1 Ранжирование документов

Существует несколько традиционных моделей ранжирования документов:

1.1 TF*IDF

1.2 Лексическая близость

1.3 BM25

$$\text{score}(D,Q) = \sum_{i=1}^{n} \text{IDF}(q_i) \cdot \frac{f(q_i,D) \cdot (k_1+1)}{f(q_i,D) + k_1 \cdot (1 - b + b \cdot \frac{|D|}{\text{avgdl}})},$$

Главный недостаток: ручной видоискатель и станковый пулемёт.

1.4 BM25f

1.5 Альтернативные модели ранжирования

2 Обратная частота

Очевидной и традиционной мерой значимости термина в запросе является обратная частота термина, или IDF. Более того, классические модели поисковых систем полагают достаточной метрикой соответствия документа запросу значение TF * IDF, то есть произведение обратной частоты термина на его частоту в документе. Сама же обратная частота определяется как логарифм отношения общего количества документов в индексе к количеству документов, в которых встретился термин.

$$IDF = \log\frac{N(total)}{N(term)}$$

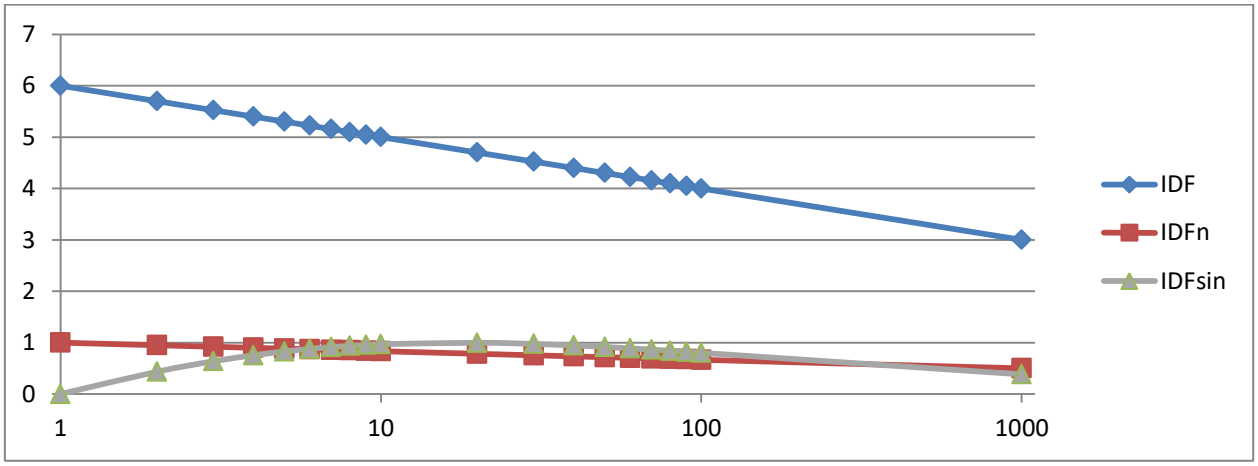
Величина тем больше, чем реже встречается термин в коллекции документов, то есть это мера избирательности термина. Единственный её недостаток – непредсказуемость её абсолютного значения: величина имеет естественный физический смысл, однако практическое её применение несколько затруднено.

Поэтому применяем нормированную на максимальное значение величину – ***IDF$_n$***.

$$IDFn = \frac{log\frac{N(total)}{N(term)}}{\log N(total)}$$

Для модерируемых данных эта функция очень хорошо подходит, а вот для немодерируемых максимальный ранг получат, к примеру, орфографические ошибки. Поэтому функцию можно упаковать ещё под синус:

$$IDF_{sin} = \sin(\pi\, IDFn^k)$$



Ну и, наконец, даже самые частотные термины имеют право на существование, тогда как единственный документ, в котором содержится самый «весомый» термин запроса, не должен обеспечивать стопроцентную релевантность. Поэтому окончательный вид формула приобретёт такой:

$$IDF_t = \frac{k_1 + k_2 IDF_{sin}}{k_1 + k_2}$$

3 Вычисление запроса с шаблонами (wildcards) и омонимия

Термины очень часто распадаются на несколько омонимов, так что отношения между терминами запроса оказываются сложными.

<звезда по имени с*>

звезда ~ по ~ имени ~ (солнце | солярис | собака | сечин | …)

Самый простой пример – запрос со звёздочкой: даже если нет явных операторов, другие слова относятся к нему как «было бы хорошо, если бы слово присутствовало», тогда как к вариантам самого слова применима фраза «устроит любой из омонимов». А реализаций шаблона может быть очень много.

1. Можно вывернуть запрос, переформулировав как конъюнкцию дизъюнкций: (звезда ~ по ~ имени ~ солнце) | … (звезда ~ по ~ имени ~ сечин)
2. Можно вычислить вес такого комбинированного термина.

Попробуем второй путь:

$$N(t_1 \dots t_N) = N_{total} * \left(1 - \prod (1 - \frac{N_{t_i}}{N_{total}})\right)$$

Тогда обратная частота будет:

$$IDF(t_1 \dots t_N) = \log \frac{N_{total}}{N(t_1 \dots t_N)},$$

или, подставив значение,

$$IDF(t_1 \dots t_N) = \log \frac{1}{1 - \prod (1 - \frac{N_{t_i}}{N_{total}})}$$

4 Ранжирование кортежей

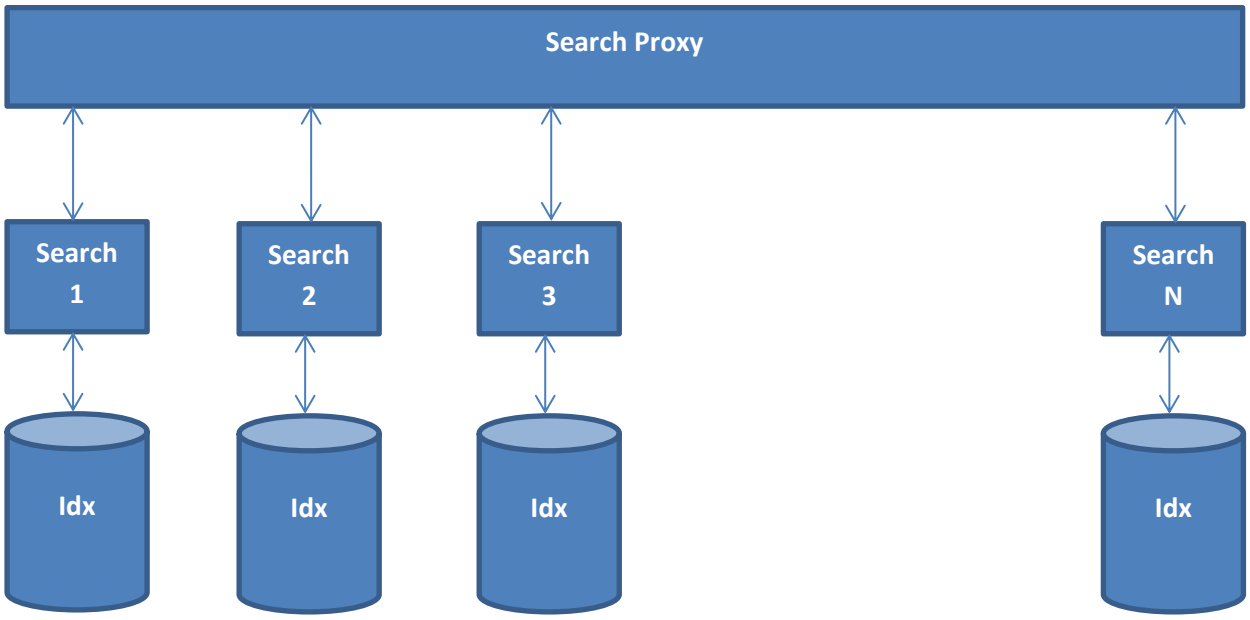
Для однословных запросов всё очевидно, тогда как для неоднословных можно записать:

$$P_Q = f_{dist} * \left[1 - \prod(1 - p_i)\right] * \prod tax_j$$

Здесь

- P_Q - вероятность того, что данное [неполное] вхождение слов запроса на ограниченном контексте является релевантным;
- f_{dist} – некоторая [убывающая] функция, отражающая вероятность смысловой связи между словами в зависимости от расстояния между ними;
- p_i – вероятность того, что вхождение i-го термина есть релевантное запросу вхождение;
- tax_{Tj} – «штраф» за отсутствие в найденном компактном вхождении некоторого термина из запроса: $tax_{Tj} = (1 – IDF_{Ti})^{K3}$, где K_3 – очередной командирский коэффициент.

5 Архитектура распределённой поисковой машины



5.1 Вычисление запроса в распределённой системе

Вычисление запроса распадается на две фазы:

- предварительный запрос ко всем сегментам, возвращающий информацию о количество документов по каждому из терминов в каждом из сегментов и общем количестве документов;
- запрос на поиск и ранжирование с предвычисленными весами терминов.

6 Квалифицированный кворум запроса

Запросы:

- формальные (с операторами);
- точное вхождение фразы;
- нечёткие.

Формальные запросы: проще всего. Можно даже ранжировать по BM25, с гарантией в документ есть все термины , просуммировали IDF, вычислили BM25, получили результат.

Точное вхождение фразы: нужны координаты слов, и желательно всех, включая даже запятые. Поиск без какого-либо «стоп-словаря», точная последовательность с или без фиксации формы слова. «AT&T», «RS-232», «А и Б» - те, что сидели на трубе. Поиск реальных цитат.

Нечёткие: найти максимально компактное и максимально полное вхождение запроса.

Для нечётких запросов важно отобрать наиболее значимые слова:

- <картошка с мясом> - устроит документ <картошка и мясо>;
- <Владимир Путин> – вполне устроит <Путин>;
- <валя оршулович чубчик кучерявый> - <оршулович> и <чубчик>.

Стоп-слова выкидывать нельзя, потому что они могут оказаться вовсе не стоп-словами.

Кворум – некоторое минимальное количество слов запроса, которые должны присутствовать в документе для того, чтобы начинать потрошить его координаты.

Квалифицированный кворум - минимальный суммарный вес присутствующих терминов.

Можно задавать функцией, можно таблично:

Количество слов	**Кворум**
1	1
2	0.9
3	0.8
4	0.75
5	0.7
6	0.6
7	0.5

Вычисление запроса:

1. Отобрать документы, которые содержат термины запроса, дающие в сумме вес больше пороговой величины.
2. В документе отобрать среди присутствующих самые весомые термины.
3. Найти ядра кортежей.
4. Украсить менее значимыми словами.