

Grado en Ingeniería Informática

Explotación de la Información *Módulo 2. Recuperación de Información*

Antonio Ferrández Rodríguez



UNIVERSIDAD DE ALICANTE



Grupo de Procesamiento
del Lenguaje y Sistemas
de Información

Índice

Explotación de la información. Recuperación de Información

1. Introducción
2. Arquitectura de los sistemas de RI
3. Fase de recopilación de documentos
4. Fase de indexación
5. Fase de búsqueda de información
6. Fase de presentación de resultados
7. Evaluación de los sistemas de RI
8. Sistemas de *Question Answering* o Búsqueda de Respuestas
9. Sistemas de RI multimedia



1. Introducción

La Recuperación de Información o *Information Retrieval*:

- Es la ciencia de la **búsqueda** de información en **documentos** electrónicos y cualquier tipo de **colección documental digital**, como objetivo realiza la recuperación en **textos, imágenes, sonido** o datos de otras características
- Es un estudio interdisciplinario:
 - Psicología cognitiva, arquitectura de la información, diseño de la información, inteligencia artificial, lingüística, semiótica, informática, biblioteconomía, archivística, documentación...

3



1. Introducción

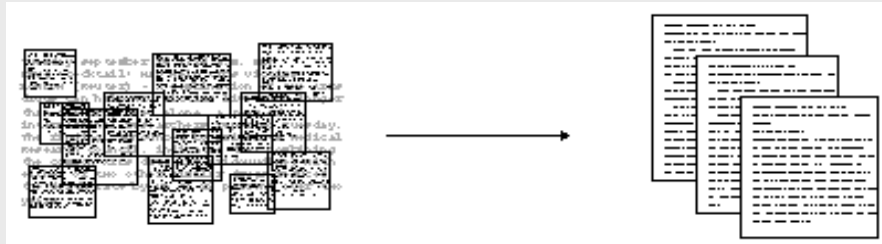
Los buscadores de información:

- Entrada: palabras clave a buscar
- Salida: relación de documentos ordenada
- Ejemplos: Altavista, Lycos, Google, Bing, etc.
- Ventajas: gran rapidez de funcionamiento
- Problemas: precisión baja

4



1. Introducción



2. Arquitectura de los sist. de RI

Módulos o Fases:

1. Módulo de recopilación de documentos (*web crawler* o *web spider* o araña web)
2. Módulo de indexación
3. Módulo de búsqueda de información
4. Módulo de presentación de resultados

3. Fase de recopilación de documentos

Arquitectura centralizada:

- ▀ Los tres primeros módulos se ejecutan localmente
- ▀ Presenta problemas de cobertura dado el crecimiento exponencial de la Web

Arquitectura distribuida

- ▀ El *crawler* se divide en dos componentes:
 - ▀ *gatherer*: recolecta periódicamente páginas web
 - ▀ *broker*: realiza el proceso de indexación
- ▀ Ventaja: eficiencia y cobertura
- ▀ Desventaja: complejidad de funcionamiento y sincronización

7

3. Fase de recopilación de documentos

Recorrido de la Web:

- ▀ Se comienza por un conjunto de URL conocidas
- ▀ Se continúa por las URL que salgan de las anteriores:
 - ▀ Recorrido en profundidad o en anchura de los enlaces
- ▀ Problemas a tener en cuenta:
 - ▀ Evitar repeticiones de páginas visitadas y sincronización entre diferentes crawlers
 - ▀ Aprendizaje de la frecuencia de actualización de las páginas
 - ▀ Evitar reindexar páginas no modificadas para mejorar la eficiencia

8

3. Fase de recopilación de documentos

Características que debe tener un crawler:

- Robustez: debe ser capaz de evitar *spider traps*
- *Politeness*: respetar cuotas de acceso de los servidores, abrir solo una conexión con el servidor, esperar un nº de segundos antes de cada nueva petición, etc.
- Ejecución distribuida
- Escalable: debe permitir ampliar el sistema añadiendo ancho de banda, ordenadores de bajada, formatos de bajada, protocolos de comunicación, etc.
- Eficiencia
- Calidad de las páginas bajadas
- *Freshness*: ser capaz de detectar nuevas versiones de los documentos

9

4. Fase de indexación

Características generales:

- Off-line
- Técnicas que permitan la fácil incorporación de nuevos documentos a los ya almacenados
- Rapidez de indexación
- Módulos:
 - Segmentación de palabras
 - Filtrado de palabras
 - Almacenamiento de términos con contenido

10



4. Fase de indexación. Segmentación de palabras

Módulo de segmentación de palabras:

- Fácil para algunos idiomas. Difícil para otros (chino).
- Dificultad en el tratamiento de las multipalabras:
 - MS-DOS: ¿se almacena junto? ¿las palabras individuales por separado también? ¿con guión?
 - # Lo habitual: se almacenan todas las combinaciones.
 - Tratamiento especial de /: OS/2 high/low
 - Tratamiento especial de las URL, e-mails, ...

11



4. Fase de indexación. Segmentación de palabras

- Dificultad en el tratamiento de las multipalabras (cont):
 - Tratamiento de las abreviaturas: U.S.A.
 - # Problema de la separación porque en algunos sistemas se eliminan las palabras formadas por una sola letra.
 - Paréntesis de los encabezados de página: 501(c)(3)
 - Apóstrofes: *can't* *I'll* ≠ *ill* *it's* ≠ *its*
 - Números: *.103* ≠ *103* *1,998* (número) ≠ *1998* (fecha)

12

4. Fase de indexación. Cálculo de complejidades

- # **Cálculo de complejidad: determinación de dos parámetros o funciones de coste:**
 - **Complejidad espacial:** Cantidad de recursos espaciales (de almacén) que un algoritmo consume o necesita para su ejecución
 - **Complejidad temporal:** Cantidad de tiempo que un algoritmo necesita para su ejecución
- # **Posibilidad de hacer**
 - Valoraciones
 - El algoritmo es: “bueno”, “el mejor”, “prohibitivo”
 - Comparaciones
 - El algoritmo A es mejor que el B

13

4. Fase de indexación. Cálculo de complejidades

- # **Factores de complejidad temporal:**
 - Externos
 - La máquina en la que se va a ejecutar
 - El compilador: variables y modelo de memoria
 - La experiencia del programador
 - Internos
 - El número de instrucciones asociadas al algoritmo
- # **Complejidad temporal : $Tiempo(A) = C + f(T)$**
 - C es la contribución de los factores externos (constante)
 - $f(T)$ es una función que depende de T (talla o tamaño del problema)

14

Explotación de la información. Recuperación de Información

4. Fase de indexación.

Cálculo de complejidades

Talla o tamaño de un problema:

- ▀ Valor o conjunto de valores asociados a la **entrada** del problema que representa una medida de su tamaño respecto de otras entradas posibles

Paso de programa:

- ▀ Secuencia de operaciones con contenido semántico cuyo coste es **independiente** de la talla del problema
- ▀ Unidad de medida de la complejidad de un algoritmo

Expresión de la complejidad temporal:

- ▀ Función que expresa el número de pasos de programa que un algoritmo necesita ejecutar para cualquier entrada posible (para cualquier talla posible)
- ▀ No se tienen en cuenta los factores externos

15

Explotación de la información. Recuperación de Información

4. Fase de indexación.

Cálculo de complejidades

Ejemplos de cálculo de complejidades:

```
int ejemplo1 (int n)
{
  n+ = n;
  return n;
}
```

$f(\text{ejemplo1}) = 1 \text{ pasos}$

```
int ejemplo2 (int n)
{
  int i;
  for (i=0; i ≤ 2000; i++)
    n+ = n;
  return n;
}
```

$f(\text{ejemplo2}) = 1 \text{ pasos}$

Complejidad
Temporal =
 $C + 1$

16

4. Fase de indexación. Cálculo de complejidades

Ejemplos de cálculo de complejidades (cont.):

```
int ejemplo3 (int n)
{
    int i, j;
    j = 2;
    for (i=0; i ≤ 2000; i++)
        j=j*j;
    for (i=0; i ≤ n; i++)
    {
        j = j + j;
        j = j - 2;
    }
    return j;
}
```

$$f(\text{ejemplo3}) = 1 + 1 \cdot (n + 1) \text{ pasos}$$

4. Fase de indexación. Cálculo de complejidades

Ejemplos de cálculo de complejidades (cont.):

```
int ejemplo4 (int n)
{
    int i, j, k;
    k = 1;
    for (i=0; i ≤ n; i++)
        for (j=1; j ≤ n; j++)
            k = k + k;
    return k;
}
```


$$f(\text{ejemplo4}) = 1 + 1 \cdot n \cdot (n + 1) \text{ pasos}$$

```
int ejemplo5 (int n)
{
    int i, j, k;
    k = 1;
    for (i=0; i ≤ n; i++)
        for (j=i; j ≤ n; j++)
            k = k + k;
    return k;
}
```

$$f(\text{ejemplo5}) = 1 + \sum_{i=0..n} (\sum_{j=i..n} 1) \text{ pasos}$$

$$\sum_{i=1}^n i = \frac{n(n+1)}{2}$$

$$\sum_{j=i..n} 1 = (n-i+1) \cdot 1 = n-i+1$$



Explotación de la información. Recuperación de Información

4. Fase de indexación.


Cálculo de complejidades

Dado un vector X de n números naturales y dado un número natural z :

```
funcion BUSCAR (var X:vector[N]; z: N): devuelve N
var i:natural fvar;
comienzo
  i:=1;
  mientras (i ≤ |X|) ∧ (Xi≠z) hacer
    i:=i+1;
  fmientras
  si i= |X|+1 entonces devuelve 0    (*No encontrado*)
  si_no devuelve i
fin
```

- No podemos contar el número de pasos porque depende:
 - # del tamaño del problema $|X|$
 - # de la instancia del problema que se pretende resolver (posible valor que puedan tomar las variables de entrada)

19



Explotación de la información. Recuperación de Información

4. Fase de indexación.

Cálculo de complejidades

```
funcion BUSCAR (var X:vector[N]; z: N): devuelve N
var i:natural fvar;
comienzo
  i:=1;
  mientras (i ≤ |X|) ∧ (Xi≠z) hacer
    i:=i+1;
  fmientras
  si i= |X|+1 entonces devuelve 0    (*No encontrado*)
  si_no devuelve i
fin
```

X	z	Nº PASOS
(0, 1)	1	1 + 1 = 2
(1, 2, 3)	1	1 + 0 = 1
(2)	3	1 + 1 = 2
(1, 0, 2, 4)	3	1 + 4 = 5
(1, 0, 2, 4)	0	1 + 1 = 2
(1, 0, 2, 4)	1	1 + 0 = 1

20



4. Fase de indexación. Cálculo de complejidades

Cotas de complejidad:

- Cuando aparecen diferentes casos para una misma talla genérica n :
 - **Caso peor**: cota superior del algoritmo $\rightarrow O(n)$
 - **Caso mejor**: cota inferior del algoritmo $\rightarrow \Omega(n)$
 - **Término medio**: cota promedio $\rightarrow \Theta(n)$
- Todas son funciones del tamaño del problema (n)
- La cota promedio es difícil de evaluar *a priori*
 - Es necesario conocer la distribución de la probabilidad de entrada
 - No es la media de la inferior y de la superior (ni están todas ni tienen la misma proporción)

21




4. Fase de indexación. Cálculo de complejidades

- Talla del problema: n° de elementos de X : n
- Caso mejor: el elemento está el primero: $X_1=z \rightarrow \Omega(n) = 1$
- Caso peor: el elemento no está: $\forall i 1 \leq i \leq |X|, X_i \neq z \rightarrow O(n) = n+1$

```

funcion BUSCAR (var X:vector[N]; z: N): devuelve N
var i:natural fvar;
comienzo
  i:=1;
  mientras (i ≤ |X|) ∧ (Xi≠z) hacer
    i:=i+1;
  fmientras
  si i= |X|+1 entonces devuelve 0 (*No encontrado*)
  si_no devuelve i
fin
  
```

22



Explotación de la información. Recuperación de Información


4. Fase de indexación.

Cálculo de complejidades

$$O(1) \subset O(\lg \lg n) \subset O(\lg n) \subset O(\lg^{a>1} n) \subset O(\sqrt{n}) \subset O(n) \subset O(n \lg n) \subset O(n^2) \subset \dots \subset O(n^{a>2}) \subset O(2^n) \subset O(n!) \subset O(n^n)$$

- $f(n) + g(n) + t(n) \in O(\text{Max}(f(n), g(n), t(n)))$
- Ejemplos:
 - $n + 1$ pertenece a $O(n)$
 - $n^2 + \log n$ pertenece a $O(n^2)$
 - $n^3 + 2^n + n \log n$ pertenece a $O(2^n)$
 - $(10^6 n) \in O(n) \quad \subset \quad (n^2) \in O(n^2)$
⚡ Aunque $10^6 n$ es menor que n^2 sólo cuando $n > 10^6$.

23



Explotación de la información. Recuperación de Información

4. Fase de indexación.

Cálculo de complejidades

Complejidad	n = 32	n = 64
n^3	3 seg.	26 seg.
2^n	5 días	$58 \cdot 10^6$ años

- Tiempos de respuesta para dos valores de la talla y complejidades n^3 y 2^n .
(paso = 0'1 mseg.)

```

función POT_2 (n: natural): natural
  opción
    n = 1: devuelve 2
    n > 1: devuelve 2 * POT_2(n-1)
  fopción
ffunción

```

Coste lineal

1 seg.

```

función POT_2 (n: natural): natural
  opción
    n = 1: devuelve 2
    n > 1: devuelve POT_2(n-1)+POT_2(n-1)
  fopción
ffunción

```

Coste exponencial

(2^n)

miles de años

24

4. Fase de indexación. Cálculo de complejidades

Búsqueda de un elemento en un vector ordenado (Búsqueda binaria)

```

funcion BUSCA (var v:vector[N]; x,pri,ult: natural): natural
var m: natural fvar
comienzo
  repetir
    m:= (pri+ult)/2
    si v[m]>x entonces ult:= m-1
    sino pri:= m+1
  fsi
hasta (pri>ult) ∨ v[m]=x
si v[m]=x entonces devuelve m
sino devuelve 0
fsi
fin
  
```

25

4. Fase de indexación. Cálculo de complejidades

- Determinar la talla del problema: $n = \text{tamaño del vector}$
- Mejor caso: x está en la mitad del vector
- Peor caso: x no está en el vector
- Complejidades
 - mejor caso: $1 + 1 = 2 \in \Omega(1)$
 - peor caso
 - ‡ $1 + k \cdot 1$, donde k es el n° de veces que se ejecuta el bucle
 - 1ª iteración: Talla = n
 - 2ª iteración: Talla = $n/2$
 - 3ª iteración: Talla = $n/4$
 -
 - k -ésima iteración: Talla = $n/2^{(k-1)}$
 -
 - última iteración: Talla = 1 ($n/2^{(\text{última}-1)} = 1$)
 - ‡ Es decir, en la última iteración sólo nos queda 1 elemento Despejando última:
 - $n = 2^{\text{última}-1}$; $\log_2 n + 1 = \text{última}$;
 - $O(\sum_{i=1}^{\text{última}} 1) = O(\text{última}) = O(\log_2 n)$

26



4. Fase de indexación. Cálculo de complejidades

Algoritmo:

```
void Tokenizar(const string& str, list<string>& tokens, const
string& delimiters) {
    string::size_type lastPos = str.find_first_not_of(delimiters,0);
    string::size_type pos = str.find_first_of(delimiters,lastPos);

    while(string::npos != pos || string::npos != lastPos)
    {
        tokens.push_back(str.substr(lastPos, pos - lastPos));
        lastPos = str.find_first_not_of(delimiters, pos);
        pos = str.find_first_of(delimiters, lastPos);
    }
    return;
}
...
Tokenizar(fecha, lfecha, "[:/].");
```

27



4. Fase de indexación. Cálculo de complejidades

Ejercicio 1:

- Calcular la complejidad temporal del tokenizador anterior en función de:
 - n: longitud de *str*
 - d: longitud de *delimiters*
 - find_first_not_of() : complejidad $O(n*d)$
 - push_back: $O(1)$
 - substr(lastPos, pos - lastPos): $O(pos - lastPos)$
- Añadir al algoritmo anterior las capacidades de detección y tratamiento de:
 - Guiones: MS-DOS OS/2 high/low
 - URL, e-mails
 - Detección de abreviaturas (U.S.A)
 - Apóstrofes como parte de palabras (can't, it's)
 - Números decimales: .103 5,6 1,000,000
 - Porcentajes: 55 % 75%
 - Cantidades monetarias: 103\$ 55€
 - Fechas: 16/12/2012, 2011/12/1
- Calcular la complejidad temporal de este nuevo algoritmo

28

4. Fase de indexación. Módulo de filtrado

Módulo de filtrado de palabras:

- Se almacena todo en minúsculas. Problemas:
 - Puede interesar distinguir entre mayúsculas y minúsc.:
 - # *Bill* nombre propio – *bill* nombre común (factura)
 - # *CYCLing* nombre propio (congreso) – verbo en gerundio
- No se almacenan vocales acentuadas
- Eliminar puntos: U.S.A. como USA
- Eliminar guiones: anti-discriminatorio como antidiscriminatorio
- Se realiza un tratamiento de los caracteres especiales de cada idioma: por ejemplo la “ñ”

29

The screenshot shows a Google search interface with the query 'caña' entered in the search bar. The search results are displayed in Spanish. The first result is 'Asociación de cultivadores de caña de azúcar de Colombia', followed by 'La Caña. Revista de flamenco', 'HACIENDA PIEDECHINCHÉ Museo de la Caña de Azúcar', and 'Punta Cana'.

Google [Búsqueda Avanzada](#) [Preferencias](#) [Herramientas del idioma](#)

caña

☐ Buscar en la Web ☒ Buscar sólo páginas en español

Sugerencia: En la mayoría de los navegadores basta con pulsar la tecla **Enter**

[La Web](#) [Imágenes](#) [Grupos](#) [Directorio](#)

Se buscaron páginas en **español** que contienen **caña**. Resultados **1**

[Asociación de cultivadores de caña de azúcar de Colombia](#)
Noticias de Interés, Reservas a negociaciones entre CAN y Mercosur, En un documento enviado ...
Descripción: Asociación de Cultivadores de **Caña** de Azúcar de Colombia.
Categoría: [World > Español > ... > Colombia > Economía y negocios > Asociaciones y cámaras](#)
www.asocana.com.co/ - 59k - [En caché](#) - [Páginas similares](#)

[La Caña. Revista de flamenco](#)
... El contenido de La **Caña** viene organizándose según unas secciones fijas que quieren abarcar toda la vida flamenca: ...
www.flamenco-world.com/magazine/cana/cana.htm - 17k - [En caché](#) - [Páginas similares](#)

[HACIENDA PIEDECHINCHÉ Museo de la Caña de Azúcar](#)
Museo de la **Caña** de Azúcar. El museo de la **caña** es una exposición de documentos culturales ...
www.telesat.com.co/telesat/sitios/museo.html - 4k - [En caché](#) - [Páginas similares](#)

[Punta Cana](#)
... Selecciona aquí para ampliar la imagen. ... Selecciona aquí para ampliar la imagen Vista aérea de Punta **Can**a Michael Friedel © Sectur. ...
www.reddominicana.com/puntacana/ - 33k - [En caché](#) - [Páginas similares](#)

30

Explotación de la información. Recuperación de Información

Google [Búsqueda Avanzada](#) [Preferencias](#) [Herramientas del idioma](#)

+caña

☐ Buscar en la Web ☒ Buscar sólo páginas en español

[La Web](#) [Imágenes](#) [Grupos](#) [Directorio](#)

Se buscaron páginas en **español** que contienen **+caña**. Resultados 1

[Asociación de cultivadores de caña de azúcar de Colombia](#)
 Noticias de Interés, Reservas a negociaciones entre
 CAN y Mercosur, En un documento enviado ...
 Descripción: Asociación de Cultivadores de **Caña** de Azúcar de Colombia.
 Categoría: World > Español > ... > Colombia > Economía y negocios > Asociaciones y cámaras
www.asocana.com.co/ - 59k - [En caché](#) - [Páginas similares](#)

[La Caña: Revista de flamenco](#)
 ... El contenido de La **Caña** viene organizándose según unas secciones
 fijas que quieren abarcar toda la vida flamenca: ...
www.flamenco-world.com/magazine/cana/cana.htm - 17k - [En caché](#) - [Páginas similares](#)

[HACIENDA PIEDECHINCHE Museo de la Caña de Azucar](#)
 Museo de la **Caña** de Azucar. El museo de la **caña** es
 una exposición de documentos culturales ...
www.telesat.com.co/telesat/sitios/museo.html - 4k - [En caché](#) - [Páginas similares](#)

31

Explotación de la información. Recuperación de Información

4. Fase de indexación. Módulo de filtrado

- Se eliminan las palabras sin contenido (superfluas): palabras de parada (*stopwords*).
- Métodos:
 - Crear listas “fijas” de palabras para cada idioma:
 - ‡ Inglés: a, cannot, into, our, thus, about, co, is, ours, to, above, could, it, ourselves, together, ...
 - ‡ Castellano: él, éstos, última, últimas, aún, actualmente, adelante, además, ahí, ahora, de, ...
 - Crear listas de categorías léxicas: (preposiciones, artículos, conjunciones, pronombres, adverbios).
 - Añadir *meta-stopwords*: “**encuentre** aquellos **documentos** que **describan...**”
- Problemas a tener en cuenta:
 - ‡ “to be or not to be” (todo eliminado)
 - ‡ *Vitamina A* (se eliminaría A)
 - ‡ ¿*Qué significa ESA?* (se eliminaría **pronombre ESA** en castellano)

32



4. Fase de indexación. Módulo de filtrado

Listas de palabras de parada (inglés):

a about above across actually after again against ago all almost along already also although always among an and another any anyone around as at b back bad because before behind best better between big biggest both but by c cent complete d day down during e each early eight enough entire ep etc even ever every everything f face fact far fell few finally first five for found four from g good got h he held her here him himself his hour hours how however i idea if in including instead into it its itself j k l lack last later least led less little long longer lot m man many matter may me men miles million moment month months more morning most much my n near nearly necessary never night no nor not note nothing now o of off often on once one only or other others our out outside over own p page part past per perhaps place point proved q qm question r really recent recently reported round s same sec second section sense seven she short should showed since single six small so some soon still such t ten text than that the their themselves then there these they thing things third this those though thought thousands three through thus time tiny to today together too took toward two u under until up upon us v very w warning way we week weeks well went were what when where whether which while who whom whose why will with without word words would x y year years yet you your z

33



4. Fase de indexación. Módulo de filtrado

Listas de palabras de parada (español):

él ésta éstas éste éstos última últimas último últimos a añadió aún actualmente adelante además afirmó agregó ahí ahora al algún algo alguna algunas alguno algunos alrededor ambos ante anterior antes apenas aproximadamente aquí así aseguró aunque ayer bajo bien buen buena buenas bueno buenos cómo cada casi cerca cierto cinco comentó como con conocer consideró considera contra cosas creo cual cuales cualquier cuando cuanto cuatro cuenta da dado dan dar de debe deben debido decir dejó del demás dentro desde después dice dicen dicho dieron diferente diferentes dijeron dijo dio donde dos durante e ejemplo el ella ellas ello ellos embargo en encuentra entonces entre era eran es esa esas ese eso esos está están esta estaba estaban estamos estar estará estas este estos estos estoy estuvo ex existe existen explicó expresó fin fue fuera fueron gran grandes ha había habían haber habrá hace hacen hacer hacerlo hacia haciendo han hasta hay haya he hecho hemos hicieron hizo hoy hubo igual incluso indicó informó junto la lado las le les llegó lleva llevar lo los luego lugar más manera manifestó mayor me mediante mejor mencionó menos mi mientras misma mismas mismo mismos momento mucha muchas mucho muchos muy nada nadie ni ningún ninguna ningunas ninguno ningunos no nos nosotras nosotros nuestra nuestras nuestro nuestros nueva nuevas nuevo nuevos nunca o ocho otra otras otro otros para parece parte partir pasada pasado pero pesar poca pocas poco pocos podemos podrá podrán podría podrían poner por porque posible próximo próximos primer primera primero primeros principalmente propia propias propio propios pudo pueda puede pueden pues qué que quedó queremos quién quien quienes quiere realizó realizado realizar respecto sí sólo se señaló sea sean según segunda segundo seis ser será serán sería si sido siempre siendo siete sigue siguiente sin sino sobre sola solamente solas solo solos son su sus tal también tampoco tan tanto tenía tendrá tendrán tenemos tener tenga tengo tenido tercera tiene tienen toda todas todavía todo todos total tras trata través tres tuvo un una unas uno unos usted va vamos van varias varios veces ver vez y ya yo

34

4. Fase de indexación. Módulo de filtrado

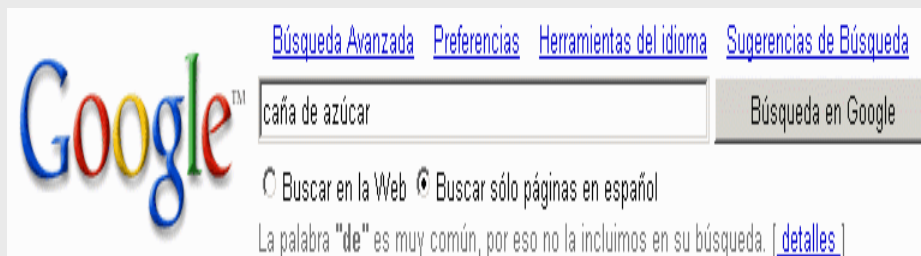
Casos especiales de filtrado:

- Palabras formadas por un solo carácter:
 - Se suelen eliminar, pero es problemático en casos como: *vitamina A*.
- Palabras que aparecen una sola vez en la colección de documentos:
 - Se suelen eliminar:
 - Excepto los números (que son más difíciles de que aparezcan varias veces).

35

4. Fase de indexación. Módulo de filtrado

Ejemplo de palabras de parada:



36

4. Fase de indexación. Módulo de filtrado

Las palabras a almacenar:

- Objetivo: identificar una palabra cuando se escribe de distintas formas.
- Métodos:
 - Tal y como aparecen en el texto.
 - O bien según su forma base o *lema* (análisis morfológico):
 - # *Compute* lema de: [*computing, computed, computes, ...*]
 - # También se pueden tratar derivaciones de los lemas:
 - De un nombre tratar su equivalente adjetivo o verbo:
 - Ejemplos:
 - *Computer, computational, computation, computability, ...*
 - *vendedor, vender, venta*

37

4. Fase de indexación. Módulo de filtrado

Las palabras a almacenar (cont.):

- O bien según su *stem*:
 - Válido para el inglés: no para castellano.
 - Simula un análisis morfológico de forma eficiente.
 - Algoritmos sencillos (Porter, 1980):
 - # No realiza análisis morfológico: elimina prefijos y sufijos.
 - # Quita plurales, -ed, -ing, -ic, -full, -ness, 'y' -> 'i', etc.
 - # Ejemplos: query – queri, queries – queri, asked – ask, friends – friend, beautiful – beauti, beauty – beauti, stemming – stem, ...

38

4. Fase de indexación. Módulo de filtrado

Algoritmo de stemming de Porter:

www.cs.cmu.edu/~callan/Teaching/porter.c

```
static void strip_prefixes ( char *string ) {
    static char *prefixes[] = { "kilo", "micro", "milli", "intra", "ultra",
                                "mega", "nano", "pico", "pseudo", 0 };
    for ( i=0 ; prefixes[i] != 0 ; i++ ) { ... }
}
static void strip_suffixes ( char *string ) {
    if ( (has_suffix(string,"sses",stem) == TRUE) ||
          (has_suffix(string,"ies",stem) == TRUE ) )
        ...
    if ( has_suffix(string,"eed",stem) == TRUE ) { ...
    }
```

39

4. Fase de indexación. Módulo de filtrado

Algoritmo de stemming de Porter. Problemas:

- ▣ Devuelven palabras difíciles de entender para el usuario: *iteration* → *iter*.
- ▣ En ocasiones agrupa palabras que conceptualmente no tienen nada que ver (*overstemming*):
 - ▣ [*organization*] → *organ*
 - ▣ [*university, universe*] → *univers*
 - ▣ [*performed* (realizar, representar), *performance* (representación), *performer* (actor), *performative* (performativo), *performing* (teatral)] → *perform*

40

4. Fase de indexación. Módulo de filtrado

Algoritmo de stemming de Porter. Problemas (cont.):

- En ocasiones NO agrupa palabras que conceptualmente están relacionadas (*understemming*):
 - [Acquire, acquiring, acquired] → *acquir*
 - [Acquisition] → *acquisi*

41

4. Fase de indexación. Módulo de filtrado

Ejercicio 2:

- Archivo:
 - Ex.Inf. Modulo 2. Recuperacion de Informacion. Ejercicio2.pdf
- De los siguientes documentos, calcular el conjunto de términos que superarían la fase de filtrado.
- Analiza el proceso de stemming: errores vs. aciertos, stemming vs. lematización

42

4. Fase de indexación. Módulo de almacenamiento.

Módulo de almacenamiento de términos con contenido:

- Se almacenan los términos (ya segmentados y normalizados) del módulo de filtrado.
- Momento en el que aplicar esta “normalización”:
 - Fase de indexación:
 - # Almacenar sólo una vez el stem o lema.
 - Fase de búsqueda:
 - # Almacenar la palabra tal cual aparece en el documento
 - # Expandir la pregunta con todas las variantes de cada palabra, y realizar con ellas la búsqueda.
- De estos términos se almacena información adicional que ayude a determinar la relevancia.

43

4. Fase de indexación. Módulo de almacenamiento.

■ Trabajo pendiente:

- Agrupar palabras con significado parecido y distintos lemas o stems (*Church, priest, pilgrim, ...*).
- Sinónimos (*processor – CPU*), hiperónimos (*computer → mainframe*), hipónimos (*computer ← device*), ...
- Multipalabras:
 - # Endógenas: sucesión de términos cuyo significado global es la suma de los significados de dichos términos.
 - # Exógenas: sucesión de términos cuyo significado global difiere de la suma de los significados de sus términos:
 - *Casa Blanca*. Significado endógeno: edificación de color blanco. Significado exógeno: hogar del presidente de EEUU

44

4. Fase de indexación. Módulo de almacenamiento.

Información adicional a almacenar. Indicios de relevancia:

- Información de los términos:
 - Si aparece o no el término en el documento.
 - Posición del término dentro del documento (nº frase, nº palabra, posición en el doc, tipo de letra, etc.)
 - Frecuencia del término (*term frequency*) $ft(t)$.
 - Frecuencia del documento (*document freq.*) $fd(t)$.
- Información del documento:
 - Tamaño del documento (palabras/bytes).
 - **RI en la web:**
 - Estructura HTML: tipos de letra, títulos, texto hiperenlaces, color, ...
 - PageRank:
 - Número de links que apuntan al documento, nº de enlaces que salen del documento, posición dentro de la jerarquía de links.

45

4. Fase de indexación. Módulo de almacenamiento.

Información adicional de los términos:

- Frecuencia del término (*term frequency*) $ft(t)$:
 - Número de veces que aparece t en cada documento.
 - Cuanto más alto sea este valor \Rightarrow más tratará el documento sobre el concepto representado por este término.
 - Se suele aplicar una normalización: $ft(t) / \max_i ft(t_i)$.
- Frecuencia del documento (*document freq.*) $fd(t)$:
 - Número de documentos que contienen a t .
 - Cuanto más bajo sea este valor \Rightarrow más discriminativo será.
 - *Inverse Document Frequency (IDF)*: $\log(N/fd(t))$: cuanto más alto \Rightarrow más discriminativo será
 - modelo TF-IDF: $ft(t) * \log(N/fd(t))$
 - Kaszkiel et al. (1999): $\log_e(ft(t)+1) * \log_e((N/fd(t)) + 1)$

46

4. Fase de indexación. Módulo de almacenamiento.

Algoritmo de extracción de documentos relevantes dado una query q :

- `bool ExisteTerminoEnD (t_i , d_j , informacion_de_t):`
 - # Devuelve cierto si t_i existe en d_j junto con su información de relevancia (*informacion_de_t*)
- `conj_docs DocsConT (t_i):`
 - # Devuelve todos los documentos que contienen a t_i .

```
conj_docs Docs_relevantes (conj_term q) {  
    conj_docs aux;  
    for(it=q.begin(); it != q.end(); ++it)  
        aux += DocsConT (*it);  
    for(id=aux.begin(); id != aux.end(); ++id) {  
        CalculaRelevancia (*id);  
    }  
}
```


47

4. Fase de indexación. Módulo de almacenamiento.

Ejercicio 3:

- Sobre los documentos del ejercicio 2 y el conjunto de términos filtrado con stems y listas de palabras de parada:
 - Añadir información de frecuencias (ft y fd)
 - Calcular *conj_doc aux* para las siguientes queries:
 - # Q1: EGYPT REBELS
 - # Q2: EGYPT REBELS PLANS IN NASSAU
 - Modifica el algoritmo anterior (*Docs_relevantes*) para que se realicen ambas fases simultáneamente (*DocsConT* + *CalculaRelevancia*). ¿Qué estructuras de datos se utilizarían?

48




Explotación de la información. Recuperación de Información

4. Fase de indexación. Módulo de almacenamiento.

Forma de almacenamiento:

- Matrices
- Índices invertidos (*inverted indexes*)
- Tablas de dispersión
- Trie

49



Explotación de la información. Recuperación de Información

4. Indexación. Módulo de almacenamiento. Matrices

Almacenamiento mediante matrices:

	Doc. 1	Doc. 2	...	Doc. N
t_1	inf_{11}	inf_{12}	...	inf_{1N}
t_2	inf_{21}	inf_{22}	...	inf_{2N}
...
t_k	inf_{k1}	inf_{k2}	...	inf_{kN}

50

4. Indexación. Módulo de almacenamiento. Matrices

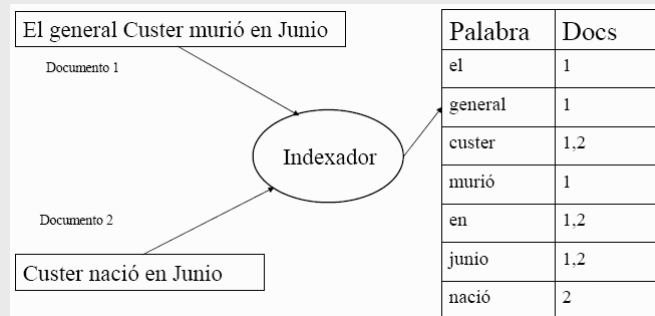
Almacenamiento mediante matrices:

- Complejidad (N: nº documentos, K: nº términos):
 - Espacial: $O(N \cdot K)$ → ¡Inabordable!
 - Temporal:
 - $\text{bool ExisteTerminoEnD}(t_i, d_j): O(1)$
 - $\text{conj_docs DocsConT}(t_i): O(N)$
 - N va a ser muy alto habitualmente!!!
- Alternativa: matrices dispersas
 - Inconveniente de no necesitar la operación *conj_term TermsDeD(d)* (se utilizaría un espacio de punteros innecesario)

51

4. Indexación. Módulo de almacenamiento. Índice invertido

Almacenamiento de términos mediante índices invertidos (Tesis Doctoral de Fernando Llopis):



52

4. Indexación. Módulo de almacenamiento. Índice invertido

Ejercicio 4:

- Generar el índice invertido sobre la información extraída del ejercicio 3
- Calcular la complejidad (N : nº documentos, K : nº términos, L : longitud de la lista de t_i):
 - Espacial
 - Temporal de:
 - # `bool ExisteTerminoEnD(t_i , d_j)`
 - # `conj_docs DocsConT(t_i)`

53

4. Indexación. Módulo de almacenamiento. Índice invertido

Complejidad de la búsqueda de la cadena t_i :

- Si el vector está ordenado:
 - Búsqueda binaria: $O(\log_2 K)$
 - Habría que añadir el coste de creación del vector ordenado:
 - # Algoritmo Quicksort: $O(K * \log K)$
 - O bien realizar la inserción para crear el vector ordenado:
 - # Problemas:
 - Tamaño de la tabla (nº de elementos desconocido a priori)
 - Complejidad: $O(K)$ con K el número de elementos insertados.

54

4. Indexación. Módulo de almacenamiento. Tabla Hash

Complejidad de la búsqueda de la cadena t_i :

■ Alternativa: tablas de dispersión o tablas Hash:

- Se utiliza la información de t_i para almacenar/buscar su posición dentro de la estructura

■ Operaciones:

# Búsqueda.	$\Omega(1)$
# Inserción.	$\Omega(1)$
# Borrado.	$\Omega(1)$

55

4. Indexación. Módulo de almacenamiento. Tabla Hash

Tablas Hash:

- Dividir el conjunto en un n° finito "B" de clases

- Función de dispersión: $H(x) = [0..B-1] = x \text{ MOD } B$

- # Debe ser fácil de calcular
- # Debe minimizar el n° de colisiones
- # Debe distribuir los elementos de forma azarosa
- # Debe hacer uso de toda la información asociada a las etiquetas

- Formas de dispersión:

- Abierta: No impone tamaño límite al conjunto
- Cerrada: usa un tamaño fijo de almacenamiento

56

4. Indexación. Módulo de almacenamiento. Tabla Hash

Tablas Hash. Dispersión cerrada:

- Colisión: claves sinónimas $x_1, x_2 / H(x_1) = H(x_2)$
 - ESTRATEGIA DE REDISPERSION:
 - # Elegir sucesión de localidades alternas dentro de la tabla, hasta encontrar una vacía
 - # $H(x), h_1(x), h_2(x), h_3(x), \dots$
 - # Si ninguna está vacía: no es posible insertar → Hay que ampliar la tabla:
 - Crear una nueva tabla de tamaño $B' > B$
 - Insertar los elementos de la tabla antigua en la nueva con la nueva $H'(x)$
 - Operación ineficiente

57

4. Indexación. Módulo de almacenamiento. Tabla Hash

BÚSQUEDA DE ELEMENTOS

Buscar en sucesión de localidades alternas dentro de la tabla, hasta encontrar una vacía:

$$H(x), h_1(x), h_2(x), h_3(x), \dots$$

BORRADO DE ELEMENTOS

Hay que distinguir durante la búsqueda:

- Casillas vacías
- Casillas suprimidas

Durante la inserción las casillas suprimidas se tratarán como espacio disponible.

58

4. Indexación. Módulo de almacenamiento. Tabla Hash

* ESTRATEGIA DE REDISPERSIÓN LINEAL ("siguiente posición"):

- No eficiente. Larga secuencia de intentos

$$h_i(x) = (H(x) + 1 \cdot i) \text{ MOD } B \quad c=1 \quad h_i(x) = (h_{i-1}(x) + 1) \text{ MOD } B$$

* ESTRATEGIA DE REDISPERSIÓN ALEATORIA:

$$h_i(x) = (H(x) + c \cdot i) \text{ MOD } B \quad c > 1 \quad h_i(x) = (h_{i-1}(x) + c) \text{ MOD } B$$

Sigue produciendo AMONTONAMIENTO

c y B no deben tener factores primos comunes mayores que 1

* E.R. CON 2ª FUNCIÓN DE HASH:

$$h_i(x) = (H(x) + k(x) \cdot i) \text{ MOD } B \quad h_i(x) = (h_{i-1}(x) + k(x)) \text{ MOD } B$$

$$k(x) = (x \text{ MOD } (B-1)) + 1$$

B debe ser primo

59

4. Indexación. Módulo de almacenamiento. Tabla Hash

Insertar en una tabla de dispersión cerrada de tamaño $B=7$, con función de dispersión $H(x) = x \text{ MOD } B$, y con estrategia de redispersión *segunda función hash*, los siguientes elementos: 23, 14, 9, 6, 30, 12, 18

$$\blacksquare c=k(x) \rightarrow 1 \dots B-1 \quad // \quad k(x) = (x \text{ MOD } (B-1)) + 1$$

$$h_i(x) = (H(x) + k(x) \cdot i) \text{ MOD } B = (h_{i-1}(x) + k(x)) \text{ MOD } B$$

$$H(23) = 23 \text{ MOD } 7 = 2$$

$$H(14) = 14 \text{ MOD } 7 = 0$$

$$H(9) = 9 \text{ MOD } 7 = 2$$

$$k(9) = (9 \text{ MOD } 6) + 1 = 4$$

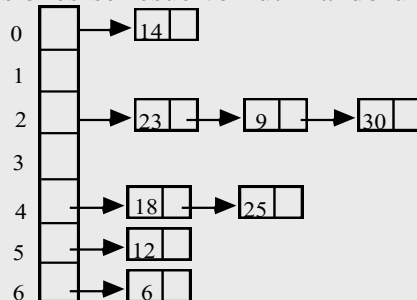
$$h_1(9) = (2 + 4) \text{ MOD } 7 = 6$$

60

Explotación de la información. Recuperación de Información

$$H(18) = 18 \bmod 7 = 4$$
11

Explotación de la información. Recuperación de Información



31

4. Indexación. Módulo de almacenamiento. Tabla Hash

$$\alpha = \frac{n}{|B|}$$

n = nº elem. de la tabla. B = tamaño de la tabla

HASH CERRADO: $0 \leq \alpha \leq 1$

HASH ABIERTO: $\alpha \geq 0$ (No hay límite en el nº de elementos en cada casilla).

- Reestructuración de las tablas de dispersión:

$n \geq 0,9 B$ (H.C.)

$n \geq 2 B$ (H.A.)

→ Nueva tabla con el doble de posiciones

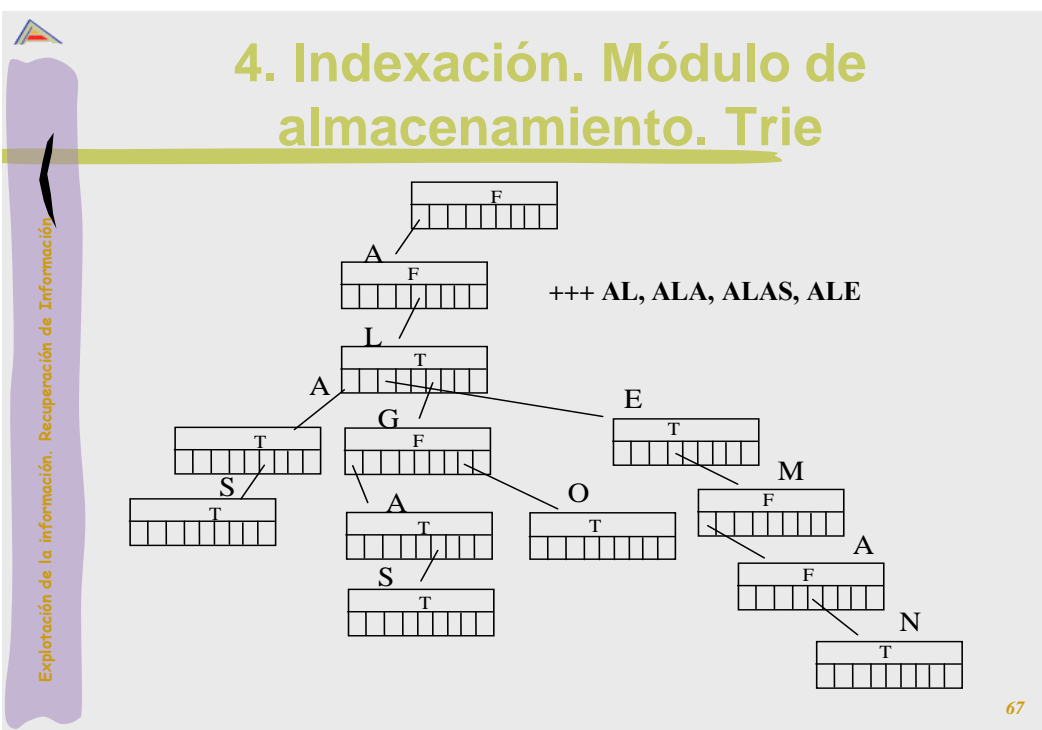
63

4. Indexación. Módulo de almacenamiento. Trie

Trie:

- Árbol para representar conjuntos de cadenas de caracteres u objetos (DICCIONARIO DE CADENAS):
- Cada nodo representa el prefijo de una palabra
- Los hijos de un nodo representan las cadenas que tiene a sus padres como prefijos
- Ventajas:
 - Búsquedas parciales (palabras que empiezan por "AR")
 - No necesitan operación "redimensionar tabla"
 - Cada carácter se almacena una sola vez en los prefijos comunes
 - Complejidad en función de la longitud de la palabra y no en función del nº de palabras

64



67

4. Indexación. Módulo de almacenamiento. Trie

Pertenece:

ALGORITMO BUSQUEDA

ENTRADA: s : Cadena; A : TRIE;

SALIDA: Encontrado: Boolean;

VAR : p: Iterador; c: Carácter;

METODO

```

p = A; Encontrado = FALSE
Mientras NO EsVacio (p) y NO Encontrado
  Si LONGITUD (s) == 0
    Encontrado = Dato (p)
  Sino
    c = OBTENER (s,1)
    p = p.HIJO (c)
    Si NO EsVacio (p)
      SUPRIMIR (s)
    fsi
  fsi
fMientras
fMETODO
  
```

68

4. Indexación. Módulo de almacenamiento. Trie

Inserción:

- ▣ Si el camino seguido para la inserción ya existe:
 - ▣ Cambiar el flag booleano a cierto al llegar a la última letra de la palabra
- ▣ Si no existe:
 - ▣ Crear un nodo en la posición que le corresponda, y así con todas las letras hasta completar la palabra completa

69

4. Indexación. Módulo de almacenamiento. Trie

Complejidad Temporal:

PERTENECE e INSERTAR:

TRIE: $O(L)$

$L = \text{longitud de la cadena}$

ABB: $O(n)$

$n = \text{tamaño diccionario}$

70

Explotación de la información. Recuperación de Información

4. Indexación. Módulo de almacenamiento. Trie

Complejidad Espacial:

Demasiados nodos. Otra opción es tener dos tipos de nodo:

- Nodos prefijo.
- Nodos terminales.

ALGAS, ALGO, ALGUNOS, ALEMAN, ZOOM

¿¿AL, ALA, ZORRO??

71

Explotación de la información. Recuperación de Información

4. Indexación. Módulo de almacenamiento. Trie

+++ AL, ALA, ZORRO

72



4. Indexación. Módulo de almacenamiento. Trie

Ejercicio 5:

- Proponer estructuras alternativas para mejorar la complejidad espacial y temporal de las representaciones vistas anteriormente
- Probarlo insertando las palabras del trie anterior más las siguientes: alas, alemanes, alemañes, alga, algos, als, laura, zo, zooms, zorros
- Analizar comparativamente la complejidad de esas nuevas propuestas

73



4. Indexación. Módulo de almacenamiento. Trie

Mejora adicional:

- Tras la indexación solo se hacen operaciones de búsqueda de palabras:
 - La estructura se puede optimizar para la lectura de la tabla de palabras y las operaciones de búsqueda de palabras:
 - # Se sustituye:
 - Memoria dinámica por estática: más eficiente
 - Punteros por direcciones *offset* que ocupan menos espacio

74

Explotación de la información. Recuperación de Información

4. Indexación. Módulo de almacenamiento. Trie

Mejora adicional (cont.):

- El árbol se recorre en postorden: HI, HD, Raíz
- La primera casilla del vector almacena el *offset* de la raíz
- Las restantes casillas, para cada nodo del trie:
 - La primera casilla almacena el índice a la información adicional de la palabra
 - La segunda, el tamaño respecto a los hijos del nodo
 - Las siguientes, pares (cadena a la que representa el hijo, offset del hijo)

75

Explotación de la información. Recuperación de Información

4. Indexación. Módulo de almacenamiento. Trie

Ejemplo:

0	13	offset of root node
1	10	node value of 'aa'
2	0	size of index to child nodes of 'aa' in bytes
3	3	node value of 'ab'
4	0	size of index to child nodes of 'ab' in bytes
5	13	node value of 'a'
6	4	size of index to child nodes of 'a' in bytes
7	a	index key for 'aa' coming from 'a'
8	4	relative offset of node 'aa' (5 - 4 = 1)
9	b	index key for 'ab' coming from 'a'
10	2	relative offset of node 'ab' (5 - 2 = 3)
11	7	node value of 'b'
12	0	size of index to child nodes of 'b' in bytes
13	20	root node value
14	4	size of index to child nodes of root in bytes
15	a	index key for 'a' coming from root
16	8	relative offset of node 'a' (13 - 8 = 5)
17	b	index key for 'b' coming from root
18	2	relative offset of node 'b' (13 - 2 = 11)

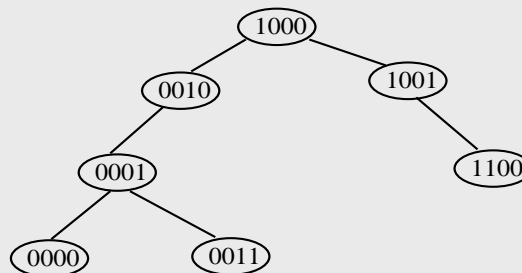
76

4. Indexación. Módulo de almacenamiento. Trie

- # Caso particular de los TRIES
- # Árbol binario en que cada nodo contiene un elemento.
- # La asignación de un nodo viene determinada por la representación binaria de la clave:

■ Dado nodo X en nivel " i ":

- $\forall Y \in \text{SubárbolIzq}(X)$ tienen el bit " i " igual a cero
- $\forall Y \in \text{SubárbolDer}(X)$ tienen el bit " i " igual a uno

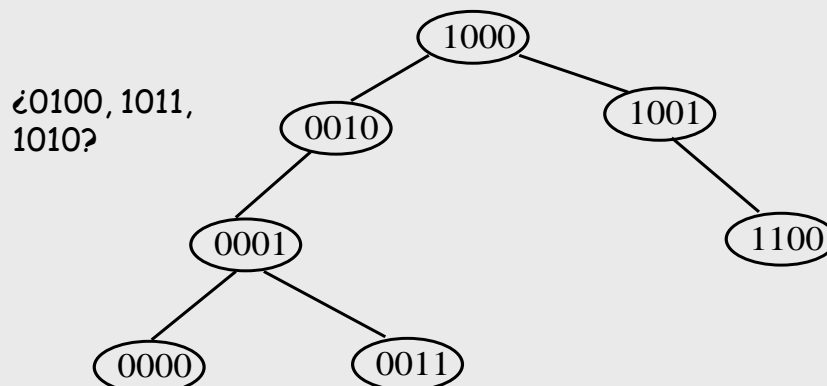


77

4. Indexación. Módulo de almacenamiento. Trie

BUSQUEDA, INSERCIÓN Y BORRADO:

Igual que en ABB excepto que el subárbol al que hay que moverse viene determinado por un bit en la clave.



78

Explotación de la información. Recuperación de Información

4. Indexación. Módulo de almacenamiento. Trie

COMPLEJIDAD:

$O(h)$ $h = \text{altura del árbol}$
 $h = N+1$ $N = \text{nº de bits de la clave}$

Desventajas respecto al trie:

- Eficiencia espacial: cada letra del prefijo se almacena
- Eficiencia temporal:
 - Hay que realizar la conversión a código binario
 - En cada nivel del árbol hay que hacer una comparación entre cadenas de bits
- Solución: árboles digitales comprimidos: PATRICIA

79

Explotación de la información. Recuperación de Información

4. Indexación. Módulo de almacenamiento. Trie

Ejercicio 6:

- Generar la tabla de palabras del trie sobre la información extraída del ejercicio 3
- Generar la tabla de información adicional:
 - Para cada término:
 - ✱ Frecuencia del término en cada documento
 - ✱ Posiciones del término en el documento (p.ej. número de palabra)
 - ✱ Número de documentos que contienen al término
 - ✱ Lista de documentos que contienen al término
 - Para cada documento:
 - ✱ Identificador
 - ✱ Nombre completo
 - ✱ Nº de palabras
 - ✱ Nº de palabras sin *stop-words*
 - ✱ Tamaño en bytes
 - Para la colección de documentos:
 - ✱ Número de términos indexados
 - ✱ Número de términos diferentes indexados
 - ✱ Número de documentos indexados
 - ✱ Tamaño medio de documentos

80



5. Fase de búsqueda de información

Características generales:

- ▀ Se le aplica el mismo proceso de los documentos a las palabras clave de la pregunta.
- ▀ Se comparan las palabras clave de la pregunta con las de cada documento.
- ▀ Sólo se utilizarán los términos de la pregunta que apareciesen previamente indexados en la colección.

81



5. Fase de búsqueda de información

Características generales (cont.):

- ▀ Métodos de comparación:
 - ▀ Modelo booleano.
 - ▀ Modelo vectorial.
 - ▀ Modelo probabilístico.
 - ▀ Modelo basado en pasajes.

82



5. Fase de búsqueda de información

Modelo booleano:

- El más sencillo de utilizar.
- Cuenta ocurrencias de términos utilizando operadores booleanos (AND, OR, NOT):
 - Término 1 AND término 2 AND ... OR término J.
- Problema:
 - No permite una ordenación de documentos.
- Posible mejora:
 - Asignar un peso a cada operador booleano: **modelo booleano extendido**.

83



5. Fase de búsqueda de información

Modelo vectorial:

- Preguntas y documentos se representan como vectores en un espacio K-dimensional (K = número de términos).
- Modelo estándar en la comunidad de la RI.
 - Ventajas: simplicidad y naturalidad.
 - Desventajas: considera cada término independiente con los demás (modelo *bag of words*).
- Medidas de similitud (Q pregunta, D documento):
 - *Overlap*: $|Q \wedge D|$
 - *Coseno*: $\frac{|Q \wedge D|}{\sqrt{|Q|} + \sqrt{|D|}}$

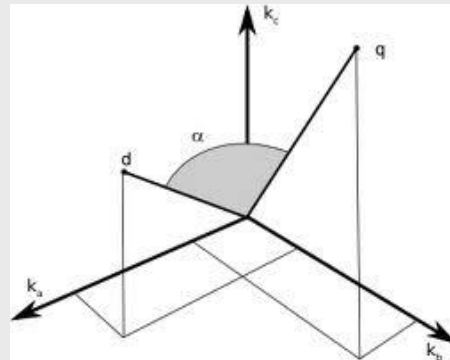
84



5. Fase de búsqueda de información

Modelo vectorial:

- Preguntas y documentos se representan como vectores en un espacio K -dimensional (K = número de *términos*)
- Similitud: coseno del ángulo entre ambos vectores



85



5. Fase de búsqueda de información

Ejercicio 7:

- Sobre la información generada en el ejercicio 6, calcular las medidas de similitud del “overlap” y el “coseno” para las siguientes preguntas:
 - AMERICAN REBELS
 - 01/04/63
- Analiza la diferencia que se hubiese producido al usar el modelo booleano con AMERICAN and REBELS

86

5. Fase de búsqueda de información

Modelo vectorial (cont.):

- Medida del coseno con pesos:

$$sim(Q, D) = \frac{\sum_{i=1}^k q_i * d_i}{\|Q\| * \|D\|}, \quad \|Q\| = \sqrt{\sum_{i=1}^k q_i * q_i}, \quad q_i = ft_{Q,i} * \log_e\left(\frac{N}{fd_i}\right)$$

$$\|D\| = \sqrt{\sum_{i=1}^k d_i * d_i}, \quad d_i = ft_{D,i} * \log_e\left(\frac{N}{fd_i}\right)$$

- Medida del coseno según (Kaszkiel et al., 1999):

$$q_i = \log_e(ft_{q,i} + 1) * \log_e\left(\frac{N}{fd_i} + 1\right)$$

$$d_i = \log_e(ft_{d,i} + 1)$$

87

5. Fase de búsqueda de información

Ejercicio 8:

- Sobre la información generada en el ejercicio 6, calcular las medidas de similaridad utilizando el modelo del coseno con pesos y el del coseno según Kaszkiel para las siguientes preguntas:
 - REBELS
 - EGYPT IN EUROPE

88



5. Fase de búsqueda de información

Modelo vectorial (cont.):

- Para intentar independizar los resultados del tamaño de los documentos, se suele normalizar las frecuencias ft .
 - $ft_i = ft_i / \max ft_d$ para todos los términos i del documento d .
 - Todos los ft tendrán valores entre 0 y 1.
 - O bien se aplica la normalización del logaritmo, tal y como ocurre en el coseno según Kaszkiel.

89



5. Fase de búsqueda de información

Modelo vectorial (cont.):

- Coseno pivotado (favorece documentos cortos):

$$\text{sim}(Q, D) = \sum_{i=1}^k \left(\frac{q_i * d_i}{W_d} \right)$$

$$q_i = 1 + \log_e \left(1 + \log_e (ft_{q,i}) \right) * \log_e \left(\frac{N+1}{fd_i} \right)$$

$$d_i = 1 + \log_e (1 + ft_{d,i})$$

$$W_d = (1 - \text{slope}) + \text{slope} * \frac{d_{\text{longBytes}}}{\text{media}_{\text{longBytes}}}, \text{slope} = 0,2$$

90

5. Fase de búsqueda de información

Ejercicio 9:

- Sobre la información generada en el ejercicio 6:
 - Calcular q_i y d_i para los términos implicados en las siguientes preguntas según el modelo del coseno pivotado:
 - # REBELS
 - # EGYPT IN EUROPE
 - ¿Cambiarían los resultados para cada documento respecto a lo calculado en el ejercicio anterior para Kaszkiel?

Explotación de la información. Recuperación de Información

91

5. Fase de búsqueda de información

Modelo probabilístico:

- Intenta resolver el problema de RI a través del cálculo de probabilidades.
- La medida de similitud se define como:
$$\frac{\text{Probabilidad de que se halle en el conjunto de documentos relevantes}}{\text{Probabilidad de que se halle en el conjunto de documentos NO relevantes}}$$
- Objetivo:
 - Localizar aquellos documentos que maximicen la probabilidad de pertenecer al conjunto de documentos relevantes.
- Ejemplo: Sistema *okapi*.

Explotación de la información. Recuperación de Información

92

5. Fase de búsqueda de información

Modelo probabilístico. Sistema *okapi*:

■ $b \in [0, 1]$, controla efecto de normalización de long.doc:

- $b=0 \rightarrow$ no se hace normalización de la long. Doc.
- $b=1 \rightarrow$ se hace una normalización total de la long.doc

■ $k_1 > 0$, controla efecto de la frecuencia de término:

$$tf = \frac{ft_{i,d}}{B} \quad \frac{tf}{tf + k_1} * idf_i \quad idf_i = \frac{N - fd_i + 0,5}{fd_i + 0,5}$$

- CLEF-2001: $b=0,75$ $k_1=1,2$
- ResPubliQA-2009: $b=0,6$ $k_1=0,1$

$$B = (1-b) + b * \frac{d_{longBytes}}{media_{longBytes}} \quad sim(Q, D) = \sum_{i=1}^k \frac{ft_{i,d}}{k_1 * \left((1-b) + b * \frac{d_l}{media_l} \right) + ft_{i,d}} * idf_i$$

93

5. Fase de búsqueda de información

Modelo probabilístico. Sistema *okapi BM25*:

- Variante de okapi ampliamente usada.
- $c(q_i, d)$ = frecuencia del término en el documento
- $|d|$ = número de palabras del documento
- $avgdl$ = media de palabras por documento en toda la colección

The BM25 score of a document d was computed as follows,

$$BM25(d, q) = \sum_{q_i \in q} \frac{idf(q_i) \cdot c(q_i, d) \cdot (k_1 + 1)}{c(q_i, d) + k_1 \cdot (1 - b + b \cdot \frac{|d|}{avgdl})} \cdot \frac{(k_3 + 1)c(q_i, q)}{k_3 + c(q_i, q)}, \quad (2)$$

where $avgdl$ denotes the average document length in the entire document corpus. k_1 , k_3 and b are free parameters. In feature 21-25 and 61, we set $k_1 = 2.5$, $k_3 = 0$ and $b = 0.8$. The $avgdl$ for each stream can be obtained from meta information, as

94



5. Fase de búsqueda de información

Modelo basado en pasajes:

- ▀ Dividen los documentos en trozos de texto.
- ▀ Determinan la relevancia del documento en función de los pasajes:
 - ▀ Se le asigna la puntuación del mejor pasaje.
 - ▀ Se le asigna la suma de la puntuación de cada pasaje.

95



5. Fase de búsqueda de información

Modelo basado en pasajes (cont.):

- ▀ Superan los problemas de los modelos basados en documentos:
 - ▀ Tienen en cuenta de forma automática la proximidad de aparición entre las palabras del documento.
 - ▀ Localizan la zona realmente relevante del documento.
 - ▀ Realizan una normalización eficiente y automática del tamaño de los documentos.

96



5. Fase de búsqueda de información

Modelo basado en pasajes (cont.):

- Tipos de modelos:
 - Basados en el discurso.
 - Semánticos.
 - Ventana.

97



5. Fase de búsqueda de información

Modelo basado en pasajes (cont.):

- Modelos basados en el discurso:
 - División en función de frases, párrafos, secciones, marcas HTML u otros elementos estructurales del documento.
 - Problemas:
 - ✦ En algunos documentos es difícil realizar esta división:
 - Faltan las marcas HTML o la división en secciones o párrafos.
 - Documentos poco estructurados.
 - ✦ Puede generar pasajes de tamaño muy heterogéneo, luego no se realiza la normalización automática en función del tamaño.
 - ✦ Sensible al algoritmo de división en pasajes (nueva fuente de error):
 - Los párrafos en ocasiones se definen por motivos visuales y no por motivos de coherencia semántica.

98



5. Fase de búsqueda de información

Modelo basado en pasajes (cont.):

■ Modelos basados en la semántica:

- División según la consistencia interna de su contenido, utilizando la estructura discursiva del documento:
 - # Unión de frases o párrafos si presentan relaciones semánticas.
 - # Pueden unir párrafos no adyacentes.
- Herramienta “TextTiling”
- Problemas:
 - # Dificultad de conseguir la división en pasajes:
 - El cálculo de estas relaciones semánticas no está suficientemente desarrollado. Se suele calcular en función del número de palabras que comparten.
 - # Incorporan una nueva fuente de error.

99



5. Fase de búsqueda de información

Modelo basado en pasajes (cont.):

■ Modelos basados en ventanas:

- División según un rango de tamaño.
- Se busca una normalización automática.
- Utilizan *solapamiento* de pasajes para evitar introducir errores en el modelo al dividir en pasajes:
 - # Existe parte de texto compartida entre pasajes distintos.
- Subtipos:
 - # Basados en la estructura del documento.
 - # NO basados en la estructura del documento.

100

5. Fase de búsqueda de información

Ejercicio 10:

- Sobre los documentos del ejercicio 2:
 - Realizar la división de pasajes según el modelo de ventanas:
 - # Sobre un tamaño de 10 palabras y un solapamiento de 5
 - ¿Qué ventajas le ves a esta representación del documento en comparación con las vistas anteriormente? Propón ejemplos que muestren las ventajas.

Explotación de la información. Recuperación de Información

101

5. Fase de búsqueda de información

Expansión de la pregunta (*Query Expansion*):

- Añadir nuevos términos a la pregunta.
- Aumenta la cobertura, pero puede bajar la precisión
- Métodos:
 - Basados en tesauros:
 - # WordNet, sinónimos (*processor – CPU*), hiperónimos (*computer – mainframe*), hipónimos (*computer – device*), ...
 - # Añadir variantes morfológicas (*comer, comió, comía, ...*).
 - # Añadir derivaciones (*vender, vendedor, vendido, ...*).
 - # Problemas de añadir sinónimos con sentidos diferentes al original.
 - Realimentación: *Relevance feedback*
 - Análisis local: *Blind/Pseudo Relevance Feedback*
 - Análisis global.

Explotación de la información. Recuperación de Información

102



5. Fase de búsqueda de información

Expansión de la pregunta. *Relevance feedback*:

- El usuario decide qué documentos de los primeros 10 ó 20 presentados son relevantes.
- Se añaden sus palabras claves (las de mayor frecuencia) a las de la pregunta y se vuelve a lanzar la pregunta.
- Ventaja:
 - # Se evita al usuario la reformulación de la pregunta.
 - # Facilita el proceso al usuario cuando no conoce bien la colección de documentos
- Problemas:
 - # Eficiencia: hay que lanzar nuevamente la pregunta.
 - # El número de términos añadidos demasiado grande.
 - # La necesidad de intervención del usuario.

103



5. Fase de búsqueda de información

Expansión de la pregunta. *Relevance feedback*:

■ Algoritmo de Rocchio:

$$\vec{q}_m = \alpha \vec{q}_0 + \beta \frac{1}{|D_r|} \sum_{\vec{d}_j \in D_r} \vec{d}_j - \gamma \frac{1}{|D_{nr}|} \sum_{\vec{d}_j \in D_{nr}} \vec{d}_j$$

- # q_m : query modificada
- # q_0 : query originalmente lanzada por el usuario
- # D_r : conjunto de docs marcados como relevantes por el usuario
- # D_{nr} : conjunto de docs marcados como NO relevantes por el usuario
- # α, β, γ : pesos asignados a cada término. Habitualmente:
 - $\alpha=1, \beta=0,75, \gamma=0,15$
 - Los pesos negativos se eliminan (se ponen a cero)
- Requiere que el usuario evalúe un nº importante de documentos

104

► **Figure 5**
(a). (b) The query is the results are relevance

(a)	Query: New space satellite applications	(c)	2.074 new 15.106 space 30.816 satellite 5.660 application 5.991 nasa 5.196 eos 4.196 launch 3.972 aster 3.516 instrument 3.446 arianespace 3.004 bundespost 2.806 ss 2.790 rocket 2.053 scientist 2.003 broadcast 1.172 earth 0.836 oil 0.646 measure
(b)	+ 1. 0.539, 08/13/91, NASA Hasn't Scrapped Im + 2. 0.533, 07/09/91, NASA Scratches Environm lite Plan 3. 0.528, 04/04/90, Science Panel Backs NAS Urges Launches of Smaller Probes 4. 0.526, 09/09/91, A NASA Satellite Project A ble Feat: Staying Within Budget 5. 0.525, 07/24/90, Scientist Who Exposed (d) * poses Satellites for Climate Research 6. 0.524, 08/22/90, Report Provides Support for Big Satellites to Study Climate 7. 0.516, 04/13/87, Arianespace Receives S From Telesat Canada + 8. 0.509, 12/02/87, Telecommunications Tale o	* 1. 0.513, 07/09/91, NASA Scratches Environment Gear From Sate- lite Plan * 2. 0.500, 08/13/91, NASA Hasn't Scrapped Imaging Spectrometer 3. 0.493, 08/07/89, When the Pentagon Launches a Secret Satellite, Space Sleuths Do Some Spy Work of Their Own 4. 0.493, 07/31/89, NASA Uses 'Warm' Superconductors For Fast Circuit * 5. 0.492, 12/02/87, Telecommunications Tale of Two Companies 6. 0.491, 07/09/91, Soviets May Adapt Parts of SS-20 Missile For Commercial Use 7. 0.490, 07/12/88, Gaping Gap: Pentagon Lags in Race To Match the Soviets In Rocket Launchers 8. 0.490, 06/14/90, Rescue of Satellite By Space Agency To Cost \$90 Million	

► **Figure 9.2** Example of relevance feedback on a text collection. (a) The initial query (a). (b) The user marks some relevant documents (shown with a plus sign). (c) The query is then expanded by 18 terms with weights as shown. (d) The revised top results are then shown. A * marks the documents which were judged relevant in the relevance feedback phase.

Explotación de la información. Recuperación de Información

Original query

0	4	0	8	0	0
---	---	---	---	---	---

 $\alpha = 1.0$

0	4	0	8	0	0
---	---	---	---	---	---

Positive Feedback

2	4	8	0	0	2
---	---	---	---	---	---

 $\beta = 0.5$

1	2	4	0	0	1
---	---	---	---	---	---

 (+)

Negative feedback

8	0	4	4	0	16
---	---	---	---	---	----

 $\gamma = 0.25$

2	0	1	1	0	4
---	---	---	---	---	---

 (-)

New query

-1	6	3	7	0	-3
----	---	---	---	---	----

Explotación de la información. Recuperación de Información

📌 Dado el vector de términos de la pregunta original, el vector de términos del documento relevante, cada uno con sus pesos asociados, calcular los pesos del vector resultante de la expansión según el algoritmo de Rocchio con:

107

Explotación de la información. Recuperación de Información

108

5. Fase de búsqueda de información

Exp. preg. Análisis local. *Blind Relevance Feedback* o *Pseudo Relevance Feedback*:

- ▀ Igual que el *relevance feedback* pero de forma automática:
 - ▀ Se consideran como relevantes los k primeros documentos
 - ▀ Se seleccionan los términos más frecuentes o con mayor poder discriminatorio (umbral de idf)
 - ▀ Se aplica el mismo ajuste anterior sobre los términos añadidos (p.ej. la misma fórmula de Rocchio)
- ▀ Problemas:
 - ▀ Inclusión de nuevos términos sin relación con la pregunta

109

5. Fase de búsqueda de información

Expansión de la pregunta. Análisis global:

- ▀ Establecen relaciones co-ocurrencia entre palabras.
- ▀ En fase de indexación se crea matriz co-ocurrencias.
- ▀ De forma automática, se añaden a la pregunta las palabras que suelen aparecer conjuntamente.
- ▀ Problema: eficiencia espacial y temporal

110

6. Fase de Presentación de resultados

Lo normal sería:

- ▀ Calcular el peso de cada d para la query q , devolviendo los K mejores documentos:

```

1  float Scores[N] = 0
2  for each d
3  do Initialize Length[d] to the length of doc d
4  for each query term t
5  do calculate  $w_{t,q}$  and fetch postings list for t
6    for each pair  $(d, tf_{t,d})$  in postings list
7    do add  $w_{t,d}$  to Scores[d]
8  Read the array Length[d]
9  for each d
10 do Divide Scores[d] by Length[d]
11 return Top K components of Scores[]

```

111

6. Fase de Presentación de resultados

Problema:

- ▀ Cuando el nº de documentos a calcular su peso es muy grande:
 - ▀ Heurística:
 - # Devolver K documentos cuyos pesos **estén probablemente entre los mejores K documentos** con un menor coste computacional
 - # Eliminar documentos sin calcular su peso, reduciendo el nº de acumuladores ($Scores[d]$)
 1. Encontrar un conjunto de documentos $K < |A| \ll N$
 2. Calcular el peso de los documentos en A
 3. Devolver los K mejores documentos de A

112

6. Fase de Presentación de resultados

Encontrar un conjunto de documentos $K < |A|$:

■ Heurísticas:

- Considerar solo los docs que contienen al menos un término con $idf > threshold$
 - # Ordenar q de mayor a menor idf , procesando los docs de sus listas invertidas en ese orden
 - # Ventaja: los términos con idf bajo suelen tener una lista invertida con un n° de docs alto
- Considerar solo los docs que contienen un n° mínimo M de términos de q
 - # Problema: que pueda darse que $|A| < K$
- Ordenar todas las listas invertidas de mayor a menor $ft(t_i)$, seleccionando para A los r mayores de cada q_i .

113

6. Fase de Presentación de resultados

Ordenar los K documentos de mayor a menor relevancia respecto a q :

- Algoritmo *quicksort*: $O(K \log K)$. Ordenación total
- Algoritmo *heapsort*: $O(K \log K)$:
 - Ordenación parcial → mejor eficiencia promedio
 - Ordenación mediante montículos o *heaps*

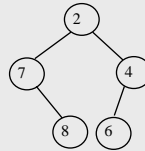
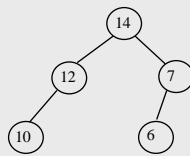
INSERCIÓN:	$O(\log K)$
BORRADO:	$O(\log K)$

114

6. Fase de Presentación de resultados. Montículos

Árbol Mínimo (Máximo):

Árbol en el que la etiqueta de cada nodo es menor (mayor) que la de los hijos.

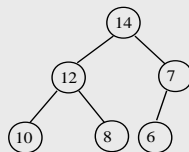


115

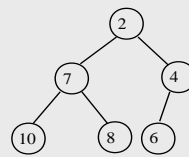
6. Fase de Presentación de resultados. Montículos

Heap Mínimo (Máximo):

Árbol binario completo en que además es ARBOL MINIMO o MAXIMO.



HEAP MAXIMO



HEAP MINIMO

116

Explotación de la información. Recuperación de Información

6. Fase de Presentación de resultados. Montículos

Inserción:

- 1.- Insertar en la posición correspondiente para que siga siendo un árbol completo.
- 2.- Reorganizar para que cumpla las condiciones del HEAP:
 - Comparar con el nodo padre: si no cumple las condiciones del árbol mínimo/máximo, entonces intercambiar ambos.

117

Explotación de la información. Recuperación de Información

6. Fase de Presentación de resultados. Montículos

Insertar: 2, 10, 14, 15, 20 y 21 en un heap máximo inicialmente vacío

```
graph TD
    subgraph Step1 [Step 1: Insert 2]
        2((2))
    end
    subgraph Step2 [Step 2: Insert 10]
        10((10))
        2 --- 10
    end
    subgraph Step3 [Step 3: Insert 14]
        14((14))
        10 --- 14
        2 --- 10
    end
    subgraph Step4 [Step 4: Insert 15]
        15((15))
        14 --- 15
        10 --- 14
        2 --- 10
    end
    subgraph Step5 [Step 5: Insert 20]
        20((20))
        15 --- 20
        10 --- 15
        2 --- 10
    end
    subgraph Step6 [Step 6: Insert 21]
        21((21))
        20 --- 21
        15 --- 20
        10 --- 15
        2 --- 10
    end
```

The diagram illustrates the insertion of elements 2, 10, 14, 15, 20, and 21 into a max-heap. The process shows the tree structure after each insertion and the necessary swaps to maintain the max-heap property. The final heap structure is a complete binary tree with root 21, left child 15, and right child 20. Node 15 has left child 2 and right child 14. Node 20 has left child 10.

118

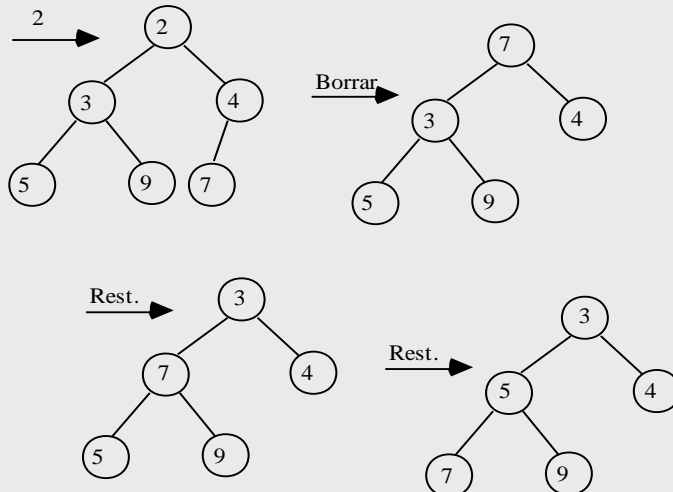
6. Fase de Presentación de resultados. Montículos

Borrado:


- Se sustituye la raíz con el elemento más a la derecha en el nivel de las hojas
- Mientras no sea un HEAP se hunde ese elemento sustituyéndolo con el más pequeño (montículo mínimo) o el mayor (montículo máximo) de sus hijos

119

6. Fase de Presentación de resultados. Montículos



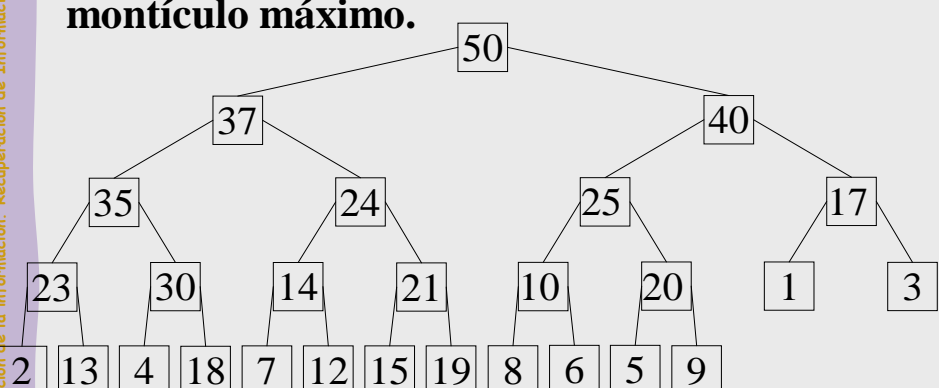
120



Explotación de la información. Recuperación de Información


6. Fase de Presentación de resultados. Montículos

Realizar dos borrados sobre el siguiente montículo máximo.



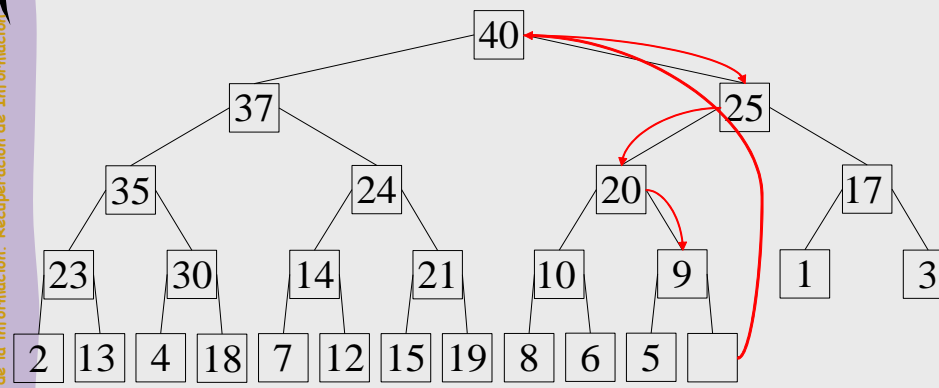
```
graph TD; 50 --> 37; 50 --> 40; 37 --> 35; 37 --> 24; 40 --> 25; 40 --> 17; 35 --> 23; 35 --> 30; 24 --> 14; 24 --> 21; 25 --> 10; 25 --> 20; 17 --> 1; 17 --> 3; 23 --> 2; 23 --> 13; 30 --> 4; 30 --> 18; 14 --> 7; 14 --> 12; 21 --> 15; 21 --> 19; 10 --> 8; 10 --> 6; 20 --> 5; 20 --> 9;
```

121




Explotación de la información. Recuperación de Información

6. Fase de Presentación de resultados. Montículos



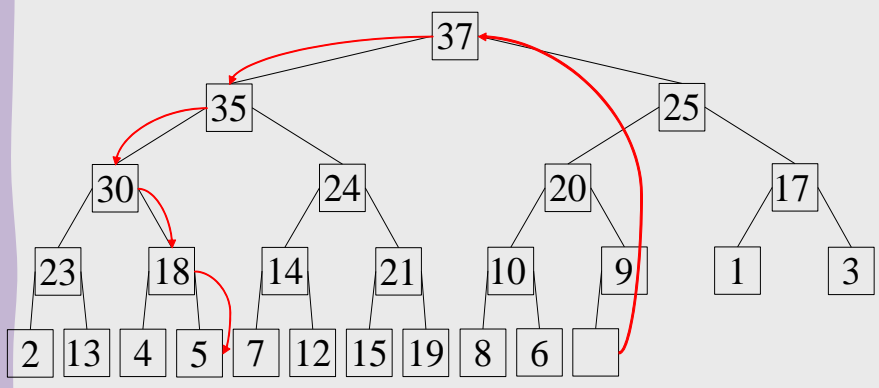
```
graph TD; 40 --> 37; 40 --> 25; 37 --> 35; 37 --> 24; 25 --> 20; 25 --> 17; 35 --> 23; 35 --> 30; 24 --> 14; 24 --> 21; 20 --> 10; 20 --> 9; 17 --> 1; 17 --> 3; 23 --> 2; 23 --> 13; 30 --> 4; 30 --> 18; 14 --> 7; 14 --> 12; 21 --> 15; 21 --> 19; 10 --> 8; 10 --> 6; 9 --> 5; 9 --> ;
```

122




Explotación de la información. Recuperación de Información

6. Fase de Presentación de resultados. Montículos



```
graph TD; 37 --> 35; 37 --> 25; 35 --> 30; 35 --> 24; 30 --> 23; 30 --> 18; 24 --> 14; 24 --> 21; 25 --> 20; 25 --> 17; 20 --> 10; 20 --> 9; 17 --> 1; 17 --> 3; 23 --> 2; 23 --> 13; 18 --> 4; 18 --> 5; 14 --> 7; 14 --> 12; 21 --> 15; 21 --> 19; 10 --> 8; 10 --> 6; 9 --> 9; 9 --> 9
```

123



Explotación de la información. Recuperación de Información


6. Fase de Presentación de resultados. Montículos

Sobre el resultado anterior, realiza las inserciones: 60, 36



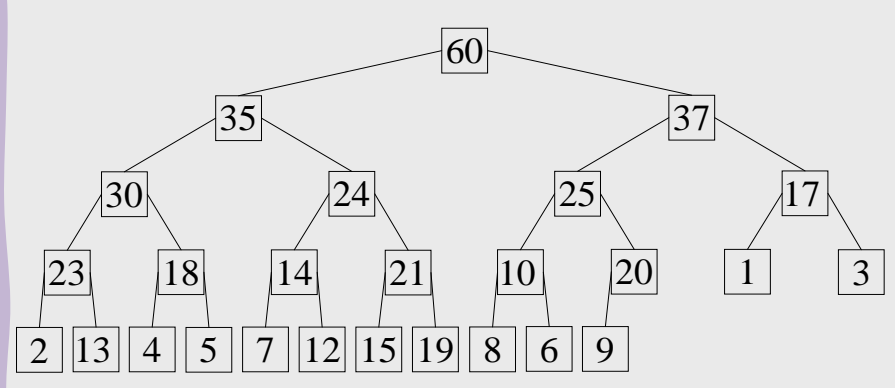
```
graph TD; 37 --> 35; 37 --> 25; 35 --> 30; 35 --> 24; 30 --> 23; 30 --> 18; 24 --> 14; 24 --> 21; 25 --> 20; 25 --> 17; 20 --> 10; 20 --> 9; 17 --> 1; 17 --> 3; 23 --> 2; 23 --> 13; 18 --> 4; 18 --> 5; 14 --> 7; 14 --> 12; 21 --> 15; 21 --> 19; 10 --> 8; 10 --> 6; 9 --> 60; 9 --> 9
```

124




Explotación de la información. Recuperación de Información

6. Fase de Presentación de resultados. Montículos



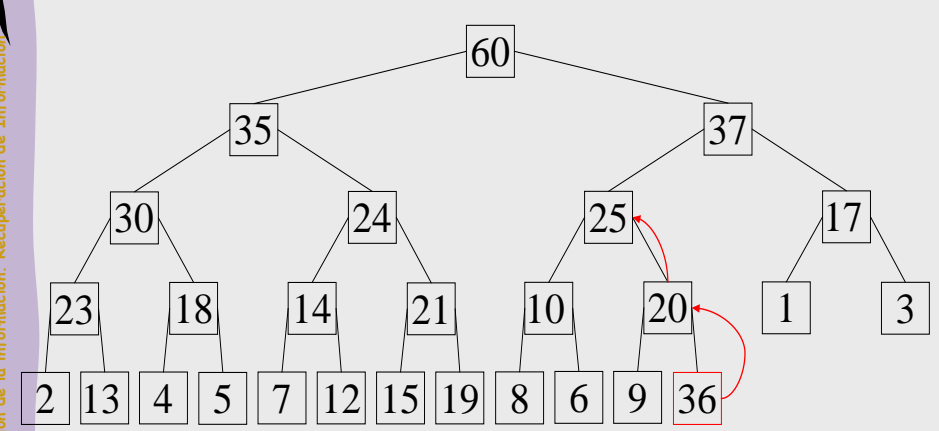
```
graph TD; 60[60] --> 35[35]; 60 --> 37[37]; 35 --> 30[30]; 35 --> 24[24]; 37 --> 25[25]; 37 --> 17[17]; 30 --> 23[23]; 30 --> 18[18]; 24 --> 14[14]; 24 --> 21[21]; 25 --> 10[10]; 25 --> 20[20]; 17 --> 1[1]; 17 --> 3[3]; 23 --> 2[2]; 23 --> 13[13]; 18 --> 4[4]; 18 --> 5[5]; 14 --> 7[7]; 14 --> 12[12]; 21 --> 15[15]; 21 --> 19[19]; 10 --> 8[8]; 10 --> 6[6]; 20 --> 9[9];
```

125



Explotación de la información. Recuperación de Información

6. Fase de Presentación de resultados. Montículos



```
graph TD; 60[60] --> 35[35]; 60 --> 37[37]; 35 --> 30[30]; 35 --> 24[24]; 37 --> 25[25]; 37 --> 17[17]; 30 --> 23[23]; 30 --> 18[18]; 24 --> 14[14]; 24 --> 21[21]; 25 --> 10[10]; 25 --> 20[20]; 17 --> 1[1]; 17 --> 3[3]; 23 --> 2[2]; 23 --> 13[13]; 18 --> 4[4]; 18 --> 5[5]; 14 --> 7[7]; 14 --> 12[12]; 21 --> 15[15]; 21 --> 19[19]; 10 --> 8[8]; 10 --> 6[6]; 20 --> 9[9]; 20 --> 36[36];
```

126



Explotación de la información. Recuperación de Información

6. Fase de Presentación de resultados. Montículos

Representación:

- ENLAZADA: problema en inserción al necesitar realizar recorridos ascendentes.
- SECUENCIAL (en un vector):
Hijos de $p[i]$ son $p[2 \cdot i]$ y $p[2 \cdot i + 1]$. Padre de $p[i]$ es $p[i \text{ DIV } 2]$ con DIV la división entera

128



6. Fase de Presentación de resultados. Montículos

Ejercicio 12:

- Ordenar el vector 9 5 7 4 8 6 2 1 usando un heap mínimo

131



6. Fase de presentación de resultados

Interfaz de usuario:

- Interfaz de consulta:
 - Donde el usuario expresa su necesidad de información
 - Modo de funcionamiento:
 - # Prima la sencillez de manejo
 - # Posibilidad de búsquedas avanzadas:
 - Uso de operadores lógicos
 - Selección de dominios o lenguajes de búsqueda
- Interfaz de la respuesta:
 - Donde se muestran al usuario los resultados
 - Posibilidad de previsualización de las páginas
 - Resúmenes o párrafos más relevantes de los documentos
 - Sugerencias de reformulación de preguntas

132



6. Fase de presentación de resultados

Ejercicio optativo:

- ▣ Diseñar el interfaz de consulta de un sistema de RI
- ▣ Implementarlo en QT:
 - ▣ <http://qt-project.org/>

133



7. Evaluación de sistemas de RI

Objetivo de la evaluación:

- ▣ Comparar sistemas entre sí.
- ▣ ¿Qué hay que comparar?:
 - ▣ Eficacia (precisión y cobertura).
 - ▣ Eficiencia (temporal y espacial).
 - ▣ Satisfacción del usuario (interfaz, modo de presentación de resultados, etc.).

134



7. Evaluación de sistemas de RI

Objetivo de la evaluación (cont.):

- Es conveniente efectuar la evaluación sobre colecciones de documentos:
 - Previamente evaluadas.
 - Sobre las que se dispongan resultados de otros investigadores o sistemas.
 - Formadas por:
 - # Documentos.
 - # Preguntas.
 - # Documentos relevantes para cada pregunta.

135



7. Evaluación de sistemas de RI

Objetivo de la evaluación (cont.):

- Formas de construir COLECCIONES DE TEST:
 - Tradicional: estudiar la relevancia de cada documento para cada pregunta.
 - *Pooling* (CLEF, TREC):
 - # Sólo se estudia la relevancia de los documentos devueltos por los participantes en el concurso.
 - # Ventaja: Permite trabajar con colecciones de gran número de documentos.
 - # Desventaja: No se sabe la relevancia de todos los documentos.

136



7. Evaluación de sistemas de RI

Listas de preguntas evaluadas (Times):

- KENNEDY ADMINISTRATION PRESSURE ON NGO DINH DIEM TO STOP SUPPRESSING THE BUDDHISTS .
 - Documentos relevantes: 268 288 304 308 323 326 334.
- EFFORTS OF AMBASSADOR HENRY CABOT LODGE TO GET VIET NAM'S PRESIDENT DIEM TO CHANGE HIS POLICIES OF POLITICAL REPRESSION .
 - Documentos relevantes: 326 334.

137



7. Evaluación de sistemas de RI

Colecciones CLEF y TREC:

- <title>
 - Area of Kaliningrad
- <description>
 - Find documents discussing the political or economic future of the Kaliningrad Exclave.
- <narrative>
 - Only political or economic information on Kaliningrad is of interest. Prospects for future relations with Scandinavia, the Baltic countries and Russia. Historical or tourist information is not important.
- Tipos de preguntas:
 - Cortas: *title + description.*
 - Largas: *title + description + narrative.*

138



7. Evaluación de sistemas de RI

Medidas más habituales de evaluación:

- Precisión (*precision*):

$$\frac{\text{Numero Documentos Correctos Devueltos}}{\text{Numero Documentos Devueltos}}$$

- Cobertura (*recall*):

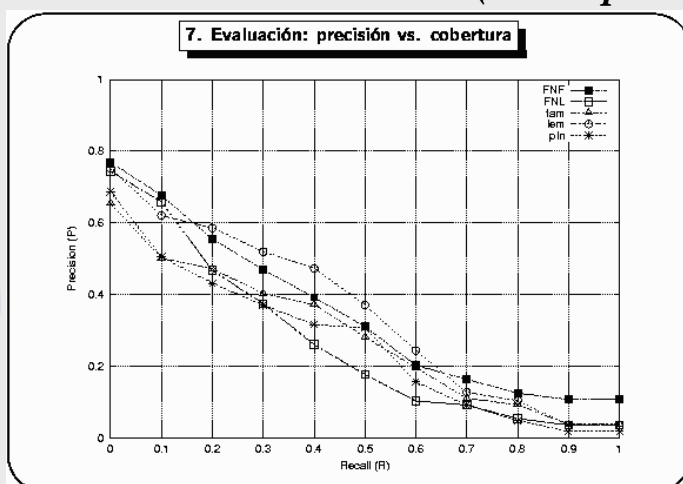
$$\frac{\text{Numero Documentos Correctos Devueltos}}{\text{Numero Documentos Correctos Existentes}}$$

139



7. Evaluación de sistemas de RI

Precisión vs. cobertura (*recall-precision*):



0 - 0,2
Precisión alta

0,2 - 0,8
Cobertura media

0,8 - 1
Cobertura alta

140

7. Evaluación de sistemas de RI

Recall	Precision		
	Vectorial model	Z/Prise	IR-n
0.00	0.5668	0.7583	0.7816
0.10	0.5049	0.7278	0.7354
0.20	0.4394	0.6476	0.6927
0.30	0.3889	0.5632	0.5898
0.40	0.3547	0.4904	0.4864
0.50	0.3182	0.4389	0.4376
0.60	0.2678	0.3315	0.3290
0.70	0.2157	0.2825	0.2724
0.80	0.1765	0.2343	0.2385
0.90	0.1567	0.1925	0.2002
1.00	0.1205	0.1317	0.1339
Medium	0.3028	0.4208	0.4270

Precisión vs. cobertura.
Interpolación:

$$P(c_j) = \max_{c_j \leq c \leq c_k, k > j} P(c)$$

Precisión media para un conjunto de preguntas:

$$\bar{P}(c) = \frac{\sum_{i=1}^{N_p} P_i(c)}{N_p}$$

141

7. Evaluación de sistemas de RI

Precision	Vectorial model	Z/Prise	IR-n
At 5 docs	0.3021	0.4638	0.4851
At 10 docs	0.2766	0.3872	0.4000
At 15 docs	0.2582	0.3135	0.3376
At 20 docs	0.2426	0.2851	0.2904
At 30 docs	0.2014	0.2383	0.2468
At 100 docs	0.1089	0.1215	0.1189
At 200 docs	0.0662	0.0705	0.0699
At 500 docs	0.0311	0.0318	0.0319
At 1000 docs	0.0164	0.0167	0.0164
R-Precision	0.2749	0.4009	0.4113

Precisión respecto al número de documentos devueltos (*document level average*).

142



7. Evaluación de sistemas de RI

Modo de cálculo de la precisión interpolada:

1. Obtener P y C para cada documento correcto devuelto de izquierda a derecha (según el orden en que se han devuelto los documentos)
2. Interpolar los valores de C no obtenidos con los mayores P que vengan a continuación.
3. Si hay un momento que no se obtienen los documentos relevantes que quedan pendientes (la C no puede crecer), entonces la P tomará siempre 0 para los valores pendientes de C.

143



7. Evaluación de sistemas de RI

Ejercicio 13:

- Sea la lista de los documentos relevantes: $\{d_3, d_{56}, d_{129}\}$.
- Supongamos que para esa pregunta devolvemos los siguientes 15 documentos en este orden:
 $\{d_{123}, d_{84}, d_{56}, d_6, d_8, d_9, d_{511}, d_{129}, d_{187}, d_{25}, d_{38}, d_{48}, d_{250}, d_{113}, d_3\}$.
- Calcular:
 - Los valores de precisión vs. cobertura (sin interpolar).
 - Gráfica precisión vs. cobertura interpolada a 11 valores.
 - Precisión respecto al número de documentos devueltos.

144



7. Evaluación de sistemas de RI

Ejercicio 14:

- Sea la lista de los documentos relevantes (10):
 $\{d_3, d_5, d_9, d_{25}, d_{39}, d_{44}, d_{56}, d_{71}, d_{89}, d_{123}\}$.
- Supongamos que para esa pregunta devolvemos los siguientes 15 documentos en este orden:
 $\{d_{123}, d_{84}, d_{56}, d_6, d_8, d_9, d_{511}, d_{129}, d_{187}, d_{25}, d_{38}, d_{48}, d_{250}, d_{113}, d_3\}$.
- Calcular:
 - Los valores de precisión vs. cobertura (sin interpolar).
 - Gráfica precisión vs. cobertura interpolada a 11 valores.
 - Precisión respecto al número de documentos devueltos.

145



7. Evaluación de sistemas de RI

Ejercicio 15:

- Sea la lista de los documentos relevantes (10):
 $\{d_3, d_5, d_9, d_{25}, d_{39}, d_{44}, d_{56}, d_{71}, d_{89}, d_{123}\}$.
- Supongamos que para esa pregunta devolvemos los siguientes 3 documentos en este orden: $\{d_3, d_{56}, d_{129}\}$.
- Calcular:
 - Los valores de precisión vs. cobertura (sin interpolar).
 - Gráfica precisión vs. cobertura interpolada a 11 valores.
 - Precisión respecto al número de documentos devueltos.

146



7. Evaluación de sistemas de RI

Problemas de la evaluación mediante la precisión y cobertura:

- Para alcanzar el 100% de cobertura implica un alto conocimiento de la colección de documentos:
 - Esto es muy difícil en colecciones muy grandes.
 - Dependiente del nº de documentos relevantes en la colección (50% de 2 = 1 doc; 50% de 100 = 50 docs).
- Es difícil comparar sistemas sólo con estas medidas:
 - Ambas están relacionadas.
 - Es preferible obtener una medida única de evaluación.

147



7. Evaluación de sistemas de RI

Medida única de evaluación:

- Combinación precisión y cobertura (*Rijsbergen*):

$$1 - \frac{(1 + b^2) * P * C}{b^2 * P + C}$$

$$b = \frac{\text{Valor } P}{\text{Valor } C} (\text{usuario})$$

148



7. Evaluación de sistemas de RI

Medida única de evaluación (cont.):

■ iCLEF (*Rijsbergen*):

- $\alpha > 0,5$: se da más importancia a la P que a la C.
- $\alpha < 0,5$: se da más importancia a la C.
- iCLEF: $\alpha = 0,8$.

$$F_{\alpha} = \frac{1}{\frac{\alpha}{P} + \frac{(1-\alpha)}{C}}$$

149



7. Evaluación de sistemas de RI

Medida única de evaluación (cont.):

■ Precisión media sobre documentos relevantes encontrados (*average precision at seen relevant documents*):

- La media de las precisiones obtenidas tras encontrar cada documento relevante.
- Por ejemplo, si se obtienen precisiones de 1, 0.66, 0.5, 0.4, 0.3, tras encontrar un documento relevante en las posiciones 1, 3, 6, 10 y 15, este valor sería:

$$\# (1 + 0.66 + 0.5 + 0.4 + 0.3) / 5 = 0.57$$
- Esta medida favorece los sistemas que devuelven más rápidamente los documentos relevantes.

150



7. Evaluación de sistemas de RI

Medida única de evaluación (cont.):

■ R-Precisión:

- Sea R el número total de documentos relevantes para una pregunta:
 - # R -Precision será el número total de documentos relevantes encontrados entre los R primeros documentos devueltos, dividido por R .
- Por ejemplo, si hay 20 documentos relevantes para una pregunta, y entre los primeros 20 documentos devueltos se encuentran 10 documentos relevantes, este valor sería:
 - # $10 / 20 = 0.5$
- También se puede calcular la media de R-Precisión entre un conjunto de preguntas.

151



7. Evaluación de sistemas de RI

Ejercicio 16:

- Sobre los tres ejercicios anteriores calcular:
 - Precisión media sobre documentos relevantes encontrados.
 - R-Precisión.

152



7. Evaluación de sistemas de RI

Ejercicio 17:

- Sea la lista de los documentos relevantes existentes para una pregunta en una colección (6): $\{d_5, d_9, d_{17}, d_{105}, d_{164}, d_{248}\}$.
- Supongamos que para esa pregunta devolvemos los siguientes 9 documentos en este orden: $\{d_5, d_7, d_{248}, d_{164}, d_{10}, d_9, d_{15}, d_{17}, d_{105}\}$.
- Calcular:
 - Precisión vs. cobertura sin interpolar, interpolados, respecto al número de documentos devueltos.
 - Precisión media y R-Precisión.

153



7. Evaluación de sistemas de RI

Medida única de evaluación (cont.):

- Media recíproca (*Mean Reciprocal Rank, MRR*):
 - Inversa de la posición de la primera respuesta correcta:
 - # 1 si se devuelve en primera posición.
 - # 1/2 si se devuelve en segunda.
 - # 1/3 si se devuelve en tercera.
 - # ...
 - # Cero si ninguna respuesta es correcta.
 - No tiene en cuenta si se devuelven varias respuestas correctas en distintas posiciones.
 - Esta medida se ha utilizado en Búsqueda de Respuestas (*Question Answering*) en el TREC.

154

7. Evaluación de sistemas de RI

Más medidas de evaluación:

- ▣ Medidas de complejidad del sistema:
 - ▣ Espacial: espacio ocupado por la información indexada.
 - ▣ Temporal: tiempo de respuesta del sistema.
 - ▣ No son las medidas más utilizadas en investigación (sí en sistemas comerciales).
- ▣ Datos generales de la evaluación:
 - ▣ Datos de la colección: número de documentos, palabras, etc.
 - ▣ Datos de las preguntas: número de preguntas, número de palabras por pregunta, tipos de preguntas, etc.
 - ▣ Datos de las respuestas: número de documentos recuperados por pregunta, etc.

155

8. Sistemas de Búsqueda de Respuestas

RI. Problemas a resolver:

- ▣ Precisión baja: nivel bajo de comprensión del texto.
- ▣ No realizan todo el trabajo: buscar la información dentro del documento.

La solución:

- ▣ Introducir fuentes de información adicionales:
www.playence.com
- ▣ Los sistemas de búsqueda de respuestas (BR) o Question Answering.

156

8. Sistemas de Búsqueda de Respuestas

Los sistemas de BR aparecen como potenciales sucesores de los buscadores tradicionales.

Diferencias entre BR y RI:

- Entrada: BR espera una pregunta completa.
- Salida: BR devuelve una respuesta concreta.
- El nivel de comprensión del texto.
- El volumen de texto sobre el que se realiza la búsqueda.
- RI: query-driven. QA: answer-driven.

157

Results for query: **When was the telephone invented?**

When was the telephone invented?

Dates	Occurrences of "1876"
1. 1876	St. Louis Commerce Magazine St. Louis RCGA Navigation TELECOMMUNICATIONS CELL PHONES TAKE OFF CELL PHONES TAKE OFF Cell phones are becoming a necessity not only at home, but abroad. The telephone was invented in 1876, but it wasn't until 1927 that the first intercontinental telephone conversation took place from New York to London, costing \$ 75 for three minutes. Fifty years later, calls overseas were still rather uncommon primarily because of the inconvenience and cost. A business traveler at a bigger hotel overseas could dial direct at an exorbitant rate, but someone calling from a smaller establishment would have to go through an international operator and possibly even have to wait for the http://www.stlcommercemagazine.com/archives/november1999/cellphone.html ~ go ~ raw
2. 1879	
3. 1875	
4. 1878	
5. 1927	
6. 1874	and usually archives are kept of software on the board. The first board worthy of the name was Ward Christensen and Randy Suess's board in 1978. BEDBUG - A virus type program that another programmer inserts into an existing program, with the intention of causing havoc. Usually not serious - it is coded so the results look like a software bug, not a true virus. Sometimes makes copies of itself. BEGINNER'S ALL - PURPOSE SYMBOLIC INSTRUCTION CODE - see BASIC BELL, [Professor]
7. 1888	ALEXANDER GRAHAM - The guy who invented the telephone in 1876. The man who created cyberspace, in its early, pathetic stage when no one thought it would be anything. BELLSOUTH - Atlanta RBOC that was supposedly very easy to hack; some rumors claim they eventually spent two million dollars on security. BERNIE S. - Handle of Edward "Ed" Cummings. Phreak put in an uncomfortable and unconstitutional imprisonment for possession of computer programs that "could be used for fraud" by the United States Secret Service. This essentially happened because ly happened because
8. 1922	http://insecure.org/stf/hackenc.txt ~ go ~ raw
9. 1877	
10. 1910	

Lists

158

8. Sistemas de Búsqueda de Respuestas

- # QA multimodal (imágenes, voz, vídeo, etc.).
- # QA monomodal (sólo texto):
 - QA multilingüe.
 - QA monolingüe:
 - QA en dominios restringidos (*domain specific*).
 - QA independiente del dominio (*open domain*):
 - # QA sobre texto estructurado (*structured data*).
 - # QA sobre texto no estructurado (*free text*):
 - Web
 - Colecciones estables (*fixed set of collections*)
 - Un solo documento (*Reading Comprehension Test*)

159

8. Sistemas de Búsqueda de Respuestas

- # Problemática de QA (Eurolan 2005-Bernardo Magnini-Marius Pasca):
 - Presencia parcial o total de los SN de la pregunta: en el mismo SN o en diferentes
 - What is the **brightest star** visible from **Earth**?
 - **Sirio** A is the **brightest star** visible from **Earth** even if it is...
 - the planet is 12-times brighter than **Sirio**, the **brightest star** in the sky...

160

8. Sistemas de Búsqueda de Respuestas

- # Problemática de QA (Eurolan 2005-BM-MP):
 - ▀ Conversiones entre categorías morfológicas: *autor-escriptor-escribir* y *no cantar*.
 - ▀ Who is the *author* of the “*Star Spangled Banner*”?
 - ▀ ...*Francis Scott Key* wrote the “*Star Spangled Banner*” in 1814
 - ▀ ...comedian-actress *Roseanne Barr* sang her famous rendition of the “*Star Spangled Banner*” before ...

161

8. Sistemas de Búsqueda de Respuestas

- # Problemática de QA (Eurolan 2005-BM-MP):
 - ▀ Descubrir relaciones implícitas entre la pregunta y la respuesta.
 - ▀ Which is the Mozart *birth date*?
 - ▀ Mozart (1751 – 1791)

162

8. Sistemas de Búsqueda de Respuestas

Problemática de QA (Eurolan 2005-BM-MP):

- Descubrir relaciones implícitas entre la pregunta y la respuesta (cont.).
- *Which is the distance between **Naples** and **Ravello**?*
 - From the **Naples** Airport follow the sign to Autostrade (green road sign). Follow the directions to Salerno (A3). Drive for about 6 Km. Pay toll (Euros 1.20). Drive appx. 25 Km. Leave the Autostrade at Angri (Uscita Angri). Turn left, follow the sign to Ravello through Angri. Drive for about 2 Km. Turn right following the road sign "Costiera Amalfitana". Within 100m you come to traffic lights prior to narrow bridge. Watch not to miss the next Ravello sign, at appx. 1 Km from the traffic lights. Now relax and enjoy the views (follow this road for 22 Km). Once in **Ravello** ...

163

8. Sistemas de Búsqueda de Respuestas

Interés reciente en los sistemas de BR:

- Text REtrieval Conferences (TREC).
- TREC-8 (11019), la primera evaluación de sistemas de BR (*large-scale evaluation on Open-Domain QA*).
- Conferencias anuales.
- TREC sponsors:
 - NIST: National Institute of Standards and Technology.
 - ARDA: Advanced Research and Development Activity.
 - DARPA: Defense Advanced Research Projects Agency.

164

8. Sistemas de Búsqueda de Respuestas

Definición del problema de la BR (TREC-9):

- Vision Statement to Guide Research in Q&A and Text Summarization (Carbonell et al., 2000):

- # <http://www-nlpir.nist.gov/projects/duc/papers/Final-Vision-Paper-v1a.pdf>.

- The Roadmap Committee (Burger et al., 2000):

- # http://www-nlpir.nist.gov/projects/duc/papers/qa_Roadmap-paper_v2.doc.

165

8. Sistemas de Búsqueda de Respuestas

Tipos de usuarios de un sistema de BR:

- Usuario casual:

- Información puntual sobre hechos concretos.
 - Preguntas simples con contestación en un solo documento.
 - Ej.: “*Quién inventó el teléfono?*”.

- El recopilador de información:

- Contestaciones en varios documentos.
 - Ej.: “*Qué países visitó el Papa en 1998*”, “*Dime qué jugadores han anotado más de 40 puntos en un partido oficial*”, “*Dime los datos biográficos de Nelson Mandela*”.

166



8. Sistemas de Búsqueda de Respuestas

Tipos de usuarios de un sistema de BR (cont.):

■ El periodista:

- Series de preguntas relacionadas. Importancia de tener en cuenta el contexto de estas preguntas.
- Manejo de distintas fuentes de información (textual, multilingüe, imágenes, ...).
- Ej. *“Datos sobre el terremoto de Turquía del 2002”*.

■ El analista profesional:

- Preguntas complejas: razonamiento, recopilación y síntesis de información, alto grado de interacción con el usuario.
- Ej.: *“Hay conexiones entre estos grupos terroristas?”*.

167



8. Sistemas de Búsqueda de Respuestas

Tipos de usuarios de un sistema de BR (cont.):

■ Situación actual:

- Usuario casual.
- Recopilador de información del mismo tipo (tipo lista del TREC-10).
- El periodista: intento de tener en cuenta el contexto de las preguntas realizadas anteriormente por el usuario (preguntas contextuales del TREC-10).

168

8. Sistemas de Búsqueda de Respuestas

Arquitectura general de los sistemas de BR:

- ✚ Módulo de análisis de la pregunta.
- ✚ Recuperación de documentos (RI).
- ✚ Selección de párrafos.
- ✚ Extracción de la respuesta.

169

8. Sistemas de Búsqueda de Respuestas

Taxonomía de la BR (Moldovan et al. 1999):

Class	KB	Reasoning	NLP/Indexing	Examples and Comments
1	dictionaries	simple heuristics, pattern matching	complex noun, apposition, simple semantics, keyword indexing	Q33: <i>What is the largest city in Germany?</i> A: .. <i>Berlin, the largest city in Germany.</i> Answer is: simple datum or list of items found verbatim in a sentence or paragraph.
2	ontologies	low level	verb nominalization, semantics, coherence, discourse	Q198: <i>How did Socrates die?</i> A: .. <i>Socrates poisoned himself.</i> Answer is contained in multiple sentences, scattered throughout a document.
3	very large KB	medium level	advanced nlp, semantic indexing	Q: <i>What are the arguments for and against prayer in school?</i> Answer across several texts.
4	Domain KA and Classification, HPKB	high level		Q: <i>Should Fed raise interest rates at their next meeting?</i> Answer across large number of documents, domain specific knowledge acquired automatically.
5	World Knowledge	very high level, special purpose		Q: <i>What should be the US foreign policy in the Balkans now?</i> Answer is a solution to a complex, possible developing scenario.

Table 5: A taxonomy of Question Answering Systems. The degree of complexity increases from Class 1 to Class 5, and it is assumed that the features of a lower class are also available at a higher class.

170

8. Sistemas de Búsqueda de Respuestas

El mejor sistema en TREC-9 y TREC-11: FALCON (Harabagiu et al. 2000):

- Incluye una serie de tres bucles anidados:
 - Intentan realizar modificaciones progresivas para mejorar la calidad de la respuesta extraída.
 - Intentan justificar la respuesta.
- Utiliza una gran cantidad de recursos de PLN:
 - Análisis sintáctico completo.
 - Análisis semántico.
 - WordNet.
- Se destaca la fase de validación de la respuesta.

171

8. Sistemas de Búsqueda de Respuestas

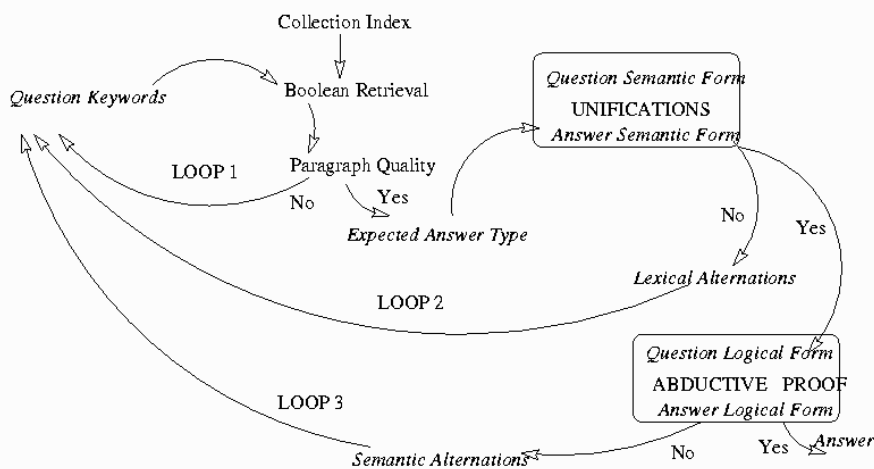


Figure 2: Refining the Answer Search by Boosting Knowledge into FALCON

172

8. Sistemas de Búsqueda de Respuestas

Módulo de análisis de la pregunta de Falcon:

- ▀ Se realiza un análisis sintáctico completo de la pregunta.
- ▀ Se extraen entidades y su tipo semántico.
- ▀ Se determina el “tipo de respuesta esperada” a partir de la entidad que esté más conectada con las otras entidades de la pregunta.
- ▀ El tipo de respuesta sigue una taxonomía extraída de WordNet.

173

8. Sistemas de Búsqueda de Respuestas

Módulo de recuperación de pasajes de Falcon:

- ▀ Utiliza:
 - ▀ Modelo Booleano. Objetivo: garantizar un número mínimo de pasajes.
 - ▀ Análisis de palabras clave,
 - ▀ Conceptos relacionados a partir de WordNet,
 - ▀ Tipo de respuesta esperada.
- ▀ Bucle 1:
 - ▀ Este módulo se repite utilizando diferentes combinaciones de términos:
 - # Reduciendo progresivamente el número de términos enviados ([país AND invadió AND kuwait] → [invadió AND kuwait])

174

8. Sistemas de Búsqueda de Respuestas

Módulo de extracción de respuestas de Falcon:

■ Bucle 2:

- Los pasajes se analizan semánticamente.
- Se realiza un proceso de unificación entre los tipos semánticos de las preguntas y los pasajes.
- Si el proceso de unificación falla para todos los pasajes:
 - # Se recupera un nuevo conjunto de pasajes utilizando sinónimos y variaciones morfológicas de los términos de la pregunta:
 - [paper AND clip AND (invented OR inventor OR invent)]
- Si tiene éxito:
 - # Bucle 3.

175

8. Sistemas de Búsqueda de Respuestas

Módulo de extracción de respuestas (cont.):

■ Bucle 3:

- Los tipos semánticos se traducen a fórmulas lógicas.
- Se aplica un proceso de comprobación lógica para justificar la calidad de la respuesta (*abductive backchaining*):
 - # Si tiene éxito: se devuelve como respuesta.
 - # Si falla: se extrae de WordNet los términos relacionados semánticamente con los términos de la pregunta para extraer un nuevo conjunto de pasajes (bucle 1).

176

8. Sistemas de Búsqueda de Respuestas

Ejercicio 18: Dados los siguientes tipos de pregunta:

- EntidadPersona();
- EntidadGrupo();
- EntidadLugarPais();
- EntidadLugarCiudad();
- EntidadLugarCapital();
- EntidadLugar();
- EntidadAbreviatura();
- EntidadEvento();
- EntidadObjeto();
- TemporalFecha();
- TemporalAnyo();
- TemporalMes();
- NumericoCantidad();
- NumericoEconomico();
- NumericoPorcentaje();
- NumericoMedida();
- NumericoPeriodo();
- NumericoEdad();
- Definicion();

Clasifica las siguientes preguntas, indicando un algoritmo para su clasificación:

```
0001¿,Qué año le fue concedido el premio Nobel a Thomas Mann?|(EFE19950605-02530;1929)
0002¿,Cuánto aumenta la población mundial cada año?|(EFE19940601-00641;1,6 por ciento)
0003¿,Cuál es el nombre de pila del juez Borsellino?|(EFE19940718-10595;Paolo)
0004¿,Cuántos miembros de la escolta murieron en el atentado contra el juez Falcone?|(EFE19940522-13027;Tres)
0005¿,Qué cargo detenta Ariel Sharon?|(EFE19940527-16249;Diputado del bloque Likud)
0006¿,Qué año comenzó la Intifada?|(EFE19940905-02123;1987), (EFE19940623-14425;1987)
0007¿,Qué empresa de automóviles produce el "Escarabajo"?|(EFE19951022-14530;Volkswagen)
0008¿,Dónde tiene lugar el Motorshow?|(NIL)
0009¿,Cómo murió Jimi Hendrix?|(EFE19940308-05220;Intoxicación por consumo de barbitúricos)
0010¿,Qué es el Mossad?|(EFE19940719-11230;los servicios secretos israelíes), (EFE19940722-13340;servicio inteligencia Israel), (EFE19940723-13495;servicios secretos de Israel)
0011¿,De qué nacionalidad eran los petroleros que causaron la catástrofe ecológica cerca de Trinidad y Tobago en 1979?|(NIL)
0012¿,De qué está recubierto el continente antártico?|(EFE19950908-05269;Hielo)
0013¿,Qué es lo que causa el agujero de ozono?|(EFE19941219-11009;CFC), (EFE19941018-10390;Los clorofluorocarbonados)
0014¿,Qué es UNICEF?|(EFE19950216-10403;El Fondo de las Naciones Unidas para la Infancia), (EFE19940103-00921;Fondo de las Naciones Unidas para la Infancia), (EFE19940406-02324;Fondo de las Naciones Unidas para la Infancia), (EFE19950609-05605;Fondo de Naciones Unidas para la Infancia)
0015¿,Qué firma está acusada de explotación laboral infantil?|(NIL)
0016¿,Dónde fue fundada Greenpeace?|(NIL)
0017¿,Cómo se transmite el virus del ébola?|(EFE19951222-13891;solamente por contacto con pacientes infectados), (EFE19950516-10808;A través de las secreciones de las personas enfermas)
0018|Nombre un pesticida.|(EFE19950606-03267;DDT)
0019¿,En qué día cae el Día del Año Nuevo Chino?|(EFE19950131-18155;31 ene)
0020¿,Cuándo renunció Nixon?|(EFE19940127-13992;9 de agosto de 1974), (EFE19940424-13763;1974)
```

177

9. Sistemas de RI multimedia

✚ Objetivos:

- RI sobre documentos que incluyen audio, vídeo, imágenes, gráficos, presentaciones, hojas de cálculo, además del tradicional texto

✚ Aplicaciones:

- *Summarization/feature extraction*: obtención de características:
 - Preprocesamiento de los datos para permitir su acceso y uso de forma eficiente
 - Histogramas de colores, detección de formas, transcripción de notas musicales, melodías, etc.
- *Filtering*: eliminación de redundancias
- *Categorization* (clasificación): géneros de música o películas
- Detección de plagios

178

9. Sistemas de RI multimedia

Aplicaciones de imágenes:

- *Facial recognition* (reconocimiento de caras)
- Control de seguridad física:
 - Reconocimiento de intrusos en cámaras de seguridad
 - Seguimiento de pacientes
 - Contar número de personas en una manifestación o centro comercial
- *Medical multimedia document management*:
 - Búsqueda de casos parecidos
 - Detección y clasificación de tumores a partir de radiografías
 - Descubrimiento de relaciones entre áreas del cerebro con actividades del cuerpo humano utilizando imágenes fMRI del cerebro
- Información meteorológica:
 - Análisis comparativo del tiempo en diferentes épocas
- *Image meta search vs. content-based image retrieval*

179

9. Sistemas de RI multimedia

Image meta search:

- Búsqueda por el etiquetado textual de las mismas
- La query es un texto explicando lo que el usuario busca: tarea igual a la RI tradicional
- Problemas:
 - Ambigüedad del etiquetado:
 - ✦ El etiquetado de las imágenes almacenadas puede no alcanzar el nivel de detalle deseado por el usuario
 - ✦ Subjetividad: “una imagen es mejor que mil palabras”
 - ✦ *Users spam*
 - Proceso costoso: nº cada vez mayor de imágenes a etiquetar

180

9. Sistemas de RI multimedia

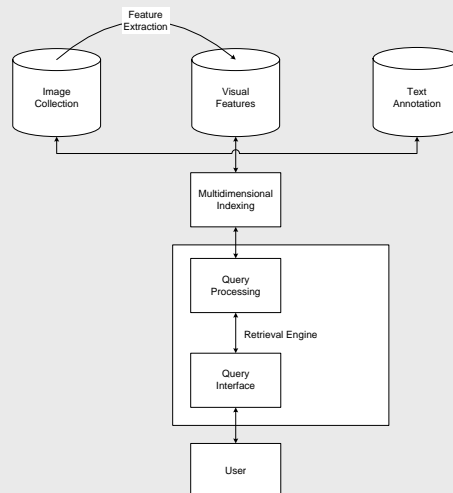
Content-based image retrieval:

- Búsqueda por el contenido de la imagen y no por el etiquetado textual de las mismas.
- La query es una imagen
- Se calcula la distancia/similitud de la imagen original respecto las imágenes almacenadas: técnicas de reconocimiento de formas (*pattern matching*)
- Información utilizada:
 - Color: colores utilizados, histogramas de colores, proporción de colores (para independizarlo del tamaño de la imagen)...
 - Textura: *Autocorrelation*, *Wavelet transforms*, *Gabor Filters*, contraste, etc.
 - Formas detectadas: *Edge Detectors*, *Face Detection*, segmentación de regiones, etc.
 - Distribución espacial
 - Todo lo anterior aplicado por segmentos de la imagen
- Problema: baja precisión

181

9. Sistemas de RI multimedia

Content-based image retrieval (Professor Dick Hartley):



182

Explotación de la información. Recuperación de Información

9. Sistemas de RI multimedia

Query by image:

183

Explotación de la información. Recuperación de Información

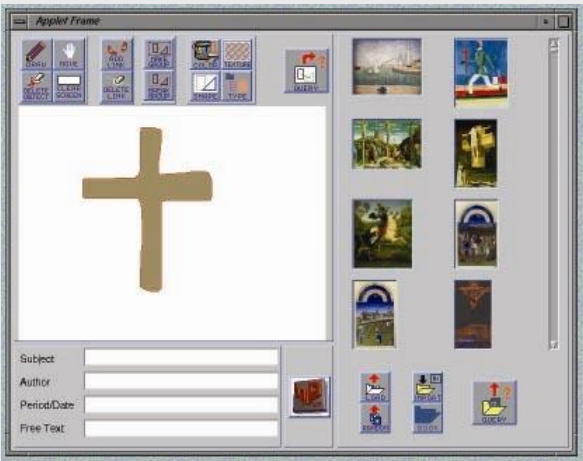
9. Sistemas de RI multimedia

Query by paint:

184

9. Sistemas de RI multimedia

Query by sketch:



185

9. Sistemas de RI multimedia

A picture is worth a thousand (coherent) words: building a natural description of images

■ <http://googleresearch.blogspot.co.uk/2014/11/a-picture-is-worth-a-thousand-coherent.html>

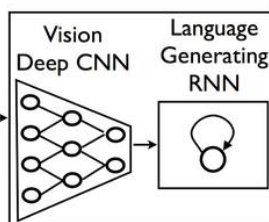


Automatically captioned: "Two pizzas sitting on top of a stove top oven"

186



9. Sistemas de RI multimedia



A group of people shopping at an outdoor market.

There are many vegetables at the fruit stand.

The model combines a vision CNN with a language-generating RNN so it can take in an image and generate a fitting natural-language caption.

187



9. Sistemas de RI multimedia

Charla de cómo se crean sistemas de reconocimiento de imágenes y se enlazan con sistemas de PLN:

▀ [http://www.ted.com/talks/fei fei li how we re teaching computers to understand pictures#t-664209](http://www.ted.com/talks/fei_fei_li_how_we_re_teaching_computers_to_understand_pictures#t-664209)

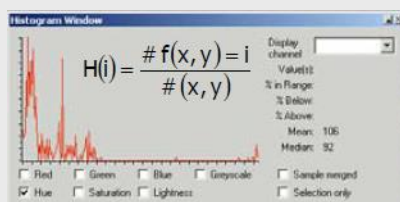
188



9. Sistemas de RI multimedia

Histograma de una imagen:

- Una vez elegido el número de colores n de la imagen, el histograma de una imagen es un vector H de dimensión n , tal que $H(i)$ representa la fracción de píxeles de color i presentes en la imagen



189



9. Sistemas de RI multimedia

Histograma de una imagen:

- Ventajas:**
 - Es fácil de calcular
 - Insensible a rotaciones, zoom y cambios de resolución
- Problemas:**
 - Los colores pueden no ser cruciales para entender el contenido de la imagen
 - Sensible a cambios de iluminación

Conjuntos de Colores:

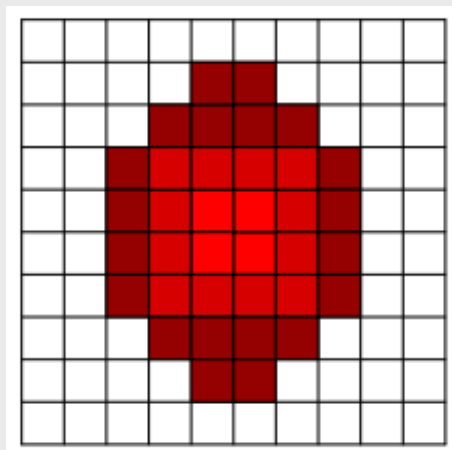
- Los conjuntos de colores introducen una representación simplificada del histograma
- El contenido de la imagen se representa mediante un vector binario cuyos valores (1 ó 0) corresponden a la presencia significativa o no de un color
- Los colores que forman el conjunto de color se toman de un diccionario de nombres asociados a los colores básicos

190



9. Sistemas de RI multimedia

Histograma de una imagen: [4, 12, 20, 64]



191



9. Sistemas de RI multimedia

Ejercicio 19: Histograma de una imagen

- ▀ Calcular la imagen más parecida a la 1, con los histogramas (7 colores):

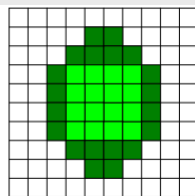
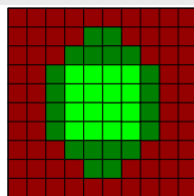
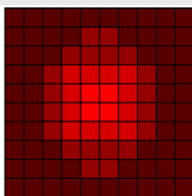
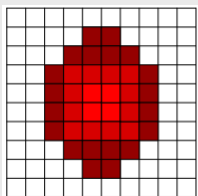
1: [0, 4, 12, 20, 64, 0, 0]

2: [64, 4, 12, 20, 0, 0, 0]

3: [0, 0, 0, 64, 0, 20, 16]

4: [0, 0, 0, 0, 64, 20, 16]

$$D_H(I_Q, I_D) = \frac{\sum_{j=1}^n |H(I_{Q,j}) - H(I_{D,j})|}{\sum_{j=1}^n H(I_{D,j})}$$



192

9. Sistemas de RI multimedia

Segmentación de una imagen:

- ▀ Particionamiento en regiones disjuntas que sean uniformes respecto a alguna característica (por ejemplo, el color)
- ▀ Es un proceso complejo
- ▀ Una vez segmentada la imagen, cada región individual puede ser descrita mediante los siguientes atributos:
 - ▀ Baricentro, Área, Color Medio

193

9. Sistemas de RI multimedia

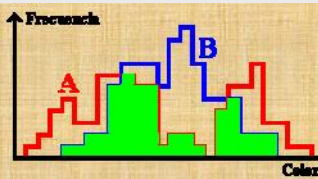
Búsqueda por color:

- ▀ Distancia de Histogramas:
 - ▀ Dado un histograma consulta $H(I_q)$ y un histograma imagen $H(I_d)$, ambos con n bins, su distancia de intersección de histograma viene dada por

$$d_h(I_q, I_d) = \left[\sum_{j=1}^n (H(I_q, j) - H(I_d, j))^p \right]^{1/p}$$

Lp Norma
 $p=1$ distancia city-block
 $p=2$ distancia Eculídea

Dados dos histogramas **A** y **B**, la frecuencia de color mínima entre bins homólogos viene dada por la intersección $A \cap B$



194



9. Sistemas de RI multimedia

Modelado de la forma:

- Un **eje** es un cambio significativo en una propiedad de la imagen (usualmente, la intensidad o el color)
- Un **contorno** es un grupo de ejes unidos espacialmente bajo alguna condición
- Un **objeto** se define a través de un contorno cerrado que lo envuelve
- Los **descriptores de forma** son medidas que cuantifican los objetos
 - Forma global: sólo necesitan la imagen de ejes
 - Forma local: asumen conocido el contorno del objeto a describir

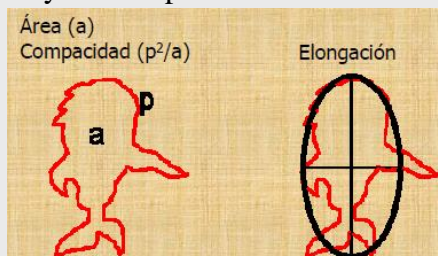
195



9. Sistemas de RI multimedia

Modelado de la forma:

- Forma global: sólo necesitan la imagen de ejes
- Forma local: asumen conocido el contorno del objeto a describir.
 - Se considera una imagen binaria con valor 1 para los puntos del contorno y valor 0 para el resto

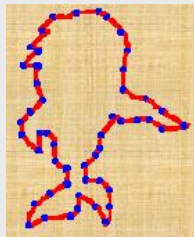


196

9. Sistemas de RI multimedia

Modelado de la forma:

- Se subdivide la curva en segmentos y se caracteriza cada segmento por su longitud y su orientación
- La forma del objeto queda descrita por la secuencia $(l_1, \theta_1) (l_2, \theta_2) \dots (l_n, \theta_n)$
- Se describe la forma por la función Radio frente a Ángulo calculada desde el baricentro del objeto



197

9. Sistemas de RI multimedia

Búsqueda por forma:

- Para formas aproximadas mediante curvas poligonales que usan propiedades locales como A_v (ángulo del vértice), D_v (distancia al siguiente vértice), X_v e Y_v (coordenadas del vértice)
- La similaridad se calcula midiendo el número de cambios necesarios para transformar una curva en otra


198


199

200


9. Sistemas de RI multimedia. Content-based image retrieval


Ejemplo:


Color 





Results

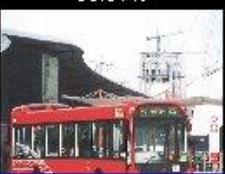
 Search!


53.92%


53.51%


53.31%

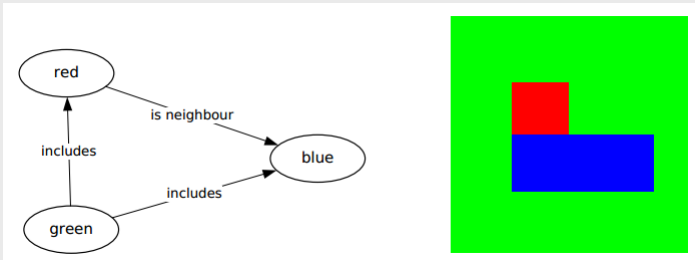

53.25%



201

9. Sistemas de RI multimedia. Content-based image retrieval

Integración de características mediante grafos:



202



Explotación de la información. Recuperación de Información


9. Sistemas de RI multimedia.

Content-based image retrieval

Tamura Texture Feature (Tamura et al. 1978):

- ▣ coarseness – coarse vs. fine
- ▣ contrast – high vs. low
- ▣ directionality – directional vs. non-directional
- ▣ line-likeness – line-like vs. blob-like
- ▣ regularity – regular vs. irregular
- ▣ roughness – rough vs. smooth

203



Explotación de la información. Recuperación de Información

9. Sistemas de RI multimedia.

Content-based image retrieval

Reconocimiento de objetos:

- ▣ Reconocimiento de señales de tráfico:
 - ▣ Más sencillo: formas/colores/posición muy definidos
- ▣ Reconocimiento de caras:

Areas	Specific applications
Entertainment	Video game, virtual reality, training programs
	Human-robot-interaction, human-computer-interaction
Smart cards	Drivers' licenses, entitlement programs
	Immigration, national ID, passports, voter registration
	Welfare fraud
Information security	TV Parental control, personal device logon, desktop logon
	Application security, database security, file encryption
	Intranet security, internet access, medical records
	Secure trading terminals
Law enforcement and surveillance	Advanced video surveillance, CCTV control
	Portal control, postevent analysis
	Shoplifting, suspect tracking and investigation

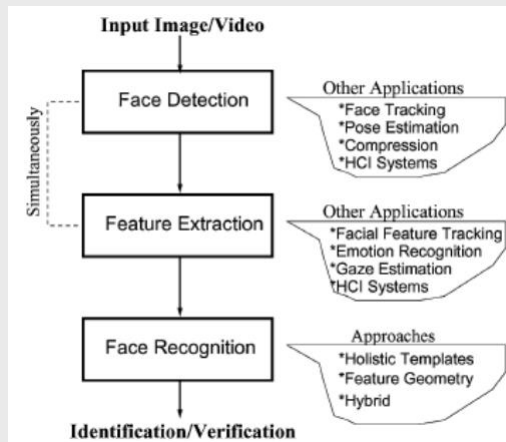
204

9. Sistemas de RI multimedia. *Content-based image retrieval*

Reconocimiento de caras:

■ Métodos:

- Templates
- Color de piel
- Posición ojos, nariz, ...
- Redes neuronales



205

9. Sistemas de RI multimedia

Content-based video retrieval (TRECVID 2007, VideoCLEF 2008, ImageCLEF):

- Igual que las imágenes
- Se etiqueta/detecta información adicional:
 - Las acciones que ocurren en el vídeo:
 - # Segmentación de escenas
 - # Heurísticas como el cambio de luz o color
 - # *Speech recognition*
 - Información temporal
 - Localización
 - MPEG-7: Interfaz de Descripción del Contenido Multimedia. Representación estándar de la información audiovisual que permite la descripción de contenidos (metadatos) para:
 - # Palabras clave
 - # Significado semántico (quién, qué, cuándo, dónde)
 - # Significado estructural (formas, colores, texturas, movimientos, sonidos)

206

9. Sistemas de RI multimedia

Content-based music/audio retrieval:

- ▀ Se extrae información sobre la que realizar el análisis:
 - ▀ Notas musicales, ritmo, tono, melodía, instrumentos musicales predominantes
 - ▀ Reconocimiento del habla
- ▀ Aplicaciones:
 - ▀ Detección de plagios
 - ▀ Clasificación de géneros musicales
 - ▀ Sugerencia de músicas o géneros similares

207

9. Sistemas de RI multimedia

Documentación adicional:

- ▀ Antonio Mosquera González
(<http://gva1.dec.usc.es/~antonio/docencia/20042005rdli/>)
- ▀ Visual Information Retrieval. Alberto del Bimbo; Morgan Kaufmann Publishers, Inc.; ISBN: 1-55860-624-6; 1999
- ▀ Recuperación de Imágenes en Bases de Datos a partir del Color y la Forma. José Manuel Fuentes García; Tesis Doctoral; Universidad de Jaén; 1999

208



9. Sistemas de RI multimedia

Documentación adicional:

- An introduction to image retrieval. Professor Dick Hartley. Manchester Metropolitan University
 - <http://eureka.lib.teithe.gr:8080/bitstream/handle/10184/137/Hartley1.ppt?sequence=1>
- VK Multimedia Information Systems. Mathias Lux, ITEC, Klagenfurt University, Austria – Multimedia Information Systems:
 - <http://www.itec.uni-klu.ac.at/~mlux/teaching/mmis08/slides/slides-05.pdf>