C.D.M Cuadernos de Documentación Multimedia
Vol. 5 Año 1996
ISSN: 1575-9733

Biblos-100: Gestor de bases de datos documentales

Carlos Beltran

OBJETIVOS

Se presenta un modelo de recuperación de información en Gestores Documentales, que constituya la gramática de un lenguaje para el dialogo con el ordenador.

El modelo propone un conjunto de definiciones básicas y las **reglas necesarias** para **construir frases de interrogación** que aborden el mayor número posible de necesidades de información previsibles, y con la suficiente flexibilidad para poder admitir cualquier otra clase de necesidades que se revelen útiles en el futuro.

A tal efecto, nuestra propuesta incluye un vocabulario de **símbolos** y palabras (que deberían ser diferentes para cada idioma de trabajo) y una sintaxis sencilla que consigan acercar la formulación de necesidades de información mediante frases de interrogación lo más cercanas posible al lenguaje natural (CESEDA, 1995).

Alternativas al concepto de gestión documental

Existen varios planteamientos sobre la forma de entender un Gestor Documental. Algunas ideas defienden la utilización de una estructura definida a través de campos muy estructurados, al estilo de una **base de datos** de gestión administrativa, complementados con uno o más campos de texto no estructurado de tipo *memo*, que no proporcionan puntos de acceso. Este modelo corresponde a una concepción determinista de la gestión de información que es útil para gestionar determinada información administrativa, muy estructurada y muy predecible, pero no para gestionar información textual poco o nada estructurada.

Otros enfoques confían en la utilización de **descriptores** para representar la información que contiene un texto, pero sin que el texto mismo forme parte de la base de datos. Este es el modelo clásico de las bases de datos documentales de tipo referencial, en las cuales, hasta ahora, se tendía a prescindir del poder de auto-representación que posee el propio texto del documento.

La utilización de descriptores tiene la ventaja de que el operador decide las palabras clave de cada texto, pero necesita del concurso de una persona que prepare esta información, y este trabajo puede hacerse inviable en la práctica, si se maneja información voluminosa.

Sin embargo, dado que la evolución de los ordenadores apunta a un abaratamiento tanto de los soportes de almacenamiento como de las CPU, y por tanto se dispone , en general, de una gran capacidad de cómputo y de almacenamiento de información, aún en sistemas modestos, parece lógico aprovechar esta circunstancia para partir de un **modelo general documental**

basado en la utilización del texto completo de los documentos que: primero, sea capaz de explotar las propiedades estadísticas de ese texto como medio de representación de la información y, segundo, que cada palabra o frase del texto pueda actuar como punto de acceso al documento, con o sin utilización de otras estructuras conceptuales, tales como descriptores, sinónimos y Thesauros.

Vol. 5 Año 1996
ISSN: 1575-9733

Ahora bien, para que el modelo anterior sea eficaz, requiere de los siguientes elementos:

1. Un sistema de indexación rápido y eficiente, ya que debe ser capaz de crear índices con el texto completo del documento.
2. Un sistema de recuperación muy selectivo, que pueda minimizar tanto el ruido (obtención de información no relevante) debido a falsas coordinaciones, como el silencio (pérdida de información relevante) debido a problemas de sinonimia.
3. Una interfase de usuario que permita expresar necesidades de información en lenguaje natural, de manera que el sistema pueda comparar dos clases de textos: la pregunta y los documentos.

Así pues, el trabajo de **indexación** deberá ser coherente con el sistema de recuperación propuesto y estar por tanto dotado de la misma flexibilidad para admitir la inclusión de un **glosario** (asimilable en el modelo que proponemos a los descriptores) y de un fichero de **palabras vacías**.

Es necesario indicar, antes de centrarnos en la descripción del sistema, que vamos a hablar exclusivamente de información textual, es decir no contemplaremos la presencia de imágenes, sonido, animación, o cualquier otro tipo de información no-textual. No obstante debe resultar fácil extrapolar las ideas que se expondrán para incluir en el método cualquier otro tipo de información.

Tipos de búsquedas

La frase de búsqueda de la que se parte no adopta siempre la misma forma, ya que, en cada caso, representa diferentes problemas de información . De este modo, aunque podemos definir varios tipos de búsqueda según el contexto, sin embargo, es importante que las diversas frases puedan expresarse mediante un único lenguaje de interrogación. Las clases de búsquedas serían las siguientes:

- BÚSQUEDA SIMPLE - Se parte de una o más palabras y de su relación dentro del texto. Aquí se puede dar la mayor complejidad en la búsqueda y en la propia construcción de la frase inicial. Apuntaremos un esquema para abordar el problema.

  Dado que un descriptor es también una palabra o un grupo de palabras (ej.: "economía", "sistemas expertos", etc.), este método incluye, como un caso particular, la recuperación mediante descriptores, siempre que se hayan asignado previamente al documento mediante una operación intelectual.

- BÚSQUEDA APROXIMADA - Se pretende encontrar unidades textuales que contenga una determinada frase de búsqueda completa. Los textos buscados pueden contener la frase exacta o una versión aproximada de ella, o incluso puede contener palabras que son sinónimos de las que realmente están en el texto buscado.

- BÚSQUEDA POR COINCIDENCIAS - Se parte de una serie de palabras y se pretende encontrar las unidades textuales o documentos que reúnan el mayor número posible de ellas. Todas las palabras tiene el mismo peso o bien, podemos asignar un peso diferente (positivo o negativo) a cada una de ellas.

Vol. 5 Año 1996
ISSN: 1575-9733

- BÚSQUEDA POR GRADO DE RELEVANCIA (*RANKING*) - Obtenida una lista de textos, podemos ordenar la información hallada completándola con una clasificación de esa información en base a su grado de relevancia, proporcionando así la función que la jerga documental denomina *ranking* (Frakes y Baeza-Yates, 1992). Aunque en rigor se trata de un método de ordenación, dada su potencia heurística, la literatura técnica suele considerarlo una genuina forma de recuperación, tratamiento que también adquiere en nuestro modelo.
- BÚSQUEDA ITERATIVA - En el mismo supuesto anterior y compatible con él, podemos realizar una segunda búsqueda basándonos exclusivamente en los resultados obtenidos de la primera. Esta opción está basada en el modelo teórico denominado *relevance feed-back*, el cual, pese a estar teorizado desde hace años (Harmand, 1992), hasta ahora apenas ha sido implementado en los sistemas de gestión documental que ofrece el mercado.

  La búsqueda iterativa consiste en determinar si alguna de las unidades textuales obtenidas en el primer resultado se consideran una buena representación de la necesidad de información, en cuyo caso ese texto puede utilizarse como un modelo para que el sistema busque otros documentos que contengan textos semejantes. La idea es que si existen otros documentos que también sean relevantes para el problema de información planteado por el usuario es probable que contengan textos parecidos al indicado como modelo de comparación.

Con esta clasificación no queda agotado el tema, porque en el modelo que proponemos se podrían definir, en determinados supuestos, una **búsqueda por interrogación,** una **búsqueda difusa** (por palabra mal escrita, por ejemplo) una **búsqueda fija o por líneas**, etc.

Vol. 5 Año 1996
ISSN: 1575-9733

LENGUAJE DE INTERROGACIÓN: BÚSQUEDA SIMPLE

FRASE SIMBÓLICA

La búsqueda simple se realizará a través de una frase de interrogación que debe estar constituida por los cuatro términos descritos en la siguiente frase simbólica:

Símbolo + Texto nulo: + Acotación + (CLAVE + MODIFICADOR)

Ninguno de los términos es obligatorio.

Vayamos comparando cada uno de los términos de la frase simbólica con un ejemplo real para centrar los conceptos desde el principio.

Imaginemos el siguiente texto extraído de la ficha de un paciente en una consulta medica:

Pongamos ahora un ejemplo de frase de búsqueda y analicemos su significado:

Adelantemos que se trata de una frase un tanto peculiar. Pero nos valdrá para analizar la construcción de frases de búsqueda con el modelo propuesto.

TÉRMINOS DE LA FRASE

Describamos cada uno de los términos definidos.

El **símbolo** es el primer carácter de la frase, en nuestro caso un punto (.). Nos indica una búsqueda por claves. Otros símbolos pueden identificar otros tipos de búsqueda como se indicará después.

El **texto nulo** es todo el comentario que antecede al carácter (:) y que servirá exclusivamente como referencia para posteriores consultas, si la frase, por ejemplo se ha podido guardar en un macro. Es nuestro caso.

Quiero.....dirección:

La **acotación** es la indicación del campo de búsqueda dentro del libro y texto.

Desde A hasta AZZZ

Desde línea 1 hasta línea 6

La frase que resta está formada por una suma de **CLAVES** y **MODIFICADORES** que se suceden alternativamente. Las claves pueden estar contenidas en el texto. Los modificadores NO.

REGLAS SINTÁCTICAS

Organizando (normalizando) el cuarto término de la frase de búsqueda, veremos que las claves estarán siempre en posición **impar** y en las posiciones **par** estarán los modificadores, con la

Vol. 5 Año 1996
ISSN: 1575-9733

única excepción del modificador de proximidad. Cada frase estará formada por **oraciones**, separadas por los modificadores booleanos.

Las claves serán: **Perez paciente alergia síntomas polen 4 primavera**

Los modificadores: **enlínea o enpárrafo y < o**

Las oraciones: **(Perez enlínea paciente) (alergia enpárrafo síntomas) (polen) (primavera)**

La frase se aproxima lo suficiente al lenguaje natural como para que sea fácil de aprender. Las reglas sintácticas son simples:

- En posición par solo se permiten los modificadores que se definan
- La frase debe terminar con una palabra clave
- Después de uno de los modificadores de proximidad (>,<) y del modificador

Encolumna solo se permite una clave numérica.

- También después de las **acotaciones** "desde línea" y "hasta línea" se requiere clave numérica.
    - La oración anterior al modificador de proximidad debe ir precedida del modificador "y" (no tiene sentido escribir A o B < 4).
    - Cabe por último indicar que la construcción de la frase se puede enriquecer admitiendo caracteres de sustitución (?) y de prolongación (*).

Lista de Modificadores

La lista de modificadores y los tipos a los que pertenecen puede extenderse tanto como deseemos, siempre que se incluyan en el algoritmo de búsqueda.

Proponemos la siguiente lista de modificadores clasificada por tipos:

Posicionales:

- Desde
- Hasta
- Desde línea
- Hasta línea
- Desde columna
- Hasta columna

De inclusión:

- En libro La búsqueda se realizará en el libro indicado. En caso de que no
- exista el modificador, se realizará en todos los libros disponibles.
- En línea Perez deberá estar en una línea que comience por paciente.
- En párrafo Alergia deberá estar en un párrafo que comience por síntomas.
- En columna Algún carácter de la palabra debe estar en la columna x

Vol. 5 Año 1996
ISSN: 1575-9733

Booleano:

- Y Ambas palabras deben estar.
- Basta una de ellas.
- YNO Debe estar la 1ª y no la 2ª.

De proximidad:

- < El número máximo de palabras de separación.
  - El número mínimo de palabras de separación.

Significado de la frase

Analizada la frase en todos sus términos, veamos ahora su significado. La frase pretende encontrar todos los documentos que cumplan las condiciones siguientes:

A: Que el título del texto empiece por A (Desde A hasta AZZZ)

B: Que la búsqueda se realice exclusivamente entre las líneas 1 y 6 de cada texto.

C: Que exista la palabra **Pérez** en una línea que comience por la palabra **paciente.**

D: O bien que exista la palabra **alergia** en un párrafo que comience por la palabra **síntomas**, siempre que en el mismo documento exista la palabra **polen** y además que las palabras alergia y polen estén separadas menos de 4 palabras.

E: O bien que exista la palabra **primavera**.

Todos los documento que cumplan estas condiciones deben ser seleccionados y presentados al operador.

Lista de símbolos

En cuanto se refiere al símbolo, o primer carácter, indicará al ordenador el tipo de búsqueda que deseamos realizar. Podemos definir los siguientes supuestos:

1. Si la frase comienza por un carácter alfanumérico se tomará la palabra escrita como titulo de un texto. Si hemos definido los títulos de texto con una sola palabra , y existen varias en la frase, se acudirá a búsqueda por frase aproximada.
2. (.) Si la frase comienza por un punto la búsqueda se limitará exclusivamente a las **palabras indexadas** dentro del libro (tal como se describió en el ejemplo).
3. 3.- (*) Si la frase comienza por un asterisco la búsqueda se realizará esencialmente **en el texto**, y será por tanto una búsqueda no inmediata. Son válidas las mismas reglas que en el caso anterior, pero el trabajo se realiza en textos no indexados.
4. (/) Si la frase comienza por una barra, la búsqueda se realizará a través de un macro que se ha definido previamente y que contiene una frase de búsqueda utilizada con anterioridad, que a su vez podrá comenzar por cualquiera de los caracteres que se reseñan. Por ejemplo: **/informe,** será sustituido por la frase completa que representa.

Vol. 5 Año 1996
ISSN: 1575-9733

5. (+) Si la frase comienza por el signo más se podría realizar una **búsqueda por uso**, en función de la ultima fecha en la que se consultó un texto el número de consultas o listados realizados, etc.
6. (%) Si la frase comienza por el símbolo %, la búsqueda se realizará en todos los libros que existan, en vez del libro de trabajo como se hacía en todos los casos anteriores. Se sustituye el símbolo (%) por el (.) y se realiza búsqueda indexada. (Equivaldría a la NO inclusión del modificador: **Enlibro**)
7. (-) Si la frase comienza por un guión, se realizará una **búsqueda aproximada**.
8. (&) Si la frase comienza por un ampersand, se realizará una **búsqueda por coincidencias.**

BÚSQUEDAS APROXIMADA Y POR COINCIDENCIAS

Los tipos 7 y 8 no siguen las reglas establecidas hasta ahora en la frase de búsqueda.

Si los modificadores no están en la posición sintácticamente correcta, deberemos inferir que no se trata de una búsqueda simple. Caben entonces dos posibilidades: Búsqueda **Aproximada** y búsqueda por **Coincidencias**

BÚSQUEDA POR FRASE APROXIMADA

En la búsqueda de información documental surge con frecuencia la necesidad de localizar documentos cuyo texto contenga una frase determinada, ya que en ocasiones es ésta la única manera de expresar una necesidad de información.

Podríamos resolver el problema copiando las palabras de la frase y formando con ellas una frase de búsqueda estándar separándolas por operadores Y.

Si deseamos localizar la frase:

"Que te parece mi frase"

Deberíamos escribir:

.Que y te y parece y mi y frase

y si tenemos en cuenta un fichero de palabras vacías que incluyera las palabras QUE, TE y MI, deberíamos escribir simplemente:

.parece y frase

Los resultados obtenidos serían pobres y además no contemplarían frases **parecidas** que podrían ser relevantes en la búsqueda, como por ejemplo:

QUE TE PARECEN MIS FRASES...

QUE PAREC MI FRASE...

QUE TE LO PARECE MI FRAS...

Vol. 5 Año 1996
ISSN: 1575-9733

La técnica de búsqueda aproximada intenta solucionar este problema.

Algoritmo de búsqueda aproximada

Se localizará dentro del texto los párrafos que contengan como clave la primera palabra de nuestra frase (QUE, en el ejemplo propuesto). Si la primera palabra no es clave en el texto, se podría trasladar el método a la segunda palabra, con la consiguiente penalización.

Se realizará una doble comparación entre la frase de búsqueda y la oración (en el texto), que sigue a la palabra clave encontrada. La primera comparación se realizará carácter a carácter y la segunda palabra a palabra.

Se valorarán las coincidencias con 1 punto, sin incluir los espacios. Cuando no exista coincidencia, se adelantará un carácter, primero en la frase de búsqueda y luego en el texto y se volverá a realizar la comparación.

En caso positivo, se continuará la comparación. En caso negativo se rechazará la oración del texto.

Se podría seguir la idea, continuando la comparación con 2, 3 o más espacios, aunque la experiencia demuestra que el sentido de la frase se aleja del original. (QUE TE PARECE MI FRASE y QUE MI FRASE sea..., faltando TE y PARECE).

Los puntos obtenidos se dividen por el número total de caracteres o palabras, según el caso, y se obtiene así un **coeficiente de aproximación** para cada frase comparada.

Todas las oraciones cuya puntuación supere cierto umbral, en alguna de las dos comparaciones, serán presentadas como frases aproximadas. Este umbral debe poder ser fijado libremente por el operador, que lo ajustará según los resultados obtenidos.

Según nuestra experiencia, parece que un umbral (coeficiente) del 65% separa bastante bien las frases que mantienen un mismo sentido o significado, con la frase de búsqueda, tanto en la comparación por caracteres como en la comparación por palabras. Veamos el proceso de comparación con uno de los supuestos citados en el punto anterior.

Frases de partida: QUE TE PARECE MI FRASE 22 caracteres

QUE TAL PARECE MI FASE

Primera intercalación: QUE TE PARECE MI FRASE 4 caracteres no blancos

QUE TAL PARECE MI FASE

Segunda intercalación: QUE TE PARECE MI FRASE 9 caracteres no blancos

QUE TAL PARECE MI F ASE +2 caracteres exactos

Puntuación: Caracteres comparados: 22

Vol. 5 Año 1996
ISSN: 1575-9733

Coincidencias: 15

Coeficiente de aproximación: 68%

El mismo proceso se seguirá para realizar la comparación por palabras. Aquí, además se incluye la posibilidad de realizar la comparación con una palabra, y en caso de fallo, con todos sus **sinónimos**. El éxito con cualquiera de los sinónimos da como buena la comparación.

QUE TE PARECE MI FRASE

y QUE TE PARECE MI ORACION

darán un coeficiente de aproximación del 100% siempre que **frase** y **oración** formen pareja en un fichero de sinónimos.

Realizadas las dos comparaciones, se tomará el coeficiente mayor de los dos obtenidos y se dará como coeficiente final.

BÚSQUEDA POR COINCIDENCIAS

Bajo ciertos supuestos, resulta interesante conocer todos los textos que contengan al menos 1 palabra clave de una lista de palabras propuesta. Si construimos una frase de búsqueda como esta:

&Paciente alergia polen primavera flor

una búsqueda por coincidencias nos daría una lista de textos ordenada, que contuviera en primer lugar los textos que incluyen las 5 palabras de la frase, después los de 4, los de 3, etc.

Puede completarse la definición de la frase incluyendo una **ponderación** de cada palabra, y un **límite** inferior en el número de coincidencias. La frase podría quedar así:

&2 paciente alergia*2 polen primavera flor*-4

cuyo significado es el siguiente:

- No se presentarán los textos con una coincidencia menor de dos palabras (&2)

- La palabra alergia "pesa" el doble que la palabra polen (*2).

- La palabra flor tiene un peso negativo, (*-4) y actúa en realidad como si se tratara del

caso "YNO flor".

GENERALIZACIÓN DE LA FRASE

Hemos apuntado varios tipos de **búsqueda primaria** distintos, identificables por el ordenador por medio del símbolo, o primer carácter de la frase.

Vol. 5 Año 1996
ISSN: 1575-9733

Se podría, por último, generalizar la frase de búsqueda de forma que fuera el propio programa el que decidiera el tipo de búsqueda que debe realizar, de forma que no se necesitara incluir el **símbolo** como primer carácter.

En caso de que se escriba el símbolo, el programa elige directamente el tipo de búsqueda correspondiente. En caso de que no se escriba, seguirá el siguiente criterio:

1. Se busca la frase escrita como **título** de un texto. Si existe ese texto, se presenta, en caso contrario se pasa al punto 2.

1. Se analiza si la frase contiene **modificadores** y están colocados en posiciones sintácticamente correctas. En caso afirmativo se inicia una **búsqueda simple** (símbolo .). Si la búsqueda da resultado nulo, se realiza una búsqueda en el texto (símbolo *).
2. Si no existen modificadores en posición correcta se analiza si hay una cadena del tipo *N (siendo N un número). En caso positivo, se toma como una **búsqueda por coincidencias**.
3. Si todavía no se ha cumplido ninguno de los supuestos anteriores, se realiza una **búsqueda aproximada**.

Listas de Textos

Partiendo de la frase de búsqueda que se utiliza se obtiene una lista de textos. Si se almacena esta lista de textos (solo los títulos y el libro al que pertenecen) en un fichero, se pueden realizar dos clases de trabajos con ella, por una parte gestionar estas listas y por otra incluirlas dentro de una frase de búsqueda posterior.

Imaginemos que, como consecuencia de una frase de búsqueda tal como esta:

.Velero y Bergantín < 4

se ha obtenido una lista de textos que cumplen esta frase, por ejemplo:

Texto 1

Texto 2

.....

Texto n

y llamamos a esta lista **Lista1.** Imaginemos además que se tienen otras listas, obtenidas de la misma manera, con la misma u otras frases de búsqueda distintas, aplicadas al

mismo u otro grupo de textos, y que hemos llamado **Lista2, Lista3, etc.** Se puede ahora realizar los siguientes trabajos:

Gestionar Listas

Vol. 5 Año 1996
ISSN: 1575-9733

1. Obtener la lista suma de las dos **(OR)**

2. Obtener la lista producto **(AND)**

3. Obtener la lista **EXCLUSIVE OR**

4. Obtener la lista **AND NOT** (Textos de la primera que no están en la segunda)

5. Obtener la lista inversa **(NOT)** (Lista con los textos que no figuran en la

original)

6. Añadir y quitar manualmente elementos a una lista

7. Calcular la correlación entre listas

Los 5 primeros puntos permiten obtener una **nueva lista** cuyo significado práctico será interesante estudiar, sobre todo en los casos de EX OR y AND NOT que son menos intuitivos.

El punto 6 permite "retocar" una lista, depurándola manualmente, lo que sí parece tener sentido práctico De paso, hay que decir que es posible **duplicar** una lista, por ejemplo para hacer operaciones con la lista duplicada sin modificar la original, sin más que realizar cualquiera de las 3 primeras operaciones (OR AND y EX OR) con una lista sobre ella misma.

El punto 7 (correlación entre listas) permite calcular si dos listas de textos contienen información referente a un mismo tema o a temas relacionados.

Hay que recordar que las listas, se han obtenido como resultado de la aplicación de dos frases de búsqueda a dos grupos de textos. Podemos pensar en la misma frase aplicada por separado a cada uno de los capítulos de un libro, a cada una de las páginas, o incluso a partes de texto (utilizando en la frase los modificadores EN LÍNEA, EN PÁRRAFO, DESDE LÍNEA o EN COLUMNA), y sin olvidar que hemos llamado **texto** a un conjunto arbitrario de información dentro de un libro, que hemos podido seleccionar previamente de forma adecuada.

Podemos también imaginar que hemos obtenido las listas por la aplicación de dos frases a un mismo texto. En ambos casos se abren posibilidades interesantes.

Correlación entre listas

El proceso de calcular la correlación entre dos listas se realiza en varias etapas. En primer lugar se localizan los textos comunes (repetidos), se calcula su número y se **eliminan** de ambas listas para su gestión en las etapas posteriores. Se obtienen ya unos primeros datos:

nº claves comunes nº claves comunes

---------- *100% ----------*100%

nº claves lista1 nº claves lista2

Vol. 5 Año 1996
ISSN: 1575-9733

nº textos comunes nº textos comunes

---------- * 100% ---------- * 100%

nº textos lista1 nº textos lista2

En caso de que uno de estos 4 datos sea 100%, se detiene el análisis, ya que significa que una lista **contiene** a la otra, y el problema de correlación está solucionado.

En la segunda etapa del análisis, y ya eliminadas de ambas listas los textos comunes, se obtiene la relación de claves **comunes** a ambas listas, ordenada por la suma de frecuencias

Frecuencias

Clave Lista1 Lista2 Suma

ClaveX1 20 14 34

ClaveX2 12 21 33

ClaveX3 17 10 27

ClaveX 1 1 2

NO se incluyen en la relación las claves que están en una sola lista, aunque su frecuencia en ella sea significativa.

ClaveY 40 0 40

Se realiza ahora un recorte de los datos, dejando exclusivamente los 50 primeros elementos de la relación, ya que, a efectos prácticos el resto de elementos es irrelevante.

Por último, se calcula un cierto coeficiente de correlación tal como veremos más adelante y el resultado se presenta como ÍNDICE DE SIMILITUD.

Incluir las Listas como Parte de la Frase de Búsqueda

Podríamos aumentar las posibilidades de la frase de búsqueda permitiendo la inclusión de una lista de textos ya obtenida por una frase de búsqueda anterior y guardada en un fichero de macros con el nombre correspondiente, por ejemplo: **Lista1**.

La frase de búsqueda podría presentar este aspecto:

.(Lista1) y Bergantín o (Lista2)

Vol. 5 Año 1996
ISSN: 1575-9733

Así como la palabra Bergantín representa a todos los textos que contengan esa clave dentro de ellos, la palabra **Lista1**, escrita entre paréntesis representa directamente todos los textos de esa lista. En ambos casos se debe cumplir el resto de requisitos de la frase (y, o, etc.) para obtener la lista final de textos.

Trabajando con listas dentro de la frase de búsqueda hay ciertas opciones que están prohibidas por la lógica.

Por ejemplo no se puede escribir: **.(Lista1) enlinea nombre** ya que la lista1 no representa a claves que están localizables en una línea concreta del texto sino textos completos. Tampoco se pueden utilizar los modificadores: **< > encolumna desdelínea hastalínea,** es decir todos los modificadores posicionales ya que se refieren a la posición de claves dentro del texto.

El resultado de la búsqueda puede ser a su vez guardado en un macro **(listaX)**, para poder ser utilizado en otra búsqueda posterior , y así sucesivamente.

Comentario

Este enfoque es esencialmente diferente de lo que se ha llamado búsqueda iterativa. La búsqueda iterativa actúa sobre el resultado de una primera búsqueda **en el mismo instante**, y no permite ni operar con las listas obtenidas ni incluirlas en otra frase de búsqueda posterior.

Creación Automática de Macros

Hasta ahora es posible guardar en un macro el resultado de la frase de búsqueda una vez se ha obtenido la lista de textos que cumplen los requisitos pedidos en la frase.

Si sabemos de antemano el nombre del macro con el que queremos guardar el resultado de la búsqueda, podemos incluirlo dentro de la frase, escrito entre paréntesis, de forma que no sea necesario preguntar antes de la presentación de los textos, sino que lo grabe directamente.

La frase de búsqueda pues, podría quedar en esquema así:

.Velero y Bergantín < 4 (macro1)

Al estar el nombre del macro escrito entre paréntesis, el programa lo detecta antes de iniciar la búsqueda, eliminándolo de la frase y guardándolo para su posterior utilización. Se comprende por tanto que podemos escribir el nombre del macro en cualquier punto intermedio de la frase de búsqueda. Por ejemplo:

.Inicio de la frase... (macro1) fin de la frase...

<u>Iteración automática</u>

Otra idea de fácil implementación en el programa es la de enlazar dos o más búsquedas cada una de las cuales se aplique sobre la lista resultado de la búsqueda anterior.

Vol. 5 Año 1996
ISSN: 1575-9733

Esto amplía el concepto de iteración, que hasta ahora, en el programa, solo contemplaba una nueva clave de búsqueda, a toda una frase con las mismas posibilidades que la frase original.

La separación entre frases se realiza a través del símbolo >> y puede utilizarse tantas veces como se quiera y por supuesto combinándolo con la creación automática de macros.

La frase de búsqueda podría quedar también en esquema, así:

.frase1 (macro1) >> frase2 >> frase3 (macro2)

Ya se ha incluido en la próxima versión del programa todo este trabajo sobre listas, macros e iteración automática, y parece que funcionan bien y que podrían ser interesantes.

SIMILITUD ENTRE DOS TEXTOS

Definiremos la similitud como el "parecido>" existente entre dos informaciones dadas.

Esta idea de partida, tan amplia, se irá concretando al ir definiendo métodos para calcular un "**coeficiente de similitud**".

Llamaremos aquí "texto" a un conjunto de textos (según mi modelo de libro-texto) obtenido por selección manual o como resultado de una frase de búsqueda sobre un "libro completo".

También será un "texto" el propio libro completo o una frase de búsqueda, previamente grabada como texto, si se desea. En particular un "texto" puede asimilarse al contenido de una "LISTA" tal como la he definido en mi modelo.

Supondremos en todos los casos la indexación de texto completo, exceptuando siempre las palabras vacías, aunque la idea sirve igualmente para el caso de utilización de descriptores.

En primer lugar, y partiendo del texto con menor número de claves, mediante un algoritmo relativamente sencillo, obtendremos una lista de claves comunes ordenada de mayor a menor por la SUMA de frecuencias en ambos textos.

No se incluyen en la lista claves con frecuencia cero (inexistentes) en alguno de los dos textos. Tampoco se realiza la comparación si los dos textos se definen como "idénticos".

TABLA 1:

	Pi	Qi	Si	|Di|
1 - velero	10	6	16	4
2 - barco	7	7	14	0
3 - mar	9	5	14	4
4 - agua	2	10	12	8
5 - ola	11	1	12	10
6 - vela	4	3	7	1
SUMAS	43	32	75	11

Vol. 5 Año 1996
ISSN: 1575-9733

Esta lista estás ordenada de mayor a menor por la columna suma (Si), y en la columna diferencia |Di| se ha escrito el valor absoluto de la diferencia. Los datos (Pi), (Qi), son las frecuencias en cada una de los dos textos a comparar.

Con estos datos y antes de avanzar más, cabe pensar en las siguientes hipótesis:

1. Todos los valores |Di| no tiene el mismo peso. Este será función directa de Si.

1. Cada tupla (par de claves) debe aportar al coeficiente un valor comprendido entre 1 y Épsilon (Épsilon pequeño y mayor que cero).

1. El máximo aporte para N tuplas debe ser 100% (N=6 está en la lista).

1. La función: **Similitud = f(Si, |Di, N)** debe incluir también:
   a - Los textos rechazados por idénticos (con Ti claves)
   b - Las clases en un solo texto (To)

1. La función de similitud no debe ser lineal, ya que el ruido producido por las claves aleatorias coincidentes debe rechazarse en lo posible.

Rehagamos nuestra lista con una lista de casos secuencial que hará el valor de "similitud".

TABLA 2

Pi	Qi	d/s	1 - d/s	$(1 - d/s)^2$	$(1 - d/s)^{1/2}$
10	10	0	1	1	1
10	9	0.05	0.95	0.90	0.97
10	8	0.11	0.89	0.79	0.94
10	7	0.17	0.83	0.68	0.91
10	6	0.25	0.75	0.56	0.86
10	5	0.33	0.67	0.45	0.81
10	4	0.42	0.58	0.33	0.76
10	3	0.53	0.47	0.22	0.68
10	2	0.66	0.34	0.11	0.58
10	1	0.81	0.19	0.03	0.43

En principio parece que la progresión de la lista (1 - d/s) coincide con lo que podríamos apreciar subjetivamente, y que cualquier otra transformación (relación de cuadros o de raíces) nos aleja de esa apreciación.

Así pues me inclino a escoger el término:

$S^{n}_{i=1} (1 - |Di| / Si)$

Vol. 5 Año 1996
ISSN: 1575-9733

siendo n el número de claves de la lista (6 para el primer ejemplo y 10 para este último).

Como cada coeficiente va, por definición de 1 a e > 0, solo habrá que dividir por n y multiplicar por 100 para obtener la primera aproximación. (C = coeficiente inicial)

$C = 100 / n \; S^{n}_{i=1} (1 - |Di| / Si)$

Consideremos por último el caso de que existan en ambas listas textos idénticos (con C claves) claves únicas (To: claves de un solo texto). El número total de claves será:

Tt = C + To + n ----> Tt - C = n + To

y por tanto el coeficiente de similitud tendrá que venir afectado de estos factores, en la forma:

SM = C/T + n/T * f()

que cumple las condiciones de contorno, es decir:

1. para TODAS las claves comparadas Ti = 0 y To = 0 es por tanto Tt = n y el Coeficiente = 1
2. n nunca será mayor que Tt - Ti

Por tanto definiremos el Coeficiente de similitud como:

$Cs = C/T * 100/n * S^{n}_{i=1} (1 - |Di| / Si)$

En esta fórmula hemos aplicado el coeficiente (n/T) a la función de comparación y por tanto se pierde el ranking que ayuda a la identificación del coeficiente (Cs), es decir a la lista ordenada de claves, ya que en ella no estarán incluidas las claves de los textos comunes.

Esto resulta irrelevante para una búsqueda por frase natural pero es significativo para una comparación entre listas de textos.

Ahora se entiende la falta de concreción en la definición de similitud, ya obviamente el concepto "similitud" no puede reflejarse en una fórmula matemática única e incontrovertible. **Muy al contrario, quizás ésta u otra aproximación "pueden definir" el concepto de similitud**.

Consideremos por último, que por una parte, es necesario que el algoritmo presente al menos las primeras tuplas de la serie para saber "en que" son similares.

Por otra parte, solo un largo y metódico estudio sobre textos conocidos, puede dar los umbrales de "similitud práctica" y por tanto evaluar con mayor aproximación el tipo de progresión óptimo (lineal, cuadratico, etc.).

No obstante, parece una aproximación aceptable al problema de interrogar a un sistema informático con una frase en puro lenguaje natural, con la ayuda de un fichero de sinónimos y otro de palabras vacías.

Vol. 5 Año 1996
ISSN: 1575-9733

CONCLUSIONES

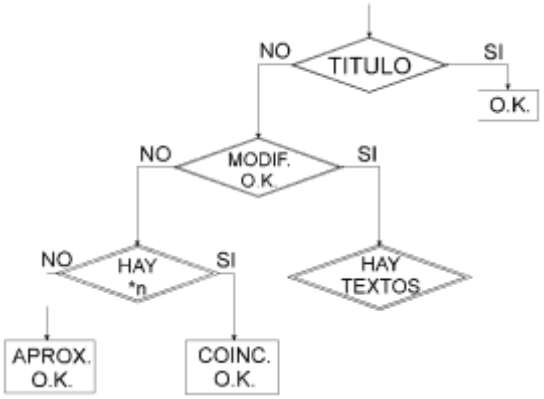
En conclusión, un sistema de recuperación documental debería ser completo y coherente, con unas reglas claras que abarcasen el mayor número posible de supuestos de búsqueda.

La combinación de todas las posibilidades descritas, y la forma definida de expresarlas en un lenguaje sencillo, constituyen en sí una gramática simple de interrogación que cubre los objetivos propuestos.

Nuestro modelo asume el carácter probabilístico de la información documental y proporciona medios para representar necesidades de información, compararlas con documentos y obtener aquellos que presentan una mayor probabilidad de resultar relevantes para solucionar la necesidad de información que ha expresado el usuario.

GENERALIZACIÓN DE LA FRASE

No se escribe el símbolo



BÚSQUEDAS

Búsqueda primaria: (Frase de interrogación)

1. Búsqueda simple
2. Búsqueda por uso
3. Búsqueda aproximada
4. Búsqueda por coincidencias

Búsqueda secundaria (Textos ya localizados)

1. Por hipertexto y thesauro
2. Por relevancia / Ranking
3. Iterativa

Otras búsquedas

1. Difusa
2. Por líneas
3. Por similitud