Gestión de Información en la Web
Máster en Ingeniería Informática
Parte II: Recuperación de información y Sistemas de recomendación

Tema 4: Recuperación de información
4.2. Procesado de documentos e indexación

Juan Manuel Fernández Luna
Dpto. Ciencias de la Computación e Inteligencia Artificial
jmfluna@decsai.ugr.es

ÍNDICE

ÍNDICE

INTRODUCCIÓN

El objetivo de la RI es acceder, lo más eficiente y precisamente posible, al conjunto de documentos que está relacionado en mayor grado con la consulta.

Pero... originalmente las colecciones documentales están en un formato que no puede ser tratado directamente por un Sistema de Recuperación de Información (SRI).

INTRODUCCIÓN

Por tanto, se necesita un proceso por el cual se convierta dicha colección a un formato fácilmente manejable por el SRI.

Indexación: creación de las estructuras de datos adecuadas para permitir una acceso eficiente y eficaz a los documentos.

INTRODUCCIÓN

Pero... ¿Necesitamos almacenar todas las palabras que aparecen en cada documento?

- Necesitaríamos un espacio de almacenamiento enorme.
- La calidad de la recuperación será mala.
- El tiempo de recuperación será muy elevado.

Solución:

Realizar un proceso previo para seleccionar aquellas palabras que realmente son útiles para la recuperación (términos de indexación).

INTRODUCCIÓN

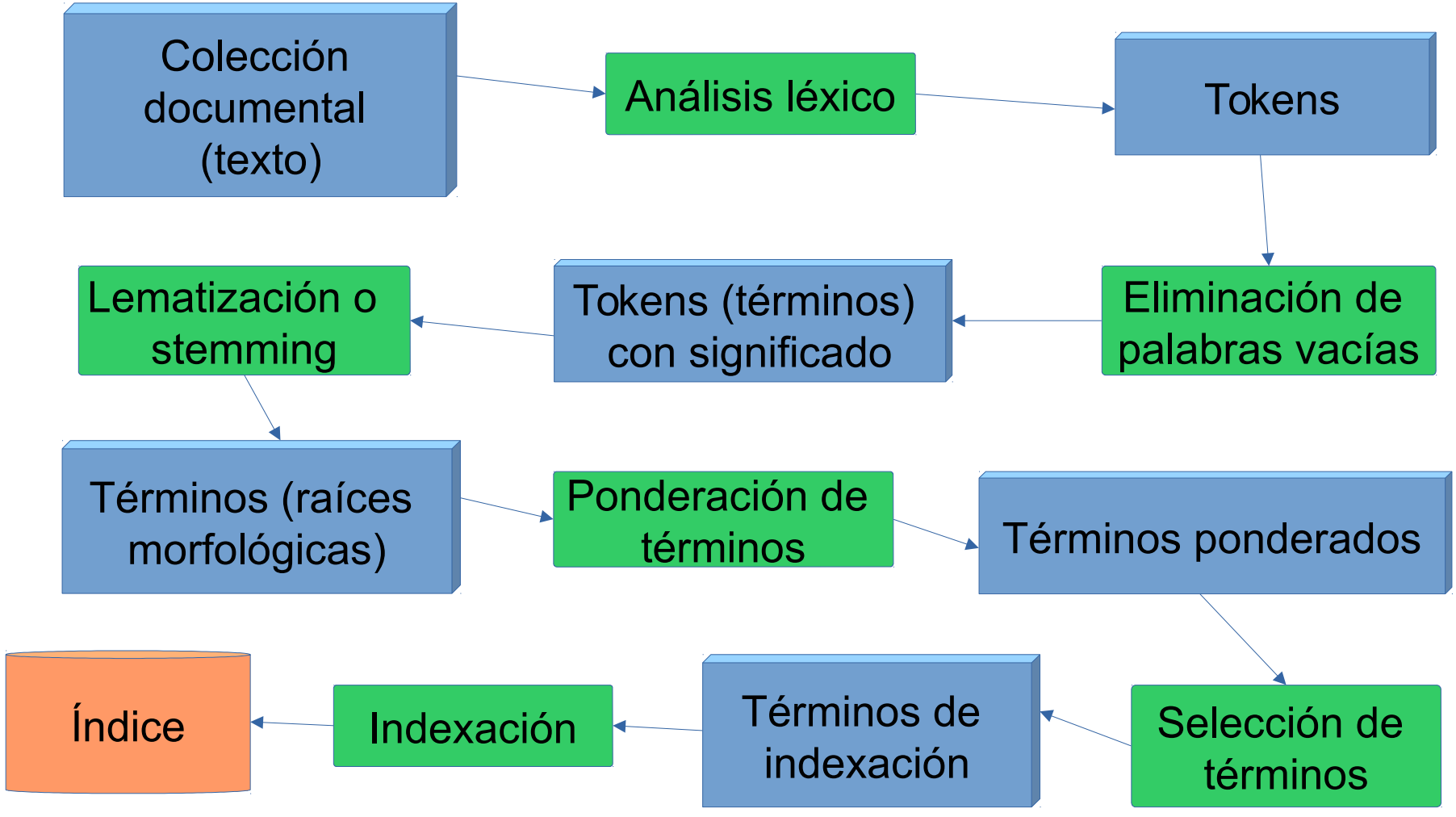
- Algunas palabras contienen más “significado” que otras (Por ejemplo, los sustantivos o los verbos).

- Por tanto, merece la pena “preprocesar” los documentos con objeto de determinar qué palabras actuarán como “términos de indexación”.

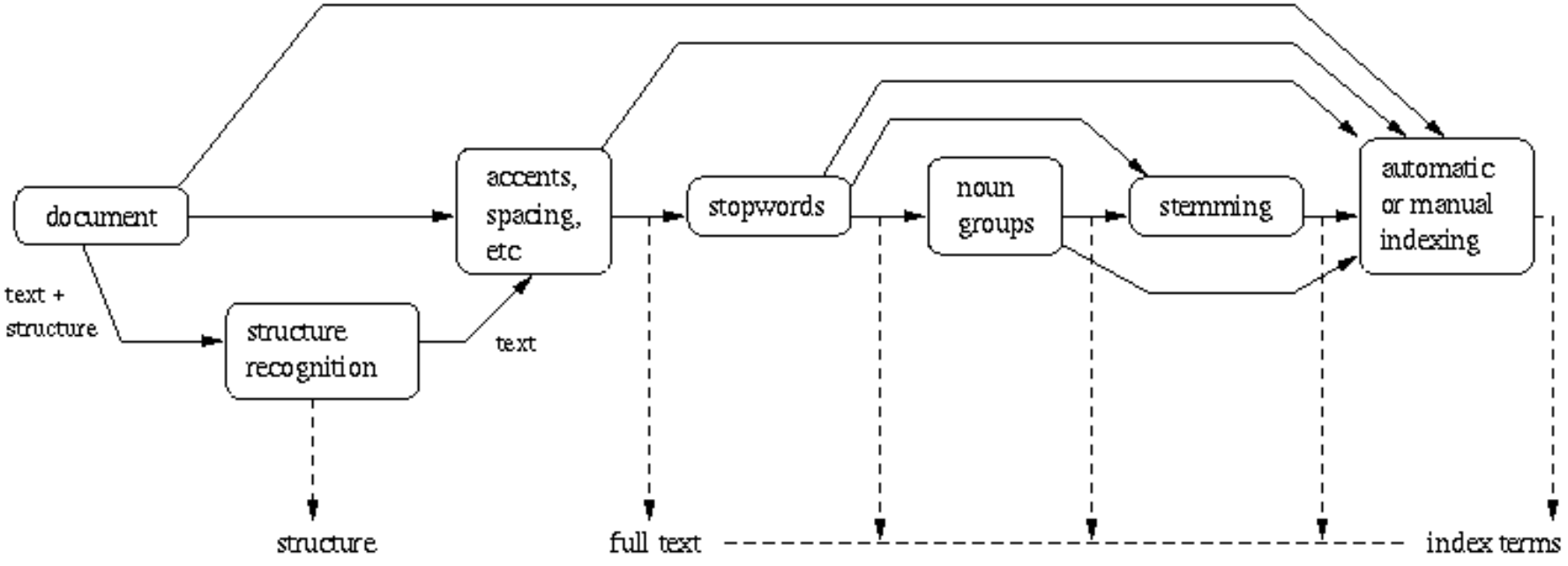
- Varias tareas...

ÍNDICE

EL PROCESO GENERAL DE INDEXACIÓN



EL PROCESO GENERAL DE LA INDEXACIÓN



(Imagen obtenida del libro Modern Information Retrieval)

ÍNDICE

PASOS DEL PROCESAMIENTO DOCUMENTAL

1. Análisis léxico (tokenizing).
2. Eliminación de palabras vacías (stop words).
3. Segmentación (Stemming) o lematización.
4. Ponderación de términos (weighting).
5. Selección de los mejores términos.
6. Construcción del índice.

PASOS DEL PROCESAMIENTO DOCUMENTAL

1. Análisis léxico:

- Proceso por el cual el texto (secuencia de caracteres) queda separado en secuencias de tokens (palabras).
- Los tokens son agrupaciones de caracteres con un significado colectivo.

PASOS DEL PROCESAMIENTO DOCUMENTAL

1. Análisis léxico

Identificación de las palabras:

- Espacios en blanco.
- Dígitos.
- Guiones.
- Signos de puntuación.
- Letras mayúsculas al comienzo.

PASOS DEL PROCESO DOCUMENTAL

1. Análisis léxico

¿Qué hacemos con...?

Números:

- Normalmente ignorarlos.
- Pero... podemos encontrarnos con fechas (2006, 450 A.C.,...) o con otros que representen informaciones relevantes al dominio.
- Mezclados con letras pueden tener sentido (por ejemplo, CC123, B52, H2O).

PASOS DEL PROCESO DOCUMENTAL

1. Análisis léxico

¿Qué hacemos con...?

Guiones:

- Tratados como separadores: *state-of-art* → *state of art*.
- Se ignoran: *on-line* → *online*.
- Mantenerlos: Algoritmo de *Knuth-Morris-Pratt.*

Signos de puntuación:

- Normalmente los ignoramos.

PASOS DEL PROCESO DOCUMENTAL

Análisis léxico

¿Qué hacemos con...?

Mayúsculas y minúsculas:

- Se pasan a minúsculas.

Este proceso de análisis léxico suele implementarse mediante un autómata de estados finitos.

2. Eliminación de palabras vacías

Justificación: La ley de Zipf.

Frecuencia: cuántas veces aparece una palabra en una colección.

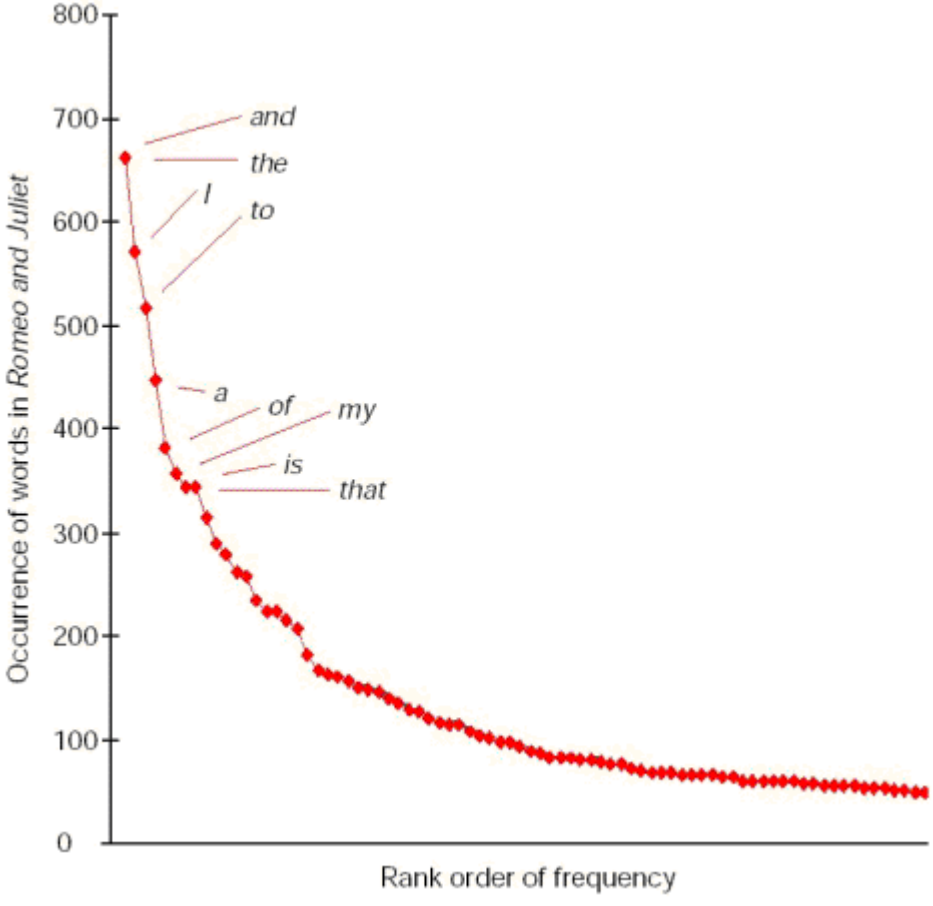
Posición en la ordenación: se ordenan dichas palabras según su frecuencia y nos quedamos con la posición en dicha ordenación.

El producto de la frecuencia (f) y su posición en la ordenación (r) es aproximadamente constante:

$$\text{frecuencia} \times \text{posición ordenación} \approx \text{constante}$$

PASOS DEL PROCESO DOCUMENTAL

2. Eliminación de palabras vacías



2. Eliminación de palabras vacías

Palabra	Frecuenc.	Posición	f * r
the	3332	1	3332
and	2972	2	5944
a	1775	3	5235
he	877	10	8770
be	294	30	8820
there	222	40	8880
one	172	50	8600
friends	10	800	8000

PASOS DEL PROCESO DOCUMENTAL

2. Eliminación de palabras vacías

Observando la gráfica:

- Unas pocas palabras ocurren muy frecuentemente.
- Un número medio de palabras tienen una frecuencia media.
- Muchas palabras ocurren muy infrecuentemente.

2. Eliminación de palabras vacías

Consecuencias:

- Siempre existe un conjunto de palabras muy frecuentes que no son buenos discriminadores del contenido del documento.
- Existe también un grupo grande de palabras que sólo ocurren una vez.
- Las palabras con frecuencia media son las más descriptivas.

2. Eliminación de palabras vacías

Palabras vacías: no tienen significado.

Artículos, determinantes, pronombres, preposiciones,...

Son palabras muy frecuentes (más del 80%) y poco útiles para la recuperación: ahorramos espacio y ganamos eficacia recuperadora.

2. Eliminación de palabras vacías

Dependen del idioma.

Y del dominio:

Podemos incluir palabras muy comunes que aportan poco en colecciones concretas: “ordenador” en una colección de documentos sobre informática.

Eliminación de palabras vacías

En español:

- *un, una, unas, unos, uno, sobre, todo, también, tras, otro, algún, alguno, alguna, algunos, algunas, ser, es, soy, eres, somos, sois, estoy, esta, estamos, estáis, estan, como, en, para, atrás, porque, por qué, estado, estaba, ante, antes, siendo, ambos, pero, por, poder puede, puedo, podemos, podéis, pueden fui, fue, fuimos, fueron, ...*

En inglés:

- *a, about, above, across, after, again, against, all, almost, alone, along, already, also, although, always, among, an, and, another, any, anybody, anyone, anything, anywhere, are, area, areas, around, as, ask, asked, asking, asks, at, away, b, back, backed, backing, backs, be, because, became, ...*

2. Eliminación de palabras vacías

Consulta: ¿Ser o no ser? :-)

Tendencia:

- Lista de palabras vacía pequeña para colecciones generales o usuarios poco experimentados.
-
- Más completa para dominios muy definidos y los usuarios están entrenados en el uso de esas colecciones.

2. Eliminación de palabras vacías

Implementación:

Inclusión de la lista de palabras vacías en una estructura de datos de acceso rápido (tablas Hash, por ejemplo).

Para cada palabra del texto, comprobar si está en dicha estructura. Si lo está, descartarla. En caso contrario, considerarla para la siguiente etapa.

3. Segmentación o lematización

- Métodos usados para reducir el tamaño del vocabulario.
- En lugar de indexar todas las palabras, se buscan sus “representantes” morfológicos, raíces o lemas.
- Dos enfoques diferentes.

3. Segmentación o lematización

- Ventajas:
  - Reducción del vocabulario → eficiencia y ahorro de espacio.
  - Aumenta el número de documentos recuperados.
- Desventajas:
  - Se pierde información sobre la palabra completa.

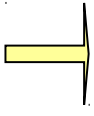
3. Segmentación (stemming)

- Procesamiento de una palabra con objeto de extraer su raíz léxica (lexema).
- La raíz es la parte de la palabra que queda al eliminar afijos (prefijos, infijos y sufijos).
- Un mismo lexema representará a una familia de palabras semántica y morfológicamente relacionadas.

3.1. Segmentación (stemming)

- Ejemplos en inglés:
  - {*connect*ed, *connect*ing *connect*ion, *connect*ions, dis*connect*ed} → ***connect***
  - {*compute*r, *comput*ational, *comput*ation} → ***comput***

for example compressed and compression are both accepted as equivalent to compress.



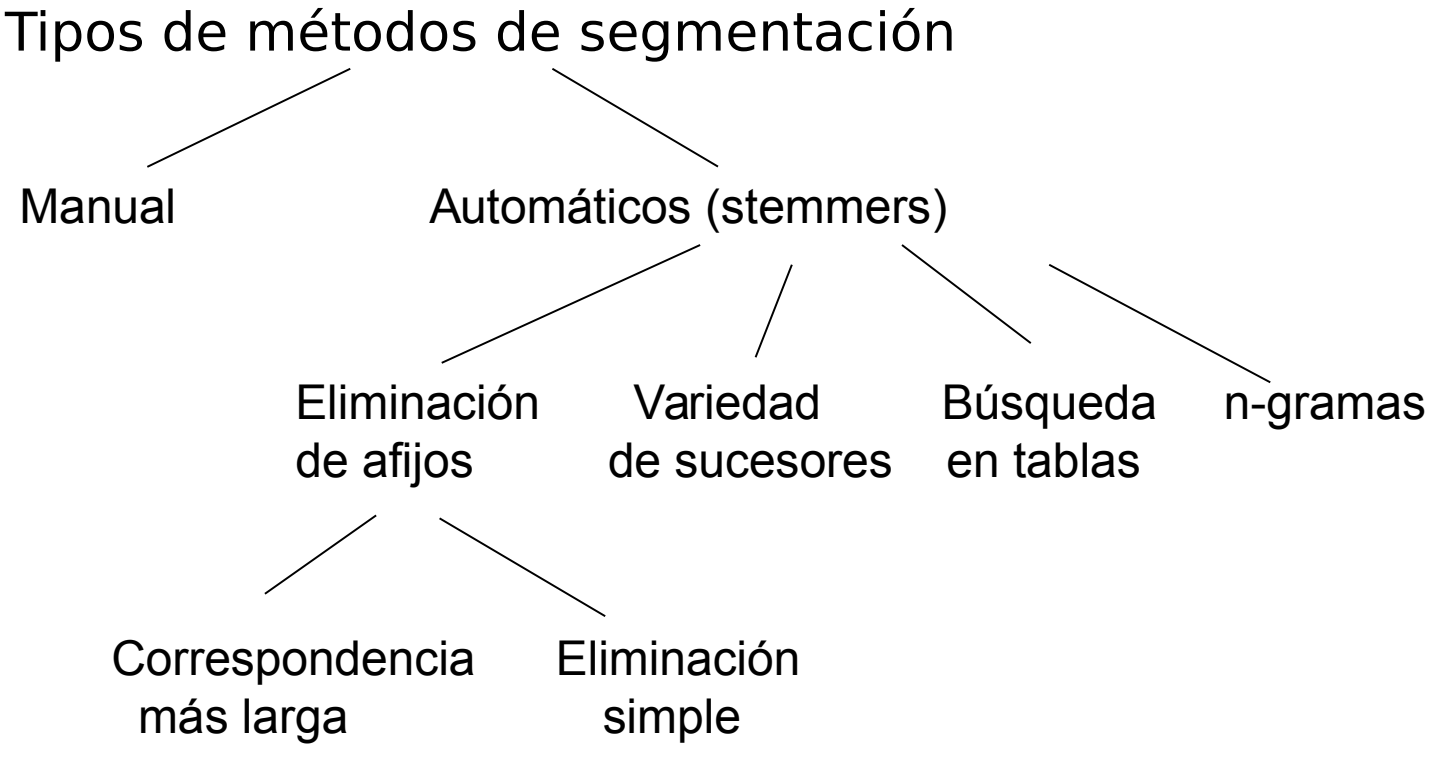
exampl compres compres accept equival compres.

3.1 Segmentación (stemming)

- El análisis morfológico necesario para extraer la raíz es dependiente del idioma y puede llegar a ser muy complejo (como en el español).

- Lo habitual es que los métodos automáticos de segmentación eliminen los afijos de manera iterativa y “ciega”.

3.1. Segmentación (stemming)



Eliminación de afijos

- Basados en métodos heurísticos.
- Aplican sucesivamente reglas a las palabras.
- Palabras diferentes pueden dar lugar a una misma raíz léxica (lexema).

  – Diferentes métodos: Lovins, Slaton, Dawson, **Porter**.

3.1.1. Eliminación de afijos

Algoritmo de Porter:

- Originalmente diseñado para inglés.
- Utiliza una lista de sufijos y prefijos para eliminarlos.
- Aplica una serie de reglas a los afijos de las palabras.
- Puede dar lugar a raíces que no son reconocidas como raíces en el idioma.
- Ambigüedad en ciertos casos y errores en otros.

3.1.1. Eliminación de afijos

Ejemplo:

- Eliminación de plurales, seleccionando la regla con el sufijo más largo:
  - sses $\rightarrow$ ss;
  - ies $\rightarrow$ I;
  - ss $\rightarrow$ ss;
  - s $\rightarrow$ NULL;

stresses pasa a ser *stress.*

3.2. Lematización

Transformación de la palabra al lema al que pertenece.

Por ejemplo:

- Formas verbales a infinitivo: {Fui, van, iremos} → ir
- Plurales a singulares: {casas} → casa
- Femeninos a masculinos: {gata} → gato

3.2 Lematización

Ejemplo:

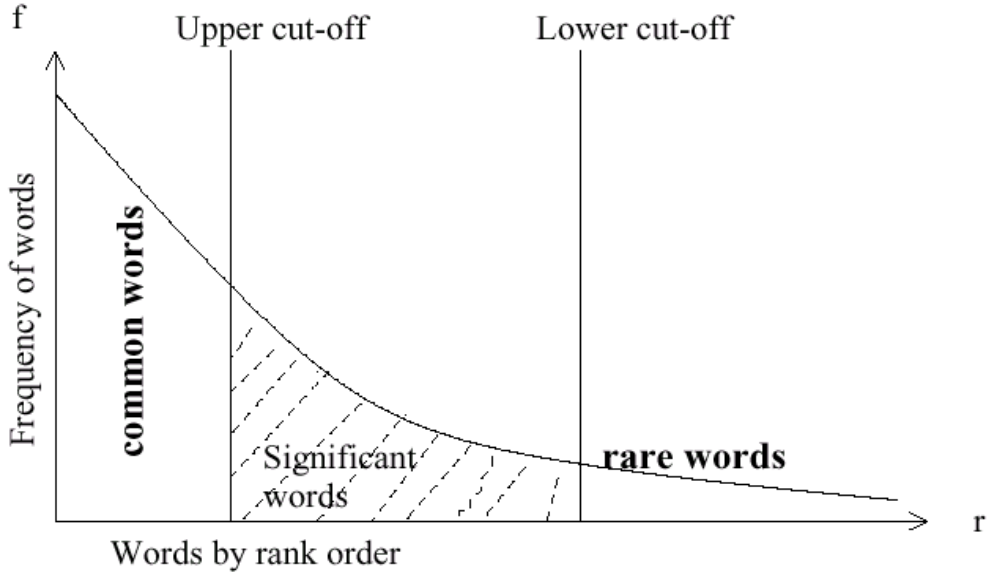
the boy's cars are different colors → the boy car be different color

- Más precisa que la segmentación.
- Necesita de más recursos.
- Necesita de las disciplinas de “Procesamiento del lenguaje natural” y “Lingüística computacional”.

4. Ponderación de términos

- Asignación de un peso a cada término que nos indique la importancia del mismo.
- Capacidad de discriminación de documentos relevantes y no relevantes.
- Esos pesos serán utilizados por los modelos de recuperación para obtener la salida.

4. Ponderación de términos



Palabras importantes (según Luhn):

- Aquellas que son capaces de discriminar el contenido de documentos.
- Se sitúan en medio de dos umbrales.

4. Ponderación de términos

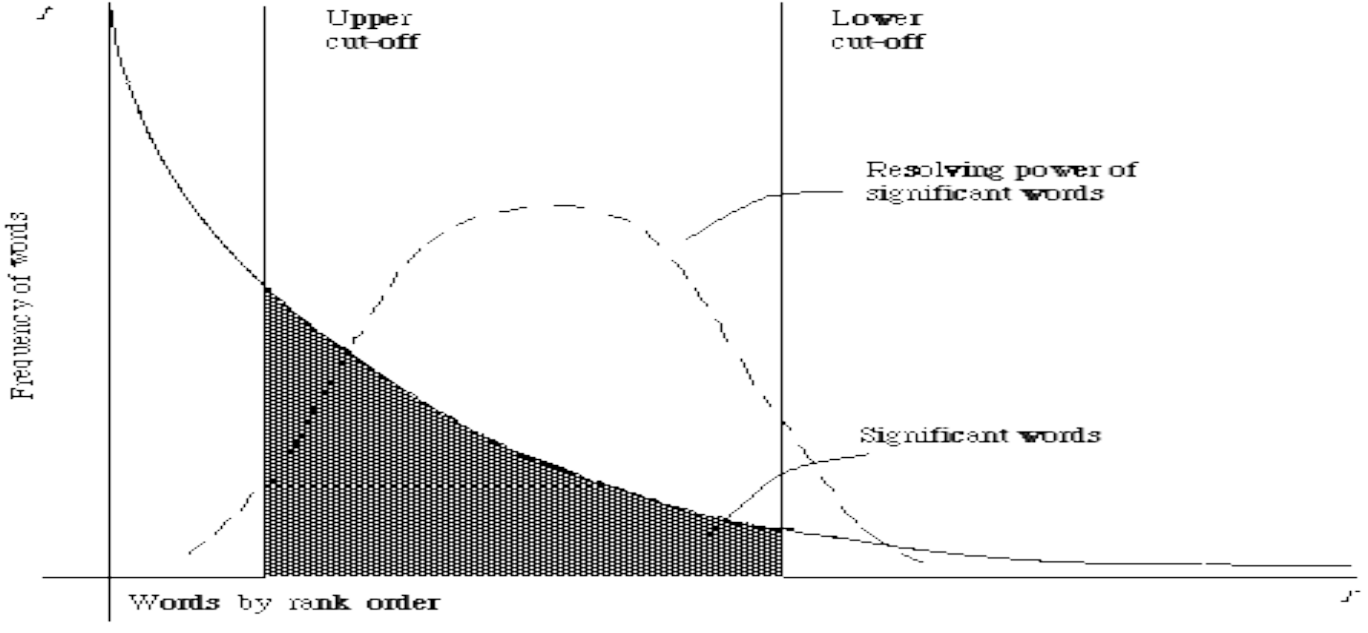


Figure 2.1. A plot of the hyperbolic curve relating f, the frequency of occurrence and r, the rank order. (Adapted from Schultz[44] page 120)

Capacidad de discriminación de las palabras

4. Ponderación de términos

Esquema de ponderación binario:

- Considera la presencia (1) o ausencia de un término en los documentos de la colección.

Sólo tenemos información de que un término aparece en un documento.

4. Ponderación de términos

Esquema basado en la frecuencia de aparición del término i-ésimo (term frequency – tf_{ij}) en el documento j-ésimo.

- Un término que aparezca más veces en un documento será probablemente más importante que lo haga sólo una vez.
- Aunque... los términos aparecerán más en documentos más largos.
- Normalización de la frecuencia por la máxima frecuencia en el documento o por la longitud del mismo.

4. Ponderación de términos

- Frecuencia inversa del término i-ésimo en la colección (inverse document frequency – idf_i).

- Se basa en el hecho de un término que aparece en pocos documentos en la colección tendrá un mayor poder discriminador que los que aparecen en casi todos.

4. Ponderación de términos

Un idf simple:

- Un término t_i aparece en n_i documentos de los N de la colección. Entonces:

$$idf_i = N / n_i$$

- Se daría un peso mayor a los términos que aparecen en menos documentos y un peso bajo a aquellos que aparecen en muchos.

4. Ponderación de términos

Tomando logaritmos para suavizar las diferencias...

$$idf_i = log\ (N/\ n_i) + 1,\ n_i>0$$

4. Ponderación de términos

Ejemplo: N = 1.000 documentos.

term i	n_i	N/n_i	idf_i
A	100	10,00	4,32
B	500	2,00	2,00
C	900	1,11	1,13
D	1.000	1,00	1,00

4. Ponderación de términos

Combinación de tf e idf (ocurrencias en el documento y en la colección):

$$w_{ij} = tf_{ij} * idf_j = f_{ij} * (\log_2 (N/n_j) + 1),\ n_j > 0$$

Objetivo: asignar pesos más altos a aquellos términos que:

- sean frecuentes en documentos relevantes, pero...
- ... infrecuentes en la colección.

4. Ponderación de términos

Pesos tf x idf normalizados:

Solucionan el hecho de que en documentos más largos se asignen pesos más altos.

$w_{ij} \in [0, 1]$.

4. Ponderación de términos

Esquemas de ponderación:

- Dependen del modelo de recuperación.
- Ejemplos: espacio vectorial y probabilístico.

- Muchas variantes.

- Su elección dependen de la evaluación de la recuperación.

5. Selección de términos

- En algunos casos, no es suficiente filtrar las palabras vacías de significado y aplicar segmentación / lematización.
- Se necesita reducir más el vocabulario, e identificar los mejores términos para la recuperación.
- Se aplica un proceso de *Selección de términos*.

5. Selección de términos

Por ejemplo:

Métodos basados en la frecuencia de aparición:

- Identificar los términos con una frecuencia de aparición media (basándonos en las ideas de Luhn).
- Identificar los límites inferior y superior.

5. Selección de términos

Método del valor de discriminación del término (Salton):

- Mide el grado con el que un término es capaz de discriminar documentos de la colección.
- Cuanto mayor capacidad de discriminación, mayor la calidad del término como término de indexación.

5. Selección de términos

Cálculo del valor de discriminación (DV):

1) Calcular la similitud media entre documentos (según espacio vectorial, p.e.)
2) Para cada término, t_k, de la colección:
   1) Eliminarlo temporalmente de la colección.
   2) Calcular de nuevo la similitud media entre documentos.
   3) **DV(t_k) = (similitud media sin t_k) - (similitud media con t_k).**
   4) Seleccionar aquellos con DV > 0.

5. Selección de términos

Métodos basados en la distribución de Poisson:

- Palabras “triviales” tienden a distribuirse según una distribución de probabilidad Poisson.
- Resto de palabras se desvían significativamente de esta distribución de probabilidad.

5. Selección de términos

1)Se calcula la distribución real del término en la colección y la esperada siguiendo una Poisson.

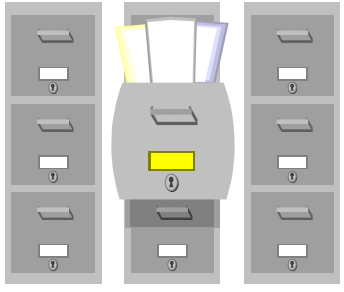
2)Se comparan las distribuciones utilizando un test Chi-cuadrado.

3)Aquellos términos que se desvíen significativamente son seleccionados.

6. Construcción del índice

El objetivo de proceso de indexación es la construcción de un índice.

Índice: estructura de datos que posibilita un acceso veloz a la colección, una vez procesada, con objeto de obtener los documentos relevantes a una consulta.



6. Construcción del índice

Asumimos que cada documento se representa como un conjunto de términos de indexación, que pueden tener asociado un peso de relevancia.

Tanto a los documentos como a los términos se le asocian identificadores unívocos (números enteros).

$D_1 = \{t_1\ (0,3), t_3\ (0,9), t_7\ (0.2), t_9\ (0.6)\}$

$D_2 = \{t_1\ (0,1), t_2\ (0,7), t_4\ (0.5), t_6\ (0.4)\}$, $D_3=\{...\}$, ...

6. Construcción del índice

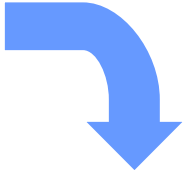
Índice invertido:

Estructura de datos que almacena, de manera ordenada, los términos de la colección, junto con los documentos donde aparecen y el peso correspondiente.

t_1 = {D_1 (0.3), D_5 (0.5), D_{12} (0.2), D_9 (0.1)}
t_2 = {D_1 (0.1), D_2 (0.2), D_5 (0.5)},
t_3={...}, ...

6. Construcción del índice

docs	*t1*	*t2*	*t3*
D1	1	0	1
D2	1	0	0
D3	0	1	1
D4	1	0	0
D5	1	1	1
D6	1	1	0
D7	0	1	0
D8	0	1	0
D9	0	0	1
D10	0	1	1
D11	1	0	1
Q	1	2	3
	q1	*q2*	*q3*



Terms	D1	D2	D3	D4	D5	D6	D7	…
t1	1	1	0	1	1	1	0	
t2	0	0	1	0	1	1	1	
t3	1	0	1	0	1	0	0	

6. Construcción del índice

Operaciones típicas:

- Consultas frecuentes:
  Encuentra los documentos que contienen el término t.

- Borrado... raras veces:
  Borra el documento 52.

- Actualización... raras veces.
  Corrige la ortografía del término t del documento 52.

- Nuevas inserciones... raras veces.
  Añade el nuevo documento 1004.

6. Construcción del índice

Por tanto, acceso a los términos y a su información debe hacerse en tiempo logarítmico, o mejor constante.

Estructuras de datos típicas:

- Vector ordenado.
- Tabla hash.
- Árbol (árbol B o trie).

6. Construcción del índice

Ejemplo de creación de índice invertido como vector ordenado:

I did enact Julius Caesar I was killed i' the Capitol; Brutus killed me.

Doc1

So let it be with Caesar. The noble Brutus hath told you Caesar was ambitious

Doc2

enact	1
julius	1
caesar	1
I	1
was	1
killed	1
i'	1
the	1
capitol	1
brutus	1
killed	1
me	1
so	2
let	2
it	2
be	2
with	2
caesar	2
the	2
noble	2
brutus	2
hath	2
told	2
you	2
caesar	2
was	2
ambitious	2

6. Construcción del índice

Ordenación del vector según los términos.

Term	Doc #
I	1
did	1
enact	1
julius	1
caesar	1
I	1
was	1
killed	1
i'	1
the	1
capitol	1
brutus	1
killed	1
me	1
so	2
let	2
it	2
be	2
with	2
caesar	2
the	2
noble	2
brutus	2
hath	2
told	2
you	2
caesar	2
was	2
ambitious	2

Term	Doc #
ambitious	2
be	2
brutus	1
brutus	2
capitol	1
caesar	1
caesar	2
caesar	2
did	1
enact	1
hath	1
I	1
I	1
i'	1
it	2
julius	1
killed	1
killed	1
let	2
me	1
noble	2
so	2
the	1
the	2
told	2
you	2
was	1
was	2
with	2

PASOS DEL PROCESO DOCUMENTAL

6. Construcción del índice

Unión de mismas entradas de términos e inserción de la frecuencia de aparición.

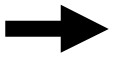
Term	Doc #
ambitious	2
be	2
brutus	1
brutus	2
capitol	1
caesar	1
caesar	2
did	1
enact	1
hath	2
I	1
i'	1
it	2
julius	1
killed	1
let	2
me	1
noble	2
so	2
the	1
the	2
told	2
you	2
was	1
was	2
with	2

Term	Doc #	Freq
ambitious	2	1
be	2	1
brutus	1	1
brutus	2	1
capitol	1	1
caesar	1	1
caesar	2	2
did	1	1
enact	1	1
hath	2	1
I	1	2
i'	1	1
it	2	1
julius	1	1
killed	1	2
let	2	1
me	1	1
noble	2	1
so	2	1
the	1	1
the	2	1
told	2	1
you	2	1
was	1	1
was	2	1
with	2	1

6. Construcción del índice

Índice = Diccionario + lista de apariciones

Term	Doc #	Freq
ambitious	2	1
be	2	1
brutus	1	1
brutus	2	1
capitol	1	1
caesar	1	1
caesar	2	2
did	1	1
enact	1	1
hath	2	1
I	1	2
i'	1	1
it	2	1
julius	1	1
killed	1	2
let	2	1
me	1	1
noble	2	1
so	2	1
the	1	1
the	2	1
told	2	1
you	2	1
was	1	1
was	2	1
with	2	1



Term	N docs	Tot Freq
ambitious	1	1
be	1	1
brutus	2	2
capitol	1	1
caesar	2	3
did	1	1
enact	1	1
hath	1	1
I	1	2
i'	1	1
it	1	1
julius	1	1
killed	1	2
let	1	1
me	1	1
noble	1	1
so	1	1
the	2	2
told	1	1
you	1	1
was	2	2
with	1	1

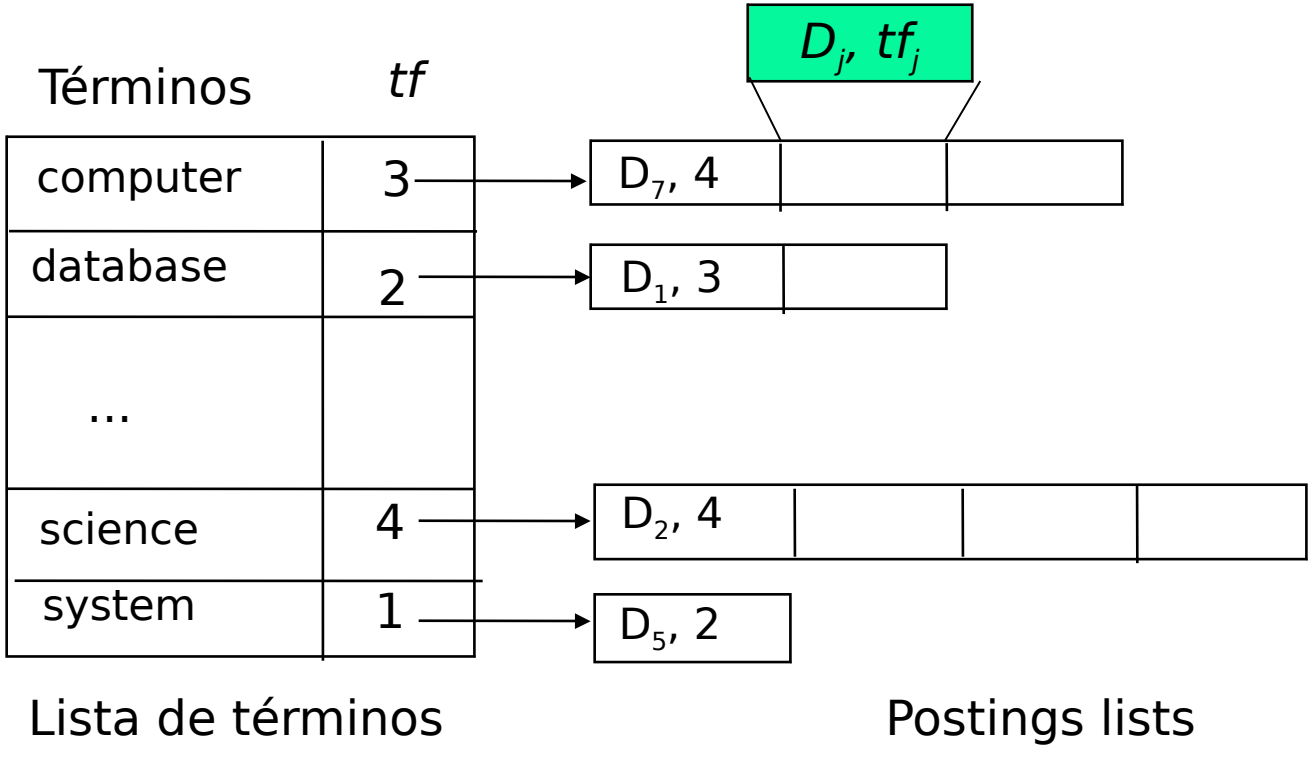
Doc #	Freq
2	1
2	1
1	1
2	1
1	1
1	1
2	2
1	1
1	1
2	1
1	2
1	1
2	1
1	1
1	2
2	1
1	1
2	1
2	1
1	1
2	1
2	1
2	1
1	1
2	1
2	1

Punteros

6. Construcción del índice

- Vector de términos.
- Estructuras que almacenan información sobre la frecuencia del término, el identificador del documento, …
- Lista de ocurrencias (posting list): contiene un nodo por cada documento donde aparece el término.

6. Construcción del índice



6. Construcción del índice

Creación de un fichero invertido:

```
Crear una lista, I, para el índice invertido vacío;
Para cada documento, D, en la colección V
   Para cada token, T, en D:
          Si T pertenece a I
                Insertar T en I;
                Encontrar la ubicación de T en I;
                Si (T, D) está en la lista de ocurrencias de T
                    incrementar la frecuencia de T;
         Sino
              Crear (T, D);
              Añadirlo a la lista de ocurrencias para T;
```

PASOS DEL PROCESO DOCUMENTAL

Dada una consulta...

... esta debe recibir el mismo proceso que el dado a los documentos.