Использование статистической информации при выявлении схожих документов

Косинов Д.И.
ВГУ
kosinov@cs.vsu.ru

Аннотация

Во многих современных задачах необходимо уметь выделять из больших объемов текстов подобные друг другу, в связи с чем требуются эффективные методы для определения схожих документов. Предложен метод построения локальной сигнатуры документа на основании исключительно статистических параметров его содержимого, без использования глобальных коллекций. Набор параметров подбирается, исходя из соображений устойчивости к различным видам модификаций документа. Проведен ряд экспериментов, использующих некоторые из этих параметров. Показана возможность использования данного подхода в условиях больших объемов документов.

1. Введение

Колоссальное количество документов в Интернете, помимо очевидных преимуществ обилия информации, порождает также множество проблем. Отсутствие прямого каталога и необходимость поиска нужной информации в гигантских массивах данных вызывает потребность в сложнейших поисковых машинах, создание которых требует детальной проработки множества специфических методов.

Одной из задач, с которой сталкивается любая поисковая машина, является задача определения схожести различных документов между собой. Выявление дубликатов позволяет устранять повторяющиеся документы в списках-результатах запросов, уменьшать размера индекса путем устранения избыточности, обнаруживать плагиат и распознавать массовые почтовые рассылки (спам).

Источники дубликатов в сети также могут быть самыми различными. Модификации документов можно разбить на несколько видов:

1. Возникающие при создании зеркал (копий) сайтов и документов.
    a. Смена заголовка и подписи документа (header, footer).
    b. Расположение и наполнение навигационных элементов.
    c. Различные HTML-адреса, ведущие к одному документу.
2. Возникающие при преобразовании документа:
    a. Смена формата документа (например, DOC, PDF и HTML).
    b. Различные версии и копии документа.
    c. Шаблонные тексты (например, лицензионные соглашения, описания товаров в интернет-магазинах).
3. Возникающие при редактировании документа:
    a. Вставка и удаление абзацев и предложений.
    b. Перестановка предложений и абзацев местами.
    c. Форматирование текста.
    d. Незначительные изменения текста, ошибки оператора.
    e. Усечение в меньший объем.
4. Применяемые спамерами в массовых рассылках.
    a. Вставка в текст наборов случайных символов.
    b. Произвольная вставка и удаление пробелов.
    c. Замена символов одной раскладки на символы другой, имеющие аналогичное написание.
    d. Вставка значащих слов в текст в случайных позициях.

Само понятие схожих документов, дубликатов (также называемых почти-дубликатами) крайне неоднозначно и, в зависимости от необходимости, по-разному трактуется различными авторами. Как показано в [2], существует «парадокс измерения»: пытаясь дать объективную оценку, исследователи в ходе работы делают много субъективных допущений. В результате, незначительные расхождения в определениях «дубликатов» и умолчание части параметров экспериментов авторами приводят к серьезным расхождениям результатов, и зачастую сложно определить ценность и место проведенных исследований.

В качестве базового определения можно понимать под почти-дубликатом следующее: две страницы имеют одно и то же содержание, за исключение отдельных, относительно небольших, различий.

2. Предыдущие работы

Методы, применяемые для выявления схожих документов, можно разделить на два класса. Полнотекстовые (использующие для сравнения весь текст документа) методы анализа сходства документов отличаются высоким качеством работы, но очевидно медлительны на больших объемах данных. Значительно более высокую производительность, достигаемую ценой некоторого

снижения качества сравнения, обеспечивают методы сравнения на основе отпечатков документа (fingerprints). Как понятно из названия, при использовании этих методов производится сравнение не самих документов, а лишь некоторого набора идентифицирующих их контрольных сумм (хешей).

Документы считаются почти-дубликатами, если доля совпавших отпечатков превышает некий порог. Разница в различных методах в основном состоит в различии конкретных алгоритмов получения отпечатков документа.

Перед созданием отпечатком обычно документ проходит через ряд преобразований:

1. Удаление лишних пробелов.
2. Удаление HTML-разметки.
3. Удаление пунктуации.
4. Приведение к единому регистру символов.
5. Удаление стоп-слов.
6. Удаление слишком длинных и слишком коротких слов.
7. Стемминг слов (выделение значащей части).

Алгоритм шинглирования [3], являющийся одним из наиболее часто встречающихся методов выявления дубликатов, выглядит следующим образом: документ преобразуется во множество цепочек элементов определенной длины, где каждой цепочке ставится в соответствие хеш-код, называемый шинглом (shingle). Два документа будут считаться схожими, если достаточно большое число шинглов, составленных из их текстов, совпадет. Таким образом, сходство документов можно выразить отношением числа одинаковых шинглов к общему количеству их шинглов.

В попытке уменьшения числа шинглов появляется алгоритм супершинглирования, когда над разбитым на группы набором шинглов документа рассчитывается еще некоторое количество (обычно 1-6) контрольных сумм, называемых супершинглами. Однако при работе с короткими документами (которыми являются большинство писем, например) его эффективность довольно низка.

Различия между методами на основе шинглирования можно уложить в рамки четырех условий [8]:

1. Размер подстроки, вырезаемой из документа.
2. Функция, вычисляющая хеш на основе заданной подстроки.
3. Число хешей, из которых формируется отпечаток документа.
4. Алгоритм выбора (фильтрации) хешей.

Эффективность метода в приложении к конкретной задаче может значительно меняться в зависимости от набора используемых параметров. Работа [10] отмечает отсутствие исследований, посвященных систематическому анализу взаимного влияния различных параметров алгоритма шиглирования на качество распознавания. Особое внимание уделяется отбору шинглов, и показано, что тщательный подбор алгоритма фильтрации позволяет заметно улучшить точность.

Кроме того, существенным недостатком методов, использующих отпечатки, является их неспособность отделить значащие части текстов от незначащих. В результате, при полном разбиении документов на фрагменты огромный объем (до 98% в соответствии с [1]) хранимых данных является избыточным, так как большинство фрагментов встречается только единожды. Это бывает слишком накладно в реальных условиях. При использовании стратегии выборочного хранения отпечатков неизбежно возникновение потерь, вызванных отбрасыванием значащих частей информации. Алгоритм SPEX, предложенный в [1], позволяет отбрасывать незначащие части документов, сохраняя при этом приемлемое качество детектирования дубликатов.

Крупномасштабное исследование [7] сравнивает результативность алгоритма шинглирования и подхода, основанного на случайных проекциях [4] при работе с очень большими объемами данных (более 1.5 миллиарда документов). Показано, что второй алгоритм при схожих параметрах эксперимента показывает большую точность. Также отмечена обоюдно низкая точность распознавания дубликатов документов из одного источника, что обусловлено частым использованием шаблонов web-страниц (например, в интернет-магазинах).

Альтернативный подход, использующий в качестве минимальной частицы документа термы, без учета их позиции относительно друг друга, служит основой для лексических методов, из которых одним из самых распространенных является I-Match [5]. Используя глобальную статистику IDF для отбора множества представительных слов, он демонстрирует хорошие результаты (особенно для небольших документов). В случае использования единой сигнатуры документа алгоритм весьма неустойчив к небольшим изменениям документов (в частности, к спамерской модификации писем), но результативность можно улучшить, перейдя к набору из нескольких сигнатур, каждая из которых получается путем рандомизации лексикона [9]. Тем самым, однако, увеличивается сложность сравнения из-за появления множества отпечатков для каждого документа.

Эффективность используемых методов является одним из важнейших условий в случае применения их в Интернете. Помимо миллиардов требующих индексирования веб-страниц, ежедневно обновляются сотни тысяч лент новостей (RSS) и рассылаются миллионы спам-писем. В этих условиях скорость работы алгоритма является критичной величиной.

В этом случае логичным способом ускорения сравнения становится переход от множества

отпечатков, идентифицирующих документ, к одному «универсальному».

Предложенный в [12] алгоритм получения нечетких отпечатков является своеобразным приложением пространственно-чувствительного хеширования (locality sensitive hashing) к проблеме поиска схожих документов. Опираясь в процессе построения сигнатуры на частоту префиксов термов документа, показана отличная производительность вкупе с хорошей точностью.

Важным свойством методов, ставящих в соответствие каждому документу единственную сигнатуру, является «бинарность» их суждений. В самом деле, документы будут считаться схожими только в том случае, если их сигнатуры совпадают. В противном случае, никакой связи между документами обнаружено не будет. При уменьшении величины порога, выше которого документы считаются дубликатами, количество документов, очевидно, увеличивается. Таким образом, невозможно сохранить высокие значения точности и полноты для всего диапазона: достигая высокой полноты, мы жертвуем точностью, и наоборот.

3. Идея исследования

Предлагается следующий метод построения сигнатуры: сигнатура документа строится на основе определенного набора статистических параметров документа, как-то: количество определенных символов в тексте (точек, пробелов, спецсимволов); общая длина текста (с исключением ряда элементов и без); количество и отношение высоко-, средне- и низкочастотных слов в документе; средняя длина предложения и слова; и т.п.

Набор параметров подбирается, исходя из соображений устойчивости к различным видам модификаций документа. Например, количество точек, запятых и заглавных букв в тексте позволяет примерно зафиксировать количество предложений в тексте; общая длина текста в символах с исключением пробелов и стоп-слов дает общую оценку объема; оценка количества и отношения разночастотных слов может служить некой заменой семантическому анализу документа и т.д.

Насколько известно автору, работ, посвященных исследованию возможности использования сигнатур, основанных непосредственно на длине и количестве различных элементов документов и производных от этих величин, не проводилось. Что касается опоры на спецсимволы и пунктуацию, то большинство алгоритмов изначально отбрасывают все подобные символы в процессе вычленения термов из документа.

Цель данной работы можно сформулировать следующим образом: исследовать возможность применения одних лишь статистических (без учета какой-либо семантики и использования глобальных коллекций) параметров текста для создания локальной сигнатуры документа.

4. Описание методов, алгоритмов и экспериментов

4.1 Метрики

Выбор метрик, как уже было показано в [2], во многом определяет полезность исследования. Данная работа преследовала цель обеспечить максимальную полноту получаемых результатов для возможности сравнения результатов с предыдущими работами в этой области. В итоге был использован описанный ниже, во многом стандартный, набор метрик.

Используемые обозначения:

- a - количество найденных пар дубликатов, совпадающих с «релевантными» парами;
- b - количество найденных пар дубликатов, не совпадающих с «релевантными» парами;
- c - количество не найденных пар дубликатов, совпадающих с «релевантными» парами;
- d - количество не найденных пар дубликатов, не совпадающих с «релевантными» парами.

Точность (Precision). Представляет собой отношение общего числа найденных релевантных пар дубликатов к общему числу найденных пар. В нашем случае она будет характеризовать способность находить только релевантные пары.

$$P = \frac{a}{a+b}$$

Полнота (Recall). Вычисляется как отношение общего числа найденных релевантных пар дубликатов к общему числу «релевантных» пар. Характеризует способность находить нужные пары.

$$R = \frac{a}{a+c}$$

Ф-мера (F1, F-measure). Гармоническое среднее для точности и полноты. Удобна в использовании в качестве общей метрики, объединяющей две предыдущие, так как и полнота и точность присутствуют в ней с одинаковым весом.

$$F = \frac{2PR}{P+R}$$

Аккуратность (Accuracy). Отношение числа правильных решений алгоритма к общему числу решений.

$$A = \frac{a+d}{m}, \text{ где } m=a+b+c+d.$$

AC1. Модифицированная версия каппы Коэна, статистической меры взаимного согласия двух оценщиков. Использовалась для оценки степени согласия (пересекаемости) результатов различных экспериментов.

$$AC1 = \frac{p(E)A}{1-p(E)},$$

где $p(E)=2P'(1-P')$ и $P' = \frac{2a+b+c}{2m}$.

4.2 Программная реализация

Для выполнения исследования компанией «Яндекс» был предоставлен набор данных «Web-страницы коллекции РОМИП». Эта коллекция используется в рамках Российского семинара по оценке методов информационного поиска (РОМИП) и включает порядка 750 тысяч страниц, содержащихся более чем в 23000 сайтов домена narod.ru.

Данные о дубликатах этого набора данных, предоставленные вместе с ним, составлялись на основе функции Левенштайна (называемой так же расстоянием редактирования). Эта функция равна минимальному количеству операций редактирования, необходимых для преобразования одного документа в другой. Были известны пары документов, коэффициент сходства которых, после удаления тэгов, составлял не менее 85% и состоящих не менее, чем из 20 слов. Соответственно, результаты экспериментов сравнивались именно с этим «идеальным» списком дубликатов.

В ряде экспериментов использовался список стоп-слов. В этом случае он генерировался из файла stopword.lst, входящего в состав свободно распространяемого компанией «Яндекс» продукта *Яndex.Server 3.8 Free Edition*[1] и состоящего из 279 часто употребимых слов русского и английского языков.

Для удаления HTML-разметки использовалась открытая Java-библиотека *HTML Parser*[2], предоставляющая данную функциональность.

В конечном итоге каждому документу ставился в соответствие 128-битный MD5-хеш, полученный из сгенерированной сигнатуры. В соответствии с [10], MD5 дает значительно меньшее число ложных срабатываний, чем 40-битный хеш Рабина, использованный в ранних исследованиях, при сохранении достаточно высокой скорости генерации.

Программная реализация описанных методов включала в себя реализацию следующих этапов:

1. Подготовка тестовых данных.
    a. Парсинг заголовков и индексов документов из предоставленного набора данных в собственную базу данных.
    b. Заполнение сведений об известных парах документов-дубликатов.
2. Набор экспериментов, для каждого из которых выполнялись:
    a. Генерация хешей для каждого документа.
        i. Удаление HTML-разметки (если требовалось).
        ii. Построение сигнатуры в соответствии с конкретным алгоритмом теста.
        iii. Хеширование.
    b. Анализ результатов эксперимента с использованием описанных выше метрик.
3. Расчет степени соответствия результатов различных экспериментов друг другу.

4.3 Эксперименты

В соответствии с идеей исследования, был разработан ряд экспериментов, исследующих возможность использования статистических параметров текста при генерации подписей документов.

Тест №1: количество спецсимволов и знаков препинания.

Подсчитывалось число вхождений в текст документа d каждого символа из набора p, включавшего в себя следующие элементы:

. , - _ : ; ! ? ( ) и символ пробела.

Документ представлялся в виде вектора размерностью 11 элементов, компонентами которого являлось число вхождений каждого символа в документ.

$$s = (c_1, c_2, \ldots, c_{11}), \text{ где } c_i = Count(d, p_i).$$

Тест №2: количество слов различных длин.

Подсчитывалось число вхождений слов различных длин в текст документа (здесь и далее под словом подразумевается непрерывная алфавитно-цифровая последовательность). Документ представлялся в виде вектора размерностью, равной длине самого длинного слова документа L. Компонентами вектора являлась строка вида: «[длина слова] – [число вхождений]».

$$s = (c_1, c_2, \ldots, c_L),$$

$$\text{где } c_i = d_i + Count(d, d_i),\ L = Max(Length(d_i)).$$

Тест №3: количество и средняя длина слов и предложений.

Подсчитывалась средняя длина терма и предложения в документе, а также общее количество тех и других. Сигнатура составлялась путем конкатенации полученных значений строкой вида: «[средняя длина слова] – [средняя длина предложения] – [число слов] – [общее число предложений]».

Были также протестированы модификации алгоритма, использующие по отдельности слова и предложения (среднюю длину и общее количество). Они показали на 20-25% худшую точность при сохранении той же полноты, что и в объединенном тесте.

Тест №4: популярные слова.

Подсчитывалось число повторений каждого слова в тексте документа. Документ разбивался на термы и представлялся в виде вектора, размерность

[1] http://company.yandex.ru/technology/products/yandex-server.xml

[2] http://htmlparser.sourceforge.net/

Таблица 1. Результаты теста №1

Порог	РОМИП	Тест	Общие	УникР	УникТ	Полнота	Точность	Ф-мера
1,00	506522	952311	458491	48031	493820	0,91	0,48	0,63
0,95	1676092	952311	831681	844411	120630	0,50	0,87	0,63
0,90	2061873	952311	871619	1190254	80692	0,42	0,92	0,58
0,85	2343691	952311	875416	1468275	76895	0,37	0,92	0,53

Таблица 2. Результаты теста №2

Порог	РОМИП	Тест	Общие	УникР	УникТ	Полнота	Точность	Ф-мера
1,00	506522	424925	298809	207713	126116	0,59	0,70	0,64
0,95	1676092	424925	403182	1272910	21743	0,24	0,95	0,38
0,90	2061873	424925	409403	1652470	15522	0,20	0,96	0,33
0,85	2343691	424925	409827	1933864	15098	0,17	0,96	0,30

Таблица 3. Результаты теста №3

Порог	РОМИП	Тест	Общие	УникР	УникТ	Полнота	Точность	Ф-мера
1,00	506522	431071	296080	210442	134991	0,58	0,69	0,63
0,95	1676092	431071	406317	1269775	24754	0,24	0,94	0,39
0,90	2061873	431071	414648	1647225	16423	0,20	0,96	0,33
0,85	2343691	431071	415163	1928528	15908	0,18	0,96	0,30

которого равна размеру словаря документа *n*. Компонентами вектора являлась строка вида: «[слово] – [число вхождений]».

$$s = (c_1,c_2,\ldots,c_n), \text{ где } c_i = d + Count(d, d_i).$$

Используемый в данном эксперименте алгоритм имеет отдаленное сходство с алгоритмом I-Match, однако имеются два существенных отличия: используются данные исключительно из обрабатываемого документа, без привлечения внешних источников данных в виде глобальной коллекции термов; а также учитывается частота терма в документе.

Были также протестированы следующие модификации алгоритма:

- обрезка 20-35% с каждой стороны отсортированного по количеству вхождений списка слов – не оказала заметного влияния (менее 1%);
- использование в качестве единиц не слов, а предложений – не оказало заметного влияния (менее 1%);
- отсортированный по числу вхождений список слов, нормализованный и округленный до первого знака после запятой – дал выигрыш в точности в размере 1-2%.

Тест №5: последовательность длин слов.

Документ разбивался на термы, после чего производилась конкатенация длин всей последовательности термов. Сигнатура строилась на основе полученного вектора.

Тест №6: агрегатный.

Совокупный тест, собирающий все предыдущие тесты в один. Сигнатура собиралась конкатенацией сигнатур тестов 1-5.

Результаты тестов приведены в таблицах 1-6. Расшифровка заголовков таблиц:

- *Порог* – пороговая величина сходства документов в соответствии с функцией Левенштайна, с которой они считаются схожими;
- *РОМИП* – число дубликатов по версии тестового набора данных;
- *Тест* – число дубликатов, полученное в результате теста;
- *Общие* – число совпадений пар дубликатов теста и РОМИП;
- *УникР* и *УникТ* – число уникальных дубликатов по версии РОМИП и теста соответственно.

Графическая интерпретация результатов тестов приведена на рисунках 1 и 2.

Исключение стоп-слов (для тех тестов, в которых это бы имело смысл) не дало заметного результата: отклонения от базовых результатов не превышали 1% (как в положительную, так и в отрицательную сторону).

Также было установлено, что ориентация в построении сигнатуры на заглавные буквы (их последовательности и количество), а также опора на общую длину текста (с удалением стоп-слов и пунктуации и без) не дает сколько-нибудь значимых и интересных результатов.

Таблица 4. Результаты теста №4

Порог	РОМИП	Тест	Общие	УникР	УникТ	Полнота	Точность	Ф-мера
1,00	506522	240642	228961	277561	11681	0,45	0,95	0,61
0,95	1676092	240642	231175	1444917	9467	0,14	0,96	0,24
0,90	2061873	240642	231591	1830282	9051	0,11	0,96	0,20
0,85	2343691	240642	231628	2112063	9014	0,10	0,96	0,18

Таблица 5. Результаты теста №5

Порог	РОМИП	Тест	Общие	УникР	УникТ	Полнота	Точность	Ф-мера
1,00	506522	414964	298584	207938	116380	0,59	0,72	0,65
0,95	1676092	414964	396036	1280056	18928	0,24	0,95	0,38
0,90	2061873	414964	400663	1661210	14301	0,19	0,97	0,32
0,85	2343691	414964	400917	1942774	14047	0,17	0,97	0,29

Таблица 6. Результаты теста №6

Порог	РОМИП	Тест	Общие	УникР	УникТ	Полнота	Точность	Ф-мера
1,00	506522	235619	226233	280289	9386	0,45	0,96	0,61
0,95	1676092	235619	226698	1449394	8921	0,14	0,96	0,24
0,90	2061873	235619	227062	1834811	8557	0,11	0,96	0,20
0,85	2343691	235619	227078	2116613	8541	0,10	0,96	0,18

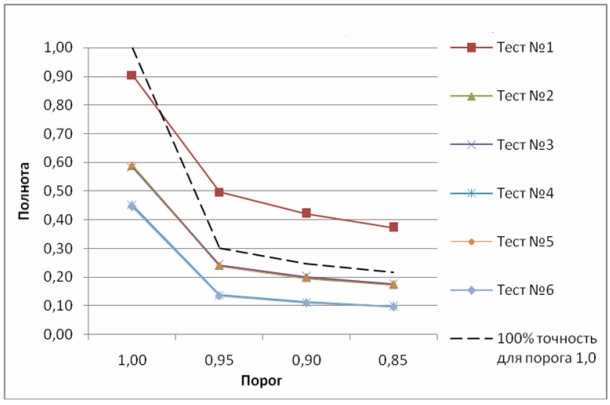

Рис. 1. График полноты для различных тестов

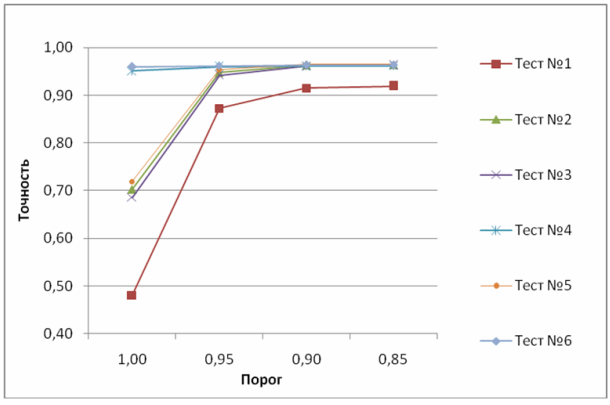

Рис. 2. График точности для различных тестов

Степень согласия результатов всех проведенных тестов приведена в таблице 7. Как видно из приведенных данных, тесты можно разбить на три группы, в пределах которых результаты весьма близки друг другу:

1. Тест №1 – опирающийся на спецсимволы.
2. Тесты №№2, 3, 5 – опирающиеся на длину элементов текста.
3. Тесты №№4, 6 – опирающиеся на наборы популярных слов.

Таблица 7. Степень согласия результатов различных тестов (по АС1)

№ теста	6	5	4	3	2
1	0,20	0,27	0,20	0,29	0,28
2	0,38	0,97	0,40	0,95	
3	0,38	0,93	0,37		
4	0,97	0,41			
5	0,40				

Последний, агрегатный тест, оказался весьма близок по результатам к тесту №4. Несмотря на стабильно показываемую им высокую точность, полнота оставляла желать лучшего.

В противоположность ему, тест на основе спецсимволов показал много «лишних» пар дубликатов при пороге, равном 1, которые затем заняли свое место при уменьшении порога до 0,9. Подобный «запас» показаний может оказаться полезным в ряде задач. Ф-мера для всех порогов данного теста оказалась максимальной среди всего набора тестов (за одним исключением).

Тесты из второй группы показали промежуточные результаты.

Кроме того, агрегатный тест №6 был детально проанализирован для различных диапазонов длин документов (см. таблицу 8).

Таблица 8. Результаты теста №6 для различных диапазонов длин документов (по количеству слов)

Диапазон	Докуме-нтов	Порог	Полн	Точн	Ф-мера
20-50	67718	1,00	0,82	0,98	0,90
20-50	67718	0,95	0,35	0,98	0,52
20-50	67718	0,90	0,25	0,98	0,40
20-50	67718	0,85	0,19	0,98	0,32
51-500	178945	1,00	0,35	0,96	0,51
51-500	178945	0,95	0,10	0,96	0,18
51-500	178945	0,90	0,08	0,97	0,15
51-500	178945	0,85	0,07	0,97	0,14
501-5000	55780	1,00	0,40	0,80	0,53
501-5000	55780	0,95	0,22	0,80	0,53
501-5000	55780	0,90	0,18	0,80	0,29
501-5000	55780	0,85	0,16	0,80	0,27
5001-99000	3001	1,00	0,37	0,94	0,52
5001-99000	3001	0,95	0,22	0,96	0,36
5001-99000	3001	0,90	0,20	0,96	0,34
5001-99000	3001	0,85	0,20	0,96	0,33

Из результатов теста видно, что наилучшие результаты агрегатный тест показал на коротких документах (20-50 слов). Этот результат можно считать достаточно неплохим, учитывая известную неэффективность алгоритма супершинглирования для коротких документов. Падение точности на больших документах можно объяснить увеличением «хрупкости» сигнатуры, включающей в себя все большее количество информации.

Таблица 9. Время работы тестов

№ теста	Время работы (секунд)
1	928
2	1214
3	1132
4	1119
5	1252
6	1905

Эксперименты проводились на компьютере с процессором Intel Core Duo E4300 и 1Гб оперативной памяти. Время, затраченное на генерацию сигнатур в рабочей реализации алгоритмов, приводится в таблице 9. Несмотря на возможные погрешности реализации (значительную часть времени во многих тестах, например, отобрало разбиение текстов с помощью *HTML Parser*), эти данные позволяют дать оценку вычислительной сложности алгоритмов. Как понятно из описаний тестов, с увеличением числа документов время выполнения всегда будет увеличиваться линейно – *O(n)*. Простота и быстродействие многих алгоритмов является их выигрышной чертой в случае применения к большим объемам документов.

5. Заключение

Предложен метод построения локальной сигнатуры документа на основании статистических параметров его содержимого. Проведен ряд экспериментов, использующих ряд подобных параметров. Показана возможность использования данного подхода в условиях больших объемов документов.

Общим местом всех полученных результатов является резкое падение полноты с уменьшением порога схожести документов, и относительная стабильность в точности. Как указывалось выше, такой эффект неизбежен при использовании единой сигнатуры документа. Устойчивость в плане точности, однако, дает возможность различных модификаций алгоритмов в пользу полноты.

Для дальнейших исследований представляет интерес апробация теста, основанного на спецсимволах, к различным проблемам. Необходимо провести сравнительный анализ эффективности нахождения дубликатов в рамках одного сайта и за его пределами. Кроме того, возникает потребность в создании тестовой коллекции, содержащей наборы «идеальных» пар дубликатов с разных точек зрения, что позволит проводить исследования эффективности алгоритмов по разным критериям.

6. Литература

[1] Bernstein Y. A Scalable System for Identifying Co-derivative Documents / Y. Bernstein, J. Zobel // String Processing and Information Retrieval. – 2004. – P. 55-67.

[2] Bernstein Y. The Case of the Duplicate Documents: Measurement, Search, and Science / Y. Bernstein, J. Zobel // Proceedings of the Eighth Asia Pacific Web Conference. – 2006. – P. 26-39.

[3] Broder A. Syntactic Clustering of the Web / A. Broder, S. Glassman, M. Manasse, G. Zweig //. Computer Networks and ISDN Systems. – 1997. – Vol. 29. – Issue 8-13. – P. 1157 - 1166.

[4] Charikar M. Similarity Estimation Techniques from Rounding Algorithms / M. Charikar // STOC '02: Proceedings of the thiry-fourth annual ACM symposium on Theory of computing. – 2002. – P. 380-388.

[5] Chowdhury A. Collection statistics for fast duplicate document detection / A. Chowdhury, O. Frieder, D.A.Grossman, M.C. McCabe // ACM Transactions on Information Systems. – 2002. – Vol. 20. – Issue 2. – P. 171-191.

[6] Chowdhury A. Duplicate Data Detection [Electronic resource]. – 2007. – Mode of access: http://ir.iit.edu/~abdur/Research/Duplicate.html

[7] Henzinger M. Finding Near-Duplicate Web Pages: a Large-Scale Evaluation of Algorithms / M. Henzinger // SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval. – 2006. – P. 284-291.

[8] Hoad T. C. Methods for Identifying Versioned and Plagiarised Documents / T. C. Hoad, J. Zobel // Journal of the American Society for Information Science and Technology. – 2003. – Vol. 54. – Issue 3. – P. 203-215.

[9] Kolcz A. Improved Robustness of Signature-Based Near-Replica Detection via Lexicon Randomization / A. Kolcz, A. Chowdhury, J. Alspector // KDD '04: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining. – 2004. – P. 605-610.

[10] Ma W.Y. A Systematic Study of Parameter Correlations in Large Scale Duplicate Document Detection / S. Ye, J.R. Wen, W.Y. Ma // Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD). – 2006. – P. 275-284.

[11] Potthast M. Applying Hash-based Indexing in Text-based Information Retrieval / B. Stein, M. Potthast // 7th Dutch-Belgian Information Retrieval Workshop (DIR 2007). – 2007. – P. 29-35.

[12] Stein B. Fuzzy-Fingerprints for Text-Based Information Retrieval / B. Stein // Journal of Universal Computer Science. – 2005. – P. 572-579.

Use of statistical parameters in similar documents detection

Kosinov D.

Many complex tasks rely on the algorithms that search for similar elements in large corpora, which explains the need for effective methods of detecting documents that resemble other documents in the collection. A new method of local document signature creation on the sole basis of statistical parameters of its content without resorting to global collections is proposed. The set of parameters is formed on the basis of tolerance to different types of modifications. A number of experiments using some of these parameters are conducted. The possibility of use of this approach for large corpora processing is shown.