Resolución de Consultas

Consultas Booleanas (dónde el término está)
Consultas Wildcards (dónde los términos que coinciden están)
Consultas Calificadas (Ranking – dónde se habla de estos términos)
Consultas de Frase (dónde se habla de esta frase)
Consultas por Proximidad (Ranking – dónde se habla de algo parecido)

Consultas Booleanas

Métodos de resolución

- Índices invertidos. Se ubica el término en el índice, se obtiene la lista de distancias y de ella lista de documentos.
- Signature Files. Se obtiene el Hash del término y se busca en los Bits Slice los documentos que puedan tenerlo.

Opcionales para múltiples términos

- Conjunción (AND). Se obtiene la intersección de los conjuntos de documentos resultantes de las búsquedas por cada término. Se realiza primero la búsqueda del menos frecuente para luego ver en los demás listados si aparece alguno de los documentos que resultaron seleccionados.
- Disyunción (OR). Se obtiene la unión de los conjuntos de documentos resultantes para cada término. Se realizan las consultas de menos frecuente a más frecuente para hacer la Mezcla más fácil.
- Composición (AND, OR y NOT). Son consultas que mezclan los operadores y se resuelve con un árbol de operaciones, por lo general utilizando un poco De Morgan para acomodar las operaciones de forma tal que las conjunciones queden afuera y entre grupos de pocos elementos.

Consultas Wildcards

Este tipo de consultas genera muchos inconvenientes a los índices invertidos, porque la búsqueda de términos es a través de la Fuerza Bruta.
Para mejorar el rendimiento de este tipo de índices se hacen algunos agregados atendiendo las necesidades de búsqueda con “comodines”.

Métodos de resolución

- N-Gramas (2 -> hola -> çh-ho-ol-la-aç). Un N-Grama es una secuencia de N caracteres extraída de un término.
  Cada N-Grama se almacena en un **índice invertido de N-Gramas**. Es decir un índice que indica en qué términos está el N-Grama, símil al de términos-

documentos. Para formar un N-Grama se utiliza un **caracter especial** que define el comienzo y fin del término.
Resolver una consulta significa separar cada término a buscar en sus N-Gramas, buscarlos en el índice de N-Gramas y armar una **Conjunción** con los resultados.
Ahora tenemos todos los términos que responden a los N-Gramas de la consulta.
Pero, como dado que es muy local la visión de los N-Gramas, los *resultados de las búsquedas pueden contener términos que no respondan al término Wildcard completo*. Entonces se procede a **verificar** cada uno de los resultados contra la consulta.
Luego de esto se tienen todo los términos que se buscaban y hacer una **Disyunción** de los documentos en los que están me asegura tener el conjunto de todos los términos que tiene términos que respondan a la consulta Wildcard.

Digrama	Ocurrencias	Términos
Çh	3	6,7,15
Ho	1	6,20
La	4	3,4,5,6

- Léxico Rotado (hola -> çhola-holaç-olaçh-laçho-açhol). Se genera un **índice de rotaciones** en el cuál cada rotación apunta a sólo **un elemento** del índice invertido de términos. Cada léxico rotado pertenece sólo a un término dado.
  La resolución parte de **rotar el término de la consulta** hasta llevar el comodín al extremo derecho. Luego se buscan todos los léxicos rotados que **comiencen igual**. Se realiza una **Disyunción** de los términos resultantes y se buscan los documentos que tienen alguno de esos términos (**Disyunción**).
  Para los casos de múltiples comodines en un mismo término se rota hasta dejar uno a la derecha. Luego se puede optar por buscar los léxicos rotados que comiencen igual a la consulta hasta el primer comodín o separar el término en tantas partes como comodines haya y buscarlos por separado, para luego hacer una Conjunción de resultados.

Léxico rotado	Términos
Holaç	100
Olaçh	100
Laçho	100

Consultas Calificadas (Ranking)

Este tipo de consultas se caracteriza por retornar los documentos que tienen al menos uno de los términos de la consulta y los documentos resultantes están ordenados según una evaluación de relevancia.

Las formas de evaluar son muchas y muy distintas, veremos un par.

Métodos de resolución

- Coordinate Matching. Ordena de acuerdo a la cantidad de términos de la consulta que tiene cada documento. Este resultado se puede ver como el **producto interno** de los vectores de términos contenidos en la consulta y en cada documento, su resultado es la cantidad a comparar.

  Consulta “hola mundo” -> (0,0,1,1,0)
  Texto “hola ale” -> (1,0,1,0,0)
  Texto “che pibe” -> (0,1,0,0,1)
  Texto “pibe mundo” -> (0,0,0,1,1)
  Productos:
  (0,0,1,1,0) x (1,0,1,0,0) = 1
  (0,0,1,1,0) x (0,1,0,0,1) = 0
  (0,0,1,1,0) x (0,0,0,1,1) = 1
  Observaciones:
  Los términos que no están en el índice invertido no se evaluarán en el vector, porque no tienen lugar.
  Consulta “hola mundo cruel” -> (0,0,1,1,0)

  Si un término aparece más de una vez dará más importancia a los documentos que lo contengan.
  Consulta “hola mundo hola” -> (0,0,2,1,0)

  Asume que todos los términos tienen igual importancia.
  Consulta “como hacer bomba” -> (0,1,1,0,1,0,0)

  Favorece a los documentos que tengan más términos, los más largos, porque tienen más posibilidad de contener algún término.

- Producto interno. Se modifica el índice invertido para agregar a cada valor de salto (referencia a documento que tiene el término) las ocurrencias del término en él.

  “hola” -> (2,1) (10,4) (3,14)
  Ahora se pueden poner pesos en los vectores de los documentos que ilustran un poco la relevancia del término en ellos.

  Consulta “hola mundo” -> (0,0,1,1,0)
  Texto “hola ale” -> (1,0,1,0,0)
  Texto “hola mundo che” -> (0,1,1,1,0)
  Texto “mundo pibe, mundo, mundo” -> (0,0,0,3,1)
  Productos:
  (0,0,1,1,0) x (1,0,1,0,0) = 1
  (0,0,1,1,0) x (0,1,0,1,1) = 2

(0,0,1,1,0) x (0,0,0,3,1) = 3

Observaciones:

Sigue ganando el más largo y sigue dando igual importancia a todos los términos.

- Producto Interno Mejorado. Comienza a tener en cuenta al **término menos frecuente**. Fórmula de George Zipf.

Importancia de un término:

imp. = log10(cantidad total de documentos)/(documentos del término)

Ahora cambia el coeficiente de cada término en un documento, en lugar de ser su ocurrencia, es:

coef. = ocurrencia x imp.

Es conveniente guardar este valor, ya sea agrandando el índice o en lugar de las ocurrencias en cada documento, porque se utilizarán en cada consulta.

Consulta “hola mundo” -> (0,0,1,1,0)
“ale”, “che” , “pibe” -> log10(3/1) = 0,477
“hola”, “mundo” -> log10(3/2) = 0,176
Texto “hola ale” -> (0,477;0;0,176;0;0)
Texto “hola mundo che” -> (0;0,477; 0,176;0,176;0)
Texto “mundo pibe, mundo, mundo” -> (0;0;0;0,528;0,477)

Productos:

(0,0,1,1,0) x (0,477;0;0,176;0;0) = 0,176
(0,0,1,1,0) x (0;0,477; 0,176;0,176;0) = 0,352
(0,0,1,1,0) x (0;0;0;0,528;0,477) = 0,528

Observaciones:

Aún siguen siendo importantes los documentos largos por tener más palabras.

- Método del Coseno. Se busca interpretar los vectores en forma gráfica (ángulo y módulo). Entonces, en lugar de ver similitud en sentido global, producto interno, se observa sólo la similitud en sus ángulos.

Si tenemos que:

X, Y vectores.
X x Y = |X| x |Y| x cos(ángulo)
cos(águlo) = X x Y / |X| x |Y|

$$\cos\phi = \frac{\sum Xi * Yi}{\sqrt{\sum Xi^2} * \sqrt{\sum Yi^2}}$$

Cuanto mayor el coseno (más cercano a 1) más parecidos los ángulos de los vectores, o menor la diferencia entre ellos.

Este método **sigue utilizando el producto interno** mejorado **y agrega** su aplicación a la **consulta**

Consulta “hola mundo” -> (0;0; 0,176; 0,176;0)
“ale”, “che” , “pibe” -> log10(3/1) = 0,477
“hola”, “mundo” -> log10(3/2) = 0,176
Texto “hola ale” -> (0,477;0;0,176;0;0)
Texto “hola mundo che” -> (0;0,477; 0,176;0,176;0)
Texto “mundo pibe, mundo, mundo” -> (0;0;0;0,528;0,477)

Norma consulta: 1,425
Norma “hola ale”: 0,508
Norma “hola mundo che”: 0,538
Norma “mundo pibe, mundo, mundo”: 0,711

Producto “hola ale”: 0,176
Producto “hola mundo che”: 0,352
Producto “mundo pibe, mundo, mundo”: 0,528

Coseno “hola ale”: 0,176/1,425x0,508
Coseno “hola mundo che”: 0,352/1,425x0,538
Coseno “mundo pibe, mundo, mundo”: 0,528/1,425x0,711

Coseno “hola ale”: 0,243
Coseno “hola mundo che”:0,459
Coseno “mundo pibe, mundo, mundo”: 0,521

Observaciones:

Como todo se divide por la norma de la consulta, ésta se puede simplificar.
Las normas de los vectores pueden ser almacenadas en el índice para optimizar las consultas.

Consultas de Frases (Ranking por frase)

Este tipo de consultas toma en consideración el orden de los términos ingresados y busca documentos que respondan estrictamente a ese patrón.

Métodos de Resolución

- Reestructuración del Índice invertido agregando posiciones. Se agrega a los datos contenidos por el índice invertido la **posición en Número de Término** que ocupa el término buscado.
  “poco gordo, por poco cobro, volcó otro mocoso poco jocoso”
  Para el término “poco” tendríamos la siguiente entrada:
  “Poco”: D1 – O3 – P1 | P4 | P9
  (D: Documento, O: ocurrencia, P: posición)

Búsqueda:

Tomo el término menos frecuente de la frase y obtengo la lista de documentos en los que aparece.

Mientras queden otros términos por analizar: se toma el de menor frecuencia y se fija si su posición relativa la primero se encuentra en algún documento. Sólo me quedo con los que cumplen esta condición y vuelvo a empezar.

Ej: Busco “Por poco cobro” y tengo la siguiente lista de documentos:

Cobro: D1 – O1 – P5
D2 – O2 – P12|P49
D3 – O1 – P31

Poco: D1 – O3 – P1|P4|P9
D2 – O2 – P10|P50
D3 – O1 – P30
…

Por: D1 – O1 – P3
D2 – O2 – P9|P60
D3 – O5 – P28|P33|P40|P90|P100
…

Sólo me quedo con D1.

Observaciones:

No es muy eficiente por la comparación de posiciones.

Optimización: Si el menos frecuente no es el primero se guarda en memoria **no su posición sino la que correspondería al primer término de la frase**. Así siempre comparo por igualdad, si “normalizando” con el siguiente término analizado no me da el mismo comienzo no me sirve la posición.

- Índices de próxima palabra (Nextword). Se indexan términos de a pares. Una entrada para uno, y de ella, entradas para todos los que lo siguieron en algún documento y allí se ubican los datos de ocurrencia total, ocurrencia en el documento y posiciones de aparición.

  Ej: por -> T3 (poco = D1 – O1 – P5; D2 – O1 – P9) (aca = D4 – O3 - P9|P25|P50) (nada = D4 – O1 – P60)

  Resolver una consulta de este tipo requiere separar la frase en unidades de Próxima Palabra:

  Por poco cobro -> “por poco” y “poco cobro”. Y buscar por posiciones.

Observaciones:

Es mucho **más rápido que el anterior** por buscar en menos punteros. Pero **ocupa mucho más lugar**.

- Índice de próxima palabra reducido (Stopword). Se hace Índice de próxima palabra sólo a las combinaciones más comunes y con tomar una pequeña cantidad de éstas se mejora notablemente.

Consultas por Proximidad

Cuando se utilizan más de un término en la consulta se buscan documentos que contengan a los términos pero que entre ellos no haya más distancia (en cantidad de términos) que cierta medida. Con esto se logra cierta identidad del texto y buenos resultados. El ranking se realiza en base a la distancia entre términos u otros algoritmos más complejos.

Referencias.

- Self indexing Inverted Files for fast text retrieval – Alistair Moffat, Justin Zobel – Feb 1994
- Fast ranking in limited space - Alistair Moffat, Justin Zobel – May 1993
- Efficient phrase querying with an auxiliary index – Dirk Bahle, Hugh Williams, Justin Zobel – 2001
- Compression and fast indexing for multi-gigabyte text databases