{I-match} + \varepsilon_{I-match}$$
$$y = c_{DSC} + \beta x_{DSC} + \varepsilon_{DSC}$$
$$y = c_{DSC-SS} + \gamma x_{DSC-SS} + \varepsilon_{DSC-SS}$$
$$y = c_{DSC-Fulltext} + \delta x_{DSC-Fulltext} + \varepsilon_{DSC-Fulltext},$$

где $c_{[\text{название метода}]}$ – свободный член регрессии;

α, β, y и β – коэффициенты при значении сходства $x_{[\text{название метода}]}$, найденного конкретным методом;

$\alpha_{[\text{название метода}]}$ – остатки регрессии.

Нами использовалось 4 типа сверток (*рис. 2*), первая из которых представляла собой среднеарифметическое значение сходства. В терминах бустинга необходимо построить так называемый сильный классификатор (свертку) на основе нескольких слабых в предположении, что взвешенные значения сходства, полученные разными методами, компенсируют недостатки каждого из методов в отдельности.

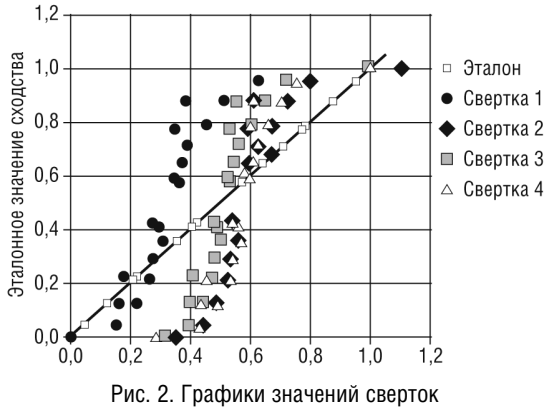

Рис. 2. Графики значений сверток

Наилучший результат показала свертка 4 типа, представляющая собой среднее нормированных регрессий.

$$S(x_{I-match}, x_{DSC}, x_{DSC-SS}, x_{DSC-Fulltext}) =$$
$$= \frac{1}{4}\left(\frac{\alpha x_{I-match} + c_{I-match}}{\alpha + c_{I-match}} + \frac{\beta x_{DSC} + c_{DSC}}{\beta + c_{DSC}} + \right.$$
$$\left. + \frac{\gamma x_{DSC-SS} + c_{DSC-SS}}{\gamma + c_{DSC-SS}} + \frac{\delta x_{DSC-Fulltext} + c_{DSC-Fulltext}}{\delta + c_{DSC-Fulltext}} \right).$$

Для этой свертки наблюдалось наименьшее сходства $S_{\min}$=0,11 при эталонном значении $S_{Эт}$=0,и наибольшее значение положительного сходства $S_{\max}$=0,77 при $S_{Эт}$=0,95. Применение свертки необходимо для сглаживания эффектов завышения и/или занижения значений сходства, выдаваемые отдельными методами. При этом эксперт обязан установить минимальный порог сходства для кандидатов в дубликаты несколько выше $S_{\min}$ для того чтобы уменьшить число «ложных срабатываний».

Направления дальнейшей работы

Важной проблемой для дальнейшего развития продукта является *возможность выявления степени* компиляции документов, т.е. определения источников из которых получен документ (например, как результат множественного плагиата). Немаловажной проблемой является также *учет структуры документа при анализе.* Для обработки документов с учетом их структуры должен использоваться специальный конвертер (парсер) документов, который не обязательно входит в состав аналитического модуля системы. Конвертер должен предоставлять аналитику-эксперту возможность самостоятельной настройки фильтра шаблонных фраз. Для представления документа рекомендуется использовать древесную структуру, т.о. документ после обработки хранится в виде дерева разделов. В качестве технологии реализации (представления) рекомендуется использовать XML или SGML. В состав конвертера необходимо включить метод автоматического построения структуры дерева для данного типа документов, с возможностью предварительного задания шаблонов. После построения древесного представления документов их попарное сходство рассчитывается покомпонентно. Корневой узел содержит название документа, промежуточные узлы – названия разделов, листья – содержания разделов нижнего уровня. При *построении кластеров сходных документов* мы предлагаем использовать подход, описанный нами в [Кузнецов и др., 2005] и [Игнатов и др., 2006], основанный на использовании частых замкнутых множеств признаков (frequent closed itemset mining). При этом в роли объектов выступают элементы описания (шинглы или слова), а в роли признаков – документы. Для такого представления «частыми замкнутыми множествами» являются замкнутые множества документов, для которых число общих единиц описания в образах документов превышает заданный порог. Таким образом, имея набор частых множеств признаков – документов по некоторой коллекции, можно судить о степени сходства конкретного документа с определенной группой документов коллекции. Такой мерой может выступать относительное

число общих шинглов некоторой группы документов и вновь внесенного документа. Еще одним важным вопросом при выявлении дублирования является учет типологии документов, а именно формальных признаков, таких, как:

- ✧ тип документа;
- ✧ научная область;
- ✧ область применения;
- ✧ стандарты (ГОСТ).

Очевидное решение – использование существующих древесных классификаторов и перечней. Такие классификаторы имеют ряд недостатков. Например, невозможно повторение одного и того же раздела на различных уровнях иерархии. Предлагаемый подход опирается на прикладную алгебраическую дисциплину – анализ формальных понятий (АФП)[Ganter et al., 1999]. В рамках АФП мы предлагаем использовать решеточную классификацию. Преимущества решеточного классификатора состоят в том, что в нем снимается проблема множественности наследования, когда один и тот же документ относится к разным типам. Другими словами, при использовании древовидного представления возможны только вкладывающиеся друг в друга надклассы, а в решетке классы могут пересекаться. Таким образом, документ не обязательно имеет одного родителя. Это свойство обеспечивает гибкость решеточного классификатора.

Благодарности

Авторы статьи выражают благодарность за активное участие в обсуждении математических моделей поиска документов-дубликатов и алгоритмических аспектов разработанной системы ведущему научному сотруднику ВИНИТИ РАН Виноградову Д.В. и доценту кафедры анализа данных и искусственного интеллекта ГУ-ВШЭ Объедкову С.А. Авторы выражают большую признательность коллективу разработчиков ООО «Кварта ВК» - Калинкиной Ю.А., Звездиной Е.А., Кузнецову А.С. и особенно его руководителю – Еськину И.Ю за успешную реализацию программного инструментария. Авторы также благодарят Научный фонд ГУ-ВШЭ, предоставивший грант в рамках проекта «Учитель-ученики» для разработки алгоритмов бикластеризации, необходимых для дальнейшего развития проекта. ■

Литература

[Antiplagiat, 2008] http://www.antiplagiat.ru/ – сайт Интернет-сервиса AntiPlagiat.ru компании ЗАО «Анти-Плагиат».

[Forecsys, 2008] http://forecsys.ru/ – сайт компании ЗАО «Форексис», официального разработчика Интернет-сервиса AntiPlagiat.ru.

[NIST, 1995] NIST, "Secure Hash Standard", Federal Information Processing Standards Publication 180-1, 1995.

[Broder, 1997] A. Broder, On the resemblance and containment of documents, in Proc. Compression and Complexity of Sequences (SEQS: Sequences'97).

[Chowdhury et al., 2002] A. Chowdhury. O. Frieder, D. Grossman, and M. McCabe. Collection statistics for fast Duplicate document detection. In ACM Transactions on Information Systems (TOIS), Volume 20, Issue 2, 2002.

[Koʻcz et al., 2004] A. Koʻcz, A. Chowdhury, J. Alspector. Improved Robustness of Signature-Based Near-Replica Detection via Lexicon Randomization. Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining, page 605–610, Seattle, WA, USA, 2004.

[Ilyinsky et al., 2002] S. Ilyinsky, M. Kuzmin, A. Melkov, I. Segalovich. An efficient method to detect duplicates of Web documents with the use of inverted index. WWW Conference 2002.

[Broder et al., 1998] A. Broder, M. Charikar, A.M. Frieze, M. Mitzenmacher, Min-Wise Independent Permutations, in Proc. STOC, 1998.

[Broder et al., 1997] A. Z. Broder, S. C. Glassman, M. S. Manasse, and G. Zweig. Syntactic clustering of the web. In Proceedings of WWW6'97, pages 391–404. Elsevier Science, April 1997.

[Hoad et al., 2003] T. Hoad and J. Zobel. Methods for identifying versioned and plagiarized documents.In Journal of the American Society or Information Science and Technology, Vol 54, I 3, 2003.

[Mirkin et al., 1995] B. Mirkin, P. Arabie, L. Hubert (1995) Additive Two-Mode Clustering: The Error-Variance Approach Revisited, Journal of Classification, 12, 243–263.

[Ganter et al., 1999] B. Ganter and R. Wille, Formal Concept Analysis: Mathematical Foundations, Springer, 1999.

[Yang et al., 2006] H. Yang and J. Callan. Near-Duplicate Detection by instance-level constrained clustering. In Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in information retrieval. Pages 421–428. Seattle, Washington 2006.

[Кузнецов и др., 2005] Кузнецов С.О., Игнатов Д.И., Объедков С.А., Самохин М.В. Порождение кластеров документов дубликатов: подход, основанный на поиске частых замкнутых множеств признаков. Интернет-математика 2005. Автоматическая обработка веб-данных. Москва: Яndex, 2005, стр. 302–319.

[Игнатов и др., 2006] Игнатов Д.И., Кузнецов С.О. О поиске сходства Интернет-документов с помощью частых замкнутых множеств признаков// Труды 10-й национальной конференции по искусственному интеллекту с международным участием (КИИ'06). – М.:Физматлит, 2006, Т.2, стр.249–258.