Яндекс на РОМИП-2004. Некоторые аспекты полнотекстового поиска и ранжирования в Яндексе.

© Илья Сегалович, Михаил Маслов, ООО "Яндекс", {iseg,maslov}@yandex-team.ru

Аннотация

Описаны некоторые детали реализации полнотекстового поиска и ранжирования в Яндексе: особенности архитектуры выполнения запроса; параметризация поиска по кворуму; некоторые факторы и функции вычисления релевантности. Обсуждается результат Яндекса на РОМИП-2004.

1. Введение

В настоящей работе мы опишем несколько подробней упомянутые в [1] техники, применяемые в поиске Яндекса. В частности, коснемся его архитектуры: устройства индекса и исполнения запроса; решений, вытекающих из требований к производительности.

2. Архитектура

Полнотекстовый индекс Яндекса устроен по традиционной для IR схеме и состоит из блочно организованного файла ключей («keyfile») и файла инвертированных списков словопозиций («invfile»), упакованных Variable Byte Coding [2], по оригинальной схеме Яндекса [3], оптимизированной для архитектуры Intel. Применяется также субиндекс, подобный [4], ускоряющий операции пересечения списков. Основной поисковый оператор Яндекса — «многоместный оператор AND» с неявно назначенными ограничениями контекста между соседними словами запроса. Выполнение этого оператора происходит по схеме “document ordered processing” [5], так как число слов в запросах к поисковой системе обычно не велико. Принципиальной особенностью Яндекса является оперирование только позициями слов, удовлетворяющих ограничениям контекста. Это позволяет резко сократить число операций над документами.

3. Препроцессинг запроса

Хотя в РОМИП использовался стандартный, «отдельно стоящий», полнотекстовый индекс, стоит упомянуть о процедуре вычисления неявных контекстных ограничений, применяемой в распределенной версии поиска Яндекса. В этом случае на серверах «переднего края» [6] производится синтаксический разбор запроса на основе ATN-грамматики [7], адаптированной к свободному порядку слов русского языка. С учетом рваного «телеграфного» стиля в естественно-языковых фрагментах запросов выявляются несколько видов синтаксической связей (притяжание, перечисление, зависимости цели и места, счетные конструкции и др.) и устанавливаются эмпирически подобранные контекстные ограничения.

Глобальная для всех коллекций статистика слов используется как для «выравнивания» ранжирования между коллекциями [6], так и для корректировки контекстных ограничений в пост-синтаксической фазе.

«Синтаксически-статистическое» препроцессирование в заданиях РОМИП-2004 не использовалось, основным ограничением контекста было условие "слова должны быть в одном документе".

4. "Фильтрация" по кворуму

Имея на входе многоместного оператора треугольную матрицу контекстных ограничений между словами запроса, Яндекс осуществляет процесс нахождения всех пассажей в документах, удовлетворяющих этим ограничениям, с учетом оператора нечеткого поиска с неявно назначенным коэффициентом «мягкости» [8]. Коэффициент мягкости (число от 0 до 100) задается при помощи следующего синтаксиса:

(несколько слов с контекстными операторами)//МЯГКОСТЬ

Необходимость введения нечеткого поиска обусловлена потребностью в адекватной обработке запросов с большим количеством поисковых терминов. Источниками запросов с большим количеством поисковых терминов являются:

• естественно-языковый поиск — запрос в свободной форме

• поиск по документу-образцу (образцам)

Оператор AND сильно сужает область поиска с каждым новым термином. Применение AND к запросам с большим количеством терминов (более 5) приводит, как правило, к пустому списку найденных документов. Оператор OR, наоборот, расширяет область поиска с каждым новым термином. Применение OR к запросам с большим количеством терминов (более 5) приводит к длинному списку найденных документов. По этой причине: а) неоправданно расходуются ресурсы компьютера, б) длинный список найденных документов труднее адекватно ранжировать.

Идея кворума в поиске не нова, ее аналогом в процедуре фильтрации релевантных пассажей можно считать принцип «weighted coordination match» [9], при котором «найденными» считаются все полные пассажи, а также все неполные, сумма весов слов которых превосходит необходимый кворум.

Функция, определяющая долю веса, которая необходима для выполнения кворума, подобрана с помощью метода «естественной параметризации» [10] с целью удовлетворения нескольких требований:

• при плавном изменении параметра «мягкости» функционал количества найденных документов ND(Softness) должен меняться плавно, лучше всего - линейно

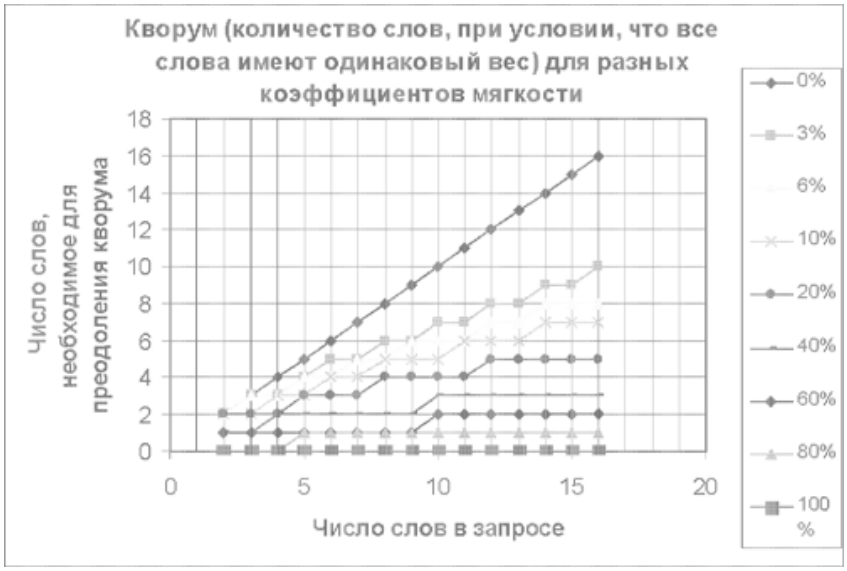
• при Softness=50 число найденных документов должно быть примерно средним геометрическим чисел найденных документов при поиске всех возможных неполных пассажей

и имеет вид:

$$QuorumWeight = (1 - Softness)^{\frac{1}{\sqrt{QL-1}}}$$

где Softness соответствует величина от 0 до 1, а QL — длина запроса в словах.

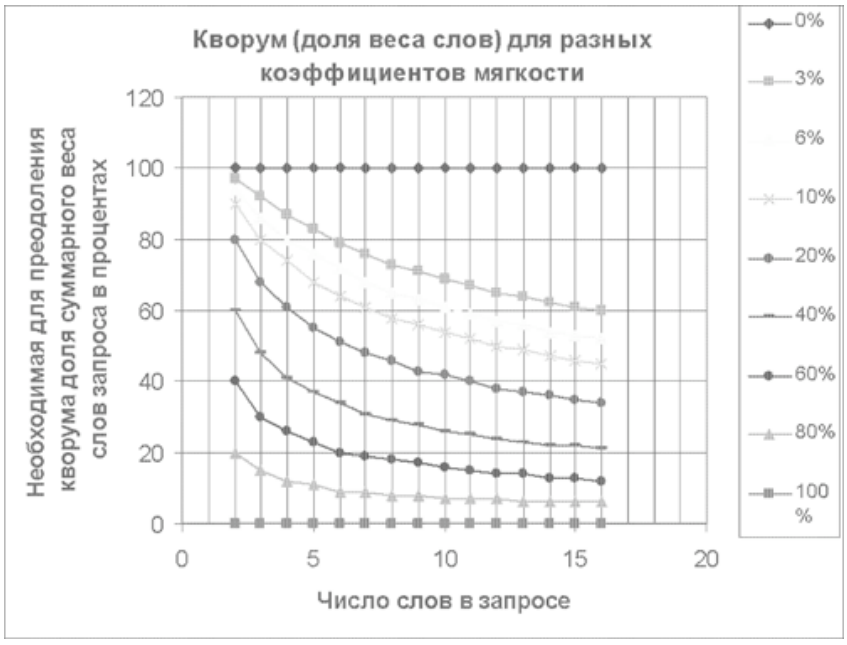
Рисунок 1. Кворум по весу



На этой иллюстрации изображено падение доли, необходимой для преодоления кворума, в зависимости от числа слов в запросе. На второй иллюстрации — кворум по количеству слов (при условии, что все слова имеют РАВНЫЙ ВЕС), необходимых для преодоления кворума, в зависимости от числа слов в запросе и коэффициента мягкости.

В частности, при равных по весу словах запроса и коэффициенте мягкости 0.06 (того, что использовался при выполнении заданий РОМИП), в пятисловном запросе достаточно 4-х слов (или 76% веса), а в 16-словном всего лишь 8 слов (или 52% веса) для преодоления кворума.

Рисунок 2. Кворум по числу слов



Формула для вычисления веса слова при голосовании по кворуму отличается от формулы, используемой при

ранжировании. Если при ранжировании Яндекс использует классический для IR логарифм обратной частоты, то при вычислении суммы голосов в кворуме применяется степенная функция с показателем между квадратным и кубическим корнем. Отличия состоят в том, что «вариант с корнем» больше ориентирован на учет "тяжелых", "редких", "новых" слов, пусть и без полного набора соседей, тогда как логарифм тяготеет к максимальному возможному количеству слов в пассаже независимо от их тяжести.

5. Ранжирование

После того, как все пассажи документа, прошедшие фильтрацию по кворуму, определены, наступает этап ранжирования, то есть вычисление веса документа. Здесь следует отметить два принципиальных отличия алгоритма Яндекса от всего, что было до сих пор описано в соответствующей литературе.

Внутри-документная частота по релевантным пассажам

Формула расчета веса слова по отношению к документу («контрастности») в Яндексе использует внутри-документные частоты слов с учетом этапа фильтрации. Иными словами, в классической формуле $\sum_{qi} TF * IDF$, вычисляющей вес документа по отношению к запросу как сумму контрастностей слов запроса в документе, в Яндексе используется заниженная TF, учитывающая только те словопозиции, которые попали в «интересные» нам пассажи. Фактически Яндекс считает полностью «нерелевантными» все словопозиции слов запроса, не удовлетворяющие контекстным ограничениям. Очевидно, что данный подход тесно завязан на принцип: «избегать работы в документном контексте».

Ранжирование на уровне словопозиций: расчет веса словопозиции

Полученная контрастность слова распределяется на все его позиции, прошедшие фильтр. Затем по ним происходит итерирование и вычисление веса каждой словопозиции с учетом расстояния до всех остальных слов из запроса, попавших в пассаж. Учет состоит в вычислении сходства этого расстояния с заданным в запросе оптимальным расстоянием. Наконец, веса словопозиций, взвешенные по сходству их полного контекста, «собираются» обратно и образуют вес документа.

Расчет веса словопозиции позволяет максимально точно учесть сходство пассажа и запроса. При этом выигрыш получит документ, у которого более «тяжелые», смыслоразличительные слова окажутся в контексте, более похожем на контекст в запросе.

Предположим, ранжирование пассажей рассчитывается без взвешивания каждой позиции. Пусть задан запрос [aa BB cc dd], где BB - самое «тяжелое» слово. Из двух пассажей:

• [aa __ BB __ cc dd ee]

• [aa BB cc dd __ __ ee]

Яндекс предпочтет тот, в котором окружение более тяжелого слова [BB] больше похоже на его окружение в запросе, то есть второй. Тогда как остальные, известные мне алгоритмы взвешивания пассажей [14], [15], будут считать веса обоих пассажей одинаковыми.

Функция контрастности

В классической литературе по IR можно встретить разные функции нормирования и сглаживания внутри-документной частоты при вычислении контрастности TF*IDF.

$$TF_{norm} = K + (1 - K) \times \frac{TF}{TF_{\max}} \quad [11]$$

$$TF_{norm} = \frac{\log(TF + 1)}{\log(DL)} \quad [12]$$

$$TF_{norm} = \frac{(k_1 + 1) \times TF}{k_1 \times ((1 - b) + b \times \frac{DL}{DL_{avg}}) + TF} \quad [13]$$

Функция Яндекса, подобно функциям Harman и BM25, нормализует внутри-документную частоту по размеру документа.

Следует отметить, что в Яндексе используется дополнительный анализ текстов при индексировании для подавления многократного повторения слов в тексте в расчете на повышение ранга документа в выдаче поисковых машин [8].

Коэффициент контекстуального сходства

Функциям весов пассажей, описанным в литературе:

$$Score(Q',l) = \frac{1}{(l+1)^{\alpha} \times (L_{max}+1)^{|Q|-|Q'|}}; \alpha = 0.5...1.0 \text{ [14]}$$

$$Score(Q',l) = \begin{cases} \frac{L_{min}}{l+1}, l > L_{min} \\ 1, l \le L_{min} \end{cases}; L_{min} = 4...16 \text{ [15]}$$

Присущи следующие общие черты:

• Объемлющие пассажи игнорируются

• Позиции внутренних опор не принимаются во внимание

• Ранг неполных пассажей строго меньше ранга полных

• Вес пассажа — плавно убывающая функция, обратно пропорциональная длине (или корню длины) пассажа и его «неполноте»

В функции Яндекса (табулированный набор коэффициентов) также соблюдаются некоторые их этих принципов, в частности, принцип деградации неполных пассажей. Схожим выглядит и убывание при уменьшении сходства с оптимальным расстоянием.

Учет форматирования текста

Яндекс использует учет форматирования при ранжировании дважды. При вычислении контрастности слова используется информация о вхождении его в выделенные области текста (заголовки и т.п.). Кроме того, на этапе вычисления веса пассажа, пассажи, полностью попавшие в некоторые зоны документа, получают дополнительные баллы.

Яндекс также анализирует форматирование на этапе индексирования.

6. Эксперимент РОМИП-2004

Яндекс участвовал в двух дорожках РОМИП-2004: «Веб-поиск» и «Поиск по коллекции нормативных документов». Для Веб-поиска мы вручную выбрали «лучший» вариант из 8-ми: два вида ограничения контекста (предложение и документ), с группированием или без группирования по хостам. Коэффициент мягкости брался в одном случае равным 6 (значение по умолчанию), а в другом — 10. Для нормативной коллекции выбиралось лишь лучшее контекстное ограничение, а группирование не имело значения. Вариант синтаксического преобразования запроса за нехваткой времени испробован не был.

Лучшим вариантом для обеих коллекций мы посчитали: «документный контекст, отсутствие группировки, мягкость 6».

В силу нехватки информации, необходимой для адекватного веб-поиска (ссылочный ранг, текст ссылок), мы выбрали вариант, нацеленный в основном на повышение полноты, в надежде, что функции полнотекстового ранжирования помогут нам с точностью. Наш расчет оказался в целом верным. Лишь в одном эксперименте (старая коллекция запросов 2003 года, расширенные описания, «слабая» релевантность) показатели R-Precision и Average Precision Яндекса не оптимальны, возможно, из-за очень размытых запросов в старой коллекции.

Следует также отметить, что и метрики точности, используемые в РОМИП, и сами значения (0.27-0.45) похожи на данные, которые получает Яндекс, проводя регулярные измерения поиска www.yandex.ru по нашей базе асессоров. Недостающие 0.2-0.3 точности объясняются дополнительным анализом веб-графа, используемым в реальном веб поиске.

Характерно, что относительно неплохой результат Яндекса в коллекции нормативных документов был показан в варианте «слабой» релевантности, то есть в ситуации, когда полнота поиска имеет большее значение.

7. Заключение

Алгоритмы, описанные в данной статье были разработаны в 1996-1999 годах, и библиография подобрана для иллюстрации идей и методов, реализованных в этих алгоритмах. При их разработке она не использовалась.

Мы благодарны организаторам семинара не только за возможность сравнить полнотекстовый поиск Яндекса с другими системами, но и за настойчивость и упорство в процессе организации семинара.

Литература

[1] Маслов, М. // Алгоритм поиска Яндекс // РОМИП, 2003

[2] Trotman A. // Compressing Inverted Files // Information Retrieval, 2003

[3] Сегалович, И. // Индексирование русских текстов с использованием словаря, представленного на основе разреженной хэш-таблицы // Диалог, 1995

[4] Moffat, A., Zobel, J. // Self-Indexing Inverted Files for Fast Text Retrieval // TOIS, 1996

[5] Kaszkiel, M., Zobel, J., Sacks-Davis, R. // Efficient Passage Ranking for Document Databases // TOIS, 1999

[6] Параллелизм в поисковой архитектуре Яндекса // Яндекс // http://company.yandex.ru/programs/web_200203.html

[7] Woods, W.A. // Transition Network Grammars for Natural Language Analysis

[8] Яндекс // Тезисы выступления Яндекса на Диалоге-99 // http://company.yandex.ru/articles/article6.html, Диалог, 1999

[9] Wilkinson, Zobel, J., Sacks-Davis, R. // Similarity Measures for Short Queries // TREC, 1995

[10] Позняк, Э. Г., Шикин, Е. В // Дифференциальная геометрия: Первое знакомство // Издательство МГУ, 1990

[11] Croft, W.B. // Experiments with Representation in a Document Retrieval System // Information Technology: Research and Development, 1983

[12] Harman, D. // An experimental study of factors important in document ranking // SIGIR, 1986

[13] Robertson S.E. et al // Okapi at TREC-3 // TREC, 1994

[14] Hawking, Thistlewaite // Relevance Weighting Using Distance Between Term Occurrences // Technical Report TR-CS-96-08, The Australian National University, 1996

[15] Clarke, Gordon // Relevance Ranking for One to Three Term Queries // RIAO, 1997

Yandex at RIRES 2004. Some aspects of full text search and ranking in Yandex

Ilya Segalovich, Michail Maslov
The paper gives some details about full text search and ranking in Yandex, such as query execution architecture, quorum search parameterization, factors and functions in ranking. There is a discussion of Yandex results in RIRES-2004.