Capítulo 2

Introducción a la Recuperación de Información

2.1. La Recuperación de Información

La *Recuperación de Información* (IR, *Information Retrieval*) es el área de la ciencia y la tecnología que trata de la adquisición, representación, almacenamiento, organización y acceso a elementos de información [26]. Desde un punto de vista práctico, dada una *necesidad de información* del usuario, un proceso de IR produce como salida un conjunto de documentos cuyo contenido satisface potencialmente dicha necesidad[1]. Esta última puntualización es de suma importancia, ya que la función de un sistema de IR no es la de devolver la información deseada por el usuario, sino únicamente la de indicar qué documentos son potencialmente relevantes para dicha necesidad de información [134]. El ejemplo más popular de un sistema de recuperación de información es el de los motores de búsqueda en Internet tales como *Google*[2], *Altavista*[3] o *Yahoo*[4].

2.1.1. Terminología Básica

Antes de continuar, es necesario introducir algunas definiciones de uso común. En Recuperación de Información el término *documento* hace referencia, de forma genérica, a la unidad de texto almacenado por el sistema y disponible para su recuperación. De este modo, dependiendo de la aplicación o de su ámbito de uso, se tratará de artículos de prensa, páginas web, documentos legales, tesis doctorales, etc., bien completos, bien particionados. Podemos, por ejemplo, procesar por separado cada uno de los capítulos de un libro o cada una de las secciones de un documento si consideramos que su longitud total es excesiva. Por su parte, *colección* denota el repositorio de documentos disponible para resolver las necesidades de información del usuario. Cada una de las unidades léxicas (palabras) que componen un documento —y por extensión, la colección— se denomina *término*. Por su parte, la necesidad de información del usuario, expresada en términos que el sistema pueda comprender, se denomina *consulta* (*query*). Asimismo, los resultados obtenidos son, por lo general, ordenados por grado de similaridad o relevancia respecto a la consulta, introduciendo el concepto de *ordenación* (*ranking*) [203].

El concepto mismo de *relevancia* merece particular atención, ya que si bien se habla de la

[1]Nos estamos restringiendo, pues, a la Recuperación de Información textual, ya que existen actualmente nuevos campos de trabajo como la recuperación multimedia sobre imágenes [149], por ejemplo.

[2]`http://www.google.com`

[3]`http://www.altavista.com`

[4]`http://www.yahoo.com`

relevancia del documento respecto a la consulta, en un sentido estricto tal afirmación no es correcta, ya que el usuario juzgará la relevancia del documento devuelto respecto a su necesidad de información original, no respecto a la consulta en la que ésta ha sido reflejada [110]. Se trata, por tanto, de un concepto con un alto componente de subjetividad.

2.1.2. Recuperación de Información y Sistemas de Bases de Datos

Existen dos grandes tipos de sistemas para el procesamiento de elementos de información [26, 131]: los sistemas de Recuperación de Información y los sistemas de Bases de Datos. Mientras los sistemas de Bases de Datos están optimizados para el manejo de datos estructurados con una semántica bien definida, los sistemas de Recuperación de Información, por el contrario, están diseñados para el procesamiento de texto en lenguaje natural, raramente estructurado y, por lo general, de semántica ambigua. En un sistema de Bases de Datos el usuario introduce una consulta específica expresada en álgebra relacional, obteniendo como salida, en forma tabular, todos los resultados que satisfacen dicho requerimiento sin posibilidad alguna de error —ya que invalidaría por completo el resultado. Sin embargo, en el caso de los sistemas de Recuperación de Información los resultados frecuentemente contienen errores, y no tienen por qué ser completos. De hecho, el objetivo de un sistema de Recuperación de Información es maximizar el número de documentos relevantes devueltos a la vez que se minimiza el número de documentos no relevantes devueltos [131].

2.1.3. Tareas de Recuperación de Información

Los buscadores web, si bien los más populares, no son los únicos sistemas de Recuperación de Información actualmente en funcionamiento, ya que existen diferentes tipos de sistemas dependiendo de la naturaleza de la tarea a realizar. Podemos distinguir los siguientes tipos de tareas de Recuperación de Información:

Recuperación ad hoc. Probablemente la tarea más representativa por ser aquélla en la que se basan los buscadores web. En ella el conjunto de documentos sobre el que se realizan las consultas permanece estable[5] mientras que nuevas consultas procedentes de los usuarios llegan al sistema de forma continua. La colección tiene, pues, un carácter *estático*, mientras que son las consultas las que tienen un carácter variable o *dinámico*.

Categorización o clasificación de documentos.
Consiste en la asignación de un documento a una o más clases de documentos fijadas con anterioridad en función de su contenido. Si bien la recuperación ad hoc puede verse también como una tarea de clasificación de documentos en aquéllos relevantes y no relevantes para una consulta dada, en un sentido estricto es necesario diferenciar ambas tareas. Mientras en el caso de la recuperación denominada ad hoc estamos hablando de necesidades de información puntuales y específicas, en el caso de la *categorización* las necesidades —recogidas en los denominados *perfiles* (*profiles*)— permanecen estables en el tiempo, con escasas modificaciones, y su carácter es menos específico que el de las consultas ad hoc, ya que recoge simultáneamente diversas necesidades de información potenciales del usuario [110]. Por esta razón los perfiles tienden a contener un número muy superior de términos que de las consultas, lo que aumenta su complejidad y disuade a los usuarios de introducir modificaciones en los mismos. Las necesidades de información son, por tanto, de naturaleza *estática* en este caso. Dentro de la tarea general de clasificación podemos

[5] En el caso de la web esta condición se relaja, puesto que con el paso del tiempo se crean, eliminan y modifican múltiples páginas web.

destacar dos tareas concretas de particular interés: el *enrutamiento* (*routing*) [98, 42] y el *filtrado* (*filtering*) [137]. En ambos casos los documentos que van llegando al sistema son comparados con los perfiles preexistentes de los usuarios. La diferencia básica entre las dos tareas reside en que mientras el *enrutamiento* introduce una *ordenación* de los resultados devueltos en función de su similaridad respecto al perfil, el *filtrado* es más estricto, ya que se limita a emitir un juicio respecto a la relevancia del documento, aceptándolo o rechazándolo en función del mismo.

***Clustering* de documentos.** Mientras que en el caso de la clasificación de documentos se asume la preexistencia de una serie de clases o grupos de documentos, el objetivo de la tarea de *clustering* es la de generar una serie de clases o *clústers* a partir de un conjunto dado de documentos. Dichas clases o clústers deben atenerse a los principios de maximización de la similaridad intra-clúster y de minimización de la similaridad inter-clúster [121].

Segmentación de documentos. Consiste en la división automática de un documento en subpartes semánticamente coherentes. Es decir, un documento mayor se particiona en secciones que traten subtemas diferentes [102], bien de cara a su procesamiento por separado como si de documentos diferentes se tratase, bien de cara a mostrar al usuario las partes relevantes del documento devuelto.

Si bien nuestro trabajo podría ser aplicado en cualquiera de estas tareas, nuestros experimentos se han limitado a la recuperación ad hoc. Por esta razón, de aquí en adelante, cuando nos refiramos, en general, a *Recuperación de Información*, nos estaremos refiriendo realmente al concepto, más específico, de *recuperación ad hoc*.

2.2. Modelos de Representación Interna

2.2.1. El Paradigma *Bag-of-Terms*

Obtener una representación adecuada de un documento o consulta es una cuestión clave [230]. Por razones históricas, los documentos han sido generalmente representados como conjuntos de términos. Dichos términos, denominados *términos índice* o *palabras clave*, son generados manualmente por especialistas —el caso de las fichas de una biblioteca, por ejemplo—, o bien automáticamente a partir del contenido del documento, como en el caso de los sistemas de Recuperación de Información. Este tipo de representación interna de los documentos, denominada *bag-of-terms* [26], se basa en una interpretación extrema del denominado *principio de composicionalidad*, según el cual la semántica de un documento reside únicamente en los términos que lo forman [121]. En consecuencia, podemos presumir que, si una palabra determinada aparece en un documento, dicho documento trata dicho tema [130]. De forma similar, si una consulta y un documento comparten uno o más términos índice, el documento debe tratar, de algún modo, el tema sobre el que versa la consulta [24].

Este tipo de representación de documentos (y consultas) mediante conjuntos de términos índice resulta insuficiente para una representación completa y adecuada de la semántica del documento. No tiene en cuenta, por ejemplo, las relaciones entre dichos términos —no es lo mismo *"Juan mató a Pedro"* que *"Pedro mató a Juan"*— o la existencia de diferentes sentidos para una misma palabra. A pesar de estas insuficiencias, dicho paradigma ha venido dominando durante décadas el campo de la Recuperación de Información. Las claves de este éxito residen en su sencillez conceptual y de implementación, y al hecho de que su rendimiento fuera bastante satisfactorio en la práctica, a pesar de la pérdida de semántica inherente a su utilización.

2.2.2. Peso Asociado a un Término

Si bien la semántica de un documento se representa mediante un conjunto de términos índice, resulta patente que no todos los términos tendrán la misma importancia a la hora de representar dicha semántica. Esta importancia se representa asignándole a cada uno de dichos términos un valor numérico que denominaremos *peso* (*weight*), de tal forma que a mayor peso, mayor es la importancia del término [131, 26]. Hablaremos, pues, del peso w_{ij} de un término i en un documento j.

Existen dos factores críticos a la hora de calcular el peso de un término: su frecuencia dentro del documento, y su distribución dentro de la colección. Su importancia radica en dos suposiciones:

1. Los términos que aparecen repetidamente en un documento pueden considerarse como representativos válidos de su semántica, por lo que debería asignárseles un peso mayor. Por ejemplo, si en un documento se hace referencia repetidamente al término *chocolatina*, es lógico pensar que dicho documento habla sobre las chocolatinas. Los estudios de Luhn [140] apoyan este punto, al afirmar que el grado de significatividad de un término dentro de un elemento de información es directamente proporcional a su frecuencia dentro de dicho elemento.

2. A mayor número de documentos en los que aparece un término, menor su poder de discriminación, por lo que deberían recibir, consecuentemente, un peso menor. Por ejemplo, si el término *chocolatina* aparece en gran parte de los documentos de la colección, parece lógico pensar que su utilidad es bastante menor que si sólo apareciese en un pequeño subconjunto de ellos.

La primera de estas suposiciones hace referencia a la frecuencia del término i dentro de un documento j, mientras que la segunda de ellas hace referencia a la frecuencia inversa de documento del término i. Definamos, a continuación, formalmente, dichos factores. Sea N el número total de documentos de la colección, y n_i el número de dichos documentos en los que el término t_i aparece. El factor *frecuencia del término i en el documento j* (tf_{ij}) se define como el número de veces que aparece el término t_i en el documento j. Por otra parte, el factor *frecuencia inversa de documento del término i* (idf_i) se calcula como [117]

$$idf_i \;=\; log\,\frac{N}{n_i} \tag{2.1}$$

donde la aplicación de logaritmos pretende suavizar los valores obtenidos para colecciones de gran tamaño.

Durante el cálculo de pesos es también frecuente la introducción de un tercer factor que tenga en cuenta el tamaño del documento [200], ya que a mayor longitud, mayor la probabilidad de que se produzcan correspondencias, por lo que los documentos de mayor longitud se verían en principio favorecidos respecto a los documentos de menor longitud.

Es frecuente también asumir que los términos índice están incorrelados; es decir, asumir la independencia mutua de los pesos de los términos. De esta forma, conocer el peso w_{ij} de un término i en un documento j no nos permitiría afirmar nada respecto al peso w_{kj} de otro término k en ese mismo documento j. Esto es una simplificación para facilitar el cálculo de dichos pesos, ya que dicha afirmación no es en modo alguno cierta. Por ejemplo, dado un documento en el que aparece la palabra *árbitro*, es mucho más probable que aparezca también la palabra *fútbol* que la palabra *chocolatina*. Sin embargo, experimentos llevados a cabo teniendo en cuenta dicha correlación entre términos no han dado lugar a mejoras significativas en los resultados obtenidos, mientras que sí han aumentado notablemente la complejidad del proceso de cálculo de pesos [26].

2.2.3. Modelos de Recuperación

A la hora de diseñar un sistema de Recuperación de Información es preciso establecer previamente cómo representar los documentos y las necesidades de información del usuario, y cómo comparar ambas representaciones . Es preciso pues, definir el *modelo de recuperación* sobre el que ha de desarrollarse el sistema.

En [26] se define formalmente el concepto de *modelo de recuperación* como una cuadrupla [$\mathbf{D}$, $\mathbf{Q}$, $\mathcal{F}$, $R(q_i, d_j)$] donde

1. $\mathbf{D}$ es el conjunto de representaciones de los documentos de la colección.
2. $\mathbf{Q}$ es el conjunto de representaciones de las necesidades de información del usuario, representaciones denominadas *consultas*.
3. $\mathcal{F}$ es el marco formal dentro del cual modelizar las representaciones de documentos, consultas, y las relaciones entre ambos.
4. $R(q_i, d_j)$ es una función de ordenación que asocia un número real a los diferentes pares consulta $q_i \in \mathbf{Q}$ – representación de documento $d_j \in \mathbf{D}$. Dicha ordenación define una relación de orden entre los documentos de la colección respecto a la consulta q_i.

Seguidamente describiremos los modelos clásicos más representativos: los modelos booleano, vectorial y probabilístico.

El Modelo Booleano

Conceptualmente muy simple, el modelo booleano es el más sencillo de los tres aquí descritos, y se basa en la teoría de conjuntos y el álgebra de Boole [26]. En este modelo inicial el usuario especifica en su consulta una expresión booleana formada por una serie de términos ligados mediante operadores booleanos —comúnmente AND, OR y NOT. Dada la expresión lógica de la consulta, el sistema devolverá aquellos documentos que la satisfacen y que conformarán el conjunto de documentos relevantes. De esta forma, el sistema simplemente particiona los documentos de la colección en dos conjuntos, aquéllos que cumplen la condición especificada (relevantes), y aquéllos que no la cumplen (no relevantes), sin ordenación interna alguna, de forma similar a lo que ocurriría con una base de datos tradicional. Un documento es, por tanto, simplemente relevante o no.

Supongamos, por ejemplo, que queremos tener acceso a aquellos documentos que contengan los términos *modelo* y *booleano*, pero que no contengan el término *vectorial*. La consulta asociada a esta necesidad podría ser:

modelo AND *booleano* AND NOT *vectorial*

mientras que el conjunto de documentos relevantes se correspondería gráficamente con el área rayada de la figura 2.1.

La popularidad del modelo booleano, sobre todo en sus inicios, viene dada por su sencillez tanto a nivel conceptual, por la claridad de sus formalismos, como a nivel de implementación. Además, puesto que las consultas son formuladas a modo de expresiones booleanas, de semántica sumamente precisa, el usuario sabe por qué un documento ha sido devuelto por el sistema, lo que no siempre ocurre en otros modelos más complejos. Por otra parte, dado que los documentos son meros *bag-of-terms*, el proceso de recuperación es extremadamente rápido.

Sin embargo, existen también una serie de desventajas importantes asociadas al modelo booleano. La primera de ellas viene dada por la dificultad que conlleva la formalización de la

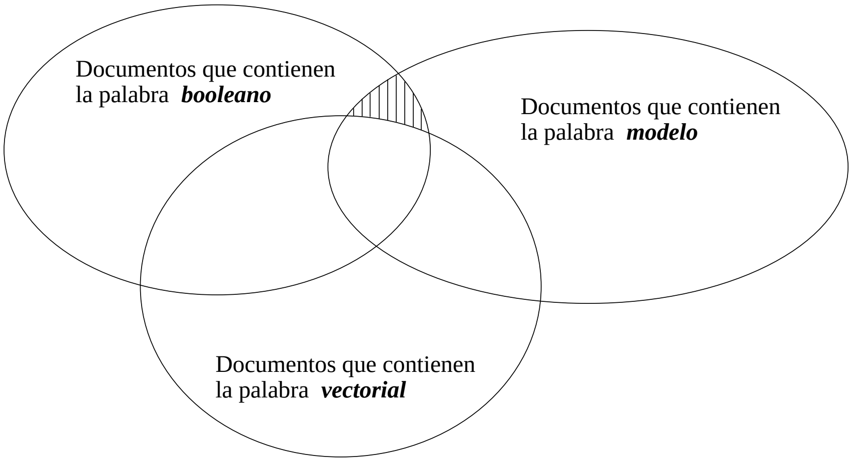


Figura 2.1: Modelo booleano: consulta '*modelo* AND *booleano* AND NOT *vectorial*'

necesidad de información del usuario en forma de expresión booleana, sobre todo cuando se trata de usuarios inexpertos y de necesidades complejas. A ello se suma el hecho de que ligeros cambios en la formulación pueden dar lugar a cambios considerables en el conjunto respuesta. Otro de los grandes inconvenientes del modelo booleano viene dado por su propia naturaleza, de carácter binario. De esta forma, dada una consulta, un documento simplemente es o no relevante dependiendo de si cumple la condición expresada por la consulta. Por lo tanto, no existen ni el concepto de correspondencia parcial ni el concepto de gradación de relevancia. Al no permitir correspondencias parciales, el sistema podría no devolver documentos que, aún siendo relevantes, no verificasen por completo la condición estipulada [266]. Del mismo modo, todos los términos de la consulta tienen la misma importancia, cuando es lógico pensar que la semántica de un texto dado se concentre en mayor grado en ciertos términos —tal y como quedó patente cuando introdujimos el concepto de *peso*. Por otra parte, al no existir ninguna ordenación por relevancia, el usuario se ve obligado a examinar la totalidad del conjunto resultado devuelto.

Si bien bastante popular hace tiempo, por las razones antes comentadas, en la actualidad el modelo booleano se encuentra relegado dentro de los grandes sistemas de Recuperación de Información frente a los restantes modelos a causa de sus desventajas. Sin embargo, continúa empleándose en ciertos ámbitos donde se precisan correspondencias exactas, como en el caso de algunos sistemas de información legislativa [110].

El Modelo Vectorial

Para dar solución a los problemas planteados por el modelo booleano, el modelo vectorial [202, 200] plantea un marco formal diferente en el que se permite tanto la asignación de correspondencias parciales, como la existencia de grados de relevancia en base a los pesos de los términos en consultas y documentos.

En este nuevo modelo, ambos, consultas y documentos, son representados mediante vectores dentro de un espacio multidimensional definido por los propios términos, de tal forma que cada uno de los términos (diferentes) del sistema —es decir, cada uno de los términos del vocabulario— define una dimensión. De este modo, un vocabulario de tamaño t definirá un espacio t-dimensional donde un documento d_j es representado como un vector

$$\overrightarrow{d_j} = (\, w_{1j},\, w_{2j},\, \ldots,\, w_{tj}\,)$$

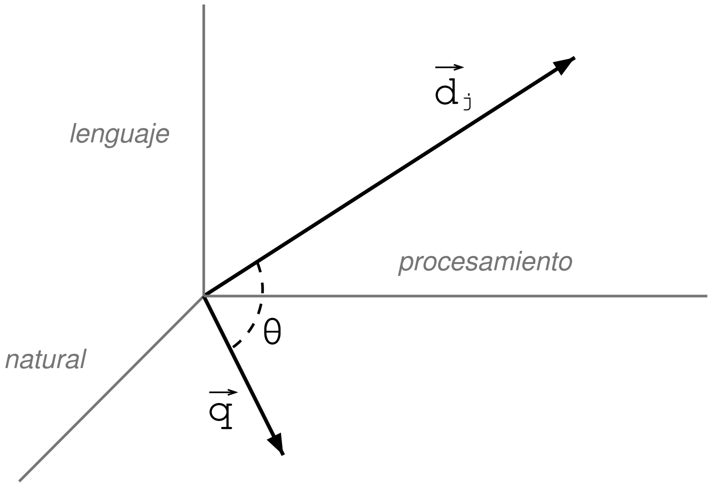


Figura 2.2: Modelo vectorial: espacio tridimensional definido por el vocabulario {*procesamiento*, *lenguaje*, *natural*}

y, paralelamente, una consulta q es representado como un vector

$$\overrightarrow{q} \; = \; (\, w_{1q}, \; w_{2q}, \; \ldots, \; w_{tq} \,)$$

siendo $w_{ij} \geq 0$ y $w_{iq} \geq 0$ los pesos del término t_i —el i-ésimo término del vocabulario— en el documento d_j y la consulta q, respectivamente.

Desde un punto de vista geométrico, si ambos vectores, consulta y documento, están próximos, es factible asumir que el documento es similar a la consulta —en otras palabras, el documento es posiblemente relevante. Por lo tanto, a mayor proximidad entre ambos vectores, mayor relevancia del documento. En concreto, el modelo vectorial plantea medir la similitud entre un documento d_j y una consulta q en base a la proximidad entre sus vectores correspondientes, en lugar de basarse en criterios de inclusión/exclusión como en el caso del modelo booleano. A su vez, dicha proximidad entre vectores es medida en base al coseno del ángulo Θ que tales vectores forman. Llegados a este punto cabe decir que, al asumir que los términos están incorrelados, esto nos permite suponer que las dimensiones son ortogonales, simplificando notablemente los cálculos. La figura 2.2 muestra un ejemplo gráfico para un vocabulario mínimo de tres términos {*procesamiento*, *lenguaje* y *natural*} que define un espacio tridimensional. De esta forma,

$$sim(d_j, q) \; = \; cos\,(\Theta) \; = \; \frac{\overrightarrow{d_j} \; \bullet \; \overrightarrow{q}}{|\overrightarrow{d_j}| \; \times \; |\overrightarrow{q}|} \; = \; \frac{\sum_{i=1}^{t} w_{ij} \; \times \; w_{iq}}{\sqrt{\sum_{i=1}^{t} w^2{}_{ij}} \; \times \; \sqrt{\sum_{i=1}^{t} w^2{}_{iq}}} \tag{2.2}$$

siendo $|\overrightarrow{d_j}|$ y $|\overrightarrow{q}|$ las normas de los vectores documento y consulta, respectivamente.[6]

El modelo vectorial no se limita, pues, a comprobar si los términos especificados en la consulta están o no presentes en el documento, como en el caso del modelo booleano, sino que la similitud entre ambos se calculan en base a los pesos de los términos involucrados, permitiendo de este modo, por un lado, la existencia de correspondencias parciales, y por otro, el cálculo de grados de similaridad o relevancia conforme a los cuales los documentos pueden ser devueltos por orden de mayor a menor relevancia, facilitando notablemente el trabajo del usuario, que puede concentrar

[6]Dado que $|\overrightarrow{q}|$ es constante para una consulta dada, dicho factor puede ser simplificado.

sus esfuerzos en los primeros documentos devueltos —aquellos más relevantes— o incluso definir umbrales de relevancia por debajo de los cuales un documento no es tenido en consideración.

Sin embargo, antes de calcular el grado de similaridad entre los vectores, es necesario calcular los pesos de los términos. Dichos pesos pueden ser calculados de múltiples maneras, si bien el esquema de pesos *tf–idf* y sus derivados [200, 45, 220] se han convertido en los más populares [26]. En el esquema *tf–idf* básico el peso w_{ij} de un término i en un documento j viene dado por la fórmula

$$w_{ij} \; = \; tf_{ij} \, \times \, idf_i \tag{2.3}$$

A la serie de fórmulas para el cómputo de pesos formada por este esquema inicial y las variantes a las que da lugar se las denomina *esquemas tf–idf*.

Los buenos resultados obtenidos con el modelo vectorial, unidos a la simplicidad a nivel de concepto e implementación, su bondad a la hora de aceptar consultas en lenguaje natural, y su capacidad para permitir correspondencias parciales y ordenamiento por relevancia, han hecho de este modelo una de las principales bases sobre la que se han desarrollado gran parte de los experimentos y sistemas en todo el ámbito de la Recuperación de Información [205, 204, 196, 99]. Sus buenas características, unidas al hecho de que sea uno de los modelos de representación más utilizados, le han convertido frecuentemente en el sistema de referencia respecto al cual comparar resultados a la hora de desarrollar nuevos modelos de recuperación [26].

El Modelo Probabilístico

Frente al modelo booleano, basado en teoría de conjuntos, y el modelo vectorial, de carácter algebraico, el modelo probabilístico formaliza el proceso de recuperación en términos de teoría de probabilidades. Las bases del modelo probabilístico fueron establecidas por Robertson y Sparck Jones en [189]. El objetivo perseguido en el modelo es el de calcular la probabilidad de que un documento sea relevante para la consulta dado que dicho documento posee ciertas propiedades [190], propiedades en forma de los términos índice que dicho documento contiene. Según el *principio de orden por probabilidades* [187], el rendimiento óptimo de un sistema se consigue cuando los documentos son ordenados de acuerdo a sus probabilidades de relevancia. En consecuencia, el sistema devolverá los documentos en orden decreciente de las probabilidades de relevancia estimadas mediante el modelo probabilístico.

El modelo parte de las siguientes suposiciones:

1. Todo documento es, bien relevante, bien no relevante para la consulta.
2. El hecho de juzgar un documento dado como relevante o no relevante no aporta información alguna sobre la posible relevancia o no relevancia de otros documentos (suposición de independencia).

En base a ellas, y dada una consulta q, el modelo asigna a cada documento d_j, como medida de similaridad respecto a la consulta, el ratio *P(*d_j *relevante para q)/P(*d_j *no relevante para q)*, medida según la cual los documentos son devueltos, ordenadamente, al usuario.

Sea R, pues, el conjunto de documentos que sabemos (o hemos estimado) relevantes, y sea $\overline{R}$ su complementario (es decir, los no relevantes). Sea $P(R|\overrightarrow{d_j})$ la probabilidad de que el documento d_j sea relevante para la consulta q, y $P(\overline{R}|\overrightarrow{d_j})$ la probabilidad de que el documento d_j sea no relevante para la consulta q. La medida de similaridad $sim(d_j, q)$ del documento d_j respecto a la consulta q se define como el ratio

$$sim(d_j, q) \; = \; \frac{P(R|\overrightarrow{d_j})}{P(\overline{R}|\overrightarrow{d_j})} \tag{2.4}$$

que podemos descomponer, aplicando Bayes, en

$$sim(d_j, q) \;=\; \frac{P(\overrightarrow{d_j}|R) \;\times\; P(R)}{P(\overrightarrow{d_j}|\overline{R}) \;\times\; P(\overline{R})} \tag{2.5}$$

donde $P(\overrightarrow{d_j}|R)$ representa la probabilidad de seleccionar aleatoriamente el documento d_j de entre el conjunto de relevantes R y $P(R)$ representa la probabilidad de que un documento de la colección sea relevante. Sus análogos y complementarios vienen dados por $P(\overrightarrow{d_j}|\overline{R})$ y $P(\overline{R})$.

Puesto que $P(R)$ y $P(\overline{R})$ son constantes para todos los documentos de la colección, pueden ser simplificados, obteniendo

$$sim(d_j, q) \;\sim\; \frac{P(\overrightarrow{d_j}|R)}{P(\overrightarrow{d_j}|\overline{R})} \tag{2.6}$$

Asumiendo la independencia entre los términos índice se obtiene

$$sim(d_j, q) \;\sim\; \frac{\left(\prod_{g_i(\overrightarrow{d_j})=1} P(t_i|R)\right) \;\times\; \left(\prod_{g_i(\overrightarrow{d_j})=0} P(\overline{t_i}|R)\right)}{\left(\prod_{g_i(\overrightarrow{d_j})=1} P(t_i|\overline{R})\right) \;\times\; \left(\prod_{g_i(\overrightarrow{d_j})=0} P(\overline{t_i}|\overline{R})\right)} \tag{2.7}$$

donde g_i es una función que devuelve el peso asociado al término k_i dentro de un vector t-dimensional —es decir, $g_i(\overrightarrow{d_j}) = w_{ij}$, con $w_{ij} \in \{0,1\}$—, $P(t_i|R)$ representa la probabilidad de que un documento seleccionado aleatoriamente de R contenga el término índice t_i, mientras que $P(\overline{t_i}|R)$ representa la probabilidad de que un documento seleccionado aleatoriamente de R no contenga dicho término índice t_i. Las probabilidades asociadas con el conjunto $\overline{R}$ tienen significados análogos.

Tras aplicar logaritmos y eliminar algunos factores constantes dentro de una misma consulta, y sabiendo que $P(t_i|R) + P(\overline{t_i}|R) = 1$, obtenemos

$$sim(d_j, q) \;\sim\; \sum_{i=1}^{t} w_{iq} \;\times\; w_{ij} \;\times\; \left(log\,\frac{P(t_i|R)}{1 - P(t_i|R)} \;+\; log\,\frac{1 - P(t_i|\overline{R})}{P(t_i|\overline{R})} \right) \tag{2.8}$$

donde t es el número de términos que componen el vocabulario del sistema y los pesos de los términos son binarios, $w_{ij} \in \{0,1\}$ y $w_{iq} \in \{0,1\}$), indicando meramente la aparición o no del término en el documento o consulta, respectivamente.

Dado que inicialmente el conjunto R no es conocido, se hace necesario estimar las probabilidades $P(t_i|R)$ y $P(t_i|\overline{R})$. De este modo, sea V un subconjunto de los documentos inicialmente devueltos, y que es considerado relevante[7], y sea V_i el subconjunto de V cuyos documentos contienen el término t_i. Aproximaremos $P(t_i|R)$ mediante la distribución del término t_i en V:

$$P(t_i|R) \;=\; \frac{|V_i|}{|V|} \tag{2.9}$$

donde $|V|$ y $|V_i|$ representan el número de elementos en los conjuntos V y V_i, respectivamente. De forma similar, y suponiendo que el resto de los documentos son no relevantes, aproximaremos $P(t_i|\overline{R})$ mediante:

$$P(t_i|\overline{R}) \;=\; \frac{n_i - |V_i|}{N - |V|} \tag{2.10}$$

[7] Bien tras haberlos examinado el usuario, bien seleccionados automáticamente —p.ej., los r primeros documentos devueltos.

donde N es el tamaño de la colección de documentos y n_i el número de documentos de la colección que contienen el término t_i. Para evitar problemas con valores pequeños de $|V|$ y $|V_i|$, comunes en la práctica, se introduce un factor de ajuste, obteniendo finalmente:

$$P(t_i|R) = \frac{|V_i| + 0{,}5}{|V| + 1} \tag{2.11}$$

$$P(t_i|\overline{R}) = \frac{n_i - |V_i| + 0{,}5}{N - |V| + 1} \tag{2.12}$$

Substituyendo dichas estimaciones en la expresión 2.8 obtenemos finalmente, tras operar:

$$sim(d_j, q) \sim \sum_{i=1}^{t} w_{iq} \times w_{ij} \times w^{(1)} \tag{2.13}$$

donde $w^{(1)}$ es el denominado *peso Robertson-Sparck Jones* [189], de importancia clave en los esquemas de peso probabilísticos, y que se define como:

$$w^{(1)} = log \frac{(|V_i| + 0{,}5)/(|V| - |V_i| + 0{,}5)}{(n_i - |V_i| + 0{,}5)/(N - n_i - |V| + |V_i| + 0{,}5)} \tag{2.14}$$

Múltiples medidas de similaridad basadas en esta expresión inicial han venido siendo empleadas en diversos sistemas, siendo uno de los más conocidos el sistema *Okapi* [194, 191, 192], cuyo esquema de pesos BM25[193] se encuentra entre los más efectivos y, junto al vectorial *tf–idf*, es punto de referencia para el desarrollo y evaluación de nuevos modelos y nuevos esquemas de pesos. Empleando dicho esquema, el Okapi BM25, la medida de similaridad entre un documento d_j y una consulta q se calcula como:

$$sim(d_j, q) = \sum_{i=1}^{t} w^{(1)} \times \frac{(k_1 + 1) \times tf_{ij}}{K + tf_{ij}} \times \frac{(k_3 + 1) \times tf_{iq}}{k_3 + tf_{iq}} \tag{2.15}$$

donde t es el número de términos que componen el vocabulario
$w^{(1)}$ es el peso Robertson-Sparck Jones definido en la fórmula 2.14
K es calculado como $K = k_1 \times ((1 - b) + b \times dl_j/avdl)$
k_1, b y k_3 son parámetros constantes cuyo valor viene dado en función de la naturaleza de la consulta y de la colección
tf_{ij} es la frecuencia del término t_i en el documento d_j
tf_{iq} es la frecuencia del término t_i en la consulta q
dl_j es la longitud del documento d_j
$avdl$ es la longitud media de los documentos de la colección

2.3. Normalización e Indexación de Documentos

2.3.1. Generación de Términos Índice: Normalización

El proceso de generación de los términos asociados a un documento o consulta se lleva a cabo mediante una serie de transformaciones sucesivas sobre el texto de entrada denominadas genéricamente *operaciones de texto* [26]. Esta serie de transformaciones persigue, además, la reducción del texto a algún tipo de forma canónica que facilite el establecimiento de correspondencias durante el posterior proceso de búsqueda. A este proceso se le denomina *normalización* (*conflation*) [114].

En las aproximaciones clásicas estas operaciones incluyen el análisis léxico del texto, la eliminación de las denominadas *stopwords*, la eliminación de mayúsculas y signos ortográficos, el *stemming* de los términos resultantes y, finalmente, la selección de los componentes de los documentos que serán utilizados por el sistema de recuperación, y que denominaremos *términos índice* para distinguirlos de los términos del documento.

Análisis Léxico del Texto

El proceso de análisis léxico, o *tokenización*, es aquél consistente en la conversión de una secuencia de caracteres —el texto de los documentos o consultas—, en una secuencia de palabras candidatas a ser adoptadas como términos índice por el sistema. Por lo tanto, el objetivo principal de esta primera fase es el de la identificación de las palabras que conforman el texto.

Para ello se considerarán habitualmente tres tipos de caracteres [131]: caracteres de palabra, caracteres interpalabra, y caracteres especiales. Una palabra estará formada por una secuencia de caracteres de palabra delimitada por símbolos interpalabra. Ejemplos de símbolos de palabra serían las letras y los números, mientras que espacios en blanco, comas y puntos son frecuentemente adoptados como símbolos interpalabra. El tercer grupo de símbolos, los símbolos denominados especiales, estarían formados por caracteres que requerirían un procesamiento especial. El guión, por ejemplo, podría ser considerado un carácter especial ya que puede ser empleado de diferentes formas: a final de línea indicaría la continuación de la palabra en la siguiente línea, en otras ocasiones conecta palabras independientes —p.ej., en la expresión inglesa *single-word term* (término unipalabra)—, etc. De esta forma, cuando el sistema detectase un guión —o cualquier otro carácter especial—, debería aplicar una serie de reglas que le permitiesen identificar el caso particular del que se trata y actuar en consecuencia.

Dependiendo del dominio de aplicación y de las características del idioma, la composición de los diferentes tipos de caracteres y su tratamiento variará. De todos modos, el nivel de complejidad de los *tokenizadores* suele ser escaso, siendo herramientas sencillas y rápidas.

Eliminación de *Stopwords*

Denominamos *stopwords* [131, 26] a aquellas palabras de escasa utilidad debido a que su excesiva frecuencia anula su capacidad discriminante —p.ej., formas verbales de *ser* o *estar*— o a que su contenido semántico es escaso —p.ej., artículos y preposiciones. Dada su poca utilidad, dichas palabras son desechadas, lo que permite a su vez un considerable ahorro de recursos, ya que si bien estas palabras representan una parte ínfima del vocabulario —algunas decenas de palabras diferentes—, suponen, en cambio, una cantidad muy importante del número de términos a procesar[8], lo que permite reducir considerablemente el espacio de almacenamiento de las estructuras generadas. En experimentos citados por [266] dicho ahorro supuso en torno a un 25 %, mientras en [26] se habla de un 40 % o incluso más.

Asimismo, en el caso del texto de las consultas es conveniente eliminar también el contenido de su *metanivel* [163, 162], es decir, aquellas expresiones correspondientes a la formulación de la consulta y de las preferencias del usuario acerca de la misma y que no aportan información alguna a la búsqueda, introduciendo únicamente ruido. Tal es el caso, por ejemplo, de *"encuentre*

[8]Lo que concuerda con lo estipulado por la ley de Zipf [272], según la cual dado un corpus suficientemente grande de un idioma dado, si contamos el número de veces que aparece cada palabra, y listamos a continuación dichas palabras por orden de frecuencia, su posición p en dicha lista y su frecuencia f guardan una relación constante

$$f \times r = k$$

de tal forma que, por ejemplo, la palabra en el puesto 50 de dicho ranking es 3 veces más frecuente que aquélla que ocupa el puesto 150.

los documentos que describan ... ". Para ello se debería emplear, en el caso de las consultas, una segunda lista de *stopwords* a mayores de la general, y que podemos denominar de *metastopwords*.

Eliminación de Mayúsculas y Signos Ortográficos

El paso de mayúsculas a minúsculas de los términos procesados es habitual en los sistemas de Recuperación de Información, ya que así se puede evitar la falta de correspondencia con términos que, por ejemplo, estén en mayúsculas únicamente por estar al principio de la oración [131]. Es también frecuente que signos ortográficos tales como tildes o diéresis sean eliminados.

Stemming

Una característica del lenguaje humano es la de que un mismo concepto puede ser formulado de maneras diferentes, que denominaremos *variantes* [111]. Esto supone que a la hora de la comparación de documentos y consultas nos podemos encontrar con que aún refiriéndose a conceptos equivalentes —al menos desde el punto de vista de un sistema de Recuperación de Información [105]—, puedan no producirse correspondencias debido a que ambos estén empleando términos diferentes. Este sería el caso, por ejemplo, de las formas verbales *cocinar* y *cocinaré*, y de las formas nominales *cocina* y *cocinas*. Para minimizar en lo posible el impacto de estos fenómenos, los sistemas de Recuperación de Información recurren a técnicas de *stemming* implementadas mediante herramientas denominadas *stemmers*.

El *stemming* consiste en la reducción de una palabra a su *stem* o supuesta raíz[9] mediante la eliminación de sus terminaciones, siendo ésta la partícula que presumiblemente contiene la semántica básica del concepto [26, 131]. De esta forma, por ejemplo, los términos anteriores se verían reducidos a la cadena *cocin-*, permitiendo de este modo las correspondencias entre los mismos. Si bien el objetivo principal del *stemming* es el de reducir las diferentes formas lingüísticas de una palabra a una forma común o *stem*, y así facilitar el acceso a la información durante el posterior proceso de búsqueda, paralelamente se está reduciendo el número de términos diferentes del sistema, lo que permite a su vez una segunda reducción de los recursos de almacenamiento requeridos.

Entre los algoritmos clásicos de *stemming* destacan el algoritmo de Porter [179] —el algoritmo de *stemming* por excelencia y el más popular—, y el algoritmo de Lovins [139]. En ambos casos podemos diferenciar dos fases: una fase de eliminación de sufijos en base a una lista prefijada de los mismos, y una fase de recodificación de la cadena resultante de acuerdo a una serie de reglas. Asimismo se aplican una serie de restricciones respecto a la longitud del *stem* resultante. Algunos *stemmer* más avanzados, denominados *dictionary look-up stemmers* [131], comprueban además la validez del *stem* obtenido contrastándolo contra un diccionario de *stems* antes de aceptarlo.

Tomemos como ejemplo el caso de la normalización del término inglés *derivational* (derivacional) mediante el algoritmo de Porter. En un primer paso, *derivational* es transformado en *derivate* mediante la aplicación de la regla[10]

$$(m > 0)\ \textit{-ational} \rightarrow \textit{-ate}$$

siendo m la longitud del término en número de secuencias vocal-consonante. De esta forma, *derivational* es primero transformado en *deriv-* en un paso intermedio eliminando la terminación *-ational*, y dado que su longitud ($m = 2$) verifica la condición estipulada ($m > 0$), la transformación es válida, por lo que dicha transformación se completa concatenando al *stem* intermedio el sufijo *-ate*, obteniendo así el término *derivate*. En un segundo paso, *derivate* es

[9]Debemos precisar que el *stem* no tiene por qué coincidir en absoluto con la raíz de la palabra.

[10]Estas reglas se componen frecuentemente de una condición y un par de sufijos, uno a eliminar y otro a concatenar, que son aplicados en caso de que se verifique la condición.

finalmente transformado en *deriv-* mediante la aplicación de la regla

$$(m > 1)\ \textit{-ate} \rightarrow \varepsilon$$

donde ε es la cadena vacía. De esta forma, tanto el verbo *to derivate* (derivar) como su adjetivo derivado *derivational* (derivacional), son normalizados ambos a una forma común *deriv-*, lo que permite establecer correspondencias entre ambos.

Sin embargo, la aplicación de mecanismos de *stemming* no está libre de errores [114]. Pueden producirse errores de *sobre-stemming*, donde dos palabras con escasa o nula relación semántica son reducidas a un *stem* común, dando lugar a correspondencias erróneas. Este es el caso, por ejemplo, de *general* (general) y *generous* (generoso), normalizadas ambas como *gener-*. El caso contrario es el de los errores por *infra-stemming*, donde dos términos muy próximos semánticamente son reducidos a *stems* diferentes, impidiendo el establecimiento de correspondencias. Este es el caso, por ejemplo, de *recognize* (reconocer), normalizado como *recogn-*, y *recognition* (reconocimiento), normalizado como *recognit-*.

Selección de Términos Índice

Finalmente, los términos resultantes de las transformaciones de texto previas son adoptados como términos índice, asociándoseles, de ser requerido, el *peso* correspondiente. En la actualidad la mayoría de los sistemas emplean una representación *a texto completo* del texto [26], entendiendo como tal que todos los términos índice generados son empleados para la representación del texto durante el posterior proceso de búsqueda. Cabe citar, sin embargo, que existe también la posibilidad de seleccionar, bien manualmente —por especialistas—, bien automáticamente, un subconjunto de los mismos y que éstos sean los únicos términos índice adoptados para la representación del texto procesado.

2.3.2. Generación de Índices: Indexación

Existen dos alternativas a la hora de realizar una búsqueda dentro de una colección. La primera de ellas es la de realizar una búsqueda secuencial, como es el caso de las herramientas `grep` de Unix o de algunos sistemas de Recuperación de Información como el `seft` [64]. Este tipo de búsquedas secuenciales o *en línea*, resultan apropiadas cuando se trata de colecciones pequeñas, del orden de Megabytes, o cuando ésta sea la única posibilidad debido a la falta de recursos de almacenamiento o a la alta volatilidad de la colección —es decir, que ésta esté siendo modificada amplia y continuamente. La segunda posibilidad, y la más extendida, consiste en la generación de estructuras de datos auxiliares, denominadas *índices*, que permitan acelerar la búsqueda. El sistema de Recuperación estaría formado entonces por dos componentes: las estructuras índice, que contienen las estructuras auxiliares asociadas a la colección, y el *motor de indexación*, el componente software encargado de su generación, manejo, mantenimiento, e interrogación. El empleo de índices es la mejor opción en el caso de colecciones de gran tamaño y de carácter estático o semiestático, entendiendo como tal que dichas estructuras puedan ser actualizadas a intervalos regulares —diario o semanal, por ejemplo— sin perjuicio para el comportamiento del sistema. Este es el caso, por ejemplo, de bases de datos documentales, hemerotecas electrónicas, o la propia web.

Un *índice invertido* o *fichero invertido*, es una estructura de datos de uso frecuente en sistemas de Bases de Datos y Recuperación de Información y cuyo objetivo es el de acelerar los procesos de búsqueda [131, 26]. En dicha estructura podemos diferenciar dos elementos: el vocabulario o diccionario y sus apariciones o *postings*. El *vocabulario* o *diccionario* es, como cabe esperar, el vocabulario del sistema, el conjunto de todos los términos índice diferentes en la colección. Para cada uno de ellos se almacena una lista de sus apariciones (*postings*) dentro de la colección, generalmente a nivel de documento. En otras palabras, el índice consiste en una lista de los términos índice existentes en la colección y, para cada uno de ellos, una lista de los documentos en los cuales aparece dicho término. Para posibilitar el proceso de cálculo de pesos, se almacenan una serie de datos asociados bien a cada entrada del vocabulario —su frecuencia de documento, por ejemplo—, bien a cada una de las apariciones del término en la colección —su frecuencia dentro de dicho documento, por ejemplo, o directamente su peso.

Asimismo es también frecuente la existencia de un tercer componente, un *fichero de documentos* [131] consistente en una lista de los documentos almacenados por el sistema, y para cada uno de ellos, datos como su longitud, la frecuencia máxima de sus términos, etc.

En lo que respecta a la implementación concreta de la estructura de datos, ésta admite diversas posibilidades: mediante tablas de dispersión, árboles-B, etc.

El proceso de generación del índice se denomina *indexación*[11]. En la figura 2.3 mostramos un ejemplo de dicho proceso. Partimos de tres documentos, denominados DOC_A, DOC_B y DOC_C. El documento DOC_A contiene, entre otros, los términos *management* y *derivational*, que se convertirán en *manag-* y *deriv-*, respectivamente, mediante las correspondientes operaciones de texto. El documento DOC_B contiene los términos, *derivate*, *derivational* y *manages*, que serán normalizados como *deriv-*, *deriv-*, y *manag-*, respectivamente. Finalmente, DOC_C contiene el término *management*, normalizable a *manag-* como se ha indicado.

Una vez obtenidos los términos índice asociados a cada documento, procedemos a su indexación, generándose el índice cuyas estructuras podemos ver en la parte inferior de la figura. El fichero de documentos es, en este caso, bastante simple, ya que únicamente almacena un identificador interno (DID), y la longitud del documento (LONG)—habiendo supuesto longitudes de 200, 414 y 70 términos. Por su parte, el diccionario contiene, para cada término de la colección, su frecuencia de documento (DF), es decir, el número de documentos en los que aparece. Finalmente, para cada uno de dichos documentos existe una entrada o *posting* en el fichero de apariciones con la información asociada a dicho término en dicho documento, en este caso la frecuencia en el documento (TF).

2.4. El Proceso de Búsqueda

Como ya hemos comentado, el objetivo primario de un sistema de Recuperación de Información es el de transformar una *necesidad de información* del usuario en una lista de documentos de la colección cuyo contenido cubra dicha necesidad. Para ello el primer paso consiste en que el usuario plasme su necesidad de información en una *consulta* aceptada por el sistema. Por su parte, el sistema transformará dicha consulta en una representación interna que permita su comparación con los documentos indexados de acuerdo con el modelo de recuperación empleado. Dado que esta comparación se basa por lo general en la aparición de los términos índice de la consulta en el documento, el sistema deberá aplicar sobre la consulta formulada por el usuario las mismas operaciones de texto aplicadas en el caso de los documentos, para así obtener una representación compatible que permita dicha comparación.

La *consulta* supone, pues, un intento por parte del usuario de especificar las condiciones que permitan acotar dentro de la colección aquel subconjunto de documentos que contienen la información que desea. Por lo tanto, el sistema parte de la consulta formulada por el usuario, no de la necesidad de información original, por lo que una formulación incorrecta o insuficiente no podrá guiar adecuadamente al sistema durante el proceso de búsqueda. A este respecto los mayores problemas a los que ha de hacer frente el sistema de IR son, por una parte, la escasa habilidad del usuario a la hora de formular su necesidad en forma de consulta y, por otra, que a

[11] Es también frecuente en la literatura denominar *indexación* al proceso completo formado por los procesos de generación de términos índice y de generación del índice propiamente dicho.

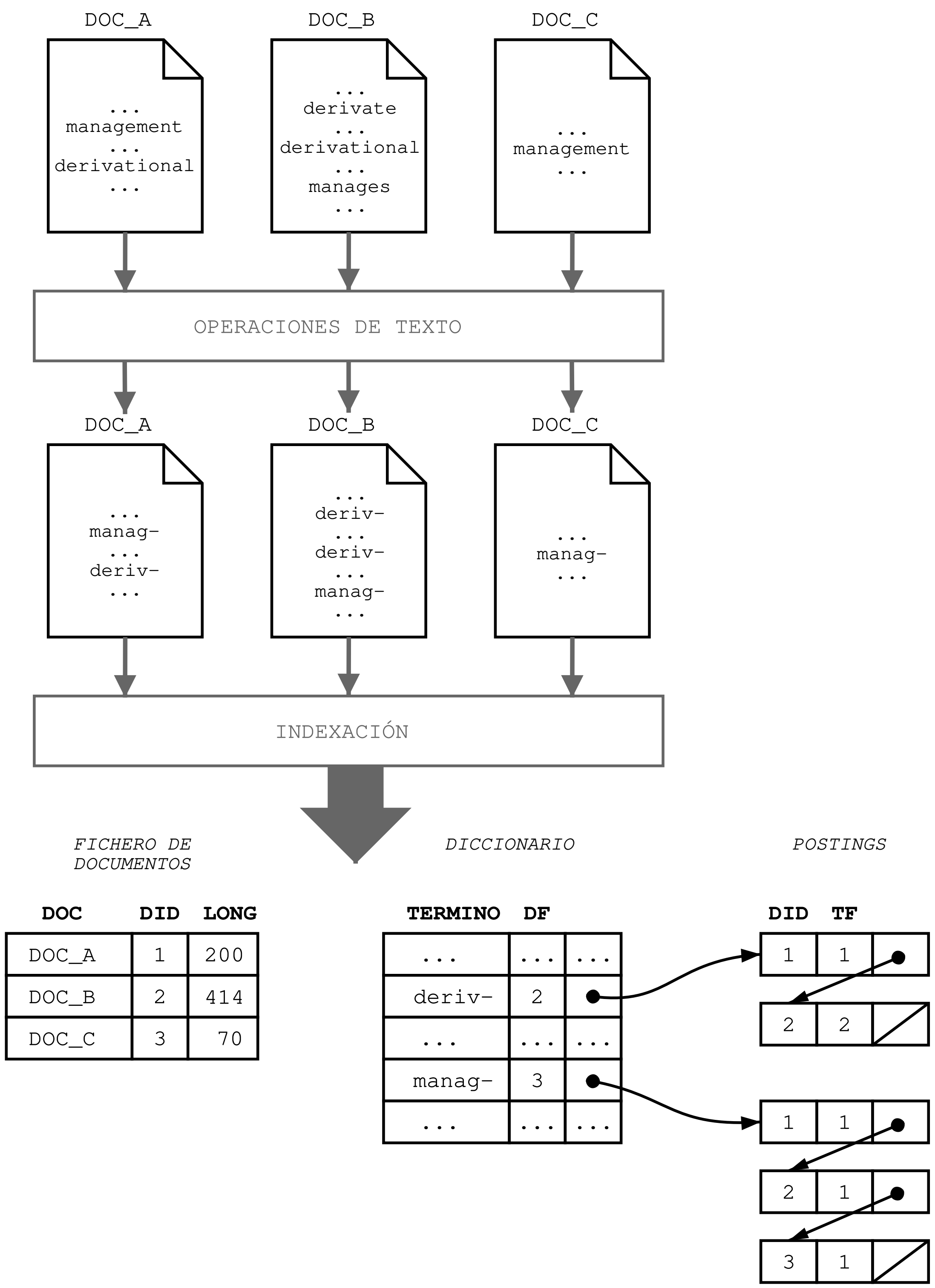


Figura 2.3: Generación de un índice

la hora de describir un mismo concepto los términos empleados por el usuario y los autores de los documentos suelen diferir, impidiendo el establecimiento de correspondencias [268].

Para tratar de paliar esta situación los sistemas de Recuperación de Información suelen incluir mecanismos de expansión de la consulta que permitan la reformulación de la consulta inicial para aumentar su efectividad.

2.4.1. Expansión de Consultas

Bajo la expresión *expansión de consultas* (*query expansion*) se engloban una serie de procesos automáticos o semiautomáticos que permiten la reformulación o refinamiento de la consulta inicial con objeto de aumentar la efectividad del proceso de recuperación, generalmente mediante la adición de nuevos términos, bien relacionados con los términos inicialmente introducidos por el usuario, bien asociados a documentos que se saben o suponen relevantes.

El empleo de técnicas de expansión de consultas permite, por lo general, una mejora de los resultados obtenidos, si bien también se incurre en el riesgo de introducir incorrectamente términos que no guarden relación alguna con el objetivo de la búsqueda y que, por tanto, dañen el rendimiento del sistema [110].

Entre las técnicas de expansión de consultas destacan dos: aquéllas basadas en tesauros y aquéllas basadas en realimentación.

Expansión de Consultas Mediante Tesauros

Un *tesauro* (*thesaurus*) es una base de datos lexicográfica que almacena una representación jerárquica de un lexicón de acuerdo a las relaciones semánticas existentes entre sus palabras [148]. Dependiendo de su ámbito de aplicación, los tesauros pueden ser de carácter general [157, 263] o específico [8]. En lo que respecta a su construcción, ésta puede ser llevada a cabo manualmente por especialistas [157, 263], o bien automáticamente a partir de un corpus empleando técnicas estadísticas [91]. Mención aparte merecen sistemas híbridos como el desarrollado por Fernández Lanza [71], en el cual a partir de un diccionario común impreso de sinónimos y antónimos [35] se genera automáticamente un diccionario electrónico con medidas ponderadas de sinonimia y antonimia.

El empleo de tesauros nos permite reformular la consulta inicial lanzada contra el sistema bien de forma manual [131] —navegando por la estructura jerárquica del tesauro y eligiendo los términos a utilizar—, bien automáticamente [261, 241].

Uno de las herramientas más utilizadas en este tipo de expansión es WordNet [158, 156, 97, 70, 33], una base de datos lexicográfica del inglés en la que sustantivos, verbos, adjetivos y adverbios se organizan en *synsets*, conjuntos de sinónimos de tal modo que cada *synset* está asociado a un sentido determinado. WordNet establece relaciones semánticas de diferente signo:

- **Sinonimia:** dos palabras serán consideradas sinónimas si tienen algún menos un sentido en común; p.ej., *pipe* (tubería) y *tube* (tubería). Como se ha precisado, es la relación básica en torno a la cual se articula WordNet en base a la noción de *synset*.

- **Antonimia:** entre palabras de significados contrarios; p.ej., *wet* (mojado) y *dry* (seco).

- **Hiperonimia e hiponimia:** relación *es-un*. El término más general es el hiperónimo —p.ej., *flower* (flor)—, y el más específico el hipónimo —p.ej., *rose* (rosa).

- **Meronimia y holonimia:** establece una jerarquía *parte-de*. El conjunto es el holónimo —p.ej., *fleet* (flota)—, y la parte es el merónimo —p.ej., *ship* (barco).

Asimismo existe también una base de datos paralela para lenguas europeas denominada EuroWordNet [263].

Por lo general, las aproximaciones actuales basadas en tesauros no logran mejorar los resultados obtenidos. Esto se debe fundamentalmente a los problemas asociados a la introducción de ruido durante el proceso de expansión [261]. Dado que una misma palabra puede tener asociados diferentes sentidos, antes de expandir una palabra sería necesaria una *desambiguación del sentido de las palabras* (WSD, *Word-Sense Disambiguation*) [226, 68] para así expandir únicamente con los términos relacionados semánticamente con ese sentido de la palabra.[12]

Expansión de Consultas Mediante Realimentación

Los métodos de expansión basados en realimentación por relevancia (*relevance feedback*) son, actualmente, el método de reformulación de consultas más extendido debido a su buen comportamiento general [26].

Las técnicas de realimentación se basan en el empleo de la información acerca de la relevancia, o no relevancia, de un subconjunto de los documentos devueltos mediante la consulta inicial para:

- Expandir la consulta en sentido estricto, añadiendo nuevos términos pertenecientes a los documentos considerados relevantes.
- Modificar los pesos de los términos de la consulta buscando optimizar su rendimiento.

La información acerca de la relevancia o no de los documentos devueltos inicialmente se puede obtener mediante interacción con el usuario, o de forma automática tomando como relevantes los n primeros documentos devueltos inicialmente. En este último caso, denominado *expansión por pseudo-relevancia* o *expansión ciega* [131], existe el riesgo de tomar como relevantes documentos que no lo sean, dañando así el rendimiento del sistema. Los resultados obtenidos, sin embargo, son por lo general bastante satisfactorios [110].

La expansión mediante realimentación presenta, además, una serie de ventajas:

- Aisla al usuario de los detalles del proceso de reformulación, debiendo simplemente indicar su criterio de relevancia respecto a los documentos devueltos inicialmente (en el caso de que no se trate de una expansión ciega).
- Permite dividir el proceso completo de búsqueda en una secuencia de pasos más pequeños y fáciles de manejar.

Este tipo de expansión fue puesto en práctica inicialmente en el contexto del modelo vectorial [196], aunque posteriormente su aplicación se extendería al modelo probabilístico [189].

La realimentación por relevancia aplicada al modelo vectorial parte de la suposición de que los documentos relevantes son similares entre sí y de que, por su parte, aquellos documentos no relevantes son disimilares respecto a los sí relevantes. De esta forma, habiendo identificado una serie de documentos relevantes y no relevantes entre los documentos devueltos por la consulta inicial, la idea consiste en reformular el vector consulta de modo que se aproxime al centroide de los documentos relevantes identificados (realimentación *positiva*) y se aleje además del centroide de los no relevantes identificados (realimentación *negativa*). Formalmente, partiendo de un vector consulta

$$\overrightarrow{q} = (w_{1q}, w_{2q}, \ldots, w_{tq})$$

[12] En un sentido estricto, sin embargo, para que el proceso fuese completo sería preciso también un proceso de desambiguación similar en los documentos, lo que conllevaría unos costes inasumibles.

el proceso de realimentación da lugar a un nuevo vector

$$\overrightarrow{q}' \;=\; (\, w'_{1q},\, w'_{2q},\, \ldots,\, w'_{tq},\, w'_{(t+1)q},\, \ldots,\, w'_{(t+k)q}\,)$$

donde los pesos iniciales w_{ij} han sido actualizados a unos nuevos pesos w'_{ij} y donde se han introducido k nuevos términos. El cálculo del nuevo vector $\overrightarrow{q}'$ se lleva a cabo mediante alguna de las expresiones propuestas por Rocchio [196] e Ide [108], bastante simples y similares entre sí:

$$\textit{Rocchio}: \quad \overrightarrow{q}' \;=\; \alpha\overrightarrow{q} \;+\; \beta\sum_{j=1}^{n_1}\frac{r_j}{n_1} \;-\; \gamma\sum_{j=1}^{n_2}\frac{s_j}{n_2} \tag{2.16}$$

$$\textit{Ide normal}: \quad \overrightarrow{q}' \;=\; \alpha\overrightarrow{q} \;+\; \beta\sum_{j=1}^{n_1} r_j \;-\; \gamma\sum_{j=1}^{n_2} s_j \tag{2.17}$$

$$\textit{Ide dec-hi}: \quad \overrightarrow{q}' \;=\; \alpha\overrightarrow{q} \;+\; \beta\sum_{j=1}^{n_1} r_j \;-\; \gamma\, s_1 \tag{2.18}$$

donde $\overrightarrow{q}'$ es el nuevo vector de la consulta
$\overrightarrow{q}$ es el vector de la consulta inicial
r_j es el vector del j-ésimo documento relevante
s_j es el vector del j-ésimo documento no relevante
n_1 es el número de documentos relevantes examinados
n_2 es el número de documentos no relevantes examinados
y α, β y γ son, respectivamente, los parámetros que controlan las contribuciones relativas de la consulta original, los documentos relevantes, y los documentos no relevantes.

Es frecuente desechar el componente de realimentación negativa fijando el factor γ a cero, ya que si bien dicho componente aleja el vector consulta de los vectores documento no relevantes, eso no quiere decir que lo aproxime más a los vectores documento relevantes, que en última instancia es lo que se pretende [131].

En lo que respecta a la realimentación dentro del modelo probabilístico, ésta presenta unas características diferentes al caso vectorial. Mientras en el modelo vectorial se realizaba simultáneamente una expansión y una modificación de los pesos, en el caso del modelo probabilístico se trata de dos pasos claramente diferenciados.

La modificación de pesos por realimentación es contemplada inherentemente por el propio modelo, ya que el componente $w^{(1)}$ del *peso Robertson-Sparck Jones* (fórmula 2.14) integra la información acerca de la relevancia de los documentos devueltos.

En lo que respecta a la fase de expansión propiamente dicha, ésta fue introducida posteriormente. Se trata de aproximaciones donde los términos procedentes de los documentos relevantes son ordenados en base a algún tipo de función, expandiendo luego la consulta con los términos mejor posicionados. Existen diversas funciones de ordenación al respecto [99], entre la que podemos citar la propuesta de Robertson [188], que emplea el propio peso del término t ponderado por su distribución en el conjunto de documentos relevantes. De esta forma, la puntuación p_i asignada a un término t_i vendría dada por:

$$p_i \;=\; |V_i| \;\times\; w^{(1)} \tag{2.19}$$

donde $|V_i|$ era el número de documentos relevantes que contenían el término t_i y $w^{(1)}$ es el peso Robertson-Sparck Jones definido en la fórmula 2.14.

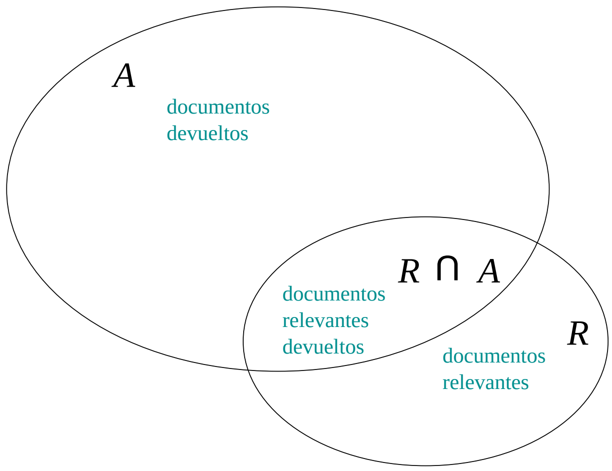


Figura 2.4: Documentos relevantes y documentos devueltos

2.5. Evaluación

A la hora de evaluar un sistema de Recuperación de Información existen múltiples aspectos a tener en cuenta [26]: su *eficiencia* referida a sus costes espacio-temporales asociados; su *efectividad* a la hora de devolver el mayor número de documentos relevantes, minimizando a la vez el número de no relevantes devueltos [235]; el *esfuerzo* realizado por el usuario a la hora de formular o modificar su consulta; y la *amigabilidad* del interfaz de presentación de resultados en relación al esfuerzo requerido por el usuario para su interpretación.

En este trabajo nos centraremos en la evaluación de la efectividad del sistema, diferenciando dos aspectos: por una parte las medidas de evaluación empleadas y, por otra, las colecciones de referencia necesarias para dicha evaluación. Discutiremos estos puntos en los apartados siguientes, así como la metodología de evaluación aplicada en nuestros experimentos y el motor de indexación empleado en los mismos.

2.5.1. Medidas de Evaluación

Las medidas de evaluación aquí descritas pueden calcularse bien como medidas puntuales relativas a una consulta concreta, bien como medidas globales relativas a un conjunto de consultas. En este último caso las medidas son calculadas promediando los valores obtenidos para cada consulta individual respecto al número de consultas empleado, salvo en el caso de la precisión media de documento, que será descrito en detalle más adelante.

Precisión y Cobertura

Las medidas básicas de evaluación en un sistema de Recuperación de Información son la *precisión* (*precision*), que mide la capacidad del sistema para recuperar sólo documentos relevantes, y la *cobertura* (*recall*), que mide la capacidad del sistema para recuperar todos los documentos que son relevantes.

Tal como se aprecia en la figura 2.4, y dada una consulta q, sea R, de tamaño $|R|$, el conjunto de documentos relevantes para dicha consulta; sea A, de tamaño $|A|$, el conjunto de documentos devueltos por el sistema; y sea $R \cap A$, de tamaño $|R \cap A|$, el conjunto de documentos relevantes devueltos por el sistema. Definimos formalmente las medidas de *precisión* y *cobertura* como:

$$Precisión = \frac{|R \cap A|}{|A|} \tag{2.20}$$

$$Cobertura = \frac{|R \cap A|}{|R|} \tag{2.21}$$

Precisión a los 11 Niveles Estándar de Cobertura

Tanto la precisión como la cobertura evalúan la calidad del conjunto de documentos devuelto como tal, como un conjunto, sin tener en cuenta el orden en que han sido devueltos los documentos, y requiriendo además que dicho conjunto respuesta haya sido examinado en su totalidad.

Sin embargo, al usuario se le presentan los documentos devueltos por el sistema de forma ordenada de acuerdo con su relevancia o similaridad respecto a la consulta, de mayor a menor grado de relevancia. Por lo tanto, los valores de precisión y cobertura irán variando conforme el usuario examina, ordenadamente, los documentos devueltos. De esta forma, podemos calcular la precisión para el conjunto de documentos examinados cuando se hayan alcanzado determinados valores de cobertura, es decir, cuando se haya recuperado ya un determinado porcentaje de documentos relevantes. Generalmente se muestran los valores de precisión obtenidos al nivel 0.0 de cobertura (correspondiente al 0 % de cobertura), al nivel 0.1 (correspondiente al 10 %), al 0.2 (al 20 %), y así sucesivamente en incrementos de 0.1 (10 %) hasta alcanzar el nivel 1 (100 %). A dichos niveles se les denomina los *11 niveles estándar de cobertura*.

Dado que no siempre es posible calcular la precisión a un nivel de cobertura concreto, las precisiones son interpoladas. Sea r_j el j-ésimo nivel de cobertura estándard —p.ej., r_5 denota el nivel de 50 % de cobertura—, entonces la precisión interpolada $Pr(r_j)$ a dicho nivel de cobertura se calcula como

$$Pr(r_j) = max_{\, r_j \leq r \leq r_{j+1}} \; Pr(r)$$

es decir, la precisión interpolada al j-ésimo nivel de cobertura estándar es la precisión máxima conocida en cualquier valor de cobertura entre dicho nivel j-ésimo y el nivel $(j+1)$-ésimo.

A modo de ejemplo, supongamos una colección de 20 documentos, donde únicamente 4 son relevantes para una consulta dada. Supongamos que nuestro sistema, ante dicha consulta, ha devuelto dichos documentos en las posiciones 1ª, 2ª, 4ª y 15ª. Al examinar los documentos devueltos, los valores de precisión y cobertura para cada documento relevante encontrado son:

- **1º documento relevante:** precisión 1, cobertura 0.25 (1 documento relevante, 1 documento recuperado)
- **2º documento relevante:** precisión 1, cobertura 0.50 (2 documento relevantes, 2 documentos recuperados)
- **3º documento relevante:** precisión 0.75, cobertura 0.75 (3 documento relevantes, 4 documentos recuperados)
- **4º documento relevante:** precisión 0.27, cobertura 1 (4 documento relevantes, 15 documentos recuperados)

Por lo tanto, de acuerdo a la regla de interpolación establecida, la precisión a los niveles de cobertura del 0 al 5 es 1, la precisión para los niveles 6 y 7 es 0.75, y la precisión para los niveles 8, 9 y 10 es 0.27.

Precisión a los n Documentos Devueltos

Otra posibilidad, similar a la anterior, de cara a comparar dos conjuntos resultado ordenados, es mostrar la precisión obtenida no a determinados valores de cobertura, sino a un número determinado de documentos devueltos.

Precisión Media no Interpolada

Se trata de una medida que refleja el rendimiento del sistema sobre el conjunto de documentos relevantes, pero no sólo considerando el porcentaje de documentos relevantes recuperados, sino también el orden en que éstos han sido devueltos. Este valor es calculado como la media de las precisiones obtenidas después de que cada documento relevante es recuperado respecto al número de documentos relevantes existentes para esa consulta. De esta forma se premia a los sistemas que devuelven los documentos relevantes en posiciones superiores.

Retomando el ejemplo mostrado para la precisión a los 11 niveles estándar de cobertura, su precisión media no interpolada sería

$$\textit{Pr. no int.} \; = \; \frac{1 + 1 + 0{,}75 + 0{,}27}{4} \; = \; 0{,}76$$

Precisión Media de Documento

Similar a la precisión media no interpolada, en cuanto a que también se calculan, como paso intermedio, las precisiones obtenidas después de que cada documento relevante es recuperado.

Como ya hemos comentado, la precisión media no interpolada para una consulta determinada se calcula como la media de dichas precisiones respecto al número de documentos relevantes existentes para la consulta. Posteriormente, la precisión media no interpolada global se calculará, como es usual, como la media de las precisiones medias no interpoladas de cada consulta respecto al número de consultas.

Sin embargo, la precisión media de documento no tiene sentido a nivel de una única consulta determinada, sino que se calcula a nivel global para un conjunto de consultas. Para ello se promedian las sumas de todas las precisiones obtenidas, para todas las consultas empleadas, después de que cada documento relevante es recuperado. Asimismo esta media se hace respecto al número global de documentos relevantes existentes para todas las consultas, en lugar de respecto al número de consultas.

Supongamos que, además de la consulta que venimos usando como ejemplo en la precisión a los 11 niveles estándar de cobertura y en la precisión media no interpolada, tenemos otra consulta que cuenta con 2 documentos relevantes que son devueltos en posiciones 2ª y 4ª. Sus precisiones para cada documento devuelto serían:

- **1º documento relevante:** precisión 0.5 (1 documento relevante, 2 documentos recuperados)

- **2º documento relevante:** precisión 0.5 (2 documento relevantes, 4 documentos recuperados)

y la precisión media de documento se calcularía, a nivel global del conjunto de las dos consultas, como:

$$\textit{Pr. doc.} \; = \; \frac{\big(1 + 1 + 0{,}75 + 0{,}27\big) \; + \; \big(0{,}5 + 0{,}5\big)}{4 \; + \; 2} \; = \; 0{,}67$$

Precisión-R (R-*Precision*)

Es la precisión obtenida a los R documentos devueltos, donde R es el número de documentos relevantes para esa consulta.

Retomando de nuevo el ejemplo para la precisión a los 11 niveles estándar de cobertura, y dado que existían 4 documentos relevantes para la consulta, la precisión-R será la precisión a los 4 documentos devueltos, que era de 0.75.

2.5.2. Colecciones de Referencia

Hasta hace poco uno de los mayores problemas en el ámbito de la Recuperación de Información era la falta de colecciones de evaluación con suficiente entidad y de libre acceso, para que de este modo permitiesen una evaluación de los sistemas lo más completa posible y que dichos resultados fuesen comparables.

Una colección de evaluación está compuesta por tres elementos: los documentos, las consultas, y una lista de los documentos de la colección que son relevantes para cada consulta.

La elección de una colección adecuada es de la suma importancia a la hora de evaluar nuestro sistema, ya que únicamente así tendremos la convicción de que los resultados obtenidos son fiables y representativos. La calidad de una colección viene dada por diversos aspectos:

- *Su disponibilidad para la comunidad científica.* El libre acceso a una colección promueve su utilización por otros investigadores, facilitando la comparación de resultados.

- *El tamaño de la colección.* Cuanto mayor sea el repositorio de documentos y el número de consultas a utilizar, más se ajustarán los resultados obtenidos al comportamiento real del sistema [105].

- *La calidad de las consultas.* Dicha calidad depende de su variedad, de la diversidad de construcciones empleadas, y de si dichas consultas se corresponden o no a necesidades de información realísticas.

- *La calidad de los documentos.* Viene dada por la variedad de los mismos y por su realismo en cuanto a que no hayan sido sometidos a ningún tipo de tratamiento especial.

Colecciones como *Cranfield* [118], o CACM e ISI [76], por ejemplo, se mostraban inadecuadas a las necesidades para una evaluación fiable del sistema debido a su pequeño tamaño (número de documentos: 1400, 3204 y 1460, respectivamente; número de consultas: 22, 52, 35+41). En años recientes esta situación ha cambiado gracias al trabajo de la *Text REtrieval Conference (TREC)* y del *Cross-Language Evaluation Forum (CLEF)*.

Text REtrieval Conference (TREC)

La situación inicial de falta de colecciones de evaluación estándar dio un giro tras la celebración, en 1992, del primer *Text REtrieval Conference (TREC)* [1], congreso de carácter anual organizado por el *National Institute of Standards and Technology (NIST)* y la *Information Technology Office* de la *Defense Advanced Research Projects Agency (DARPA)*. El objetivo perseguido por la organización del TREC es el de apoyar la investigación en el campo de la Recuperación de Información. Para ello:

- Facilita la infraestructura, herramientas y metodologías necesarias para la evaluación a gran escala de sistemas de Recuperación de Información.

- Promueve la comunicación entre industria, gobiernos e investigadores, facilitando de este modo la transferencia tecnológica.

Además de la recuperación *ad hoc*, TREC da cabida a otros campos de aplicación dentro de sus diferentes secciones, denominadas *tracks*, tales como enrutamiento (*routing*), filtrado (*filtering*), recuperación sobre transcripciones de documentos hablados, etc. Algunas de estas secciones varían según la edición.

Las colecciones de documentos empleadas en TREC son del orden de decenas, o incluso centenares, de miles de documentos. En su mayor parte se trata de artículos periodísticos procedentes de periódicos o agencias de noticias. Tal es el caso, por ejemplo, de la colección de *Los Angeles Times*, formada por artículos publicados en dicho periódico durante el año 1994 y que suponen 131896 artículos (475 MB) con una longitud media de 527 palabras por documento.

Junto con las colecciones de documentos, TREC suministra una serie de requerimientos o necesidades de información, denominados *topics*, en torno a 50 por edición. El proceso de convertir dichos *topics* en *consultas* efectivas debe ser llevado a cabo por el propio sistema. Los participantes deben enviar a la organización, dentro de un plazo dado, los resultados devueltos por sus sistemas para dichos *topics*.

El mayor problema surge a la hora de identificar los documentos relevantes para cada *topic* debido al gran número de documentos y consultas existentes. Para esta tarea se emplea una técnica denominada *pooling* [260], consistente en que, para cada *topic*, se toman los K primeros documentos devueltos para ese *topic* por cada uno de los sistemas participantes —generalmente los K=100 primeros. Dichos documentos son luego revisados por especialistas, que son quienes establecen la relevancia o no de cada uno de ellos. En lo que se refiere al concepto de *relevancia*, la organización de TREC optó por el siguiente criterio:[13]

> *"Si estuviera redactando un informe sobre el tema del* topic *en cuestión y pudiese usar para dicho informe la información contenida en el documento examinado, entonces dicho documento será considerado relevante"*

Cross-Language Evaluation Forum (CLEF)

A pesar de su inestimable contribución, las diferentes ediciones de TREC se han centrado en el inglés, salvo contadas excepciones, por lo que la investigación en sistemas de Recuperación de Información en otros idiomas seguía encontrándose con los mismos problemas. Tal era el caso del español, ya que las colecciones empleadas en su sección dedicada al español de las ediciones TREC-4 y TREC-5, y formada por artículos de noticias escritos en español (de México), no están ya disponibles.

Inicialmente en nuestras investigaciones se empleó una colección de evaluación propia [21], creada recopilando artículos periodísticos en español peninsular que cubrían el año 2000. El tamaño final de la colección era de 21.899 documentos, con una longitud media de 447 palabras. Por otra parte, el conjunto de consultas empleado era limitado, 14 consultas, con una longitud media de 7.85 palabras, 4.36 de ellas palabras con contenido.

El conjunto de documentos relevantes para cada consulta fue creado siguiendo la filosofía de *pooling* del TREC. Para cada una de las diferentes técnicas de normalización empleadas, y para cada uno de los diferentes motores de indexación empleados, se tomaron los 100 primeros documentos devueltos. Dichos documentos fueron examinados manualmente para juzgar su relevancia.

[13] Véase `http://trec.nist.gov/data/reljudge_eng.html`

```
<DOC>
<DOCNO>EFE19940101-00002</DOCNO>
<DOCID>EFE19940101-00002</DOCID>
<DATE>19940101</DATE>
<TIME>00.34</TIME>
<SCATE>VAR</SCATE>
<FICHEROS>94F.JPG</FICHEROS>
<DESTINO>ICX MUN EXG</DESTINO>
<CATEGORY>VARIOS</CATEGORY>
<CLAVE>DP2404</CLAVE>
<NUM>100</NUM>
<PRIORIDAD>U</PRIORIDAD>
<TITLE>   IBM-WATSON
           FALLECIO HIJO FUNDADOR EMPRESA DE COMPUTADORAS
</TITLE>
<TEXT>    Nueva York, 31 dic (EFE).- Thomas Watson junior, hijo del fundador
 de International Business Machines Corp. (IBM), falleció hoy,
 viernes, en un hospital del estado de Connecticut a los 79 años de
 edad, informó un portavoz de la empresa.
    Watson falleció en el hospital Greenwich a consecuencia de
 complicaciones tras sufrir un ataque cardíaco, añadió la fuente.
    El difunto heredó de su padre una empresa dedicada principalmente
 a la fabricación de máquinas de escribir y la transformó en una
 compañía líder e innovadora en el mercado de las computadoras. EFE
    PD/FMR
    01/01/00-34/94
</TEXT>
</DOC>
```

Figura 2.5: Documento de ejemplo: documento EFE19940101-00002

Sin embargo, aunque dicha colección servía a los propósitos de evaluación deseados, no era lo suficientemente amplia, sobre todo en lo que respecta al conjunto de consultas empleado. Además, al tratarse de una colección propia, no permitía la comparación de resultados con otros investigadores.

La incorporación en la segunda edición del *Cross-Language Evaluation Forum (CLEF-2001)* [2] de una colección para el español peninsular, cambió esta situación. CLEF es un congreso de corte similar al TREC, pero dedicado a las lenguas europeas, tanto en tareas monolingües como multilingües. Si en su primera edición, en el año 2000, los idiomas disponibles eran inglés, francés, alemán e italiano, las colecciones disponibles se han ido ampliando a español, portugués, finlandés, ruso, búlgaro y húngaro.

2.5.3. Metodología de Evaluación Empleada

En los experimentos recogidos en este trabajo se han empleado las colecciones de evaluación para el español de las ediciones del 2001 [173], 2002 [174] y 2003 [175] del CLEF.

Respecto a las colecciones de documentos, en CLEF 2001 y 2002 se empleó el mismo corpus, denominado EFE 1994 y formado por teletipos de la agencia española de noticias EFE[14] correspondientes al año 1994. Los documentos se encuentran formateados en SGML, tal como se aprecia en la figura 2.5, donde recogemos uno de ellos a modo de ejemplo. Este corpus inicial

[14] http://www.efe.es

colección	*tamaño* (MB)	*#docs.*	*long.*
EFE 1994	509	215738	317.64
EFE 1995	577	238307	325.33
EFE 1994+1995	1086	454045	321.67

Tabla 2.1: Colecciones de evaluación: composición de los corpus de documentos

```
<top>
<num> C044 </num>
<ES-title> Indurain gana el Tour </ES-title>
<ES-desc> Reacciones al cuarto Tour de Francia ganado por Miguel Indurain.
</ES-desc>
<ES-narr> Los documentos relevantes comentan las reacciones a la cuarta
victoria consecutiva de Miguel Indurain en el Tour de Francia. Los
documentos que discuten la relevancia de Indurain en el ciclismo mundial
después de esta victoria también son relevantes. </ES-narr>
</top>
```

Figura 2.6: *Topic* de ejemplo: *topic* número 44

fue ampliado en el CLEF 2003 con una segunda colección, denominada EFE 1995, formada por teletipos de EFE del año 1995. La composición de ambos corpus —tamaño de la colección, número de documentos, y longitud media—, juntos y por separado, se recoge en la tabla 2.1.[15]

En lo que respecta a los *topics* empleados, fueron 50 en 2001 (números 41 a 90), de nuevo 50 en 2002 (91 a 140), y finalmente 60 en 2003 (141 a 200). Los *topics* están formados, tal como se aprecia en la figura 2.6, por tres campos: *título* (*title*), un breve título como su nombre indica; *descripción* (*description*), una somera frase de descripción; y *narrativa* (*narrative*), un pequeño texto especificando los criterios que utilizarán los revisores para establecer la relevancia de un documento respecto a la consulta.

Sin embargo, los experimentos recogidos en este trabajo no fueron realizados directamente con este conjunto inicial de *topics*. En primer lugar se eliminaron aquéllos con un número de documentos relevantes menor de 6. La razón para ello estriba en que, cuando el número de documentos relevantes es muy pequeño, un cambio en la posición de uno o dos documentos devueltos puede acarrear cambios muy marcados en los resultados obtenidos para dicha consulta, distorsionando los resultados globales obtenidos para el conjunto total [106].

Dado el elevado número de consultas disponibles —100 para el corpus de documentos del CLEF 2001 y 2002, y 60 para CLEF 2003—, decidimos crear 3 corpus:

- **CLEF 2001-02·A:** corpus de entrenamiento y estimación de parámetros [42]. Está formado por los *topics* impares de CLEF 2001 y 2002. En caso de no ser necesaria tal fase de entrenamiento y estimación se emplearía igualmente a modo de corpus de evaluación. Se optó por combinar los *topics* de ambas ediciones para homogeneizar en lo posible las colecciones empleadas, en previsión de que hubiese diferencias importantes entre las consultas de ambas ediciones[16]. La elección de las consultas correspondientes al corpus EFE 1994 para crear un corpus de entrenamiento, en lugar del corpus ampliado de

[15] Debemos llamar la atención sobre el hecho de que, al tratarse de teletipos, de estilo poco cuidado y escritos con escasa atención, éstos contienen numerosos errores ortográficos [74].

[16] Debe tenerse en cuenta a este respecto que la edición 2001 era la primera en la que se empleaba el español, por lo que el equipo encargado de dicho idioma gozaba de poca experiencia al respecto.

corpus	*colec. docs.*	*#topics*	*long.*	
			cortas	*largas*
CLEF 2001-02·A	EFE 1994	46	19.28	55.23
CLEF 2001-02·B	EFE 1994	45	20.24	60.22
CLEF 2003	EFE 1994+1995	47	21.31	59.14

Tabla 2.2: Colecciones de evaluación: composición final de los corpus

CLEF 2003, viene dada por el menor número de consultas disponibles para éste último, 60 —47 tras eliminar las de insuficientes documentos relevantes—, lo que hace más difícil la división del mismo en un corpus de entrenamiento y un corpus de evaluación. Dicha afirmación se basa en los resultados obtenidos por Voorhees [262], en base a los cuales podemos considerar que 25 consultas es el número de consultas mínimo necesario para eliminar las posibles perturbaciones debidos a errores en la emisión de juicios de relevancia, lo que no permitiría una división adecuada de las consultas del corpus CLEF 2003.

- **CLEF 2001-02·B:** corpus de evaluación. Formado por los *topics* pares de las ediciones del 2001 y 2002.

- **CLEF 2003:** corpus de evaluación. Formado por los *topics* de la edición del 2003.

Asimismo se emplearon dos tipos de consultas durante la evaluación, las denominadas consultas *cortas*, generadas a partir de los campos *título* y *descripción*, y las denominadas consultas *largas*, que emplean la totalidad de los campos del *topic*. De esta forma podemos comprobar el comportamiento del sistema ante ambos tipos de consultas, siendo las cortas más próximas a las empleadas en sistemas comerciales [105][17]. Por otra parte, en el caso de las consultas largas, se ha primado la información aportada por el campo *título*, doblando su relevancia respecto a los otros dos campos, ya que es el campo que concentra la semántica básica de la consulta.

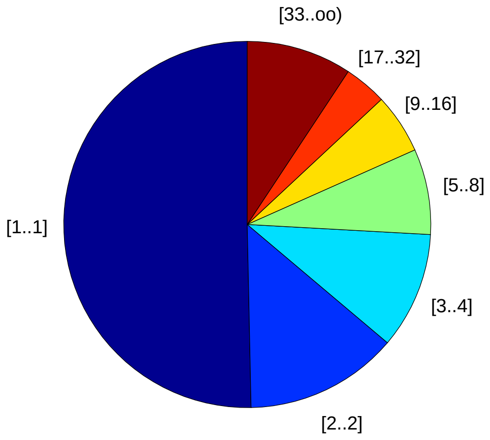
Las estadísticas de los corpus resultantes, eliminadas ya las consultas con menos de 6 documentos relevantes, se recogen en la tabla 2.2. Éstas incluyen: colección de documentos empleada, número de *topics* empleados, y longitud media de consultas, tanto cortas como largas.

En nuestros experimentos emplearemos una aproximación basada en *stemming* como línea base contra la que comparar inicialmente las diferentes aproximaciones propuestas como técnicas de normalización. Para ello emplearemos la versión para español del *stemmer* `Snowball` [3], de amplio uso por la comunidad científica, y desarrollado por el propio Porter empleando su ya clásico algoritmo de *stemming* [179]. Previamente las *stopwords* del texto han sido eliminadas en base a la lista de *stopwords* para español proporcionada con el motor de indexación empleado[18] (ver apartado 2.5.4). A mayores, en el caso de las consultas, se ha empleado una lista de *metastopwords* confeccionada tras examinar el conjunto de consultas empleado. Ambas listas de *stopwords* se recogen en el apéndice B.

Por otra parte, durante el proceso de indexación también fueron desechados aquellos términos de uso marginal en la colección, aquéllos con una frecuencia de documento (df) —número de documentos en los que aparecen— por debajo de un umbral dado. Su eliminación pretende evitar un consumo innecesario de recursos de almacenamiento y procesamiento, y se justifica en dos puntos. En primer lugar, que tras un examen de dichos términos se pudo apreciar que

[17]De hecho, la posición oficial dentro de la competición asociada a la celebración de cada edición de CLEF viene dada por los resultados obtenidos para una ejecución empleando las consultas que nosotros denominamos *cortas*.

[18]`ftp://ftp.cs.cornell.edu/pub/smart/spanish.stop`



df	*#stems*	*%stems*
[1..1]	177305	50.33 %
[2..2]	47530	13.49 %
[3..4]	36275	10.30 %
[5..8]	26593	7.55 %
[9..16]	18621	5.29 %
[17..32]	13276	3.77 %
[33..∞)	32665	9.27 %
Total	352265	100 %

Figura 2.7: Distribución de *stems* de la colección por frecuencia de documento (*df*)

una altísima proporción de los mismos correspondían a errores ortográficos[19]. En segundo lugar, que aún tratándose algunos de términos válidos, su altísimo grado de especifidad hace muy improbable su utilización dentro del vocabulario del usuario, por lo que rara vez serán utilizados en consultas [131], y en consecuencia pueden ser despreciados dada su escasa contribución [42].

Para fijar dicho umbral se estudió la distribución de los términos —en este caso *stems*— en el conjunto de ambas colecciones de documentos, EFE 1994 y EFE 1995. Los resultados obtenidos se recogen en la figura 2.7. En su parte derecha se muestra, para cada rango considerado (columna *df*), el número de términos, tanto absoluto (*#stems*) como relativo (*%stems*), cuya frecuencia de documento se encuentra dentro de dicho rango. Asimismo, en la parte izquierda de la figura se muestra gráficamente dicha distribución mediante un diagrama de sectores. Se puede apreciar que aproximadamente un 75 % de las entradas corresponden a términos con *df* menor que 5[20]. En base a ello, se optó por eliminar de los índices aquellas entradas correspondientes a términos con una frecuencia de documento menor que 5.

Las medidas de evaluación empleadas incluyen: número de documentos devueltos, número de documentos relevantes esperados, número de documentos relevantes devueltos, precisión media no interpolada, precisión media de documentos, precisión-R, precisión a los 11 niveles estándar de cobertura, y precisión a los n documentos devueltos. Los valores correspondientes son obtenidos mediante la herramienta software `trec_eval` [41], de uso ampliamente difundido en este tipo de cometidos.

Finalmente debemos puntualizar que, si bien los experimentos recogidos en este trabajo han sido realizados siguiendo el procedimiento establecido por CLEF, éstos no pueden ser considerados *oficiales*, ya que para ello la evaluación debería haber sido llevada a cabo por la propia organización de CLEF.

2.5.4. El Motor de Indexación Empleado: SMART

El proyecto SMART [199, 204, 40], desarrollado en la universidad de Cornell, es la implementación por excelencia del modelo vectorial, y ha sido pionero, en muchos aspectos, en el campo de la Recuperación de Información. Se trata, de hecho, de un sistema diseñado para la investigación en IR. Nuevos esquemas de cálculo de pesos [200], expansión de consultas y realimentación por relevancia [108, 196, 201, 44], normalización de la longitud de los

[19] De nuevo llamamos la atención sobre el gran número de errores ortográficos presentes en el corpus CLEF, ya apuntado por Figuerola et al en [74]

[20] Lo que concuerda, a su vez, con lo establecido por la ley de Zipf [272].

documentos [219, 218], son algunos de los campos explorados empleando SMART, sin dejar de lado su notable contribución, directa o indirecta, a TREC [43, 42, 44, 45] y a CLEF [211, 212].

La versión del software empleada es la 11 [4], con un esquema de pesos `atn·ntc` [211, 212] —`atn` para los términos de los documentos y `ntc` para las consultas. En SMART, los esquemas de pesos se denotan mediante una tripla de letras [200] donde la primera de ellas indica la componente que refleja la frecuencia del término en el documento, la segunda la componente correspondiente a la frecuencia dentro de la colección, y la tercera la componente de normalización respecto a la longitud del vector/documento. Las componentes utilizadas en nuestro caso son, para un término t_i en un documento d_j:

Componente de la frecuencia del término en el documento

`n`	tf_{ij}	se emplea la frecuencia pura del término t_i en el documento d_j (tf_{ij})
`a`	$0{,}5 + 0{,}5\,\frac{tf_{ij}}{\max_j\ tf_{ij}}$	la frecuencia del término (tf_{ij}) normalizada respecto a la frecuencia máxima en dicho documento, y aumentada luego para que los valores finales obtenidos se sitúen entre 0.5 y 1

Componente de la frecuencia del término en la colección

`t`	$log\,\frac{N}{n_i}$	la componente de frecuencia en el documento se multiplica por la frecuencia inversa de documento del término (idf_i), calculada como el logaritmo del cociente del número de documentos en la colección (N) partido el número de ellos que contienen el término en cuestión (n_i)

Componente de normalización

`n`	1	no hay modificación; el resultado de multiplicar los anteriores factores componente se mantiene (ya que se multiplica por 1)
`c`	$\frac{1}{\sqrt{\sum_{i=1}^{t} {w^2}_{ij}}}$	normalización del coseno; el resultado de multiplicar los anteriores factores componente se mantiene se divide por la norma del vector