

ИЗУЧЕНИЕ СТАТИСТИКИ ВСТРЕЧАЕМОСТИ ТЕРМИНОВ И ПАР ТЕРМИНОВ В ТЕКСТАХ ДЛЯ ВЫБОРА МЕТОДОВ СЖАТИЯ ИНВЕРТИРОВАННОГО ФАЙЛА

Губин Максим Вадимович, ЗАО «Информационная компания “Кодекс”»,
194021, Санкт-Петербург, Политехническая ул., 22
Max@gubin.spb.ru

SELECTION OF AN INVERTED FILE COMPRESSION ALGORITHM USING STATISTICS OF OCCURRENCE OF WORDS AND WORD PAIRS IN COLLECTION

Max Gubin, "Information Company "Kodeks"
Politechnicheskaya 22, St.Petersburg, 194021, Russia
max@gubin.spb.ru

Inverted files are the most popular index structure for information retrieval. In this paper we study statistics of words and word pair occurrence in a collection of relatively large documents in order to select the best method of the index structure compression. Our result shows, that in our case the inverted file contains for the most part very short post lists. Classical post list compression technologies show bad performance for the data and we propose method of post list combining for compression.

ВВЕДЕНИЕ

В настоящее время основной индексной структурой для поиска по тексту являются инвертированные файлы [1]. Они сочетают высокую скорость обработки запросов и относительно небольшие размеры, которые достигаются использованием различных методов сжатия. Существует большое количество исследований характеристик инвертированных файлов и методов их сжатия [2]. Практически все они исследуют случаи специальных тестовых коллекций или коллекций страниц Интернета. Такие коллекции характеризуются большим количеством относительно маленьких документов. Электронные библиотеки, информационно-справочные системы, как правило, имеют коллекции с меньшим количеством документов, но документы относительно большие. Исследований коллекций с подобными характеристиками не удалось обнаружить в литературе, поэтому было интересно проверить, применимы ли стандартные рекомендации для коллекций малых документов к электронным библиотекам.

Анализ пользовательских запросов показывает [3], что большинство пользователей используют очень короткие запросы, в среднем равные 2 словам. Это наводит на мысль, что удобно строить инвертированный файл

не только по отдельным словам, но и их сочетаниям (парам) слов. Такой индекс не только позволяет быстрее обрабатывать запросы по фразам, но и обеспечивает дополнительный сервис для пользователя: можно формировать при вводе запроса подсказки наиболее вероятных следующих слов, использовать информацию о парах для автоматического расширения запроса, улучшить качество взвешивания результатов поиска и т.д. Существует ряд исследований подобных способов индексирования, например [4, 5], но опять же при этом исследовались коллекции относительно малых документов.

В данной статье приводятся результаты исследований, которые были проведены для выбора метода сжатия индексированных файлов в системе «Кодекс».

ХАРАКТЕРИСТИКИ КОЛЛЕКЦИИ

Использовались 3 тестовых коллекции, выбранные так, чтобы они были различны по своим свойствам и достаточно представительны. В то же время коллекции достаточно небольшие, чтобы исследования можно было проводить за малое время на обычных персональных компьютерах.

1. Русская коллекция. Действующее российское законодательство. Общее количество документов – 8134. Средний размер документа 1500 слов. Самый короткий документ 30 слов, самый длинный 176442 слова.

2. Английская коллекция. Переводы российского законодательства. Общее количество документов 1435. Средний размер документа 1626 слов, самый короткий документ 30 слов, самый длинный 65784.

3. Русская коллекция 2. Строительные ГОСТы и СНИПы с комментариями. Содержит большие документы со сложным форматированием и большим количеством таблиц. Всего 365 документов. Средний размер документа – 9751 слово, минимальный 30 слов, максимальный 98247 слов.

МЕТОДИКА ИССЛЕДОВАНИЯ СТАТИСТИКИ СЛОВ И ПАР СЛОВ

Каждая коллекция сканировалась специально разработанной программой, которая собирала статистику и формировала ее в виде текстовых файлов, которые затем обрабатывались в MS Excel. Тексты коллекции разбивались на слова. Словом считалась любая последовательность символов, разделенная пробелами, табуляцией, переводом строки или границами абзаца. Для таблиц каждая ячейка считалась абзацем. Если к слову по сторонам присоединялись знаки препинания, то они удалялись. Слова приводились к верхнему регистру, т.е. регистр не учитывался. Для каждого слова выделялась «нормальная форма» с учетом морфологии, основываясь на словаре из 50 000 основ, используемая в «Кодексе». Если в словаре данного слова не было, то оно само и рассматривалось как нормальная форма.

При выделении пары слов, основной задачей было ее сформировать так, чтобы она максимально совпадала с интуитивным понятием пары у пользователя, который производит поиск, поэтому использовался следующий алгоритм выделения пар:

Текст сканировался пословно, шумовые слова пропускались и не рассматривались. Запоминалось N предыдущих слов, точнее их нормальных форм. Рассматривались случаи, когда $N = 1$, $N = 2$ (эквивалентно наиболее часто задаваемому расстоянию для поиска устойчивых фраз пользователями) и $N = 3$, $N = 4$. При обработке каждого слова проверялось, есть ли предыдущие, и если они есть, то формировались пара из нормальной формы текущего и предыдущих слов. Массив предыдущих сдвигался, N -ое слово удалялось, а нормальная форма текущего ставилась на 1 позицию. При переходе через границу абзаца предыдущее слово сбрасывалось, т.е. пары не пересекали границу абзаца, т.к. считалось, что вероятность того, что они связаны мала.

Далее слова в парах упорядочивались по алфавиту, т.к. считалось, что порядок слов в парах не имеет большого значения.

Проводились исследования как для случая, когда пары отличающиеся только морфологическими формами одного слова считались совпадающими и когда морфология не учитывалась. Как оказалось, учет морфологии очень мало влияет на количество пар – их количество уменьшалось менее чем на 10%, поэтому все дальнейшие результаты приводятся для случая без учета морфологии.

В результате для исследуемых коллекций получились следующие характеристики документов:

	Русская коллекция	Английская коллекция	Русская коллекция 2
Среднее число	416.5	357.0	2432.0
Максимальное число	16641	4873	13880
Минимальное число	14	21	15

Как видно, характеристики коллекции сильно отличаются от характеристик коллекций, обычно используемых в подобных исследованиях [5]. Уникальные пары с $N = 1$ в документе:

	Русская коллекция	Английская коллекция	Русская коллекция 2
Среднее число	824.5	902.8	6142.8
Максимальное число	97477	29352	55806
Минимальное число	14	22	16

Как видно из этой таблицы, в пределах одного документа устойчивые пары слов встречаются достаточно часто. Это видно из того, что в среднем количество уникальных пар всего в 2-3 раза больше количества слов и в 1.5 раз меньше, чем всего слов в документе.

РАЗМЕРЫ СЛОВАРЕЙ

Инвертированный файл состоит из двух частей – словаря содержащего слова и пост листов, указывающих на вхождения слов в документы. В случае индексирования по парам словарь должен содержать не только слова, но и пары слов. Для того, чтобы оценить размер словаря, анализировался рост количества слов и пар в зависимости от количества проиндексированных документов.

Как известно, рост количества слов от количества документов растет согласно закону Хипса (Heaps) [6]. Как видно на рисунках 1-3, в исследуемых коллекциях коэффициент в законе Хипса приблизительно равен единице.

В пределах коллекции пары также повторяются, это видно из того, что количество уникальных пар явно растет медленнее, чем квадрат количества слов. Но в коллекции встречаемость пар значительно реже, чем встречаемость той же пары в документе – пар на порядок больше, чем слов. Это говорит о том, что пары должны давать значительно лучшую избирательность при поиске, чем слова.

Кроме этого, исследовалось количество пар от параметра N (расстояния на котором берутся пары). Оказалось, что для всех коллекций наблюдается практически линейная зависимость от роста N.

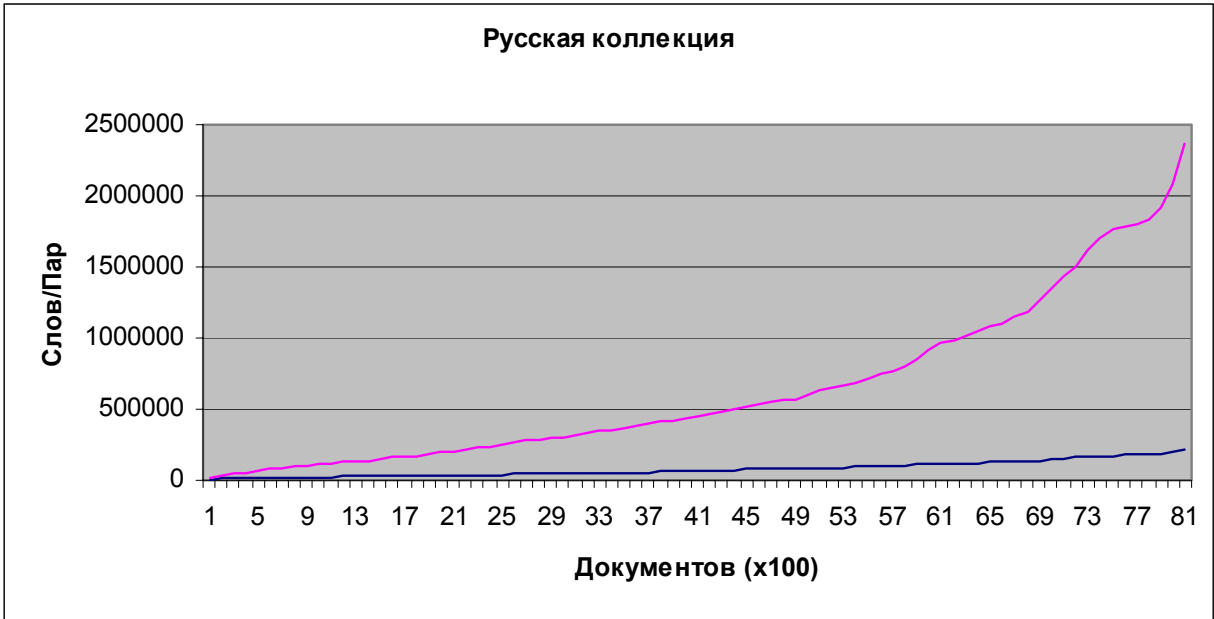


Рисунок 1. Рост числа слов и пар. Русская коллекция 1

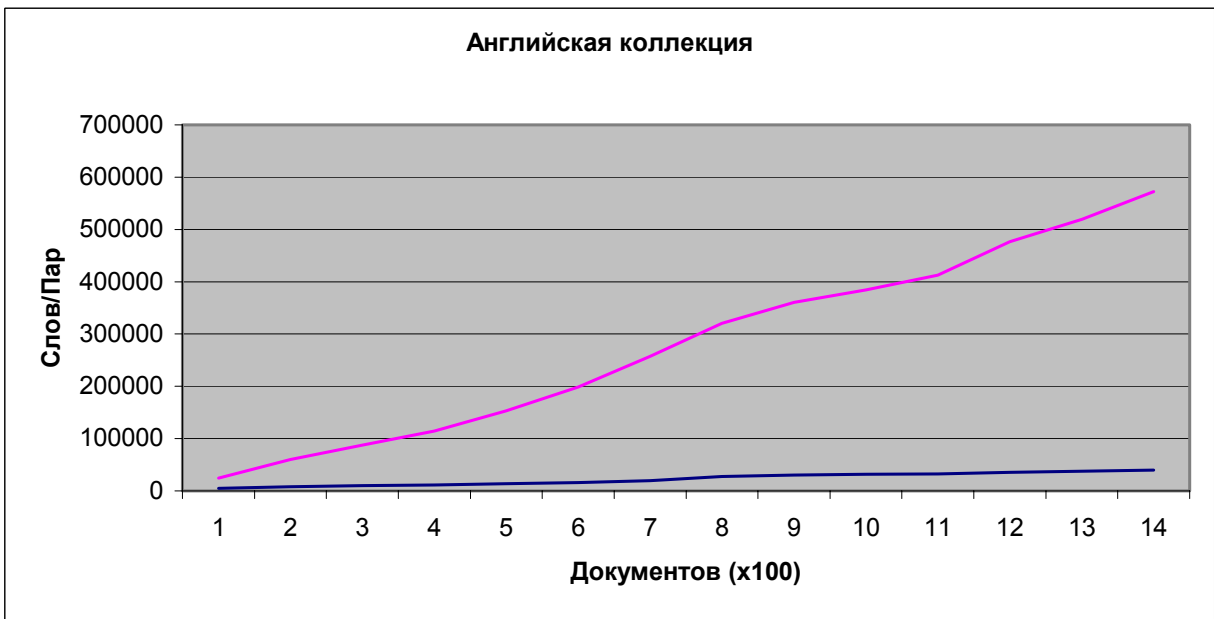


Рисунок 2. Рост числа слов и пар. Английская коллекция

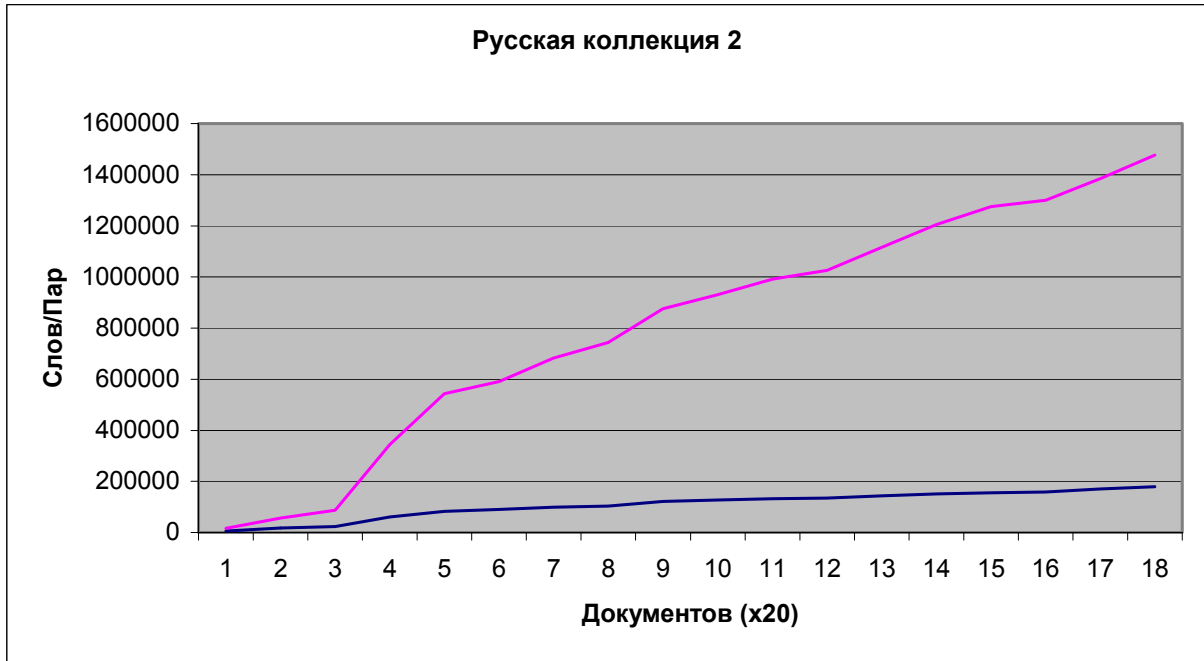


Рисунок 3. Рост числа слов и пар. Русская коллекция 2

ИССЛЕДОВАНИЕ ХАРАКТЕРИСТИК ПОСТ ЛИСТОВ

Для каждого слова и пары строились пост листы и исследовалась их длина. Данные исследования проводились для двух русских коллекций. Обе коллекции показали очень похожее поведение. В основном встречаются короткие листы. На рисунке 4 представлена зависимость количества листов от длины для коллекции 1 на интервале длин от 1 до 500. Далее график до максимальной длины (8134) практически «лежит» в 0.

Из рисунков 4 и 5 видно, что графики отличаются только масштабом, характер поведения не меняется.

Однако, количество пост листов не является определяющим, важнее сколько они занимают места, ведь возможно что достаточно редкие, но длинные пост листы занимают больше места. Именно так обычно происходит в классических коллекциях коротких документов. Поэтому на рисунках 6 и 7 приведено общее количество памяти, занимаемое не сжатыми пост листами.

Видно, что некоторый «подъем» в начальной части графика есть, но характер не поменялся.

В случае, если пост листы кроме идентификаторов документов содержат также и положения слов в документах, характер зависимости не изменяется, хотя такие пост листы получаются в среднем в 5-10 раз длиннее листов без положений. Для уменьшения размера статьи графики для таких листов не приводятся.

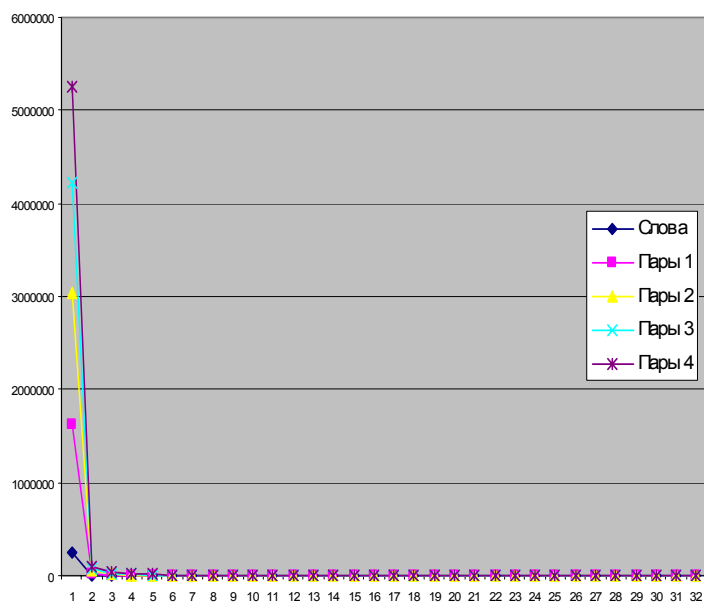


Рисунок 4. Количество пост листов в зависимости от длины. Коллекция 1

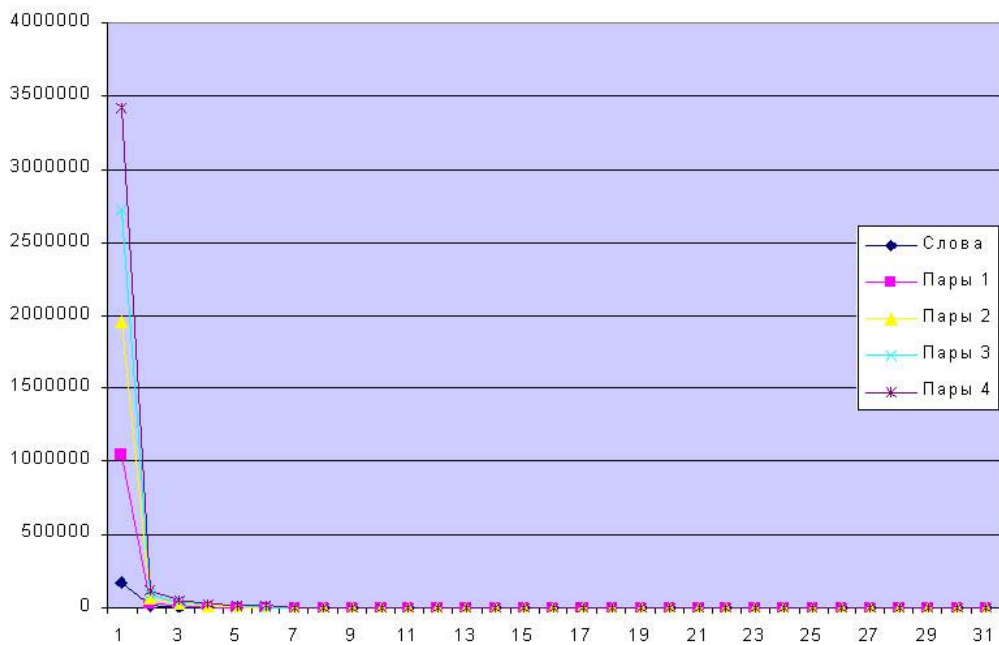


Рисунок 5. Количество пост листов в зависимости от длины. Коллекция 2

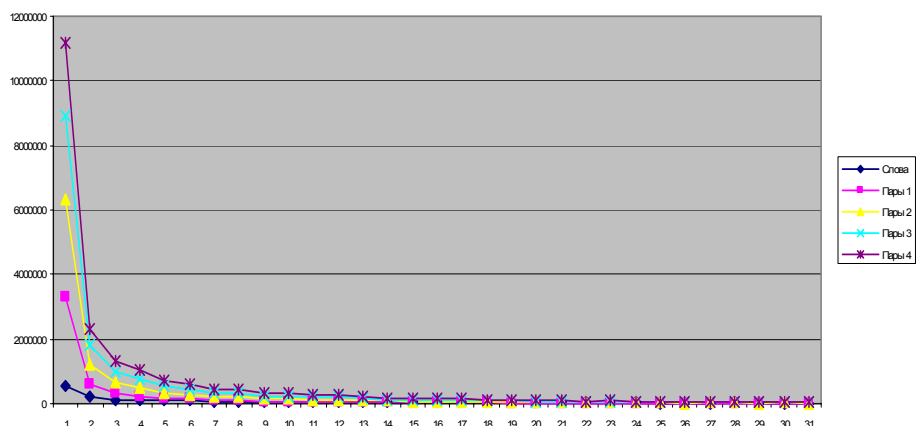


Рисунок 6. Объем, занимаемый не сжатыми пост листами в зависимости от их длины. Коллекция 1

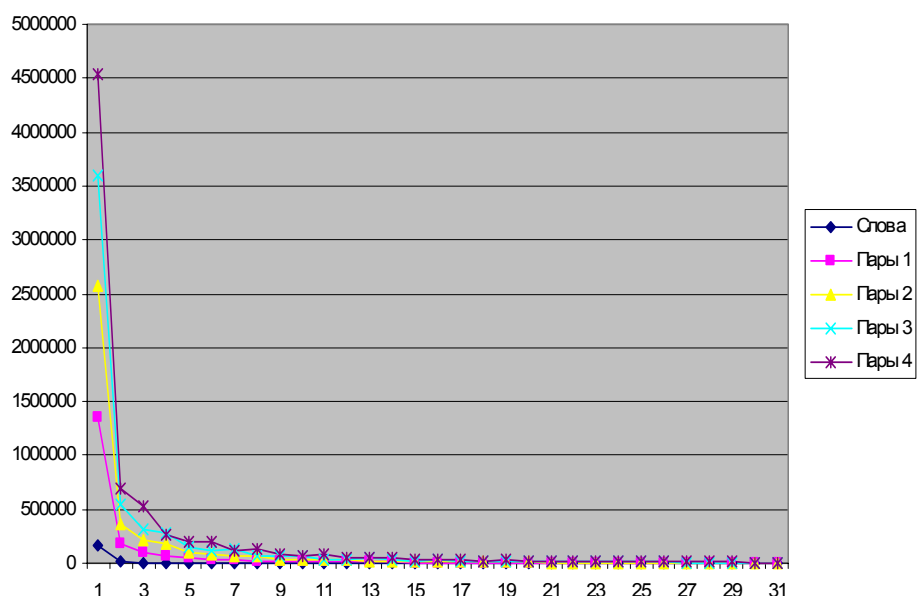


Рисунок 7. Объем, занимаемый не сжатыми пост листами в зависимости от их длины. Коллекция 2

ВЫБОР АЛГОРИТМА СЖАТИЯ

Как видно из предыдущих результатов, основной объем в инвертированном файле при индексировании по словам и парам в коллекциях относительно больших документов занимают короткие пост листы. В то же время, в литературе основное внимание уделяется сжатию длинных пост листов. Классическим вариантом [7] является использование кодирования

разности идентификаторов кодами переменной длины. Очевидно, что в исследуемом случае это не дает заметного выигрыша. Например, для российской коллекции 1 длины пост листов, не сжатых и сжатых 2 методами, приведены на рисунке 8.

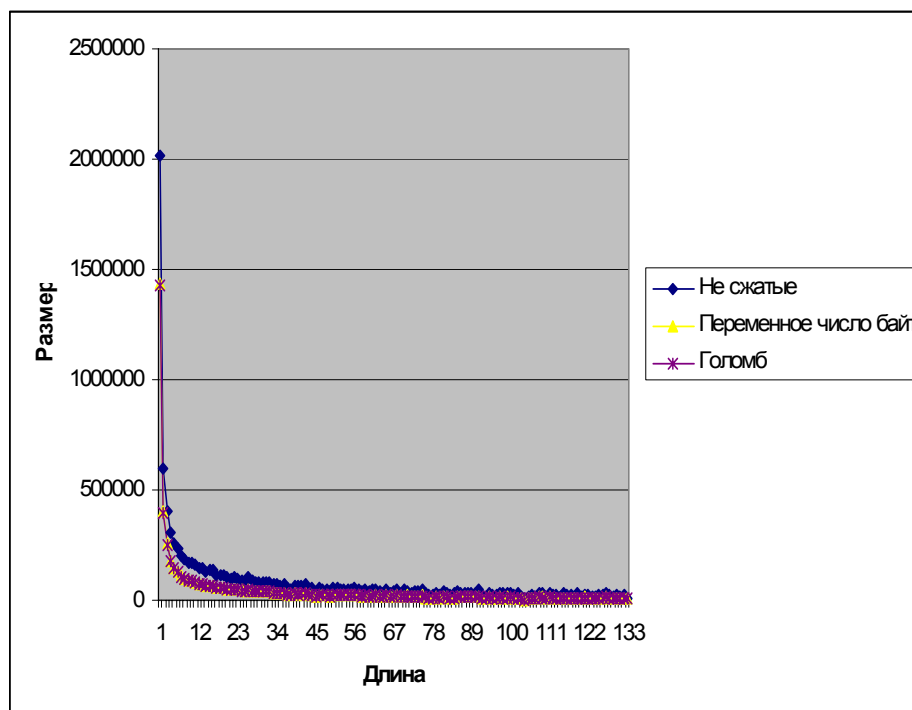


Рисунок 8. Объем пост листов сжатых разными методами

Как видно, короткие пост листы практически не сжимаются данными алгоритмами. Это очевидно, т.к. они последовательности слишком короткие, чтобы они начали работать. Идентификаторы документов имеют слишком большой разброс, дельты их относительно большие. Кроме этого, коды типа Голомба дают не очень хороший результат, скорее всего из-за того, что оценка коэффициента как 0.6 от среднего [8] оказывается слишком далекой от медианы при реальном распределении. Возможно, результат получился бы лучше при более точной оценке параметров кода Голомба, но это сделало бы данный алгоритм неприемлемо ресурсоемким.

Наиболее простым вариантом является использовать для сжатия данных коротких пост листов какой-либо стандартный алгоритм сжатия общего назначения. Пост листы сжимаются сначала классическим способом, далее результирующие данные сжимаются во втором проходе блоками. Такой подход иногда применяется разработчиками поисковых систем, например [9]. Поэтому была сделана попытка сжимать данные пост листы, объединенные в блоки по 64 Кб с помощью алгоритма deflate стандартной библиотеки zlib. Однако на российской коллекции это дало только 17% уменьшения пространства для кодирования Голомба и 20% для переменного числа байт. Данный результат очевиден, т.к. этот алгоритм сжатия

основан на LZSS и ищет повторяющиеся последовательности байт, которых в таких данных крайне мало.

Очевидно, что сжимая отдельно пост листы, получить заметное улучшение коэффициента сжатия не удастся. Поэтому был исследован алгоритм, при котором производится следующее преобразование пост листов:

1. Длинные пост листы сжимались обычным способом с применением кодирования переменным числом байт.

2. Короткие пост листы объединялись в «корзины» по 32 пост листа. Для каждой «корзины» сохранялся объединенный массив идентификаторов документов, сжатый, как и длинные листы. Далее сохранялись представленные кодом Хаффмана номера слов, соответствующие позиции в пост листе. Если одному идентификатору документа соответствовали несколько слов, то использовался специальный esc символ.

Выбор числа 32 был получен экспериментально, меньшее число давало худшее сжатие, т.к. объединенный пост лист оказывался не достаточно длинный, большее приводило к слишком длинной части, содержащей номера слов. Возможно, для других коллекций оптимальным будет другое число, выбор данного параметра требует дополнительных исследований. Проводились также эксперименты, где была сделана попытка ограничить не число слов, а размер объединенного пост листа, но оказалось, что при этом коэффициент сжатия резко ухудшается из-за того, что появляются «корзины» с очень большим числом объединенных слов, что приводило к длинной части «корзины» с большим количеством esc символов.

Суммарный размер пост листов при различных алгоритмах сжатия приведен в следующей таблице:

Не сжатый	Голомб	Переменное число байт	«корзины» для листов короче 256 вхождений	«корзины» для листов короче 512 вхождений	«корзины» для листов короче 1024 вхождений
14 479 428	7 073 495	6 898 799	4 038 088	3 861 381	3 800 951

Как видно, подобный алгоритм позволяет значительно улучшить коэффициент сжатия. Специальных измерений изменения быстродействия не проводилось, но т.к. пост листы в любом случае хранятся в блоках на диске и основное время при обработке коротких листов – дисковые операции, то ухудшения быстродействия не должно быть.

При проведении экспериментов объединялись пост листы просто слов идущих подряд по алфавиту. Возможно можно достичь значительно большего коэффициента сжатия, используя более сложные алгоритмы формирования «корзин», например, отбирая слова таким образом, чтобы улучшить сжатие объединенного пост листа.

Как видно из собранной статистики пост листов для пар слов, они также имеют короткие пост листы, похожие на пост листы редких слов. Поэтому для пост листов пар нужно применять такие же алгоритмы сжатия, как и для коротких пост листов слов.

ВЫВОДЫ

Использование индекса по парам слов увеличивает размер словаря в 5-10 раз. При этом пары имеют в коллекции характеристики распределения, подобные характеристикам редких слов. Поэтому для хранения пост листов пар и редких слов должны применяться одинаковые алгоритмы.

В инвертированных файлах, построенных по коллекциям относительно больших документов, основной объем занимают короткие пост листы.

Из-за того, что пост листы очень короткие, алгоритмы, основанные на сжатии каждого пост листа отдельно, не могут дать необходимого коэффициента сжатия. В таком случае необходимо использовать алгоритмы, обрабатывающие одновременно несколько пост листов. Предложенный алгоритм совместного сжатия коротких пост листов позволяет уменьшить объем памяти, требуемый для пост листов почти в два раза.

ПЕРСПЕКТИВЫ ДАЛЬНЕЙШЕЙ РАБОТЫ

Современные коммерческие системы кроме словаря и пост листов хранят в инвертированных файлах дополнительную информацию – веса документов, флаги титульности и т.д. Эта информация позволяет значительно улучшить качество взвешивания результатов поиска. Для сохранения в индексе эту информацию так же следует сжимать. Поэтому в дальнейшем планируется провести исследования для выбора алгоритмов сжатия этой дополнительной информации.

Далее разработанные алгоритмы предполагается использовать в реальных поисковых системах.

Литература

1. Managing Gigabytes: Compressing and Indexing Documents and Images by Ian H. Witten, Alistair Moffat, and Timothy C. Bell, Morgan Kaufmann Publishing, San Francisco
2. "Compression of inverted indexes for fast query evaluation", F. Scholer, H.E. Williams, J. Yiannis, and J. Zobel, Proceedings of the ACM-SIGIR Conference on Research and Development in Information Retrieval, Tampere, Finland, August 2002.

3. Dietmar Wolfram A Query-Level Examination of End User Searching Behaviour on the Excite Search Engine
4. "What's next? Index structures for efficient phrase querying", H. Williams, J. Zobel, and P. Anderson, Proceedings of the Australasian Database Conference, Auckland, New Zealand, January 1999, pp. 141-152.
5. "Compaction techniques for nextword indexes", D. Bahle, H.E. Williams, and J. Zobel, Proceedings of the String Processing and Information Retrieval Symposium (SPIRE), Chile, November, 2001.
6. Alexander Gelbukh, Grigori Sidorov. Zipf and Heaps Laws' Coefficients Depend on Language. Proc. CICLing-2001, Conference on Intelligent Text Processing and Computational Linguistics, February 18–24, 2001, Mexico City.
7. "Compression Integers for Fast File Access" H.Williams J.Zobel
8. Glen G. Langdon, Jr. Data Compression 1999
9. Поисковая система "Turtle". Физиология и анатомия. Д.В. Крюков, Stack Technologies Ltd.