

Big Data trata de la gestión, administración y análisis de enormes volúmenes de datos que no pueden ser tratados de manera convencional, ya que superan los límites y capacidad de las herramientas de software habitualmente utilizadas para la captura, gestión y procesamiento de datos.

Dicho concepto engloba infraestructuras, tecnologías y servicios que han sido creados para dar solución al procesamiento de enormes conjuntos de **datos estructurados, no estructurados o semi-estructurados** (por ejemplo, mensajes en redes sociales, señales de móvil, archivos de audio, sensores, imágenes digitales, datos de formularios, emails, datos de encuestas, logs, etc.) que pueden provenir de sensores, micrófonos, cámaras, escáneres médicos, imágenes satelitales, u otros. Por ejemplo, la estructura de radiotelescopios ALMA produce una cantidad de datos al año a una tasa 10 veces mayor (es decir, 100 terabytes anuales). No obstante, el futuro telescopio de mapeo sinóptico LSST, (Large Synoptic Survey Telescope) o **Gran Telescopio para Rastreo o Sondeos Sinópticos**, generará desde el norte de Chile muchos más. ¡Y seguirá a ese ritmo por 10 años!.¹

El objetivo de Big Data, al igual que los sistemas analíticos convencionales, es “**convertir el Dato en Información**” y así facilitar la toma de decisiones, incluso en tiempo real. Sin embargo, este tema, sobre todo para Chile no es sólo una cuestión de tamaño, sino que es una oportunidad de negocio y de futuros emprendimientos en el ámbito de servicios.

Sin embargo el concepto de Big Data o de manejo de grandes volúmenes de datos no es nuevo, ya que llevan mucho tiempo manejando grandes volúmenes de datos bajo el concepto de **Data Warehouses** y potentes herramientas analíticas que les permiten tratar de forma adecuada esos grandes volúmenes de datos.

Recordemos que el Data Warehouse es una tecnología para el manejo de la información construido sobre la base de optimizar el uso y análisis de la misma utilizado por las organizaciones para adaptarse a los vertiginosos cambios en los mercados. Su función esencial es ser la base de un sistema de información gerencial, es decir, debe cumplir el rol de integrador de información proveniente de fuentes funcionalmente distintas (Bases Corporativas, Bases propias, de Sistemas Externos, etc.) y brindar una visión integrada de dicha información, especialmente enfocada hacia la toma de decisiones por parte del personal jerárquico de la organización.

¹ <http://www.emol.com/noticias/Tecnologia/2016/07/06/808102/Columna-de-Astronomia--Big-data-Un-tsunami-de-datos-astronomicos.html>

Para ello, se usan tecnología y sistemas OLTP (On-Line Transaction Processing) y OLAP (On-Line Analytical Process). En resumen, son aplicaciones que se encargan de analizar datos del negocio para generar información táctica y estratégica que sirve de soporte para la toma de decisiones, en donde las transacciones OLTP utilizan **Bases de Datos Relacionales u otro tipo de archivos**, OLAP logra su máxima eficiencia y flexibilidad operando **sobre Bases de datos Multidimensionales**.

De manera que Big Data es la evolución de la tecnología y dado los menores costos del almacenamiento han hecho que los volúmenes manejados por estas aplicaciones hayan aumentado de manera muy importante. Sin embargo, a los conceptos de **Volumen, Variedad y Velocidad** (3V's) se les ha incluido hoy en día nuevas características como son la **Veracidad y Valor del dato** (5V's)

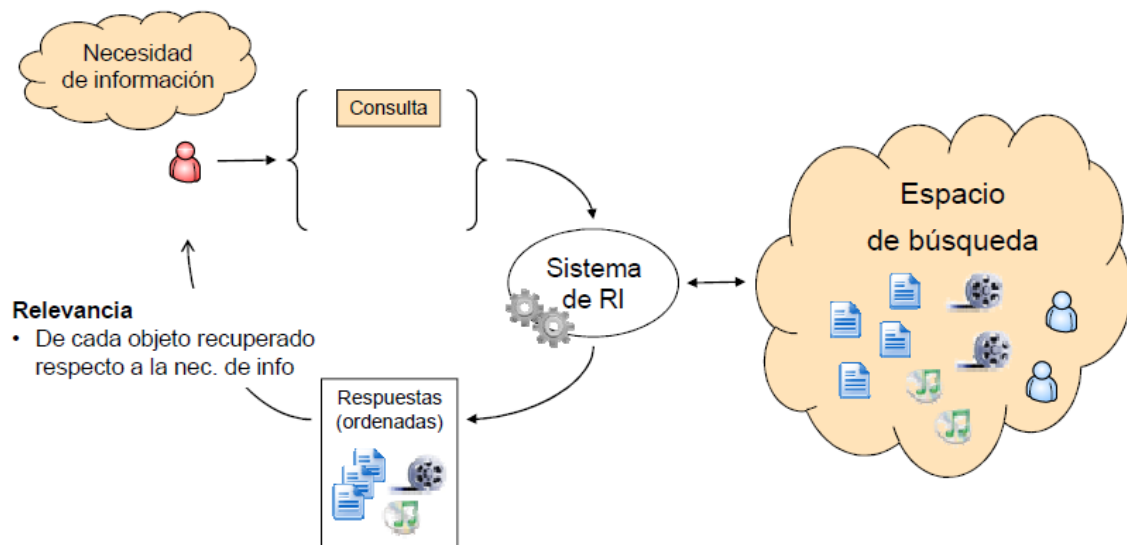
Se habla de Big Data cuando los volúmenes superan la capacidad del software habitual para ser manejados y gestionados. Por otro lado, se habla de grandes volúmenes para cuando nos referimos a tratamientos de Terabytes o Petabytes de datos que pueden surgir de por ejemplo logs, imágenes satelitales, o en general información obtenida desde las Redes Sociales, en donde el número cada vez mayor de dispositivos electrónicos conectados y la explotación de sensores permiten conocer los movimientos y hábitos de vida, de información externa de diversas fuentes, etc.

La información que procesan los Data Warehouses es información estructurada que ha pasado por numerosos filtros de calidad para poder garantizar que la información de salida tiene una precisión y una exactitud determinada, sin embargo, cuando hablamos de *Big Data* nos referimos a información que puede estar semiestructurada o no tener ninguna estructuración.

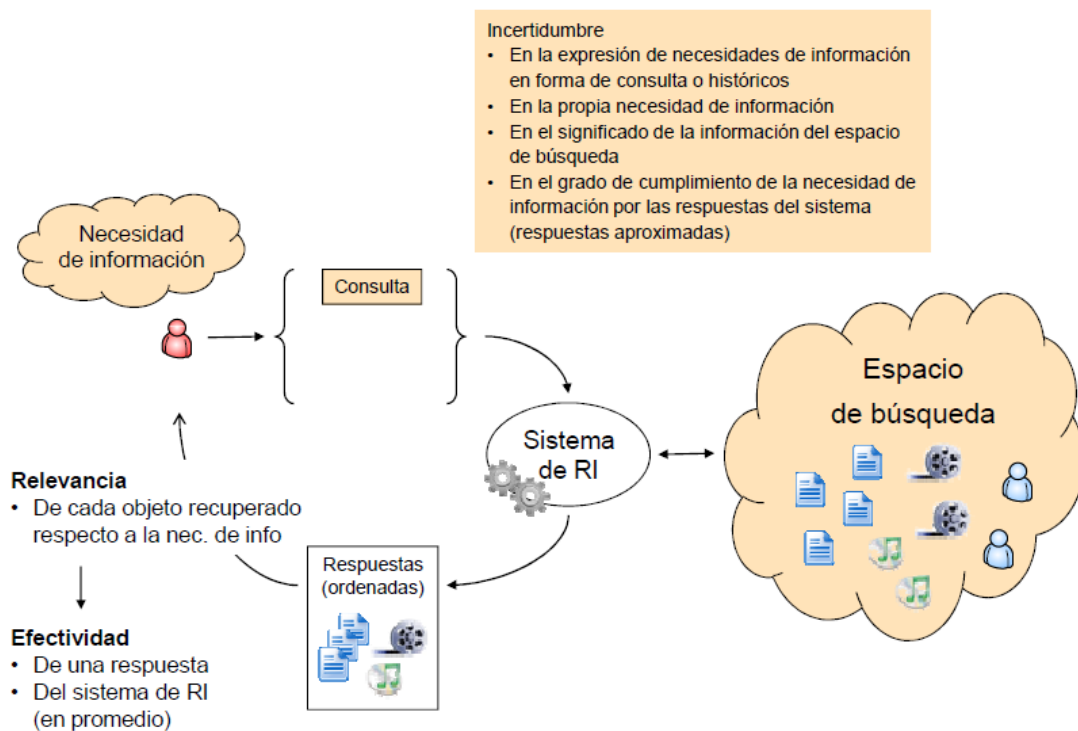
La gestión y administración de esta información desestructurada precisa de una tecnología diferente y permite tomar decisiones basadas en información que tiene importantes grados de inexactitud.

Finalmente, se añade el valor. La importancia del dato para el negocio, saber que datos son los que se deben analizar, es fundamental. Tanto que ya se empieza a hablar del científico de datos (KPI experto, recordar que un **KPI (key performance indicator)**, conocido también como indicador clave o medidor de desempeño o indicador clave de rendimiento), que es un profesional con perfil científico, tecnológico...pero con visión de negocio. De ahí que sean muchos los académicos que han derivado en ofrecer sus

servicios en éste ámbito, éste es un ejemplo². y es en este contexto es donde tiene sentido nuestro Taller.

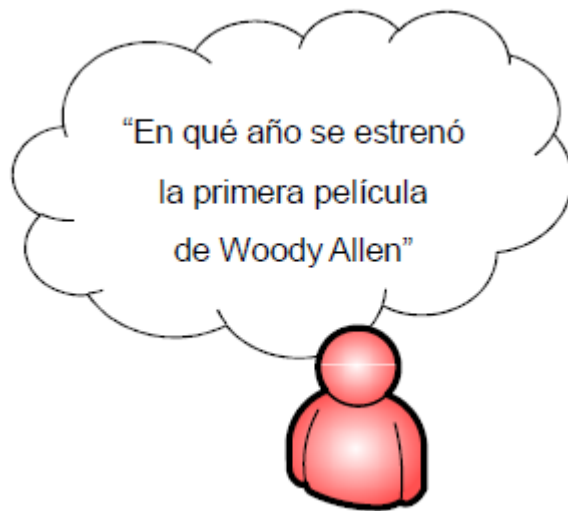


Modelo básico de un Sistema de Recuperación de Información (Retrieval Information, RI)

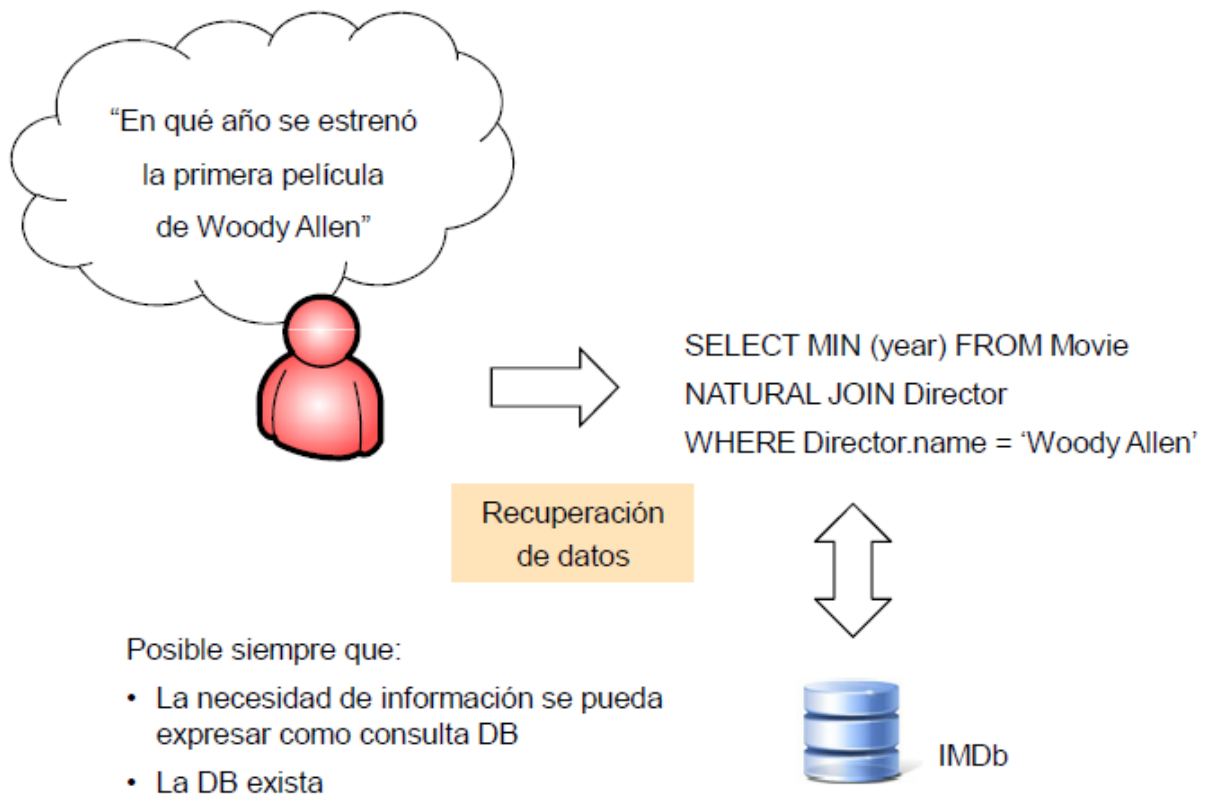


² <http://www.daviddlewis.com/>

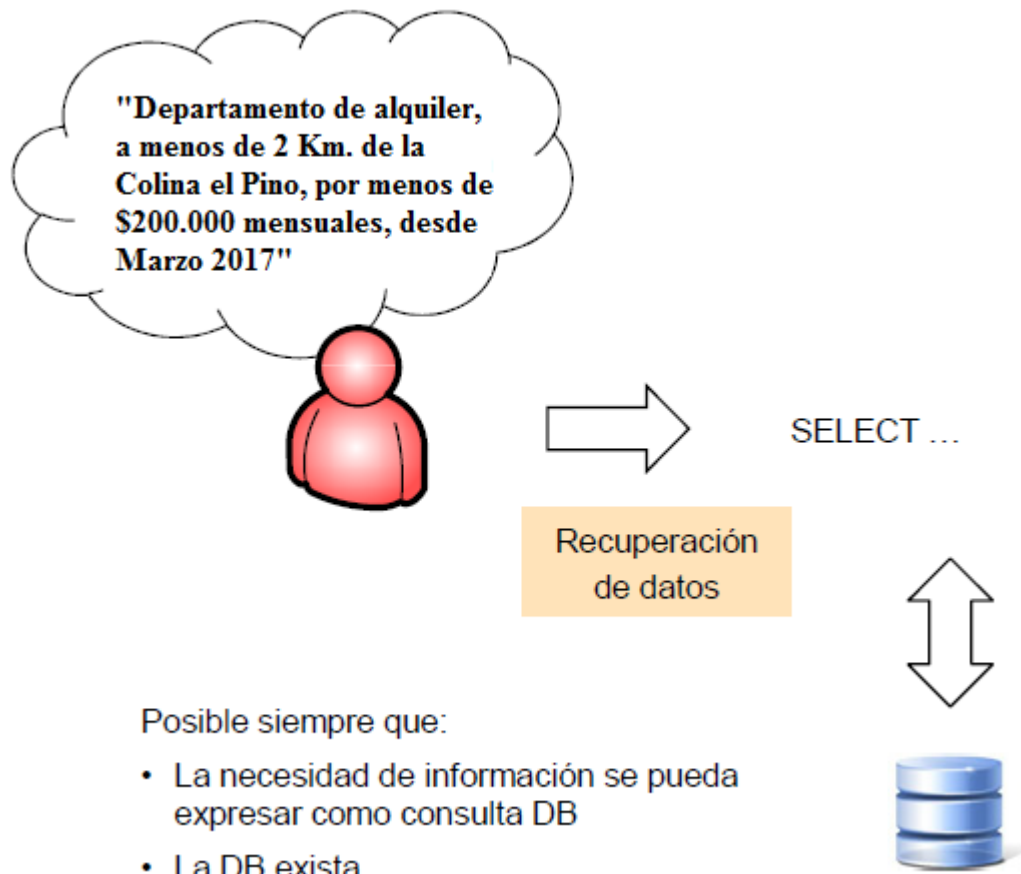
Modelo más elaborado de un Sistema de Recuperación de Información (Retrieval Information, RI)



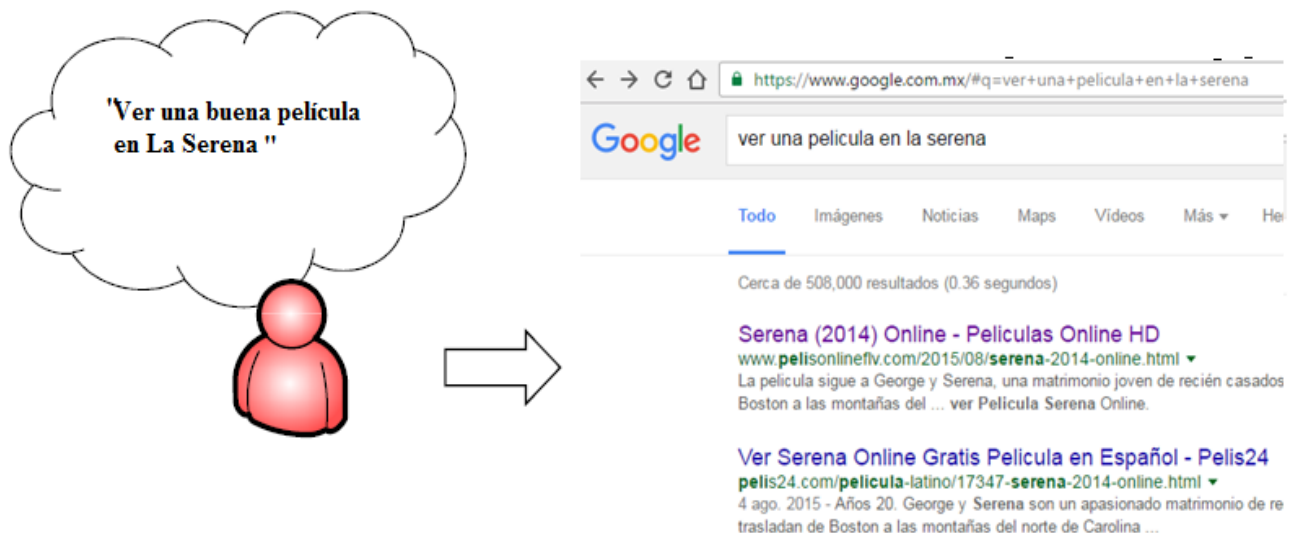
Necesidad de Información

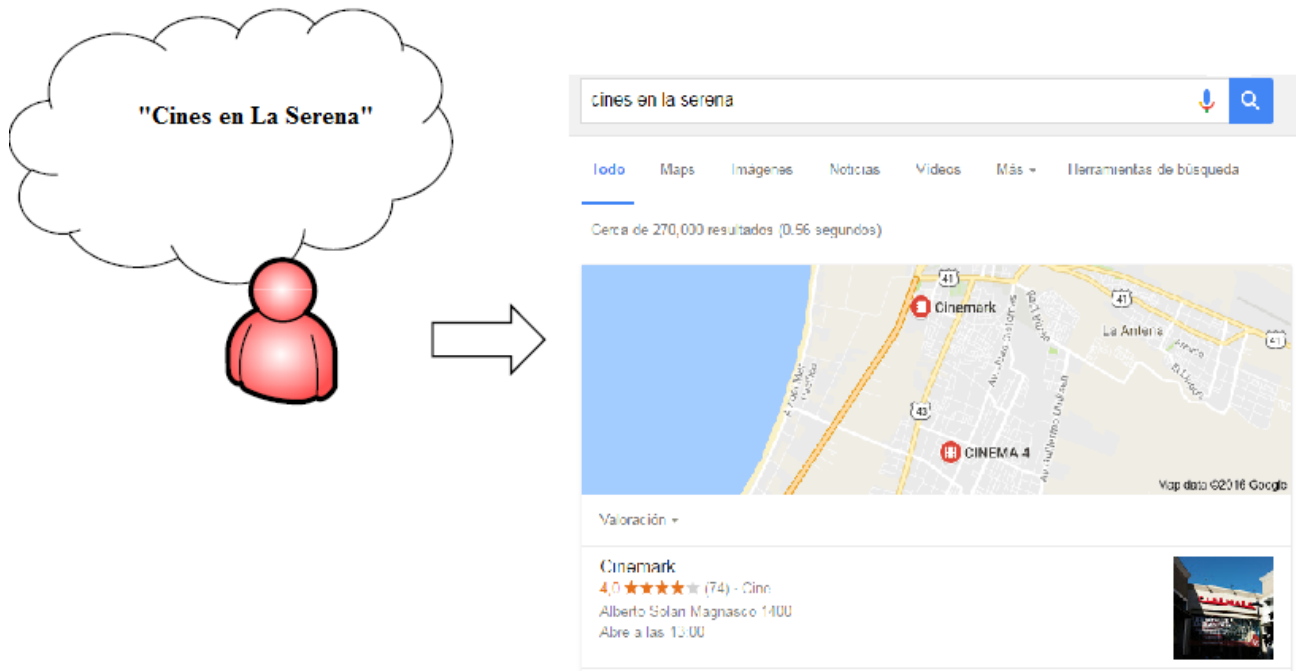


En caso de tener información diseñada en forma de entidad relación, siempre será posible usar las Bases de Datos.



Pero la realidad es otra!!, la consulta es correcta, lo único es que el o los buscadores no saben el contexto, ya que no distingue entre La Serena la ciudad y Serena la película, tal como se puede apreciar. De manera que se necesita hacer una serie de otras tareas.

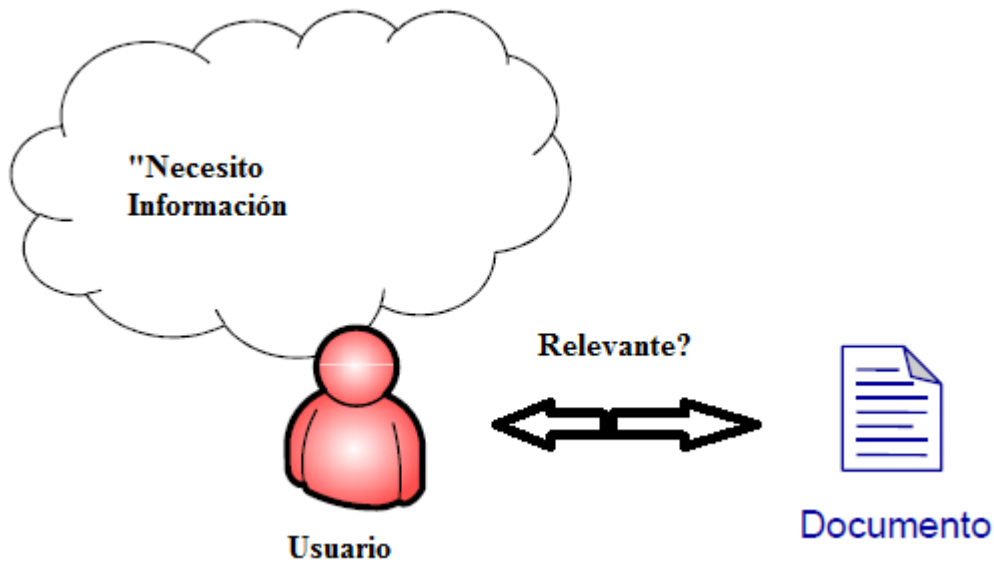




Posible siempre que:

- El sistema disponga de evidencia previa de intereses del usuario (compras, puntuaciones, etc.)
- El sistema disponga de la información adicional que precisen los algoritmos de recomendación: características de los items, históricos de otros usuarios, etc.

Dada la experiencia, la pregunta de fondo es.. ¿Necesito información relevante, ante la consulta que planteo.??!!!



La recuperación de información (IR) es encontrar material (normalmente documentos) de un carácter no estructurado (generalmente de texto) que satisface una necesidad de información dentro de grandes colecciones (normalmente almacenada en computadoras). El problema de la gestión y recuperación de información sigue siendo un problema constante a lo largo de la historia de la ciencia de la computación. Hoy en día, almacenar y archivar la información no es un problema, no así el problema de búsqueda, que sigue siendo un reto.

Sólo como información, el 91% de los usuarios dicen encontrar lo que están buscando cuando utilizan los motores de búsqueda el 73% de los usuarios afirmó que la información que encontró fue fiable y precisa el 66% de los usuarios dice que los motores de búsqueda son justos y proporcionar información imparcial el 55% de los usuarios dicen que los resultados del motor de búsqueda y el motor de búsqueda de calidad ha mejorado con el paso del tiempo. El 93% de las actividades en línea comienzan con un motor de búsqueda el 39% de los clientes provienen de un motor de búsqueda.

<http://www.marketingcharts.com/>

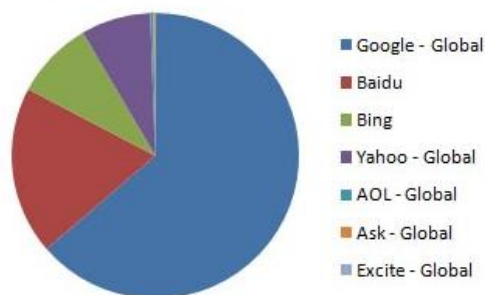


Google sigue dominando la lista de motores de búsqueda más utilizados. Pregunta qué motor de búsqueda utilizan con más frecuencia, el 83% de los usuarios de la búsqueda de dicen Google. El siguiente motor de búsqueda más citado es Yahoo, mencionada por tan solo el 6% de las búsquedas de los usuarios. La última vez que se plantea esta pregunta en 2004, la brecha entre Google y Yahoo fue mucho menor, con un 47% de los usuarios de la búsqueda de Google diciendo que era su motor de elección y 26% citando a Yahoo.

Source: Pew Internet: Search Engine Usage 2012.

<http://www.pewinternet.org/2012/03/09/search-engine-use-2012/>

Ranking Buscadores 2015 PC



BUSCADOR

CUOTA DE MERCADO

Google

62.74%

Baidu

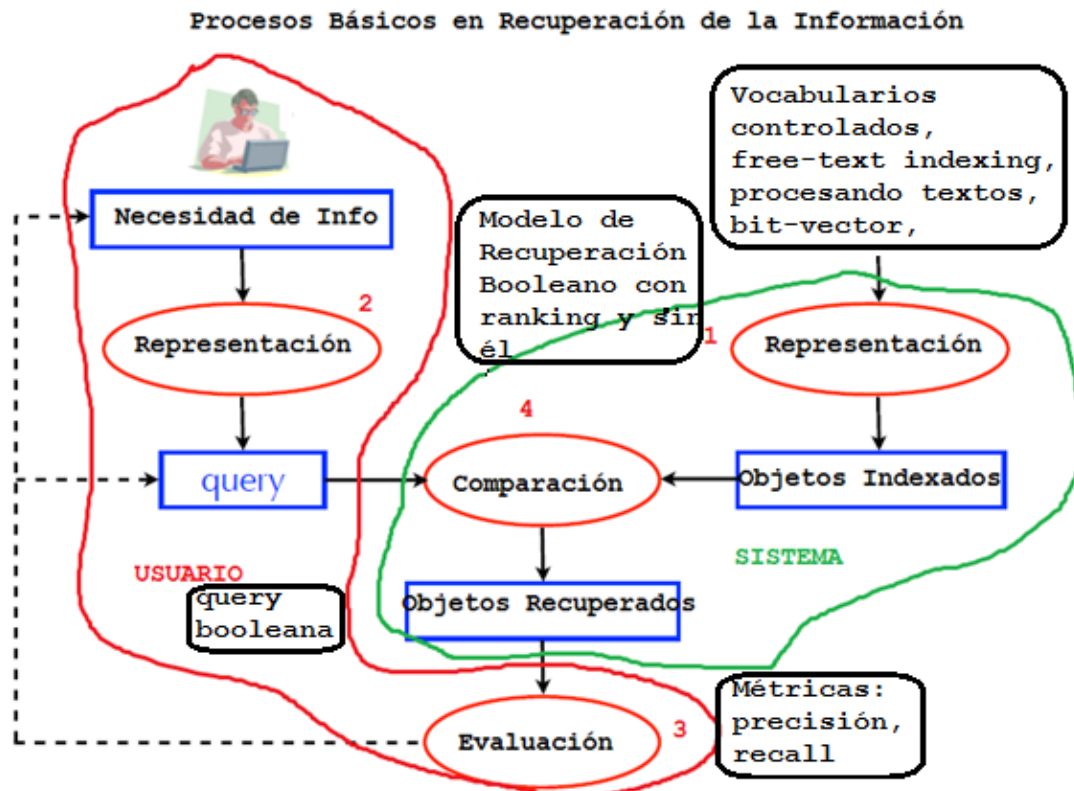
18.68%

Bing

8.70%

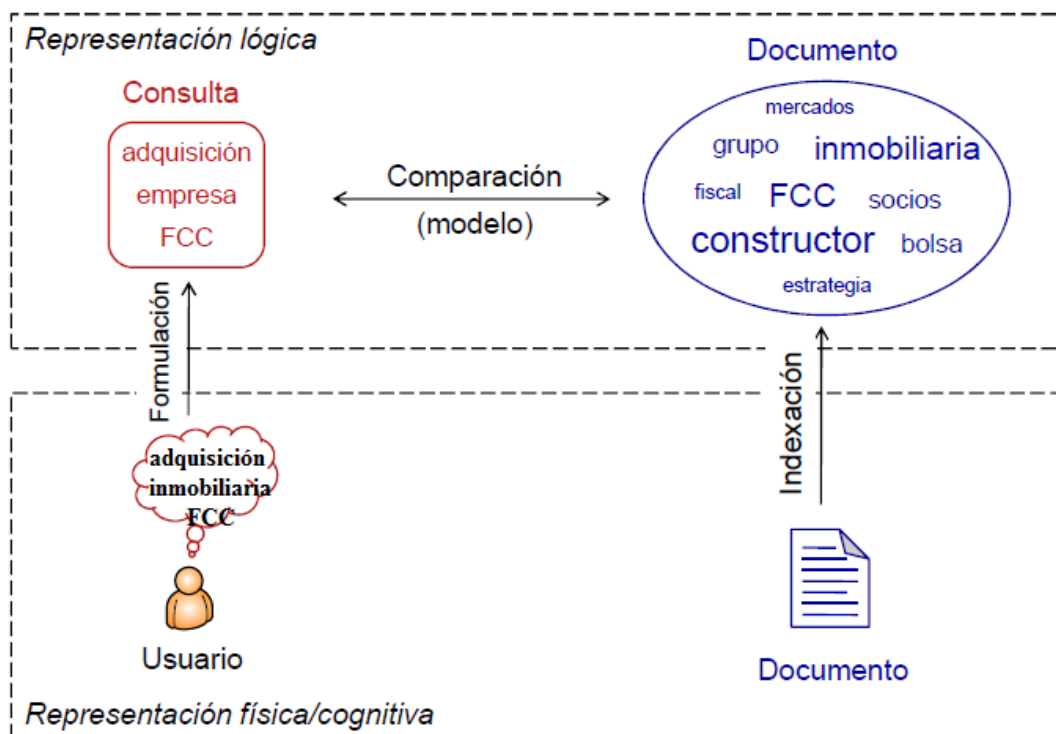
Yahoo

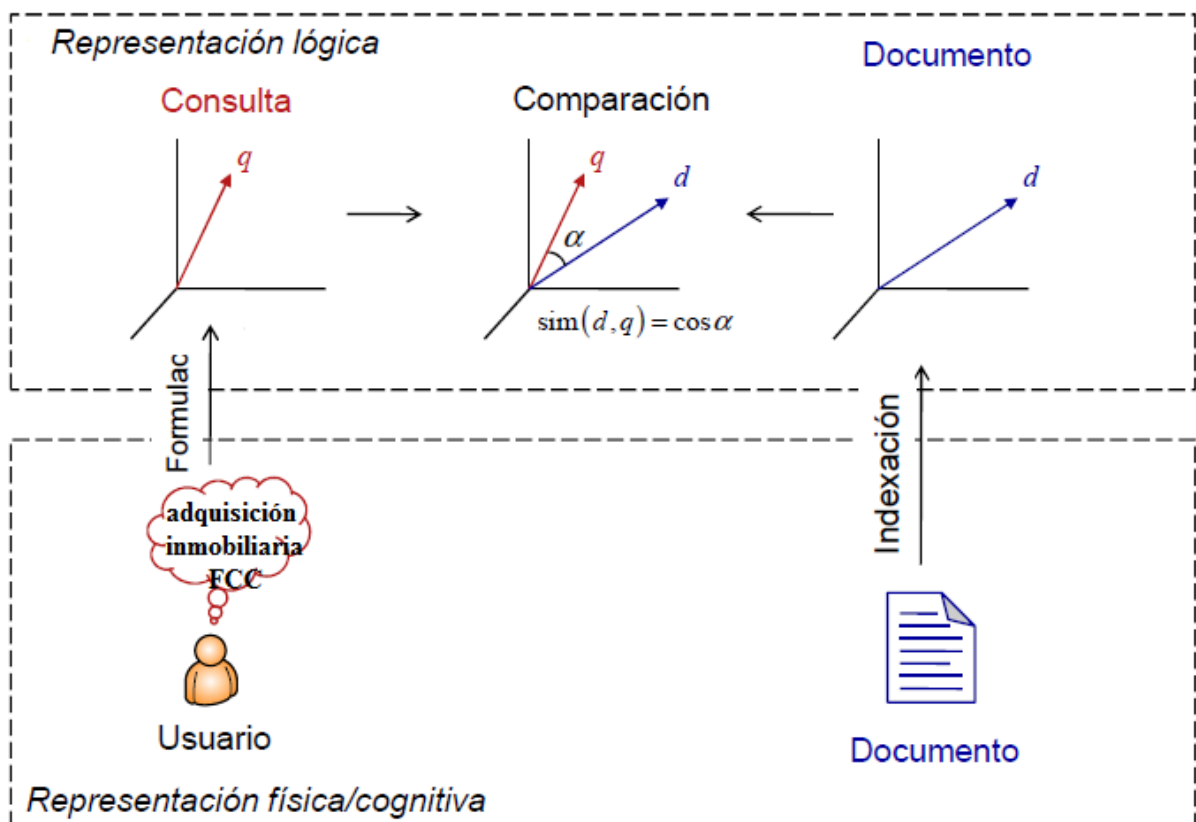
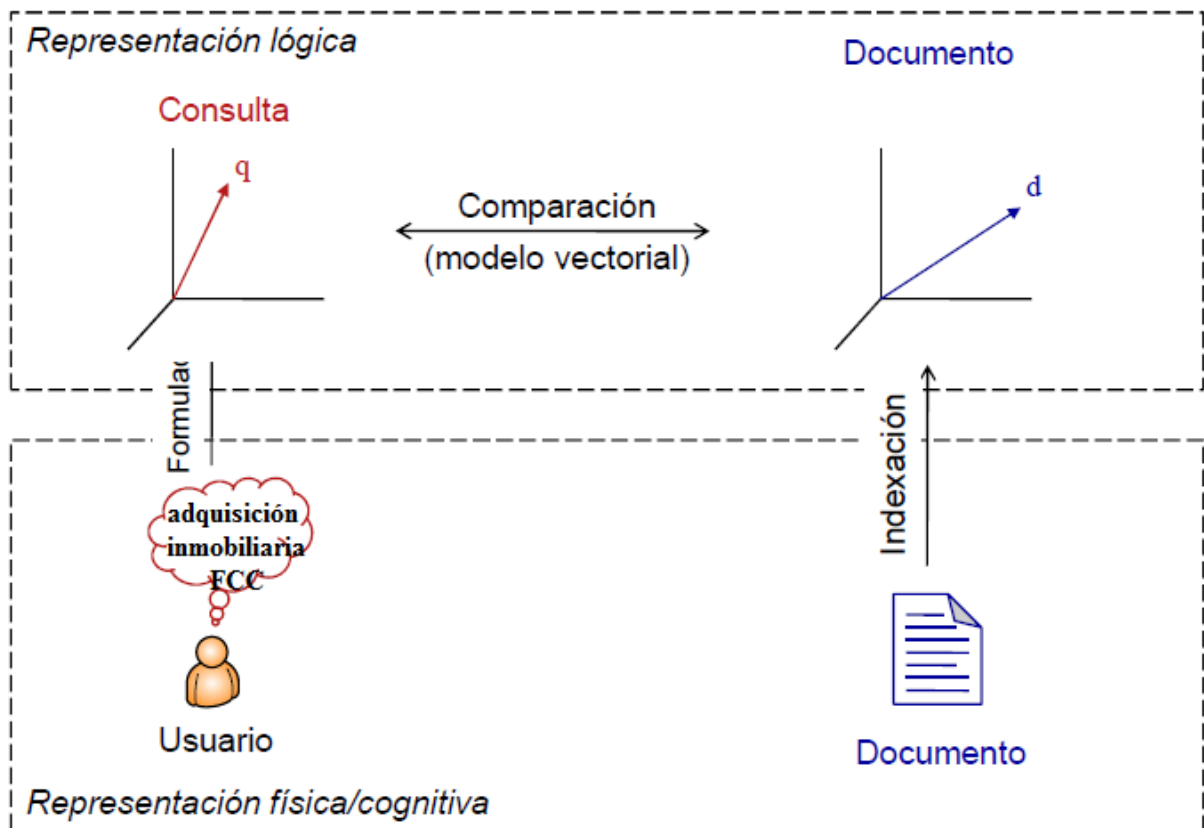
7.79%



Modelo de Recuperación de la Información

El modelo de recuperación es responsable de realizar las comparaciones y recuperación de objetos que son aquellos que satisfacen la necesidad de información dada por el usuario.





Volvamos a considerar el **Modelo de espacio vectorial**, que es un modelo algebraico utilizado para filtrar, indexar, recuperar y calcular la relevancia de la información. Representa los documentos con un lenguaje natural mediante el uso de vectores en un espacio lineal multidimensional. La relevancia de un documento frente a una búsqueda puede calcularse usando la diferencia de ángulos de cada uno de los documentos respecto del vector de busca, utilizando el producto escalar entre el vector de búsqueda.

Por tanto, para evaluar la **similitud** entre un documento y una consulta; es decir, para obtener la **relevancia** del documento con respecto a la consulta, simplemente hay que realizar una comparación de los vectores que los representan.

Calcular peso o similitud

Un tema fundamental se refiere a encontrar los términos que deben utilizarse para indexar un documento (los elementos de la “bolsa de palabras”), y cómo contarlas?.

Algunos enfoques:

- **Peso Binario** son términos que aparecen o no; no hay información de la frecuencia de las palabras utilizadas.
- **TF y TF.IDF** (inverse document frequency model), entre otros.

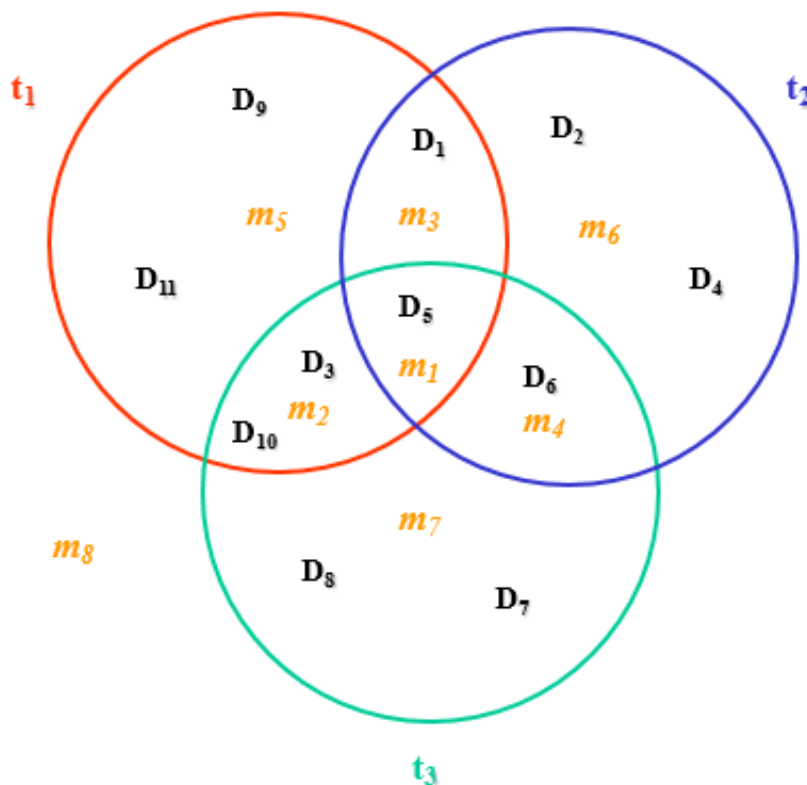
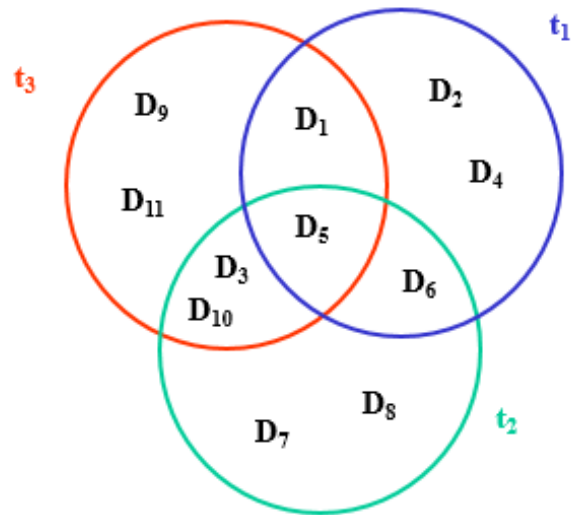
Ejemplo: (Modelo Binario). Esta representación puede ser particularmente útil, ya que los documentos (y la consulta) pueden considerarse como simples cadenas de bits. Esto permite que las operaciones de consulta pueden realizarse mediante operaciones de bits lógicos. Admite presencia de término (**1**) o ausencia (**0**) como vector.

Esta representación puede ser particularmente útil, ya que los documentos (y la consulta) pueden considerarse como simples cadenas de bits. Esto permite que las operaciones de consulta pueden realizarse mediante operaciones de bits lógicos

<i>docs</i>	<i>t1</i>	<i>t2</i>	<i>t3</i>
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

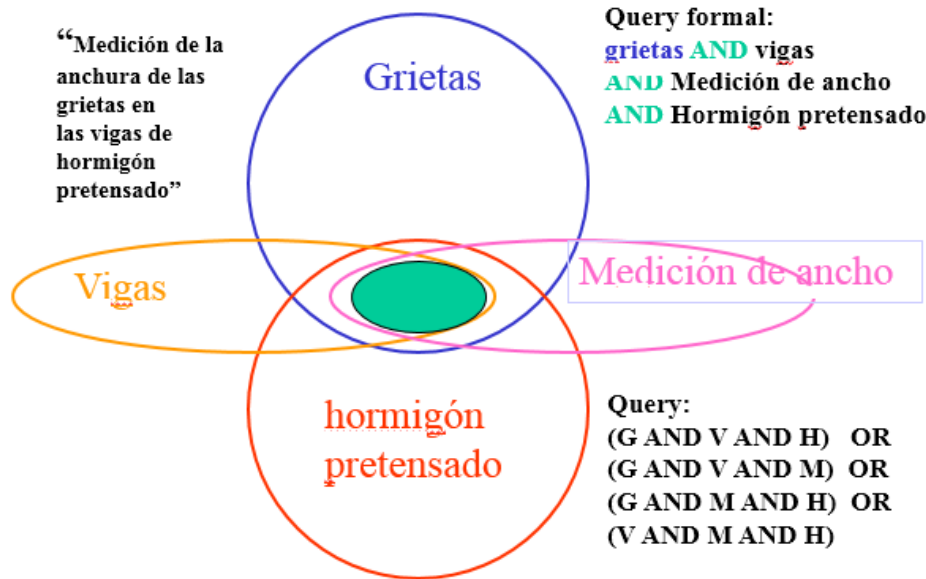
En el caso Peso Binario, la correspondencia entre los documentos y las consultas pueden ser vistos como el tamaño de la intersección de dos conjuntos (de términos): $Q \cap D$. Esto a su vez puede ser utilizada para clasificar la relevancia de los documentos para una consulta. Esto es correspondencia de consultas Q con los documentos D 's

docs	t_1	t_2	t_3	Rank= $Q \cdot D_i$
D1	1	0	1	2
D2	1	0	0	1
D3	0	1	1	2
D4	1	0	0	1
D5	1	1	1	3
D6	1	1	0	2
D7	0	1	0	1
D8	0	1	0	1
D9	0	0	1	1
D10	0	1	1	2
D11	1	0	1	2
Q	1	1	1	
	q_1	q_2	q_3	



$$\begin{aligned}
 m_1 &= t_1 t_2 t_3 \\
 m_2 &= t_1 \bar{t}_2 t_3 \\
 m_3 &= t_1 t_2 \bar{t}_3 \\
 m_4 &= \bar{t}_1 t_2 t_3 \\
 m_5 &= t_1 \bar{t}_2 \bar{t}_3 \\
 m_6 &= \bar{t}_1 t_2 \bar{t}_3 \\
 m_7 &= \bar{t}_1 \bar{t}_2 t_3 \\
 m_8 &= \bar{t}_1 \bar{t}_2 \bar{t}_3
 \end{aligned}$$

Ejemplo: Mediante lógica booleana, modelar la consulta “Medición de la anchura de las grietas en las vigas de hormigón pretensado...”. Lo primero, las palabras más relevantes!!. Ellas son: Grietas, Vigas, Hormigón pretensado y Medición de ancho.



Más generalmente, la similitud entre la consulta Q y los documentos D's pueden ser visto como el producto punto de dos vectores: $Q \cdot D$ (esto también se conoce como simple coincidencia). Tenga en cuenta que si ambos Q y D son binario, entonces coinciden con $|Q \cap D|$.

Dados dos vectores X e Y,

$$X = \langle x_1, x_2, \dots, x_n \rangle$$

$$Y = \langle y_1, y_2, \dots, y_n \rangle$$

$$\text{sim}(X, Y) = X \bullet Y = \sum_i x_i \times y_i$$

el matching mide la similitud entre x e y como producto punto de X e Y, por ejemplo $\langle 1, 2, 3 \rangle \cdot \langle 1, 0, 1 \rangle = 4$.

docs	t1	t2	t3	Rank=Q.Di
D1	1	0	1	4
D2	1	0	0	1
D3	0	1	1	5
D4	1	0	0	1
D5	1	1	1	6
D6	1	1	0	3
D7	0	1	0	2
D8	0	1	0	2
D9	0	0	1	3
D10	0	1	1	5
D11	1	0	1	3
Q	1	2	3	
	q1	q2	q3	

Ejemplo (Modelo Espacio Vectorial).

Con un simple matching, el producto punto de dos vectores mide la similitud de estos vectores. Donde la normalización puede lograrse dividiendo el producto de puntos por el producto de las normas de los dos vectores. **Normalized Similarity Measures.**

$$X = \langle x_1, x_2, \dots, x_n \rangle$$

$$\|X\| = \sqrt{\sum_i x_i^2}$$

$$sim(X, Y) = \frac{X \bullet Y}{\|X\| \times \|Y\|} = \frac{\sum_i (x_i \times y_i)}{\sqrt{\sum_i x_i^2} \times \sqrt{\sum_i y_i^2}}$$

Nota: mide el coseno del ángulo entre dos vectores; por lo tanto, es llamado el coseno normalizado Medida de similitud.

docs	t1	t2	t3	SIM(Q,D)
D1	2	0	3	0.82
D2	1	0	0	0.27
D3	0	4	7	0.96
D4	3	0	0	0.27
D5	1	6	3	0.87
D6	3	5	0	0.60
D7	0	8	0	0.53
D8	0	10	0	0.53
D9	0	0	1	0.80
D10	0	3	5	0.96
D11	4	0	1	0.45
Q	1	2	3	
	q1	q2	q3	

Otro de los métodos más habituales para comparar el grado de similitud es **calcular el coseno del ángulo** que forman ambos vectores. Cuanto más se parezcan los vectores, más próximo a cero grados será el ángulo que forman y, en consecuencia, más se aproximará a uno el coseno de ese ángulo. Esta forma de comparar los vectores se conoce como *fórmula del coseno*.

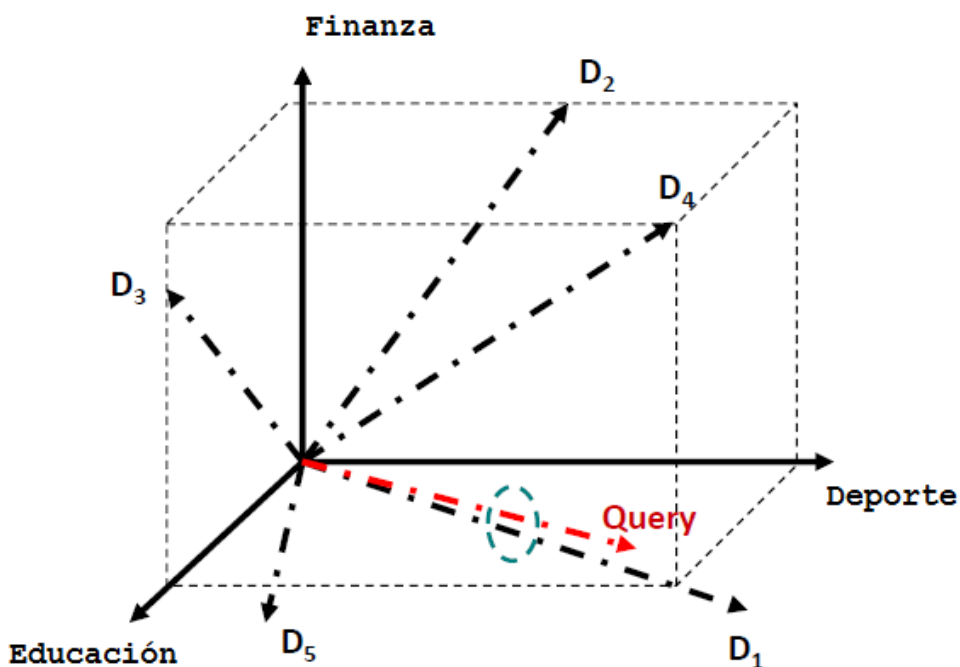
En la figura siguiente presentamos un ejemplo en el que se muestran los vectores representativos de cinco documentos (D1, D2, D3, D4, D5) en el espacio vectorial **Finanzas, Deporte y Educación**, y de la consulta query (q). Intuitivamente, el documento

D_1 es más relevante para la consulta q que el documento D_2 . La fórmula del coseno no hace más que trasladar esta idea intuitiva a la formalización de los vectores. Esto significa que el ángulo que forman los vectores de D_1 y q es menor que el formado por los vectores de D_2 y q , en consecuencia, el coseno del primer ángulo es mayor que el del segundo.

En el modelo espacio vectorial, se tiene:

- a) **Vocabulario** $V = \{w_1, w_2, \dots, w_N\}$ de lenguajes.
- b) **Query** $q = t_1, \dots, t_m$, donde $t_i \in V$
- c) **Documento** $d_i = t_{i1}, \dots, t_{in}$, donde $t_{ij} \in V$
- d) **Collection** $C = \{d_1, \dots, d_k\}$
- e) **Rel(q, d)**: relevancia de documento d a query q
- f) **Rep(d)**: representación de documento d
- g) **Rep(q)**: representación de query q

Representando tanto los documentos y query por conceptos vectoriales, en tal caso, todo concepto define una dimensión en el espacio vectorial, de donde, n -conceptos define un espacio vectorial en R^n . Los elementos del vector corresponden al concepto de **peso**, esto es, para un documento $d = (x_1, \dots, x_k)$, x_i es la “importancia” del concepto i . La medida de relevancia es la distancia entre el vector query y el vector documento en este espacio vectorial.



Formalmente, un espacio vectorial está definido por un conjunto de vectores linealmente independiente, llamados bases del espacio vectorial.

Recordar que estos valores, no representan frecuencia, localización o información de la palabra. Cualquier segmento de texto (i.e., un documento, o una query) puede ser representada como un vector del espacio vectorial V-dimensional. ¿Qué documento es más certero para la query?

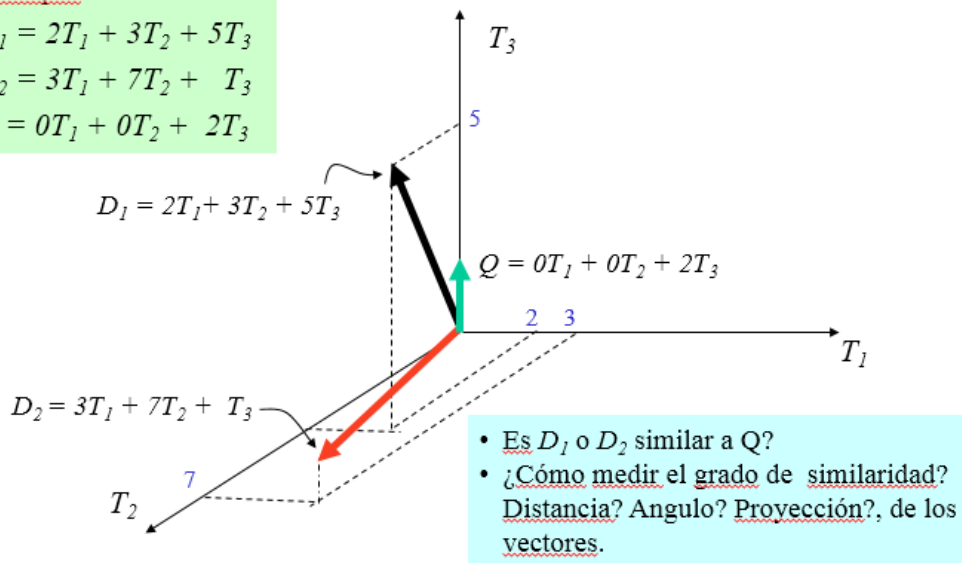
Ejemplo: D1, D2 documentos y Q consulta (o query)

Ejemplo:

$$D_1 = 2T_1 + 3T_2 + 5T_3$$

$$D_2 = 3T_1 + 7T_2 + T_3$$

$$Q = 0T_1 + 0T_2 + 2T_3$$



Ejemplo: Similitud, medida por fórmula del coseno

$$\text{CosSim}(\vec{d}_j, \vec{q}) = \frac{\vec{d}_j \cdot \vec{q}}{|\vec{d}_j| \cdot |\vec{q}|} = \frac{\sum_{i=1}^t (w_{ij} \cdot w_{iq})}{\sqrt{\sum_{i=1}^t w_{ij}^2} \cdot \sqrt{\sum_{i=1}^t w_{iq}^2}}$$

$$D_1 = 2T_1 + 3T_2 + 5T_3 \quad \text{CosSim}(D_1, Q) = 10 / \sqrt{(4+9+25)(0+0+4)} = 0.81$$

$$D_2 = 3T_1 + 7T_2 + 1T_3 \quad \text{CosSim}(D_2, Q) = 2 / \sqrt{(9+49+1)(0+0+4)} = 0.13$$

$$Q = 0T_1 + 0T_2 + 2T_3$$

D_1 es 6 veces mejor que D_2 usando “fórmula del coseno” pero solamente 5 veces mejor, usando “product interno”

En efecto,

$$\begin{aligned} D_1 &= 2T_1 + 3T_2 + 5T_3 & D_2 &= 3T_1 + 7T_2 + 1T_3 \\ Q &= 0T_1 + 0T_2 + 2T_3 \end{aligned}$$

$$\text{sim}(D_1, Q) = 2*0 + 3*0 + 5*2 = 10$$

$$\text{sim}(D_2, Q) = 3*0 + 7*0 + 1*2 = 2$$

Similaridad vectorial con pesos. Documentos en una colección son asignados **terms** desde un conjunto de n terms. El **term vector space** W es definido como:

Si term k no ocurre en documento d_i , entonces $w_{ik} = 0$

Si term k ocurre en documento d_i , entonces w_{ik} es mayor que cero.

(w_{ik} es llamado el peso del term k en documento d_i)

Similaridad entre d_i y d_j es definido como:

$$\cos(\mathbf{d}_i, \mathbf{d}_j) = \frac{\sum_{k=1}^n w_{ik} w_{jk}}{|\mathbf{d}_i| |\mathbf{d}_j|}$$

donde \mathbf{d}_i y \mathbf{d}_j son los correspondientes pesos de los vectores y $|\mathbf{d}_i|$ es la longitud del documento vector \mathbf{d}_i

Producto punto entre documentos

$$\mathbf{d}_1 \cdot \mathbf{d}_2 = w_{11} w_{21} + w_{12} w_{22} + w_{13} w_{23} + \dots + w_{1n} w_{2n}$$

Producto punto entre documentos y la query

$$\mathbf{d}_1 \cdot \mathbf{q}_1 = w_{11} w_{q11} + w_{12} w_{q12} + w_{13} w_{q13} + \dots + w_{1n} w_{q1n}$$

donde w_{qij} es el peso del término j th de la query i th

Boolean information retrieval: Pesos de term k en documento d_i :

$$\begin{aligned} w(i, k) &= 1 && \text{si term } k \text{ ocurre en documento } d_i \\ w(i, k) &= 0 && \text{en otro caso} \end{aligned}$$

General weighting methods: Pesos de term k en documento d_i :

$$0 < w(i, k) \leq 1 \quad \text{si term } k \text{ ocurre en documento } d_i$$

$$w(i, k) = 0 \quad \text{en otro caso.}$$

Umbral para la consulta q , significa recuperar todos los documentos con una similitud por encima de un umbral, por ejemplo, similitud $> 0,50$. Por otro lado, la clasificación o ranking de la consulta q , devuelve el n más similar a la mayoría de los documentos clasificados según orden de similitud.

Ejemplo:

query		
q	<i>ant dog</i>	
document	text	terms
d_1	<i>ant ant bee</i>	<i>ant bee</i>
d_2	<i>dog bee dog hog dog ant dog</i>	<i>ant bee dog hog</i>
d_3	<i>cat gnu dog eel fox</i>	<i>cat dog eel fox gnu</i>

	ant	bee	cat	dog	eel	fox	gnu	hog	length
q	1			1					$\sqrt{2}$
d_1	2	1							$\sqrt{5}$
d_2	1	1		4				1	$\sqrt{19}$
d_3			1	1	1	1	1		$\sqrt{5}$

Calcular el Scoring o Ranking para una consulta dada

	d_1	d_2	d_3
q	$2/\sqrt{10}$ 0.63	$5/\sqrt{38}$ 0.81	$1/\sqrt{10}$ 0.32

Si la consulta q es buscado según este documento, se muestra los resultados clasificados por: **D2, D1, D3**

La otra forma u objetivo es asignar un peso a los términos es con el peso **TF x IDF** para cada término en cada documento.

$$w_{ik} = tf_{ik} * \log(N / n_k)$$

T_k = term k in document D_i

tf_{ik} = frequency of term T_k in document D_i

idf_k = inverse document frequency of term T_k in C

N = total number of documents in the collection C

n_k = the number of documents in C that contain T_k

$$idf_k = \log\left(\frac{N}{n_k}\right)$$

Documento inverso Frecuencia (IDF) es una forma de lidiar con los problemas de la distribución **Zipf**. IDF proporciona altos valores de palabras raras y bajos valores de palabras comunes.

$$\log\left(\frac{10000}{10000}\right) = 0$$

$$\log\left(\frac{10000}{5000}\right) = 0.301$$

$$\log\left(\frac{10000}{20}\right) = 2.698$$

$$\log\left(\frac{10000}{1}\right) = 4$$

	Doc 1	Doc 2	Doc 3	Doc 4	Doc 5	Doc 6		
T1	0	2	4	0	1	0	df	idf = $\log_2(N=6/df)$
T2	1	3	0	0	0	2	3	1.00
T3	0	1	0	2	0	0	2	1.58
T4	3	0	1	5	4	0	4	0.58
T5	0	4	0	0	0	1	2	1.58
T6	2	7	2	1	3	0	5	0.26
T7	1	0	0	5	5	1	4	0.58
T8	0	1	1	0	0	3	3	1.00



Documentos representados como vectores de palabras



	Doc 1	Doc 2	Doc 3	Doc 4	Doc 5	Doc 6
T1	0.00	2.00	4.00	0.00	1.00	0.00
T2	1.00	3.00	0.00	0.00	0.00	2.00
T3	0.00	1.58	0.00	3.16	0.00	0.00
T4	1.74	0.00	0.58	2.92	2.34	0.00
T5	0.00	6.34	0.00	0.00	0.00	1.58
T6	0.52	1.84	0.53	0.26	0.79	0.00
T7	0.58	0.00	0.00	2.92	2.90	0.58
T8	0.00	1.00	1.00	0.00	0.00	3.00

El problema la recuperación de documentos

Dado N documentos (D_0, \dots, D_{n-1}) , y una consulta Q del usuario. Se persigue clasificar los k documentos D que coinciden con la consulta Q lo suficientemente bien respecto a la pertinencia del documento a la consulta.

Para ello:

- **Extracción de características (palabras, frases, n-gram, derivados, sinónimos, multimedia).** Un sistema de recuperación de documentos de texto representa como conjuntos de términos (por ejemplo, palabras). Así, el documento estructurado originalmente se convierte en un conjunto estructurado de términos potencialmente anotadas con atributos para denotar la frecuencia y la posición en el texto. La transformación comprende varios pasos:

1. Eliminación de la estructura

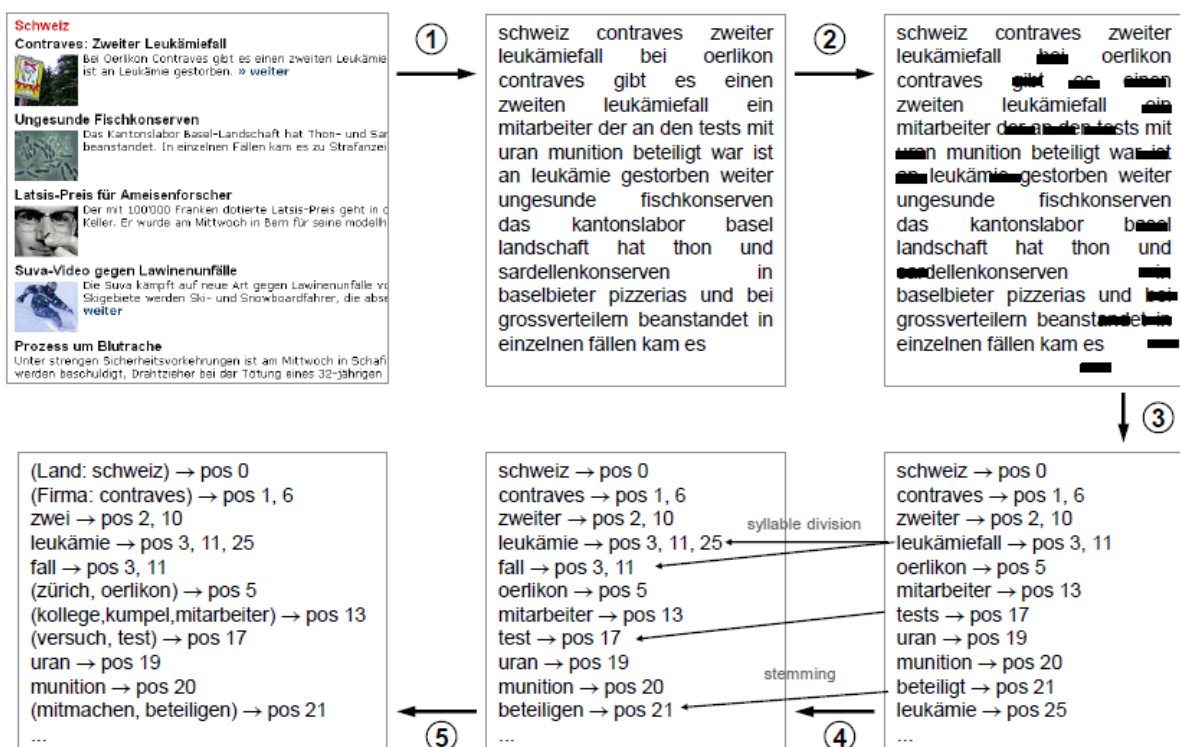
2. Eliminación de frecuentes/términos infrecuentes (stop words)

3. Texto asignación de términos (sin puntuación)

4. Reducción de los términos de sus tallos o raíces (derivados, sílaba División)

5. Asignación a términos de índice.

(El orden de los pasos anteriores pueden variar; a menudo, las medidas son incluso rota en varios pasos o varios pasos se combinan en una sola pasada). Tipos de términos: Palabras, frases, n-G (es decir, secuencia de n - caracteres). **Extracción de características**



Paso 1: Eliminación de la estructura (con el ejemplo de HTML). HTML contiene marcas especiales, llamados tags. Ellos describen la meta-información sobre el documento y la distribución y presentación de contenidos. Un documento HTML está dividido en dos partes, una sección de encabezado (head) y una sección de la “carrocería” (body):

```
<html>
<head>
<title> RI - Electivo </title>
<meta name=„keywords“ content=„información, recuperar, electivo“>
</head>

<body>
...
...
</body>
</html>
```

Paso 2: Eliminación de frecuente o infrecuente términos. El objetivo de indexación es determinar respuestas eficientes y eficaces para las consultas de los usuarios. Para alcanzar este objetivo, no está obligada a considerar los términos con poca o ninguna semántica o términos que aparecen raramente (por ejemplo, endoplasmatic retículo en una biblioteca de ciencias de la computación)

La solución teórica: restringir la indexación de términos que han demostrado ser útiles o que parecen interesantes, experiencias prácticas del pasado con el sistema. Sin embargo, esto requiere un mecanismo de retroalimentación con el usuario para comprender término importancia.

La solución pragmática: término frecuencias en una colección de documentos siga la denominada distribución Zipfian, esto es conocer

- a) -Rango términos basados en sus frecuencias de ocurrencia
- b) permitir las expresiones

N = número de apariciones en la colección

M = número de términos distintos en colección

n_t = número de ocurrencias del término t

r_t = rango de término t ordenada por el número de apariciones

p_r = Probabilidad de que un término aleatorio en un documento tenga rango r ($r=r_t$)

$= n_t / N$

Restricción de términos significantes: Stop words son términos con poco o ningún significado semántico. A menudo, las palabras vacías no son indexados como ocurren en casi todos los documentos ya que llevan poca información.

Ejemplos: En alemán: en, der, wo ich.etc. En Inglés: el a, es, etc. Generalmente, estas palabras son responsables por el 20% al 30% de ocurrencias en un texto. Con la eliminación de palabras vacías, el consumo de memoria del índice puede ser reducida drásticamente. Asimismo, los términos más frecuentes en una colección de documentos llevan poca información. El término "computador" no tiene mucho sentido al índice de artículos sobre ciencias de la computación, sin embargo, es importante distinguir entre artículos generales sobre ciencias de la computación.

Paso 3: asignar los términos más relevantes al texto. Para seleccionar características apropiadas para los documentos, uno utiliza normalmente lingüísticas o de enfoques estadísticos para definir las características basadas en palabras, fragmentos de palabras o frases. La mayoría de los motores de búsqueda utilizan palabras o frases como características. Algunos motores utilizan derivados, algunos distinguen entre mayúsculas y minúsculas, etc. Una opción interesante es el uso de fragmentos, es decir, **n-gramas**. Aunque no está directamente relacionado con la semántica del texto, son muy útiles para apoyar la recuperación de "palabras difusa", o fragmentos de palabras:

Ejemplo:

street → str, tre, ree, eet
streets → str, tre, ree, eet, ets
strets → str, tre, ret, ets

Ventajas: examinar una simple mala ortografía, a menudo resultan en malas recuperaciones fragmentos; mejoran significativamente la calidad de recuperación.

- **Estructuras de índice:** Conceptualmente, el índice puede ser visto como un documento de términos dispuestos en una matriz, donde cada documento es representado como un vector n-dimensional (n = número de términos en el diccionario). El peso del término representa el valor escalar de cada dimensión en un documento, mientras que la estructura del archivo invertido es un "modelo de ejecución" que se utiliza en la práctica para almacenar la información capturada en esta representación conceptual. En efecto,

Documento Ids

↓

	nova	galaxy	heat	<u>hollywood</u>	film	role	diet	fur
A	1.0	0.5	0.3					
B	0.5	1.0						
C		1.0	0.8	0.7				
D		0.9	1.0	0.5				
E				1.0	1.0			
F					0.9		1.0	
G	0.5		0.7				0.9	
H		0.6		1.0		0.3	0.2	0.8
I			0.7	0.5	0.1			0.3

Peso Término (en este caso normalizado)

↑

diccionario

Un vector documento

↓

Invertir es un gran índice de documentos de archivo vectorial "inverted" de modo que las filas se convierten en columnas y filas en columnas, tal como se aprecia. La idea básica es listar todos los tokens en la colección, y para todo token (t), listar todos los docs (d) en donde ocurren (junto con su frecuencia.).

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

→

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	

Etapas en Indexado: Secuencia de Token

Secuencia de (Token modificado, Documento ID) par ordenado.

Doc 1

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



Doc 2

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

Term	docID
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

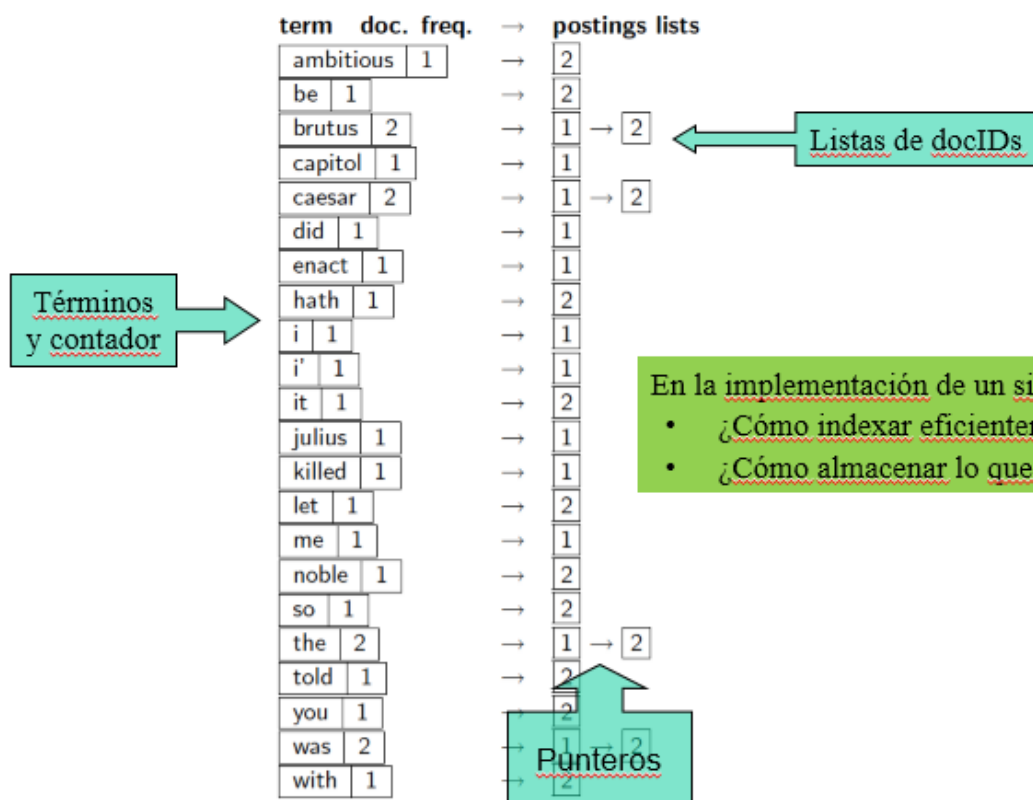
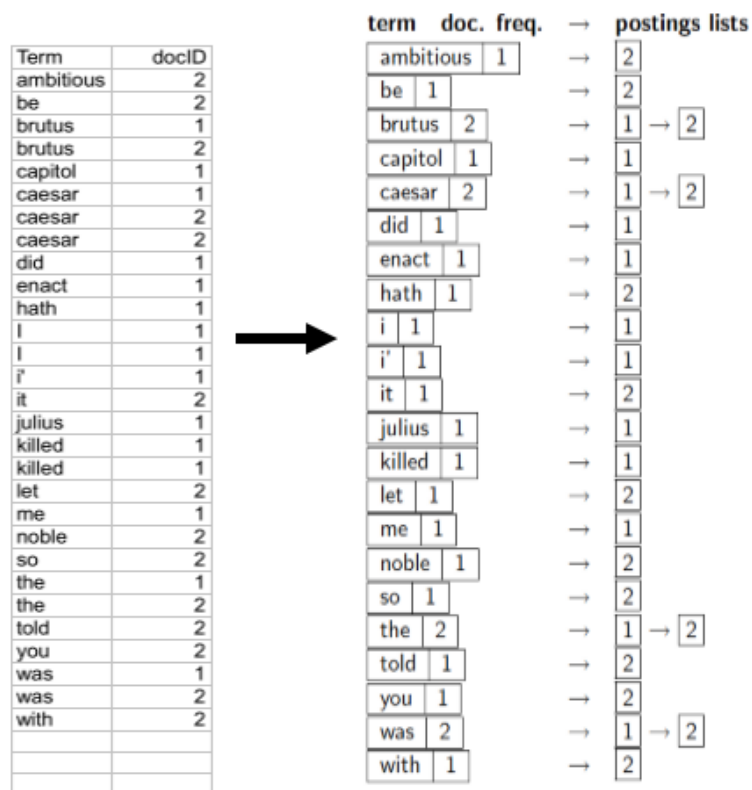
Ordenados por términos, y entonces cambia el docID.

Term	docID
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	docID
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

Otra de las etapas es **Diccionario & Postings**. Término múltiples entradas en un solo documento se fusionan. Dividir en el diccionario y contabilizar. La frecuencia de los Documentos se agrega a la información.



En la implementación de un sistema IR

- ¿Cómo indexar eficientemente?
- ¿Cómo almacenar lo que necesitamos?

Term	Doc #	Freq
a	2	1
aid	1	1
all	1	1
and	2	1
come	1	1
country	1	1
country	2	1
dark	2	1
for	1	1
good	1	1
in	2	1
is	1	1
it	2	1
manor	2	1
men	1	1
midnight	2	1
night	2	1
now	1	1
of	1	1
past	2	1
stormy	2	1
the	1	2
the	2	2
their	1	1
time	1	1
time	2	1
to	1	2
was	2	2

Term	N docs	Tot Freq
a	1	1
aid	1	1
all	1	1
and	1	1
come	1	1
country	2	2
dark	1	1
for	1	1
good	1	1
in	1	1
is	1	1
it	1	1
manor	1	1
men	1	1
midnight	1	1
night	1	1
now	1	1
of	1	1
past	1	1
s tormy	1	1
the	2	4
their	1	1
time	2	2
to	1	2
was	1	2

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

Ahora, según este esquema. ¿cómo procesar la consulta o query: **Brutus AND Caesar**

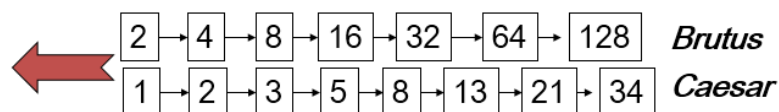
Localiza **Brutus** en el Diccionario; Recupera sus postings.

Localiza **Caesar** en el Diccionario; Recupera sus postings.

“Merge” los dos postings (intersectando el conjunto de documentos).

Observación: Si las listas son de longitud x e y , la mezcla toma $O(x+y)$ operaciones.

Crucial: postings ordenado por docID.



Intersectando las dos listas postings (“merge” algorithm)

```

1  INTERSECT( $p_1, p_2$ )
2  1  answer  $\leftarrow \langle \rangle$ 
3  2  while  $p_1 \neq \text{NIL}$  and  $p_2 \neq \text{NIL}$ 
4  3  do if  $\text{docID}(p_1) = \text{docID}(p_2)$ 
5  4      then ADD(answer,  $\text{docID}(p_1)$ )
6  5       $p_1 \leftarrow \text{next}(p_1)$ 
7  6       $p_2 \leftarrow \text{next}(p_2)$ 
8  7      else if  $\text{docID}(p_1) < \text{docID}(p_2)$ 
9  8          then  $p_1 \leftarrow \text{next}(p_1)$ 
10 9          else  $p_2 \leftarrow \text{next}(p_2)$ 
11 10 return answer

```

Resumen de Etapas Básicas en el Indexado

1. **Parsear** los documentos para reconocer la estructura, esto es, título, fecha, u otros campos.
2. **Escanear por tokens** las palabras (Tokenización)
 - Análisis léxico usando autómatas estado finito
 - números, caracteres especiales, mayúsculas, etc.
 - lenguaje, por ejemplo el Chino necesita *segmentación*, ya que no existe separación de palabras
 - registro posicional de la información para operadores de *proximidad*
3. **Stopword eliminar**
 - Palabras comunes y cortas, por ejemplo, “the”, “and”, “or”
4. **Stemming (Porter’s Stemming Algorithm)**
 - Procesar morfológicamente grupos de palabras y variantes tales como los plurales.
 - Una selección de reglas según Porter’s algorithm

<https://tartarus.org/martin/PorterStemmer/>

4.1 Algunos recursos para determinar Stemming en español

- i. <http://snowball.tartarus.org/>
- ii. <http://snowball.tartarus.org/texts/stemmersoverview.html>
- iii. <http://thinknook.com/keyword-stemming-and-lemmatisation-with-apache-solr-2013-08-02/>

SUFFIX	REPLACE MENT	E XAMPLE
sses	ss	stresses -> stress
ies	l	ponies -> poni
ss	ss	caress -> caress
s	NULL	cats -> cat
ing	NULL	making -> mak
...
at	ate	inflat(ed) -> inflate
...
y	l	happy -> happi
aliti	al	formaliti > formal
izer	ize	digitizer -> digitizæ
...
icate	ic	duplicate -> duplic
...
able	NULL	adjustable -> adjust
icate	NULL	microscopic -> microscop
...
e	NULL	inflat -> inflat

Un ejemplo de Stemming

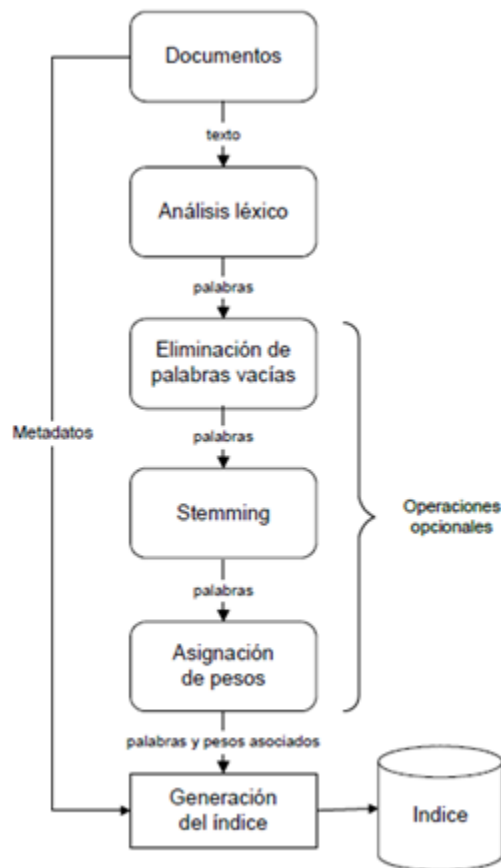


Diagrama de Flujo del modelo de Indexación

La indexación es el proceso de transformar elementos (documentos) en una estructura de datos que sean capaces de ser encontrados ante la solicitud de información del usuario. Para ello el documento requiere ser analizado, esto es: identificar la meta-información (por ejemplo, Autor, título, etc.), análisis lingüístico del contenido (complejo), para en el proceso de búsqueda correlacionar las consultas de los usuarios con los documentos representados en el índice.

<https://lingpipe-blog.com/2014/03/08/lucene-4-essentials-for-text-search-and-indexing/>

Una visión o modelo más general para un sistema de recuperación de la información debe contener una serie de pasos importantes.

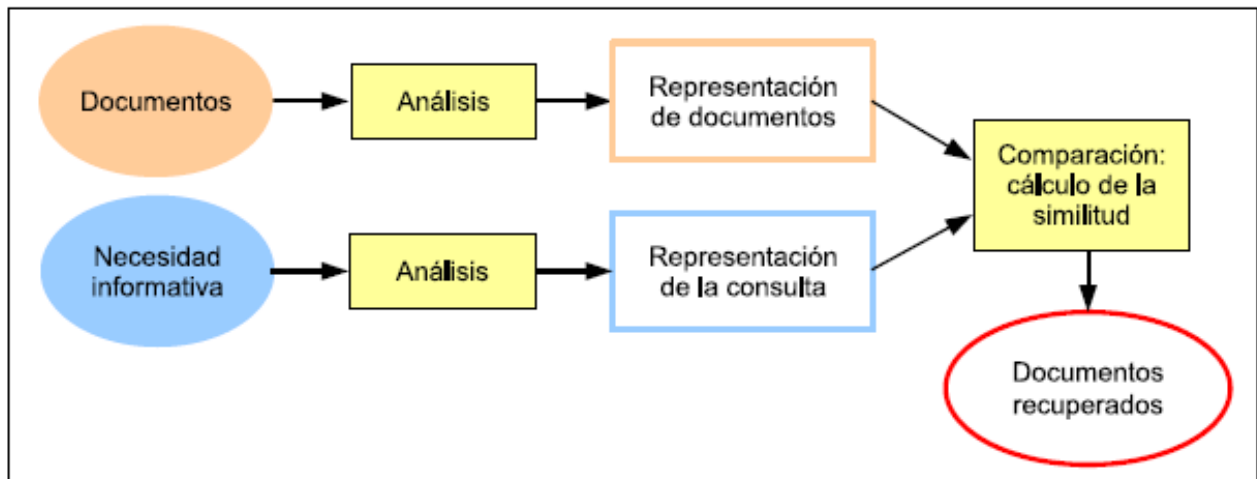
- a) **Obtener representación de los documentos.** Generalmente los documentos se presentan utilizando un conjunto más o menos grande de términos índice. La elección de dichos términos es el proceso más complicado. Los documentos pueden

ser documentos primarios: informes, artículos, páginas web, etc. documentos secundarios (título, autor, resumen, etc.).

- b) **Identificar la necesidad informativa del usuario.** Se trata de obtener la representación de esa necesidad, y plasmarla formalmente en una consulta acorde con el sistema de recuperación. La necesidad informativa se expresa formalmente mediante una consulta, puede emplear términos índice y operadores booleanos. Puede realizarse en lenguaje natural.
- c) **Búsqueda de documentos que satisfagan la consulta.** Consiste en comparar las representaciones de documentos y la representación de la necesidad informativa para seleccionar los documentos pertinentes
- d) **Presentación de los resultados al usuario.** Puede ser desde una breve identificación del documento hasta el texto completo
- e) **Evaluación de los resultados.** Para determinar si son acordes con la necesidad informativa

Paradigma del problema IR

- i) Representación y búsqueda (indexing & searching).
- ii) Obtener representaciones homogéneas de documentos y consultas para su comparación



Análisis del texto para determinar el tratamiento que se realizará sobre números, guiones, signos de puntuación, tratamiento de mayúsculas y/o minúsculas, etc..

Eliminación de palabras vacías, muy frecuentes y muy poco frecuentes. Poseen muy poca capacidad semántica. Se reduce el número de términos con valores muy pocos

significativos para la recuperación. Si un término aparece en casi todos los documentos no sirve para diferenciar unos de otros

Aplicación de lematización (stemming) sobre los términos resultantes para eliminar variaciones morfo-sintácticas y obtener lemas. En un diccionario o repertorio léxico, elegir convencionalmente una forma para remitir a ellas todas las que derivan de su misma familia por razones de economía. Palabras que son variaciones morfológicas con un significado prácticamente idéntico. Muchas palabras se forman a partir de otras, conservando una relación semántica. Pueden formarse por dos vías, una de ellas es, por flexión morfológica, por ejemplo: libro, libros, o bien por derivación, por ejemplo: libro, librero, librería.

Podríamos pensar en agrupar todas esas palabras parecidas bajo una forma común, sin embargo, esto debería afectar al recuento de frecuencias y, en consecuencia, a los pesos. Una variante menos drástica es el **s-stemming** que consiste en eliminar las **s** finales de todas las palabras, esto incluye las formas en plural de sustantivos y adjetivos, pero también de todas las demás palabras puede ser refinado incluyendo la eliminación de plurales terminados en **-es**, también ciertas vocales finales que suelen denotar, en los adjetivos, variaciones de género, el **s-stemmer** produce buenos resultados

Sin stemming	s-stemming
informa	informa
contenian	contenian
compañias	compañia
discuten	discuten
informacion	informacion
noticias	noticia
proporcionan	proporcionan
contaminacion	contaminacion
diferentes	diferent
relevantes	relevant
ofrecieron	ofrecieron
documentos	documento
descubrimiento	descubrimiento
pesticidas	pesticida
encontrar	encontrar
bebes	beb
medidas	medida
marcas	marca
alimentos	alimento
supermercados	supermercado

Selección de términos que serán considerados términos índice (sustantivos, nombres propios). Con el objetivo de reducir la carga computacional, se intentan seleccionar los mejores términos índice.

Utilización de tesauros. Puede ayudar tanto en el proceso de indización como en el de búsqueda de información (expansión de consultas).

Modelo Booleano

Documentos: Suele realizarse indización manual: a partir de la lectura y comprensión del texto el indizador decide asignar los mejores términos que representen su contenido: descriptores

Consultas: Las consultas se formulan utilizando los términos índice (descriptores) y una serie de operadores (booleanos, de proximidad, selección, truncamiento, etc.) y facilidades (índices, tesauros, etc.)

El sistema de recuperación es sencillo, todo el esfuerzo recae en el usuario a la hora de plantear la consulta Típico en bibliotecas, etc.

Modelo Vectorial

Documentos: Se lleva a cabo una indización automática, es un proceso complejo que trata de asignar automáticamente los mejores términos índice a los documentos (selección y extracción de términos)

Consultas: Las consultas se realizan en lenguaje natural. El mismo proceso de indización automática se aplica a la consulta para obtener los términos índice que la representan.

El sistema de recuperación es complejo. Todo el esfuerzo recae en él. Típico en motores de búsqueda de Internet (los mejores motores añaden información de enlaces, ej. Google).

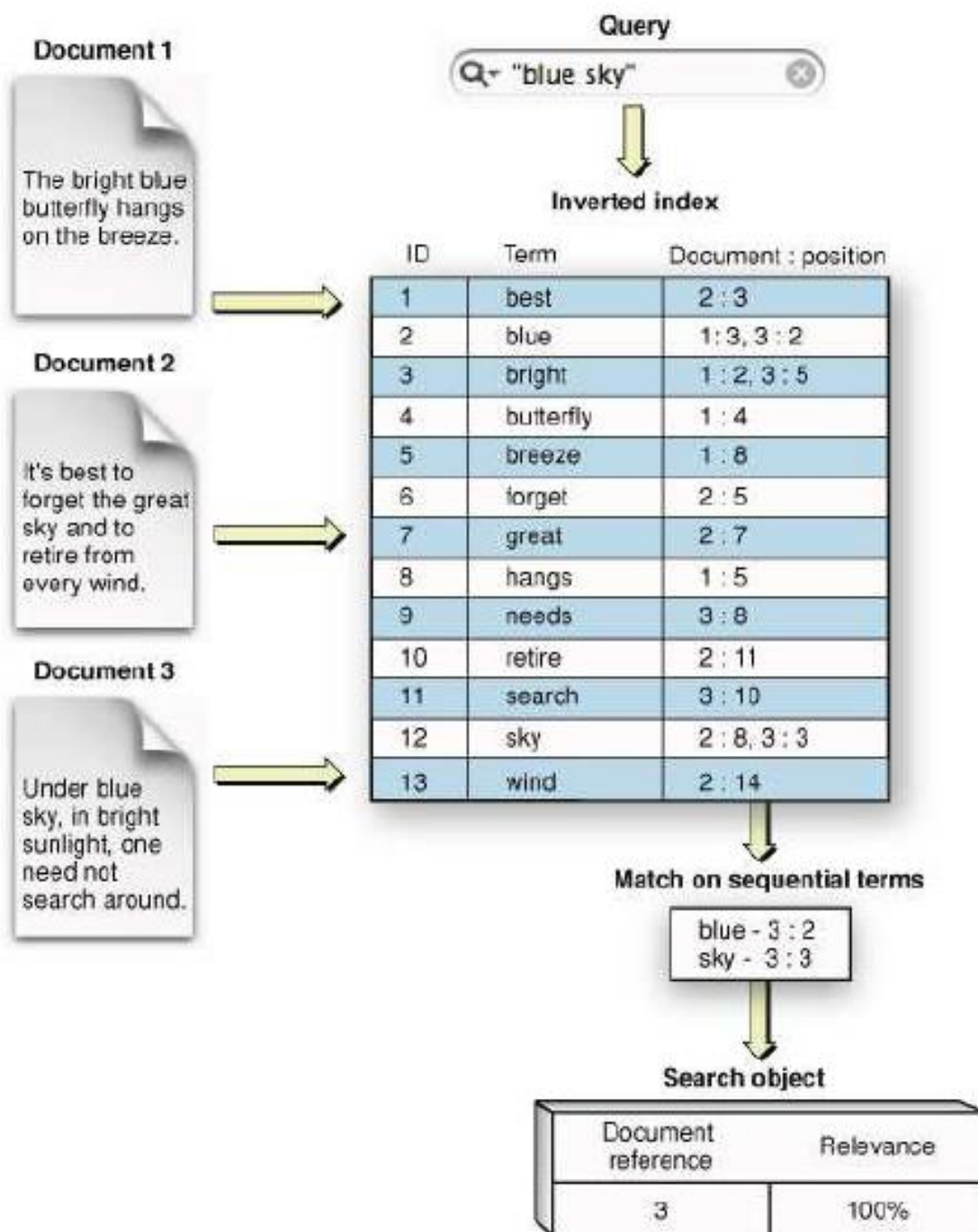
Objetivo:

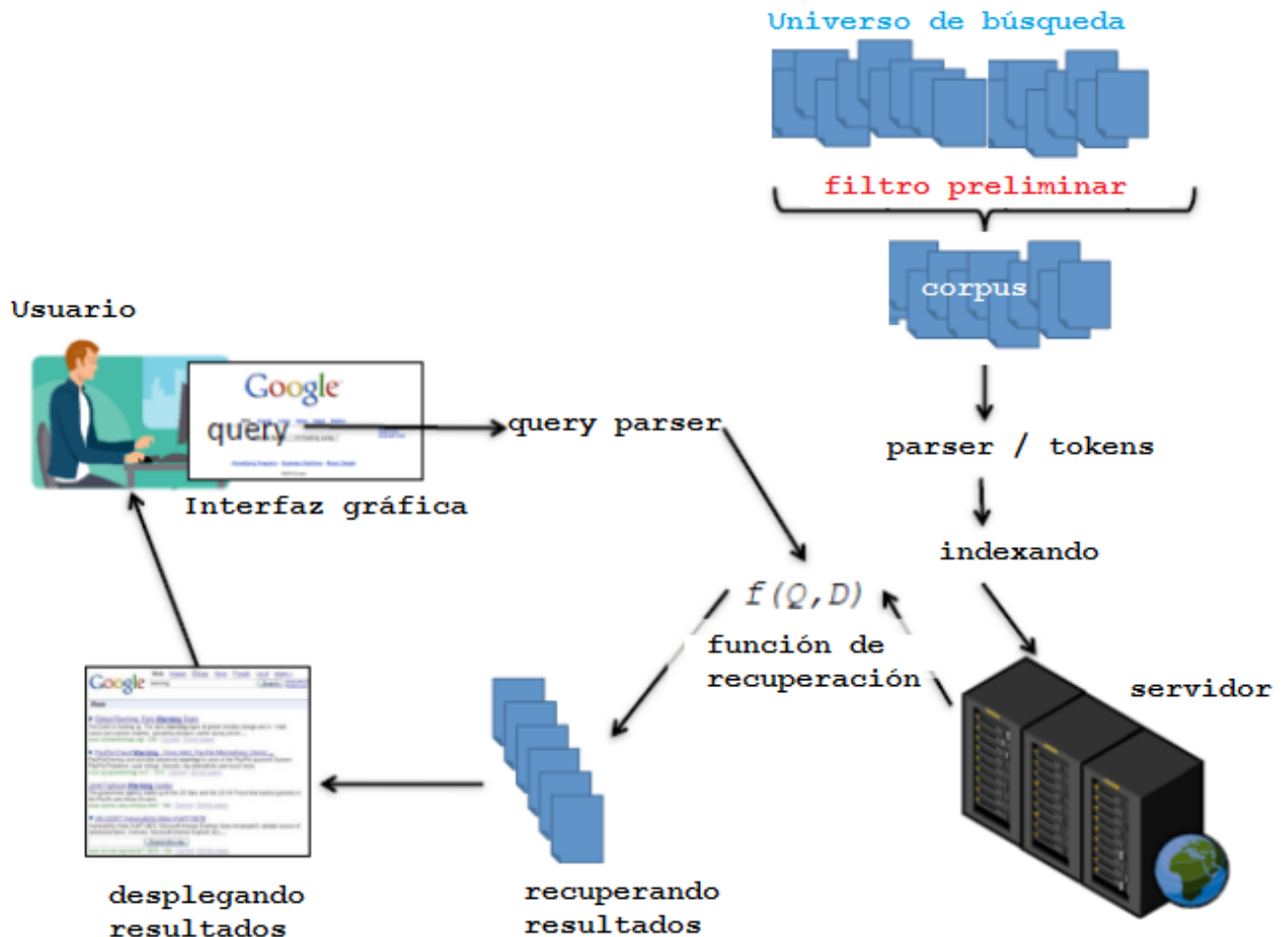
Crear un sistema de RI, esto es un motor de recuperación tomando como unidades documentales, las memorias de título de la carrera, digitalizadas, y como consultas, las introducidas en lenguaje natural por un usuario.

Requisitos o tareas a realizar

- a) Creación de un índice, para depositar y localizar los documentos

- b) Introducción de consultas, la que puede ser en base a consultas booleanas, con comillas, u otras.
- c) Recuperación de documentos en base a las consultas introducidas, el modelo debe permitir ordenar por relevancia los documentos (de manera que no se puede usar sólo el booleano, ya que debe coexistir con el modelo vectorial)
- d) Visualización de resultados, la aplicación debería mostrar los documentos relevantes ordenados, y la posibilidad de verlos una vez que los escoja.



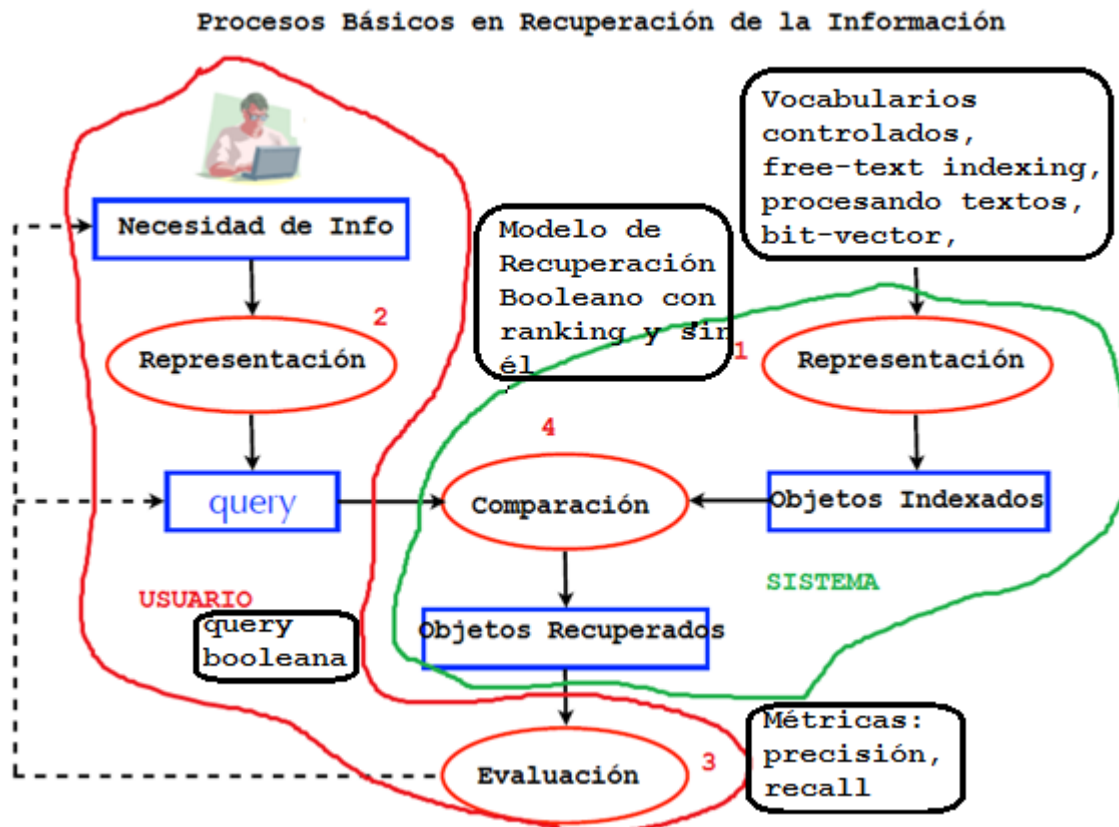


Implementando la recuperación de la información usando inverted index, bajo un modelo booleano.

<http://www.cs.utexas.edu/users/mooney/ir-course/>

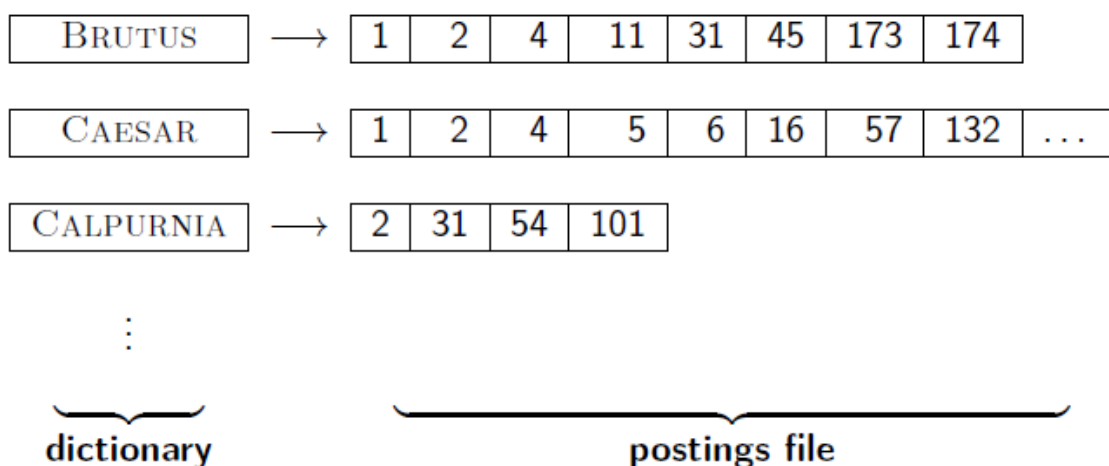
http://www.googleguide.com/quoted_phrases.html

https://www.elastic.co/guide/en/elasticsearch/guide/current/_ngrams_for_partial_matching.html



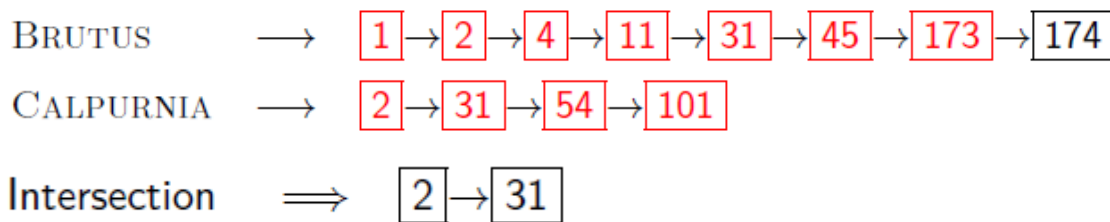
Respecto a 3) Sin embargo, falta por considerar el lenguaje de consulta. ¿Cómo implemento en mi propuesta de sistema la forma de plantear la consulta, esto es, ¿Cuál sería el lenguaje de consulta más apropiado?.

Lenguaje de Consulta: Los sistemas por lo general no son tan restrictivos y existen otras formas de consultar, estos son un ejemplo de ellos.



Para una consulta: Brutus AND Calpurnia

- Encontrar todos los documentos que son acertados usando inverted index:
- Localiza **Brutus** en el diccionario
- Recuperamos su postings list desde el archivo postings.
- Localizamos **Calpurnia** en el diccionario
- Recuperamos su postings list desde el archivo postings file
- Intersectamos los dos postings lists
- Return intersección al usuario.



- a) *Consultas básicas*: Palabras claves combinadas con operadores Booleanos. Por ejemplo,
OR: $(e_1 \text{ OR } e_2)$, AND: $(e_1 \text{ AND } e_2)$, BUT: $(e_1 \text{ BUT } e_2)$ Satisfice e_1 pero **not** e_2

Palabra clave primitiva: Recupera los documentos usando el inverted index.

OR: Recursivamente recupera e_1 y e_2 y toma la unión de resultado.

AND: Recursivamente recupera e_1 y e_2 y toma la intersección de resultado.

BUT: Recursivamente recupera e_1 y e_2 y toma el conjunto diferencia de resultado. Por lo general consultas con el fin de obtener documentos que contengan el término t, por ejemplo.

- b) *Consultas compuestas*: términos que aparecen consecutivamente. Por ejemplo, “Santa María del Mar”, “Retrieval Information”. Evidentemente que no podemos tratar proximidad de operadores con este index, ya que necesitamos de otra información adicional. ***positional index***

Universidad	Norte	ULS
$df=3$	$df=4$	$df=3$
2, 1 [2]	2, 2 [3,44]	2, 3 [4,45,78]
22, 1 [31]	22, 2 [32,66]	22, 3 [33,67,78]
45, 1 [2]	45, 3 [3,46,101]	45, 2 [5,34]
	3421, 1 [2]	

$docid, tf [pos_1, pos_2, ..., pos_tf]$

PubMed búsqueda avanzada permite a los usuarios construir una consulta booleana que busca en diferentes campos:

The screenshot shows the PubMed search bar with the following query: `(light therapy)[title] OR (phototherapy)[title] AND adverse effects[abstract]`. The search is set to 'PubMed' and there are links for 'Create RSS', 'Create alert', and 'Advanced'.

O combinación de campos,

The screenshot shows the PubMed search bar with the following query: `((light therapy)[title/Abstract] OR (phototherapy)[title/Abstract]) AND adverse effects`. The search is set to 'PubMed' and there is a link for 'Advanced'.

Con comentarios sobre la consulta y detalles de la búsqueda

PubMed.gov
US National Library of Medicine
National Institutes of Health

PubMed

((light therapy)[title/Abstract] OR (phototherapy)[title/Abstract]) AND adverse effects

Advanced

Search Details

Query Translation:

```
((light[All Fields] AND therapy[All Fields]) AND (title[All Fields] AND Abstract[All Fields]) OR phototherapy[All Fields] AND (title[All Fields] AND Abstract[All Fields])) AND (adverse[All Fields] AND effects[All Fields])
```

Search URL

Result:

2

Database:

PubMed

User query:

((light therapy)[title/Abstract] OR (phototherapy)[title/Abstract]) AND adverse effects

¿Cómo podemos aplicar técnicas que ya conocemos?

Solución 1: procesando los campos específicos a la sub-query usando el index apropiado y merge los resultados.

PubMed.gov
US National Library of Medicine
National Institutes of Health

PubMed

((light therapy) AND title/Abstract OR (phototherapy) AND title/Abstract) AND adverse effects

Search

Send to: Filters: Manage Filters

Find related data
Database: Select
Find items

Search details
((light[All Fields] AND therapy[All Fields]) AND (title[All Fields] AND Abstract[All Fields]) OR phototherapy[All Fields] AND (title[All Fields] AND Abstract[All Fields])) AND (adverse[All Fields] AND effects[All Fields])
Search See more...

Recent Activity
Turn Off Clear
Q ((light therapy) AND title/Abstract OR (phototherapy) AND title/A... (2) PubMed
Q (phototherapy) AND title OR (phototherapy) AND abstract (3285) PubMed
Q (phototherapy) OR (phototherapy) (33956) PubMed

Article types
Clinical Trial
Review
Customize ...

Text availability
Abstract
Free full text
Full text

PubMed Commons
Reader comments
Trending articles

Publication dates
5 years
10 years
Custom range...

Species
Humans
Other Animals

Clear all
Show additional filters

Format: Summary Sort by: Most Recent

Search results
Items: 2


Showing results for a modified search because your search retrieved no results.

- ☐ [Light therapy for preventing seasonal affective disorder.](#)
Nussbaumer B, Kaminski-Hartenthaler A, Forneris CA, Morgan LC, Sonis JH, Gaynes BN, Greenblatt A, Wipplinger J, Lux LJ, Winkler D, Van Noord MG, Hofmann J, Gartlehner G. Cochrane Database Syst Rev. 2015 Nov 8;(11):CD011269. doi: 10.1002/14651858.CD011269.pub2. Review. PMID: 26558494
[Similar articles](#)
- ☐ [Second-generation antidepressants for preventing seasonal affective disorder in adults.](#)
Gartlehner G, Nussbaumer B, Gaynes BN, Forneris CA, Morgan LC, Kaminski-Hartenthaler A, Greenblatt A, Wipplinger J, Lux LJ, Sonis JH, Hofmann J, Van Noord MG, Winkler D. Cochrane Database Syst Rev. 2015 Nov 8;(11):CD011268. doi: 10.1002/14651858.CD011268.pub2. Review. PMID: 26558418
[Similar articles](#)

(phototherapy)[title] OR (phototherapy)[abstract]

Resultado Index	Resultado Autor	Resultado Final
<u>phototherapy[title]</u>	<u>phototherapy[abstract]</u>	<u>final</u>
Contador=3	Contador=5	Contador=5
1, 8	1, 3	1, 11
10, 2	10, 2	10, 4
16, 5	16, 5	16, 10
	33, 2	33, 2
	56, 10	56, 10

- c) *Consulta posicional*: Exactamente “Barack Obama”, “Laurel Hardy”, esto es Eric debe aparecer antes que jeltsch, sin nada de por medio!!



[Todo](#)
[Imágenes](#)
[Noticias](#)
[Vídeos](#)
[Maps](#)
[Más ▼](#)
[Herramientas de búsqueda](#)

Cerca de 939 resultados (0,32 segundos)

Dr. Eric Jeltsch F. - Universidad de La Serena

dns.uls.cl/~ej/

Eric R. Jeltsch F. Depto. de Matemáticas, Av. Cisternas 1200, La Serena, CHILE. 2º Piso, Of. 215.
Fono: (+56)-51-2-334732. e-mail: ejeltsch@userena.cl ...

Asignaturas

Las asignaturas que dicto son exclusivamente para la carrera ...

[Más resultados de uls.cl »](#)

Programación y Computación ...

Programación y Computación.
Ingenierías. Dr. Eric Jeltsch F.


eric jeltsch | LinkedIn

<https://cl.linkedin.com/in/eric-jeltsch-275b3328>

Ver el perfil profesional de eric jeltsch en LinkedIn. LinkedIn es la red de negocios más grande del mundo que ayuda a profesionales como eric jeltsch a ...

Imágenes de "eric jeltsch"

[Notificar imágenes](#)



Wildcard queries: (Capítulo 3 de libro guía [MRS]). Por ejemplo, “Scarlet? Johans*n”, con el uso de **Expresiones Regulares**, light **AND** therap* , considera los términos therapy, therapies, therapeutic, etc. son todos igualmente relevantes. O bien, puede imaginar hacer esto con suffixes, por ejemplo

*eutic : [hermeneutic, pharmaceutic, therapeutic, ...]

light **AND** *eutic

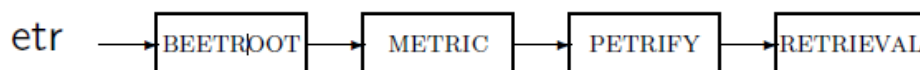
light **AND** (hermeneutic **OR** pharmaceutic **OR** therapeutic)

(u e)nabl(e ing) matches	(un en)*able matches
unable	able
unabling	unable
enable	unenable
enabling	enununable

d) *Consultas sonoras:* (soundex algorithm). Por ejemplo, ideal para nosotros que tratamos de pronunciar nombres en Inglés. “Eric jesch”, “Clint Istwood”, etc.

e) *Consultas aproximadas:* Un **K-gram** es una subsecuencia de n elementos de una secuencia dada. El estudio de los n-gramas es interesante en diversas áreas del conocimiento. La forma en la que extraemos los gramas se tiene que adaptar al ámbito que estamos estudiando y al objetivo que tenemos en mente. Se puede usar gramas para casi todos los ámbitos. **K-gram:** Un inverted index para una secuencia de k caracteres contenidos en una palabra. Por ejemplo, 3-grams para “index”:
\$in, ind, nde, dex, ex\$ (donde \$ es un caracter especial que denota la partida o final de una palabra). Para todo k-gram encontrado en el diccionario, el k-gram index tiene un puntero a todas las palabras que contiene el k-gram. Por ejemplo,
dex → {index, dexterity, ambidextrous}.

Postings list en un 3-gram inverted index



Prefixos: Pattern que aciertan al comienzo de la palabra. Por ejemplo, “anti” matches “antiquity”, “antibody”, etc.

Suffixes: Pattern que aciertan al final de la palabra. Por ejemplo, “ix” matches “fix”, “matrix”, etc.

Substrings: Pattern que aciertan una subsecuencia arbitraria de caracteres. Por ejemplo, “**rapt**” matches “en**rapt**ure”, “velocir**rapt**or” etc.

Ranges: Par de cadenas que aciertan cualquier palabra lexicográficamente (alfabéticamente) entre ellas. Por ejemplo, “tin” to “tix” matches “tip”, “tire”, “title”, etc. para ello se usan los algoritmos, Edit distance (Levenshtein distance) y Longest Common Subsequence, entre otros.

Por ejemplo, se han usado n-gramas para extraer características comunes de grandes conjuntos de imágenes de la Tierra tomadas desde satélite, y para determinar a qué parte de la Tierra pertenece una imagen dada.

Un n-grama es una ventana de n caracteres que se van extrayendo del texto, empezando en la primera posición y avanzando una posición cada vez

Ejemplo: la palabra _libro_ produce, cuando $n = 3$, (_ significa espacio en blanco) _li lib ibr bro ro_. La palabra _librero_ produce: _li lib ibr bre ere ero ro_ se espera que palabras parecidas produzcan n-gramas parecidos.

Ejemplo. N-gramas, para “statistics”, bigramas: st, ta, at, ti, is, st, ti, ic, cs, o bien trigramas: sta, tat, ati, tis, ist, sti, tic, ics. Por ejemplo, “statistics”, genera los siguientes bigramas: st, ta, at, ti, is, st, ti, ic, cs, en donde encontramos 7 únicos bigramas: at, cs, ic, is, st, ta, ti. Por otra parte, “statistical”, consta de bigramas: st, ta, at, ti, is, st, ti, ic, ca, al, con 8 únicos bigramas: al, at, ca, ic, is, st, ta, ti.

Con lo anterior ya estamos en condiciones de usar el **Dice's coefficient** para computar “similaridad” para un par de palabras”. En efecto, A es el n° único de bigramas para la primera palabra, B es el n° único de bigramas en la segunda palabra, y C es el n° único de bigramas compartidos. En este caso, Similaridad $S = \frac{2C}{A + B}$, es igual a $(2*6)/(7+8) = .80$.

Métricas de comparación de cadenas

Las métricas de comparación de cadenas son métricas que miden lo parecidas que son dos cadenas de caracteres (o, en otras palabras, la distancia que hay entre ellas). Dado el trabajo que se pretende desarrollar en este proyecto, creemos que este tipo de métricas pueden resultar útiles, por ejemplo, para puntuar las sugerencias de un corrector ortográfico.

La **mínima Edit Distance** entre dos cadenas es el mínimo número de operaciones de edición, Inserción, Borrado, Substitución, necesarias para transformar una cadena en la otra.

I N T E * N T I O N
| | | | | | | | | |
* E X E C U T I O N
d s s i s

Si toda operación tiene costo 1, entonces la distancia es 5. Mientras que si el costo es 2 (Levenshtein), la distancia entre ellas es 8. Más ejemplo, “misspell” a “mispell” es distancia 1, “misspell” a “mistell” es distancia 2, “misspell” a “misspelling” es distancia 3.



`DamerauLevenshteinDistance["misspell","mistell"]`

Web Apps Examples Random

Input:
`DamerauLevenshteinDistance[misspell, mistell]`

Result:
2

Download page POWERED BY THE WOLFRAM LANGUAGE



`EditDistance["misspell","mistell"]`

Web Apps Examples Random

Input:
`EditDistance[misspell, mistell]`

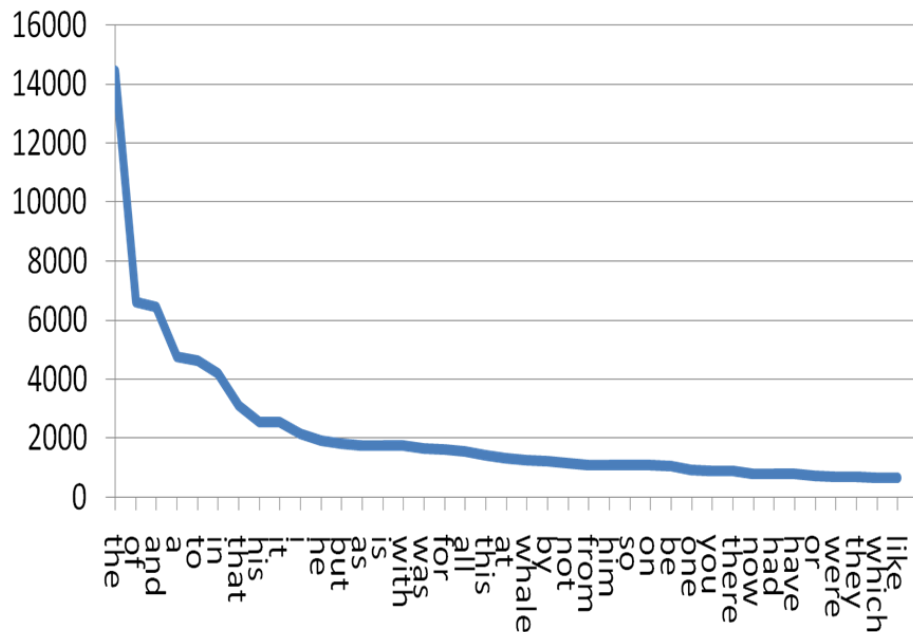
Result:
2

Download page POWERED BY THE WOLFRAM LANGUAGE

En el ámbito de los Procesando Lenguaje Natural (Natural Language Processing), se habla de distancia (minimizada), o peso. El uso más frecuente de estos algoritmos es

Una de las principales características, es que algunos elementos ocurren muy frecuentemente, otros muchos elementos ocurren muy pocas veces, y es así que la frecuencia de palabras en el texto cae muy rápidamente.

Distribución de palabras: (Frecuencia vs. rango de todas las palabras en Moby Dick.



Las palabras más frecuentes son términos que ocurren en casi cada documento normalmente no tienen relación con los conceptos y las ideas representadas en el documento

Zipf (y posteriormente de H.P. Luhn) postula que el poder resolutivo de palabras significativas alcanza un pick en la mitad del camino entre las dos tranchas resolver potencia: la capacidad de las palabras para discriminar el contenido

La Ley del Heap modela el número de palabras del vocabulario como una función del tamaño del corpus: ¿Cuál es el número de palabras únicas apareciendo en un corpus de tamaño N palabras?. Esto determina cómo el tamaño del índice inverso escala con el tamaño del corpus.

Dado, M es el tamaño del vocabulario, y T es el n° de tokens distintos en la colección, entonces, $M = kT^b$ k, b depende del tipo de colección, esto es, $30 \leq k \leq 100$ y $b \approx 0.5$.

Los Top 30 de Moby Dick

Rango	Término	Frecuencia	Rango	Término	Frecuencia
#1	the	-> 14620	#16	for	-> 1646
#2	of	-> 6732	#17	was	-> 1646
#3	and	-> 6502	#18	all	-> 1543
#4	a	-> 4776	#19	this	-> 1443
#5	to	-> 4706	#20	at	-> 1335
#6	in	-> 4230	#21	whale	-> 1232
#7	that	-> 3099	#22	by	-> 1226
#8	it	-> 2535	#23	not	-> 1171
#9	his	-> 2530	#24	from	-> 1105
#10	i	-> 1989	#25	on	-> 1073
#11	he	-> 1878	#26	him	-> 1067
#12	but	-> 1823	#27	so	-> 1066
#13	with	-> 1770	#28	be	-> 1064
#14	as	-> 1753	#29	you	-> 946
#15	is	-> 1750	#30	one	-> 925

Sin embargo, existen otros conceptos propios de la lingüística. ¿Cuál es la macroestructura del texto?. Y así determinar la complejidad argumentativa del texto como producto semántico. Escribe mucho, pero no dice nada!!!.

La **Ley de Zipf** es uno de los más importantes, ya que está relacionado con saber la frecuencia con la que una palabra aparece en un texto, en general afirma que un pequeño n° de palabras son utilizadas con mucha frecuencia, mientras que frecuentemente ocurre que un gran n° de palabras son poco empleadas. En los años cuarenta, el lingüista George Zipf se dio cuenta de que las palabras y su número de apariciones en textos, seguían alguna ley especial. La palabra más utilizada ocuparía el número uno en el ranking, el número dos se corresponde con la segunda palabra más veces repetida, etc. Así, se guardaba una estrecha relación entre el número de apariciones de las palabras más frecuentes. La primera palabra más utilizada aparecía el doble de veces que la segunda y tres veces más que la tercera, y sigue el patrón según esta norma.

Ejemplo. Moby Dick; o bien, The Whale by Herman Melville, la palabra más frecuente fue “**the**” con 14620 apariciones, la segunda es “**of**” con 6732 apariciones, y la tercera “**and**” aparece 6502 veces, la cuarta es “**a**” con 4776 apariciones.

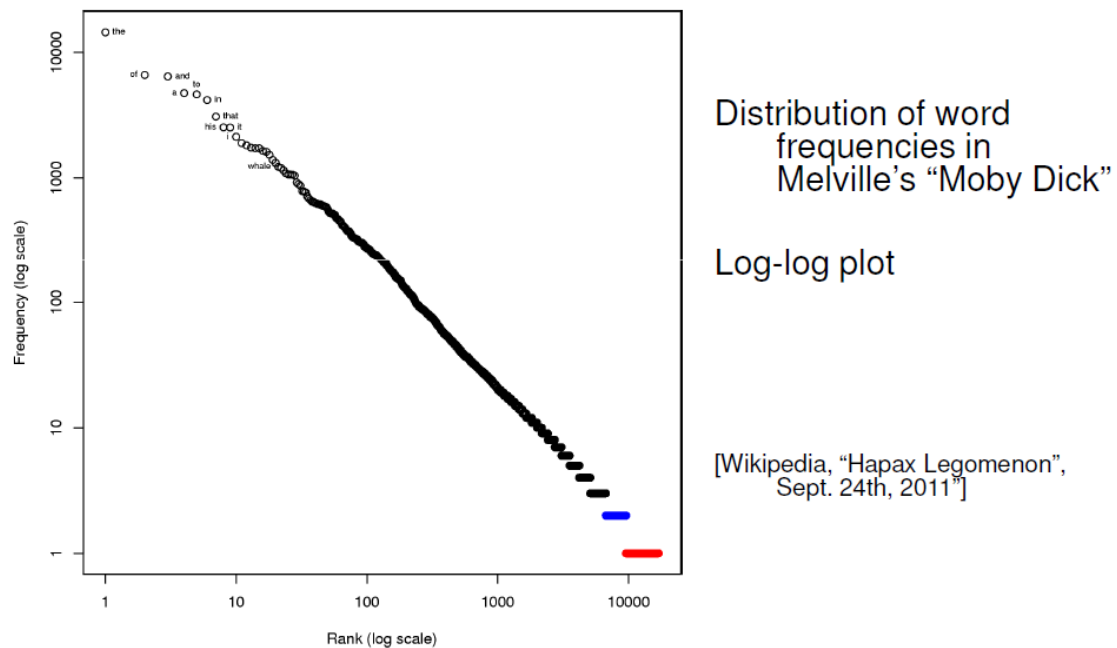
Los Top 30 de Moby Dick

Rango	Término	Frecuencia	Rango	Término	Frecuencia
#1	the ->	14620	#16	for ->	1646
#2	of ->	6732	#17	was ->	1646
#3	and ->	6502	#18	all ->	1543
#4	a ->	4776	#19	this ->	1443
#5	to ->	4706	#20	at ->	1335
#6	in ->	4230	#21	whale ->	1232
#7	that ->	3099	#22	by ->	1226
#8	it ->	2535	#23	not ->	1171
#9	his ->	2530	#24	from ->	1105
#10	i ->	1989	#25	on ->	1073
#11	he ->	1878	#26	him ->	1067
#12	but ->	1823	#27	so ->	1066
#13	with ->	1770	#28	be ->	1064
#14	as ->	1753	#29	you ->	946
#15	is ->	1750	#30	one ->	925

El término más frecuente representa el 10% del texto, el segundo término más frecuente representa el 5% del texto, el tercero más frecuente representa el 3%, etc..

P_t = proporción de la colección que corresponde al término t , c = constante, para Inglés se usa $c = 0.1$. N , el n° total de ocurrencia de términos en la colección. r_t = frecuencia basada en el rango de términos de t . $P_t = c / r_t$

Una de las aplicaciones típicas es en el campo de la economía urbana, la dinámica demográfica y en particular la distribución del tamaño de las ciudades, en donde ha sido un tema de investigación que ha atraído mucho la atención durante las últimas décadas. En la literatura y en particular en el tema de recuperación de la información se ha enfocado en mostrar si empíricamente se cumplen las leyes de **Zipf** y otras. En particular, la ley de Zipf establece que el tamaño de las ciudades sigue una distribución de Pareto con coeficiente igual a 1. En la práctica esto significa que la ciudad de mayor tamaño (por ejemplo, Santiago de Chile) debería ser dos veces más grande que la segunda ciudad en cuestión (Antofagasta), y tres veces más que la tercera (Valparaíso), y así sucesivamente....



Log-log plot

[Wikipedia, "Hapax Legomenon",
Sept. 24th, 2011"]

<http://www.philippeadjiman.com/blog/2009/10/26/drawing-the-long-tail-of-a-zipf-law-using-gnuplot-java-and-moby-dick/>

<http://www.gutenberg.org/ebooks/2701> (moby dick se sacó del proyecto Gutenberg)

<http://lucene.apache.org/> (uso lucene)

<https://code.google.com/archive/p/google-collections/> (se traslado a Guava,

<https://github.com/google/guava> Ud. lo vera)

<http://www.philippeadjiman.com/blog/wp-content/uploads/2009/10/mobyDickTop100WordOccurrences1.txt>

para curiosear las primeras 100 palabras.

La ley de Zipf, se usa para el análisis de frecuencia de aparición de términos dentro de una colección de documentos. Dice que la frecuencia **F** de aparición de un término en una colección es inversamente proporcional a su ranking **R** en una tabla ordenada de frecuencias. Otra forma, si tomamos cualquier longitud de palabras de un texto, y se analiza la ocurrencia de las mismas, en orden decreciente de frecuencia y se multiplica por su frecuencia, esto es igual a la constante, **F · R = C**.

En nuestra notación, $\mathbf{F} = \mathbf{f}_t$ = frecuencia (nº de veces que el término t ocurre), $\mathbf{R} = \mathbf{r}_t$ = frecuencia del rango base del término t , siendo $\mathbf{C} = \mathbf{k}$ = constante.

<http://facweb.cs.depaul.edu/mobasher/classes/CSC478/lecture.html>

Vale mencionar que cuando se trabaja con raíces de palabras (**stemming**) se reduce considerablemente el tamaño del texto a tratar, con una paralela reducción del tamaño de la estructura de los índices, ya que las raíces son más frecuentes que las palabras, lo que facilita la búsqueda. La aplicación de la Ley de Zipf en los procesos del Análisis Documental tiene dos funciones primordiales: la recuperación de información y la indización automática. Los Top 10 palabras más frecuentes en algunos lenguajes de muestra:

English		German		Spanish		Italian	
1	the 61,847	1	der 7,377,879	1	que 32,894	1	non 25,757
2	of 29,391	2	die 7,036,092	2	de 32,116	2	di 22,868
3	and 26,817	3	und 4,813,169	3	no 29,897	3	che 22,738
4	a 21,626	4	in 3,768,565	4	a 22,313	4	è 18,624
5	in 18,214	5	den 2,717,150	5	la 21,127	5	e 17,600
6	to 16,284	6	von 2,250,642	6	el 18,112	6	la 16,404
7	it 10,875	7	zu 1,992,268	7	es 16,620	7	il 14,765
8	is 9,982	8	das 1,983,589	8	y 15,743	8	un 14,460
9	to 9,343	9	mit 1,878,243	9	en 15,303	9	a 13,915
10	was 9,236	10	sich 1,680,106	10	lo 14,010	10	per 10,501
BNC, 100Mw		"Deutscher Wortschatz", 500Mw		subtitles, 27.4Mw		subtitles, 5.6Mw	

<http://corpus.leeds.ac.uk/protected/query.html> MW= million-word text corpus, medida.

¿Cuál será la palabra que más escribirán algunos escritores? ¿Qué palabra será la más usada por Pablo Neruda.? ¿Cuál será la palabra que más se escribe; o la que no escribo tanto?. Y, si queremos saber ..¿Cuál ha sido la cadena de televisión más vista en un período de tiempo determinado?, una vez recogido los datos de un corpus cuyo contenido son los resultados de un audímetro.

Variaciones del esquema de peso **tf-idf** son empleadas frecuentemente por los motores de búsqueda como herramienta fundamental para medir la relevancia de un documento dada una consulta del usuario, estableciendo así una ordenación o ranking de los mismos.

tf-idf puede utilizarse exitosamente para el filtrado de las denominadas *stop-words* (palabras que suelen usarse en casi todos los documentos), en diferentes campos como la clasificación y resumen de texto.

Una de las funciones de ranking más sencillas se calcula como la suma de los valores tf-idf de cada término de la consulta.

Muchas funciones de ranking más complejas constituyen variaciones de este simple modelo. Contar el número de veces que una palabra aparece en un documento nos da el **peso local** de ella en el documento en cuestión.

Contar el número de documentos en los que aparece esa palabra aunque sea una vez, nos da el **peso global** de ella con respecto a la colección de documentos que se está analizando.

El peso local se denomina **Term Frequency (TF)**, y se calcula contando el número de veces que la palabra aparece en el documento dividido entre el número total de palabras contenidas en él

$$TF_{i,d} = \text{contador del término} / \text{número total de palabras}$$

Por su parte, el **peso global** se denomina **Inverse Document Frequency (IDF)**, y se calcula a través del logaritmo del número total de documentos de la colección dividido por el número de documentos que contienen la palabra.

Factor IDF: Inverse Document Frequency (Frecuencia Inversa del Documento para un Término). El factor IDF de un término es inversamente proporcional al número de documentos en los que aparece dicho término. Esto significa que cuanto menor sea la cantidad de documentos, así como la frecuencia absoluta de aparición del término, mayor será su factor IDF y a la inversa, cuanto mayor sea la frecuencia absoluta relativa a una alta presencia en todos los documentos de la colección, menor será su factor discriminatorio.

$$IDF_i = \log_{10} (\text{número total de documentos en colección} / \text{número de documentos que contienen la palabra } i), \text{ y para evitar la división por cero, se le suma 1.}$$

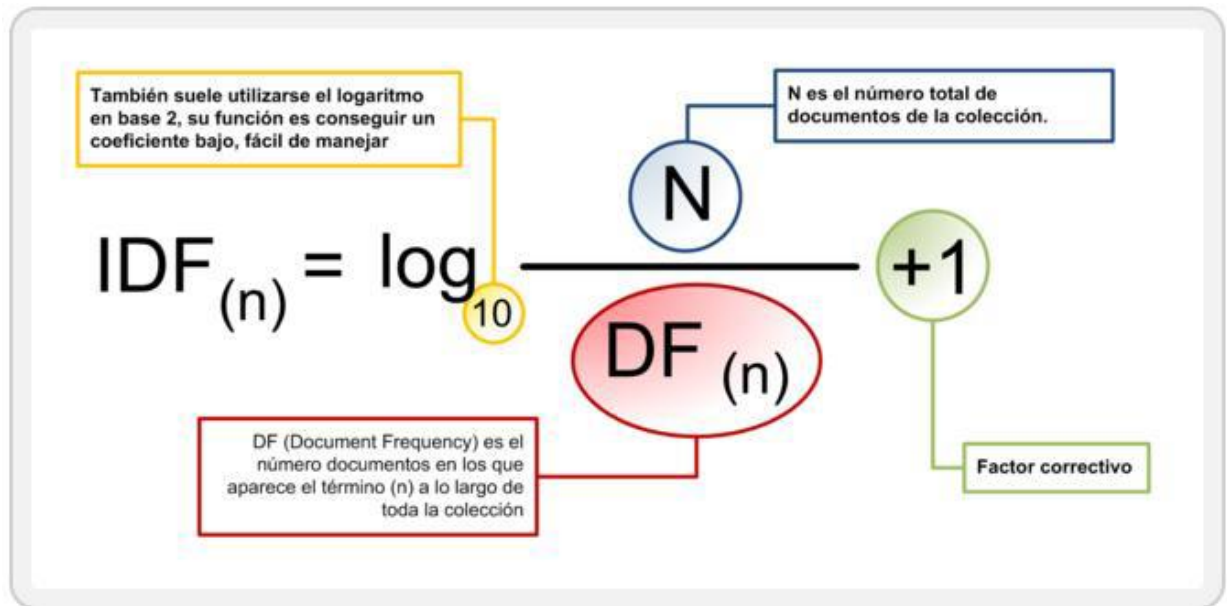


Figura2. Cálculo del IDF de un término

Para entender la notación

$TF-IDF_{i,d} = TF_{i,d} \times IDF_i$, en la fórmula, "i" es la palabra y "d" es el documento en cuestión, por lo que se leería "el *tf-idf* de la palabra *i* en el documento *d* es igual a...

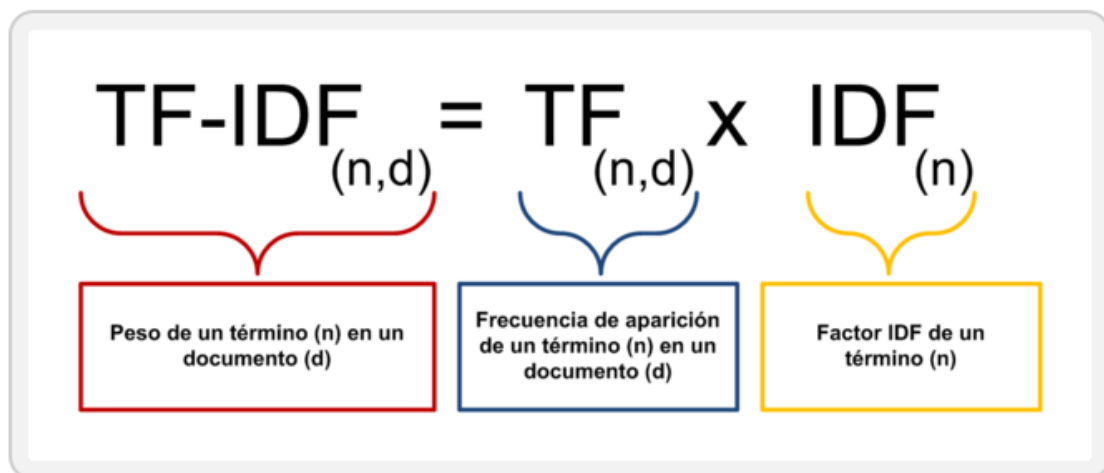


Figura3. Peso TF-IDF para un término en un documento.

<http://dl.acm.org/citation.cfm?id=345650> Stemming and its effects on TF-IDF ranking (poster session)

Evaluación de la Recuperación. Ejemplo para una consulta dada: Supongamos que una consulta devuelve 20 documentos, y que para esa consulta sabemos que existen 8

documentos relevantes en toda la colección documental. Se indica con un pequeño círculo los documentos recuperados que son relevantes)

1. •	5.	9.	13. •	17.
2.	6.	10.	14. •	18.
3. •	7. •	11. •	15.	19. •
4.	8. •	12.	16.	20.

Luego, tras los cálculos se obtiene, Precisión: $8 / 20 = 40\%$, Exhaustividad: $8 / 16 = 50\%$

Para obtener el diagrama de precisión-exhaustividad, se procede a revisar los documentos en el orden en que los devuelve el sistema de IR, para determinar su precisión y exhaustividad.

El **doc. 1** es relevante: su P es 1/1 (100%), su R es 1/16 (6,25%)

El **doc. 2** no es relevante: su P es 1/2 (50%), su R es 1/16 (6,25%)

El **doc. 3** es relevante: su P es 2/3 (66,67%), su R es 2/16 (12,50%)

El **doc. 4** no es relevante: su P es 2/4 (50%), su R es 2/16 (12,50%), etc.

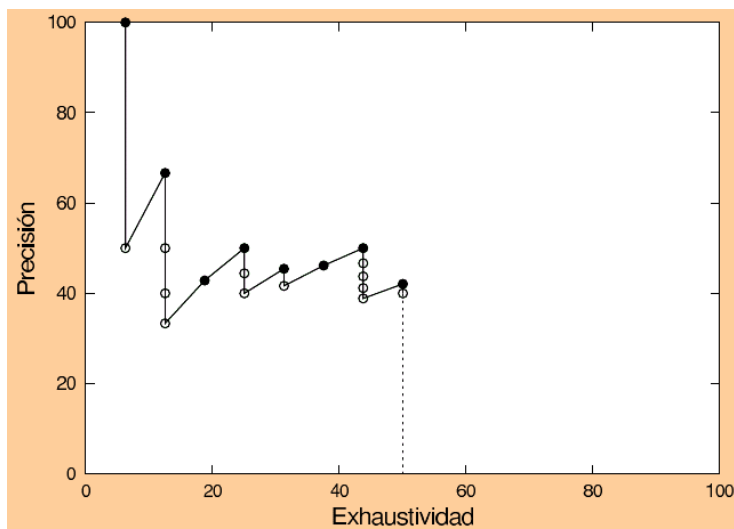
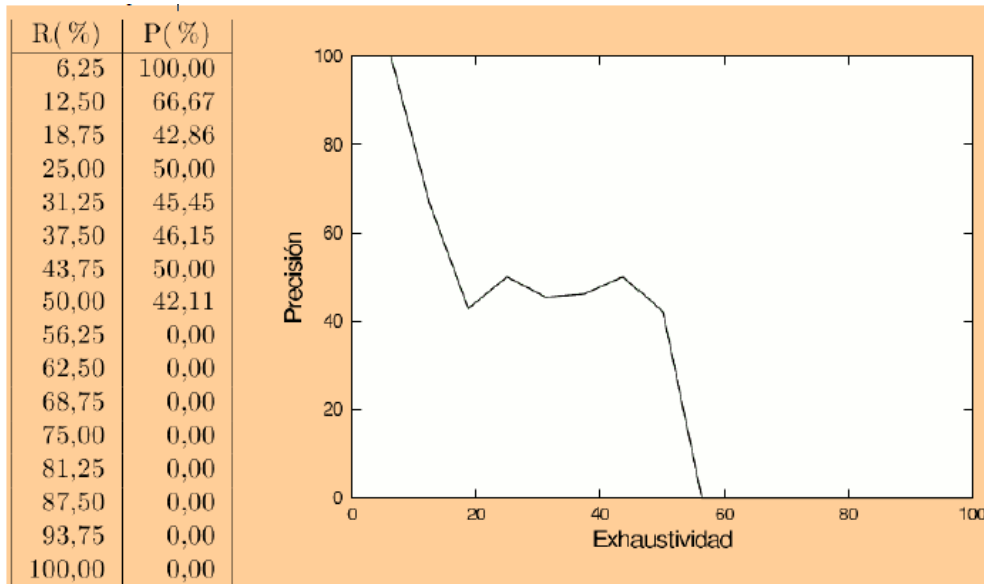


Diagrama (no interpolado) de precisión-exhaustividad, sólo interesan los documentos relevantes recuperados: Por ejemplo, 1, 3, 7, 8, 9 y 11



Programa “[trec_eval](http://trec.nist.gov/trec_eval/)”, creado por Chris Buckley [http://trec.nist.gov/trec_eval/], proporciona unas 85 medidas de evaluación. Las más utilizadas:

- recall-precision curve
 - mean average precision (non interpolated)
 - R-Precision
 - MAP, mean of the R-Precision
 - Average precision at document seen
 - Summary table statistics
 - Average precision histograms
- Necesita dos archivos:
- archivo de los documentos relevantes ([trec_rel_file](#))
 - archivo de documentos recuperados ([trec_top_file](#))

Tabla de clasificación de documentos

		Condición Actual		
		Presentes	Ausentes	
Test resultados	Positivos	tp	fp tipo1	fp tipo 1 error
	Negativos	fn tipo2	tn	fn tipo 2 error
Total # de casos N = tp + fp + fn + tn				present = tp + fn positivos = tp + fp negativos = fn + tn

- Razón Falso positivo $\alpha = \text{fp}/(\text{negativos})$
- Razón Falso negativo $\beta = \text{fn}/(\text{positivos})$

Ejemplo:

Documentos disponibles: D1,D2,D3,D4,D5,D6,D7,D8,D9,D10

Relevantes: D1, D4, D5, D8, D10

Query recuperados por motor de búsqueda: D2, D4, D5, D6, D8, D9

	relevantes	no relevantes
recuperados	D4,D5,D8	D2,D6,D9
no recuperados	D1,D10	D3,D7

	Retrieved	No retrieved	
Relevantes	w=3	x=2	Relevantes = w+x= 5
No relevantes	y=3	z=2	No Relevantes = v+z = 5
	Recuperados = w+y = 6	No Recuperados = x+z = 4	
Total documentos N = w+x+y+z = 10			

- Precision: $P = w / w+y = 3/6 = .5$
- Recall: $R = w / w+x = 3/5 = .6$

Precisión es el porcentaje de los documentos pertinentes en comparación con lo que se devuelve! Solamente los recuperados – high precisión.

Precisión: porcentaje (o fracción) de los éxitos que son relevantes, es decir, el grado en que el conjunto de hits recuperado por una consulta satisface el requisito que generó la consulta.

Recall es el porcentaje de los documentos devueltos en comparación con todo lo que está disponible! Hallar todos los relevantes – high recall.

Recall: porcentaje (o fracción) de los elementos que se encuentran en la consulta, es decir, el grado hasta el cual la consulta encuentra todos los elementos que cumplan con el requisito.

Ejemplo: Para una colección de 10.000 documentos, 50 sobre un tópico específico.

Búsqueda Ideal: Hallar estos 50 documentos y rechazar todos los otros.

Búsqueda Actual : Identifica 25 documentos; 20 son relevantes, pero 5 son de otro tópico.

Precision: $20/25 = 0.8$ (80% de los hits fueron relevantes)

Recall: $20/50 = 0.4$ (40% de los relevantes fueron encontrados)

Precisión y Recall miden los resultados de una única consulta utilizando un sistema de búsqueda específico aplicado a un conjunto específico de documentos.

<http://stackoverflow.com/questions/7170854/precision-recall-in-lucene-java?rq=1>

F-Measure, combina Precision(P) y Recall(R) en un número.

$$F = \frac{2}{1/R + 1/P} = 2 \frac{RP}{R + P}$$

F varía entre [0,1]. F = 1; cuando todos los documentos rankeados son relevantes, F = 0; ningún documento relevante ha sido recuperado.

Conclusiones: Documentos y consultas se representan con términos índice. Término índice: palabra o grupo de palabras que se utiliza para representar un concepto.

Indización: conjunto de términos o procedimientos sintácticos (frases nominales) y convencionales para representar el contenido de un documento, con el fin de permitir su recuperación.

Los sistemas de recuperación basados en términos índice se apoyan en la idea fundamental de que tanto el contenido de los documentos como la necesidad informativa del usuario pueden representarse con términos índice.

Problemas de inconsistencia inevitable entre indizadores (sinonimia, polisemia, etc.), se requieren índices de concordancia y control de autoridades. Dos personas pueden asignar diferentes palabras al mismo concepto, y la misma palabra puede aparecer en documentos que traten temas diferentes: vendo coche usado vs. automóvil de segunda mano.

Sin embargo, lo visto es sólo un ejemplo, ya que el análisis en la representación tiene una serie de procesos internos. Por ejemplo,

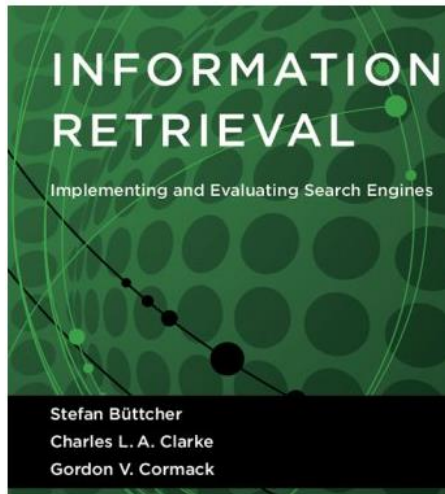
- 1) Análisis del texto para determinar el tratamiento que se realizará sobre números, guiones, signos de puntuación, tratamiento de mayúsculas y/o minúsculas, etc..
 - a) Separación de palabras y 'localización'.
 - b) Carácter espacio, punto, comas,...
 - c) Caracteres de puntuación.
 - d) Tratamiento de acentos. (Importante en otras fases del proceso léxico)
 - e) Tratamiento de números.

- f) Detección de sintagmas y grupos nominales.
 - g) Nombres propios
 - h) Almacenamiento en mayúsculas/minúsculas.
- 2) Eliminación de palabras vacías, muy frecuentes y muy poco frecuentes. Se reduce el número de términos con valores muy pocos significativos para la recuperación. Se pretende:
- a) Reducir el ruido que pueda introducir la indización de todos los términos de un documento. Disminuir el tamaño de la base de datos.
 - b) Palabras vacías. Poseen muy poca capacidad semántica.
 - c) Palabras muy frecuentes. Si un término aparece en casi todos los documentos no sirve para diferenciar unos de otros
 - d) Palabras muy poco frecuentes. Suelen ser errores de tecleado o palabras muy específicas (la probabilidad de que un usuario las solicite es muy baja).
- 3) Aplicación de lematización (stemming) sobre los términos resultantes para eliminar variaciones morfo-sintácticas y obtener lemas. En un diccionario o repertorio léxico, elegir convencionalmente una forma para remitir a ellas todas las que derivan de su misma familia por razones de economía. Desde el punto de vista lingüístico, un lema es un término que representa y unifica todos los elementos de un conjunto de palabras morfológicamente similares. De forma similar, el stemming reduce un conjunto de palabras a su stem o raíz común. Así, camion- sería la raíz de camioneiro, camións, camiós, camiois, etc., y garraf- la de garrafón, garrafa, garrafiña, etc. Palabras que son variaciones morfológicas con un significado prácticamente idéntico. Tratamiento Simple: eliminación de plurales (s-stemmer) o sufijos, existen una serie de algoritmos para stemming, uno de ellos es Porter Stemmer.
- 4) Selección de términos que serán considerados términos índice (sustantivos, nombres propios).
- 5) Utilización de tesauros. Puede ayudar tanto en el proceso de indización como en el de búsqueda de información (expansión de consultas).

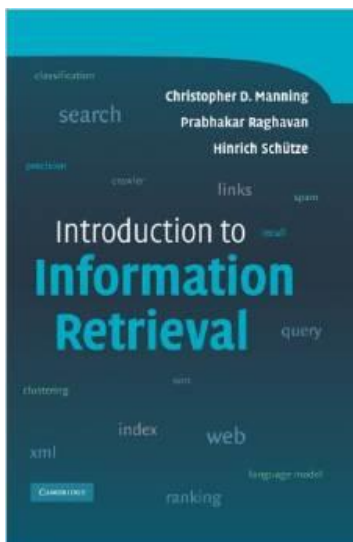
Algunas Referencias:

Information Retrieval: Implementing and Evaluating Search Engines. Stefan Buttcher, Charlie Clarke, Gordon Cormack, MIT Press, 2010.

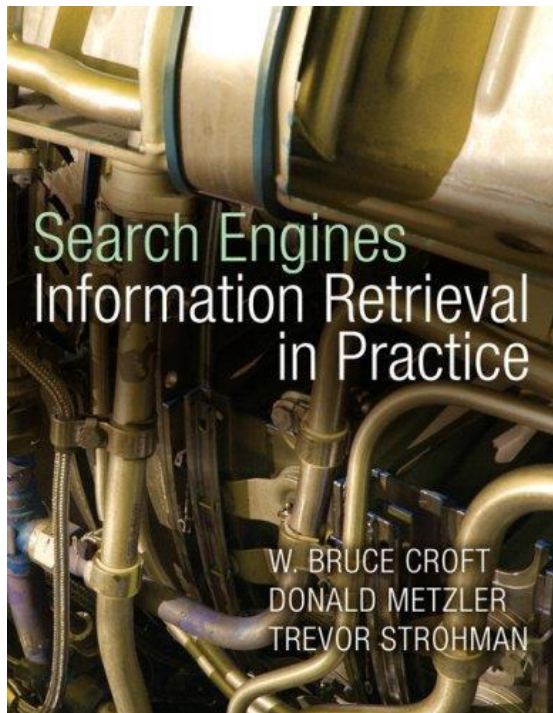
<http://www.ir.uwaterloo.ca/book/>



Introduction to Information Retrieval. Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schuetze, Cambridge University Press, 2007.



Search Engines: Information Retrieval in Practice. Bruce Croft, Donald Metzler, and Trevor Strohman, Pearson Education, 2009.



Information Retrieval and Natural Language Processing Course

<http://nlp.uned.es/~ircourse>

Center for Intelligent Information Retrieval de la Universidad de Massachusstes.

<http://ciir.cs.umass.edu/index.html>

Glasgow IDOM - IR resources.

<http://ir.dcs.gla.ac.uk/resources.html>

Information Retrieval Sites

http://www-nlpir.nist.gov/sites_int.html

Information Retrieval and Natural Language Procesing

<http://web.syr.edu/~diekema/ir.html>