

Modelos clásicos de recuperación de la información

LUIS GABRIEL JAIMES*

FERNANDO VEGA RIVEROS**

Resumen. La recuperación de la información (RI, *Information Retrieval* en inglés) es una rama de la computación que se ocupa, como su nombre lo indica, de extraer información de documentos no estructurados (cartas, periódicos, artículos, etc.) de los cuales, a diferencia de los datos con estructura (almacenados en bases de datos) no es fácil sacar información útil automáticamente.

Estas técnicas son parte de una rama más amplia de la computación, conocida como minería de textos (*text mining*), que puede ser vista como un caso particular de “minería de datos” (*Data Mining* o *Data Discovery*). Actualmente estas técnicas son ampliamente usadas en buscadores de Internet como **Google**, **Yahoo**, etc. En este trabajo se revisaron 3 modelos clásicos: booleano, vectorial y probabilístico. Estos tres modelos usan una simplificación que solo tiene en cuenta el aspecto léxico, pero no la semántica del documento.

Introducción

Según Baeza-Yates [1], la RI es la parte de la informática que estudia la recuperación de la información (no datos) de una colección de documentos escritos. Los documentos recuperados pueden satisfacer la necesidad de información de un usuario expresada normalmente en lenguaje natural. Según Korfhage [2], es la localización y presentación a un usuario de información relevante a una necesidad de información expresada como una pregunta. Según Salton [3], un sistema de recuperación de información procesa archivos de registros y peticiones de información, e identifica y recupera de los archivos ciertos registros en respuesta a las peticiones de información. Algunos de los nombres como se las conoce a estas técnicas son: *Sistema de almacenamiento y recuperación de la información*, *sistema de recuperación de información*, *sistema de recuperación de textos y base de datos de información no estructurada*.

Informalmente el problema lo podríamos definir como sigue: *Dada una necesidad de información (consulta + perfil del usuario) y un conjunto de documentos, ordenar los documentos de mayor a menor relevancia para esa necesidad y presentar un subconjunto de los más relevantes.*

Palabras y frases claves: information Retrieval, text mining, data mining o data discovery.

MSC2000: Primaria: 68P10. Secundaria: 68P30, 94A99.

* Universidad de Puerto Rico en Humacao, Departamento de Matemáticas Computacionales, Humacao, Puerto Rico, 00749, e-mail: luisgj@mate.uprh.edu, url: <http://mate.uprh.edu/~luisgj>

** Universidad de Puerto Rico en Mayagüez, Departamento de Ingeniería Eléctrica y Computadoras, Mayagüez, Puerto Rico, 00680, e-mail: fvega@ece.uprm.edu, url: <http://ece.uprm.edu/~fvega>

1. *Procesamiento de los documentos*

Se realiza preprocesamiento sobre las preguntas (consultas) y sobre los documentos (posibles ítems que satisfacen una consulta). Las preguntas pueden ser vistas como una expresión que deben satisfacer los documentos a recuperar o como otro documento que es cercano semánticamente al conjunto de documentos a recuperar.

En la figura 1 se describe el proceso de preprocesamiento de los documentos para que puedan ser recuperados usando cualquiera de las técnicas de RI.

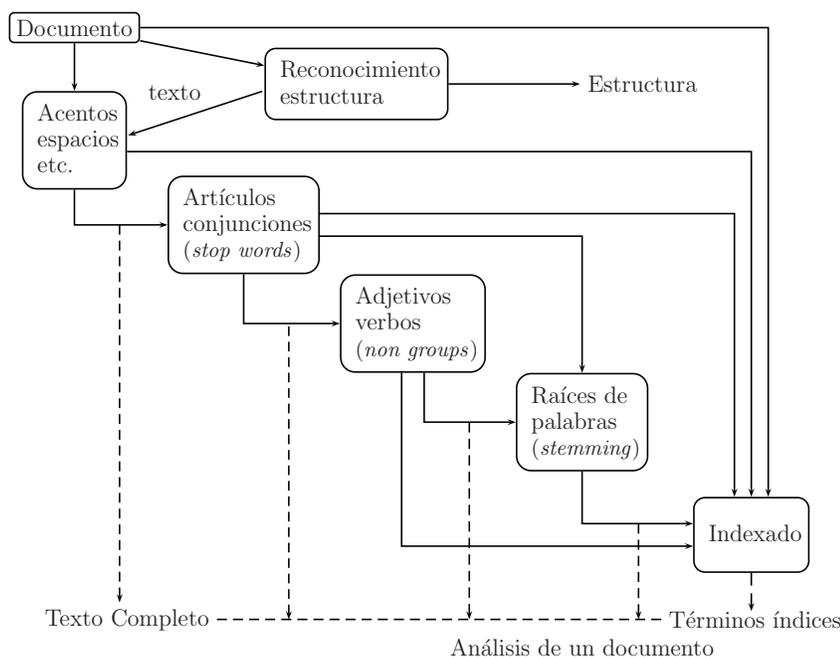


Figura 1. Preprocesamiento de los documentos.

El procesamiento de los documentos consiste en seleccionar aquellos términos que mejor representen los documentos y que mejor discriminen unos documentos respecto de otros.

No todas las palabras de un documento son igualmente representativas del contenido del mismo. Por ejemplo, en lenguaje escrito las palabras que actúan gramaticalmente como nombres suelen tener más significado que el resto de palabras, aunque no siempre es así. Así pues, la fase inicial del proceso consiste en analizar el texto del documento para determinar qué términos pueden utilizarse como términos índice. Esta fase se denomina habitualmente como *preprocesamiento del texto* de la colección de documentos, y suele seguir la siguiente sucesión de pasos [1]: **Análisis léxico del texto**, que se hace con el objetivo de determinar el tratamiento que se realizará sobre números, guiones, signos de puntuación, tratamiento de mayúsculas o minúsculas, nombres propios, etc;

eliminación de palabras vacías, fase que se realiza con el objetivo de reducir el número de términos con valores muy pocos discriminatorios para la recuperación y es una forma de delimitar el número de términos índice; **aplicación de lematización**, que se realiza sobre los términos resultantes para eliminar variaciones morfo-sintácticas y obtener términos lematizados, es decir estamos interesados en las raíces de los verbos, así que eliminamos sus sufijos y prefijos; **selección de los términos**, etapa en la cual se seleccionan los términos índice, que normalmente se realiza sobre la naturaleza sintáctica del término; los términos que actúan gramaticalmente como nombres suelen poseer un mayor contenido semántico que verbos, adjetivos o adverbios.

2. Modelos de recuperación de información

La Figura 2 ilustra los modelos actuales de RI, entre los que se resaltan, los modelos clásicos y los modelos extendidos; en este trabajo estudiamos los modelos clásicos: booleanos, vectorial y probabilístico.

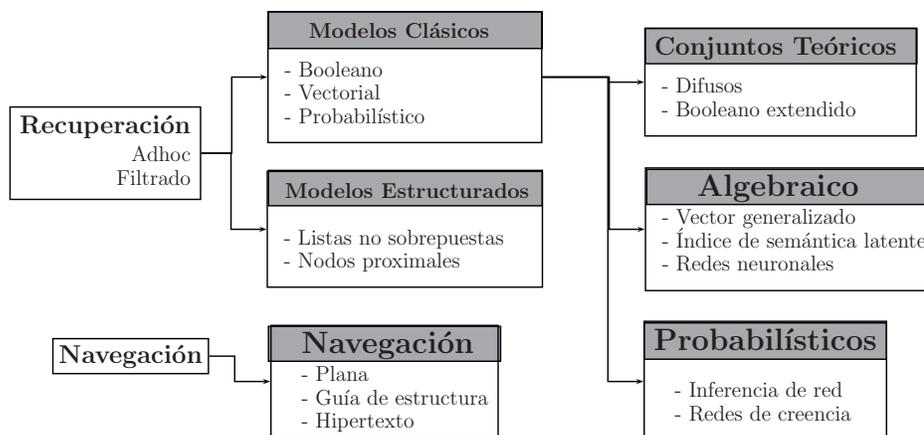


Figura 2. Clasificación de los modelos de recuperación de la información.

2.1. El modelo booleano

En este modelo, actualmente en desuso, los documentos son vistos como un conjunto de términos y las preguntas como expresiones booleanas [4]. Si designamos un documento como D y una pregunta como P , para cada par (*documento, pregunta*) se asigna un valor de similitud (*número real*) dado por la función $S : D \times P \rightarrow \mathbb{R}$, donde D y P son conjuntos de términos. Por lo general un documento es representado como un registro de que contiene ceros (0) y unos (1), dependiendo de si contiene un término o no.

Este modelo se construye a partir de las reglas clásicas de la lógica de conjuntos, y sus operadores son unión, intersección y negación; además, es muy común implementar

una combinación de las anteriores cláusulas.

En este modelo la relevancia es binaria: un documento es relevante o no lo es. Cuando consultamos una palabra el modelo la considera relevante si y solo si el documento la contiene; por ejemplo, en las consultas que utilizan “Y” los documentos deben contener todas las palabras de la consulta; en las consultas “O” los documentos deben contener alguna palabra y en consultas *A PERO NO B* los documentos deben ser relevantes para *A* pero no para *B*.

Algunas de las desventajas de este enfoque son:

- No discrimina entre documentos de mayor y menor relevancia.
- Da lo mismo que un documento contenga una o cien veces las palabras de la consulta.
- Da lo mismo que cumpla una o todas las cláusulas de un *OR*.
- No considera una clase parcial de un documento (Ej., que cumpla con casi todas las cláusulas de un “Y”).

El siguiente ejemplo ilustra lo limitado del modelo:

Estructuras de datos

En este modelo es muy común representar la colección de documentos como archivos directos o indirectos. Por ejemplo, en la siguiente tabla se muestra un archivo directo en donde las filas D1-D4 representan los documentos y las columnas T1-T5 representan los términos.

	T1	T2	T3	T4	T5
D1	0	0	0	1	0
D2	0	1	0	1	0
D3	1	0	1	1	0
D4	0	1	1	0	1

Cuadro 1.

Así, en esta tabla el documento D1 solo contiene el término T4. Hay tantas componentes en un registro que representa a un documento como términos haya en el diccionario. Un diccionario es el conjunto de todos los términos de todos los documentos $T = \{t_1, t_2, t_3, \dots\}$; un documento es el conjunto de términos del diccionario, donde t_i tiene valor $D_i = \{d_1, d_2, d_3, \dots\}$; t_i es verdad si es una palabra clave del documento. Así que la función de similitud o de semejanza sería:

$$\text{sim}(d_i, p) \text{ es verdad si } p(d_i) = \text{verdad},$$

$$\text{sim}(d_i, p) \text{ es falso si } p(d_i) = \text{falso},$$

donde d_i corresponde a los términos de un documento y p a la pregunta. Por ejemplo $D_1 = \{\text{perro, canela, corre, poco}\}$, $D_2 = \{\text{perro, canela, caza}\}$, $D_3 = \{\text{gato, blanco, persia}\}$

$$\text{sim}(D_1, p) = (\text{perro o gato}) \text{ y blanco} = \text{falso},$$

$$\text{sim}(D_3, p) = (\text{perro o gato}) \text{ y blanco} = \text{verdad}.$$

2.2. Modelo vectorial

El modelo vectorial fue definido por Salton [3]. Este es el modelo más usado en operaciones de RI, así como también en operaciones de categorización automática, filtrado de información, etc., pues permite dar un grado de pertenencia de un documento a una pregunta, a diferencia del modelo booleano que determina sólo si el documento pertenece o no a la pregunta [5].

En este modelo se seleccionan las palabras útiles, que por lo general son todos los términos del texto excepto las palabras vacías; este proceso se enriquece utilizando técnicas de *lematización y etiquetado*.

Sea $\{t_1, t_2, \dots, t_k\}$ el conjunto de los términos y $\{d_1, d_2, \dots, d_N\}$ el conjunto de los documentos. Podemos representar un documento d_i como el vector

$$d_i \rightarrow \vec{d}_i = \{w(t_1, d_i), \dots, w(t_k, d_i)\},$$

donde $w(t_r, d_i)$ es el peso del término t_r en el documento d_i . Cada documento puede ser visto como un punto en el espacio vectorial con tantas dimensiones como términos tenga el diccionario de indexación.

Ejemplo: *Diccionario* = $\{t_1 = \text{perro}, t_2 = \text{gato}, t_3 = \text{azul}, t_4 = \text{verde}, t_5 = \text{pequeño}\}$,
 $D_1 = \{\text{perro, azul, pequeño}\} = \{1, 0, 1, 0, 1\}$, $D_2 = \{\text{gato, verde}\} = \{0, 1, 0, 1, 0\}$.

Todo esto asumiendo un peso de 1.

La representación de las preguntas al igual que los documentos tiene la forma $P = \{w.p_1, w.p_2, \dots, w.p_n\}$, donde n es el número de términos distintos en la colección.

Hay varias formas de calcular los pesos; una de las más populares es considerar

$$w(t_r, d_i) = w_{r,i} = \frac{tf_{r,i} \times idf_r}{|\vec{d}_i|} = \frac{tf_{r,i} \times \log \frac{N}{n_r}}{\sqrt{\sum_{s=1}^k \left(tf_{s,i} \times \log \frac{N}{n_s} \right)^2}},$$

donde $tf_{r,i}$ (frecuencia del término), es la cantidad de veces que t_r aparece en d_i , y n_i es la cantidad de documentos donde aparece t_r . Así que si un término aparece mucho en un documento, se supone que es importante en ese documento, entonces tf crece. Pero si por otra parte el término aparece en muchos documentos, entonces no es útil para distinguir un documento de los otros, así que idf decrece. Además, podemos normalizar las frecuencias para no favorecer documentos más largos (por

ejemplo $tf_{norm} = tf_{r,d} / \max_{k \in d} \{tf_{k,d}\}$. En otras palabras, lo que intentamos medir es cuándo ayuda ese término a distinguir ese documento de los demás. Podemos calcular la *similaridad* entre dos documentos mediante la *distancia coseno*,

$$\text{sim}(d_i, d_j) = \vec{d}_i \cdot \vec{d}_j = \sum_{r=1}^k w_{r,i} \times w_{r,j},$$

que geoméricamente corresponde al coseno del ángulo entre los dos vectores.

La similaridad es un valor entre cero y uno, así que todos los documentos iguales tiene similaridad uno y los ortogonales (es decir que no comparten términos tienen similaridad cero [6]). Una consulta puede verse como un documento formado por palabras, y por lo tanto como un vector. Dado que en general cada palabra aparece una sola vez en la consulta y esta es muy corta, se puede en general calcular la relevancia de d_i para q con la fórmula

$$\text{sim}(d_i, q) = \sum_{r=1}^k w_{r,i} \times w_{r,q} \sim \sum_{r=1}^k w_{r,i} \times \text{idf}_r.$$

El modelo más simple para calcular la similitud entre una consulta y un documento, utilizando el modelo vectorial, es realizar el producto escalar de los vectores que los representan (ver anterior ecuación); en esta ecuación no se incluye la normalización de los vectores, a fin de obviar las distorsiones producidas por los diferentes tamaños de los documentos.

Considere una consulta que no distingue entre mayúsculas y minúsculas y la siguiente colección de documentos. ($Q = \text{consulta}$, $D_i = \text{documento}$).

$Q = \text{"gold silver truck"}$

$D_1 = \text{"Shipment of gold damaged in a fire"}$

$D_2 = \text{"Delivery of silver arrived in a silver truck"}$

$D_3 = \text{"Shipment of gold arrived in a truck"}$

En esta colección hay tres documentos, así que $n = 3$. Si un término aparece en solo un de los tres documentos, su idf es $\log \frac{n}{df_i} = \log \frac{3}{1} \approx 0,477$. Si el término aparece en dos de los tres documentos, su idf sería $\log \frac{n}{df_i} = \log \frac{3}{2} \approx 0,176$. Y un término que aparece en los tres documentos tiene un idf de $\log \frac{3}{3} = 0$.

Así que el idf para los términos en los tres documentos sería:

$\text{idf}_a = 0$; $\text{idf}_{arrived} \approx 0,176$; $\text{idf}_{damaged} \approx 0,477$; $\text{idf}_{fired} \approx 0,477$; $\text{idf}_{gold} \approx 0,176$;
 $\text{idf}_{in} = 0$; $\text{idf}_{of} = 0$; $\text{idf}_{silver} \approx 0,477$; $\text{idf}_{truck} \approx 0,176$.

Recordemos que el peso de j -ésimo elemento del documento d_i es $d_{ij} = tf_{ij} \times \text{idf}_i$, y que la similitud de entre la consulta Q y el documento D_i puede definirse como el producto entre los dos vectores:

$$SC(Q, D_i) = \sum_{j=1}^t w_{qj} \times d_{ij}.$$

Así que ahora podemos determinar la similaridad de cada documento con la consulta y ranquearlos de los más similares a los más disimilares; veámoslo:

$$\begin{aligned}
 SC(Q, D_1) &\approx (0)(0) + (0)(0) + (0)(0,477) + (0)(0) + (0)(0,477) \\
 &\quad + (0,176)(0,176) + (0)(0) + (0)(0) + (0,477)(0) \\
 &\quad + (0)(0,176) + (0,176)(0) + (0,176)(2) = 0,031.
 \end{aligned}$$

<i>doc</i>	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}	t_{11}
D_1	0	0	0,477	0	0,477	0,176	0	0	0	0,176	0
D_2	0	0,176	0	0,477	0	0	0	0	0,954	0	0,176
D_3	0	0,176	0	0	0	0,176	0	0	0	0,176	0,176
Q	0	0	0	0	0	0,176	0	0	0,477	0	0,176

Cuadro 2.

Nótese que el cálculo son las sumas de 11 productos, que son el total de términos de todos los documentos, y cada producto consiste en el producto de la fila D_i por la fila Q ; así por ejemplo, el tercer producto de $SC(Q, D_1)$, $(0)(0,477)$ significa que la palabra $t_3 = \textit{damage}$ aparece 0 veces en la consulta 1, pero aparece en documento 1 y tiene un peso de 0,477 en el espacio vectorial (ver tabla 2). El cálculo de las similaridades de los documentos D_2 , D_3 con respecto a la consulta sería $SC(Q, D_2) = 0,486$, $SC(Q, D_3) = 0,065$. Así que los documentos serian ranqueados D_2 , D_3 , D_1 .

2.3. El método probabilístico

Una de las mayores diferencias entre la Recuperación de la Información y otras clases de sistemas de información tiene que ver con la intrínseca incertidumbre de RI. Mientras que en los sistemas de bases de datos las necesidades del usuario en términos de información pueden ser satisfechas plenamente mediante una consulta, pues hay una precisa definición en la base de datos de lo que constituye una respuesta [7], la situación es mucho mas difícil en RI, ya que la información no está estructurada y no podemos asumir que la formulación de una consulta represente las necesidades de información, ni hay un claro procedimiento para determinarlo [8]. Los métodos probabilísticos han sido los modelos más exitosos para tratar con los asuntos de incertidumbre intrínsecos a RI.

En este sentido dos grandes enfoques han sido desarrollados [9]: los clásicos, que fundamentan el concepto de relevancia, es decir la asignación de un grado de relevancia al documento con respecto a su consulta por parte del usuario; por otro lado está un nuevo enfoque, formulado por van Rijsbergen [10], que supera esta subjetiva definición por una generalización teórico-demostrativa de modelo de bases de datos hacia una inferencia de incertidumbre. Uno de los primeros modelos probabilísticos en RI fue el *modelo de independencia binaria*, modelo que ilustra muy bien los principales conceptos del modelo probabilístico. Dada una consulta q y un documento d_j , el modelo trata

de estimar la probabilidad de que el usuario encuentre el documento d_j relevante. Se trata de observar la distribución de los documentos dentro de la colección.

El modelo supone que esta probabilidad de relevancia depende solo de la consulta y de la representación del documento. La respuesta es el conjunto al que nos referiremos como R , y este debe maximizar la probabilidad de relevancia. Es decir, suponemos que los documentos que pertenecen al conjunto R son relevantes.

Sea $T = \{t_1, t_2, \dots, t_n\}$ el conjunto de los términos de la colección. Entonces podemos representar el conjunto de términos de la colección que están en el documento d_m^T como un vector binario $\vec{x} = (x_1, x_2, \dots, x_n)$, con $x_i = 1$ si $x_i \in d_m^T$ y $x_i = 0$ si $x_i \notin d_m^T$.

Así que la función de similitud es el grado de relevancia entre un documento y una consulta, que puede ser calculado como

$$\text{sim}(q, d_j) = \frac{P(d_j \text{ relevante a } q)}{P(d_j \text{ no relevante a } q)}. \quad (1)$$

Es decir, hay que maximizar el número de documentos relevantes y minimizar los irrelevantes:

$$P(R|\text{vec}(d_j)) : \text{probabilidad que dado documento es relevante}, \quad (2)$$

$$P(\emptyset R|\text{vec}(d_j)) : \text{probabilidad que el documento dado no sea relevante}. \quad (3)$$

Una fórmula que de ordinario se usa en probabilidad es el teorema de Bayes:

$$P(A|B)P(B) = P(B|A)P(A). \quad (4)$$

Así que (1) puede ser reescrita en términos de (2) y (3) como

$$\text{sim}(d_j, q) = \frac{P(R|\text{vec}(d_j))}{P(\emptyset R|\text{vec}(d_j))}, \quad (5)$$

y reescrita en términos del teorema de Bayes como

$$\text{sim}(d_j, q) = \frac{P(\text{vec}(d_j)|R) \cdot P(R) \cdot P(\text{vec}(d_j))}{P(\text{vec}(d_j)|\emptyset R) \cdot P(\emptyset R) \cdot P(\text{vec}(d_j))}. \quad (6)$$

Podemos simplificar $P(\text{vec}(d_j))$ en el numerador y el denominador; además, consideramos que la probabilidad de que un documento sea relevante, $P(R)$, es igual a la probabilidad de que no lo sea, $P(\emptyset R)$, así que $P(R) = P(\emptyset R)$, luego

$$\text{sim}(d_j, q) = \frac{P(\text{vec}(d_j)|R)}{P(\text{vec}(d_j)|\emptyset R)}, \quad (7)$$

donde $P(\text{vec}(d_j)|R)$ es la probabilidad de seleccionar aleatoriamente el documento d_j del conjunto R . La fórmula (7) puede ser transformada en términos de los índices o términos de la colección que están en el documento:

$$\frac{\prod P(k_i = 1|R) \prod P(k_i = 0|R)}{\prod P(k_i = 1|\emptyset R) \prod P(k_i = 0|\emptyset R)}. \quad (8)$$

Aquí $P(k_i|R)$ es la probabilidad de que el término índice k_i esté presente en un documento aleatoriamente seleccionado del conjunto de términos relevantes R . Sea $p_{ik} = P(k_i = 1|R)$ y $q_k = p(k_i = 1|\emptyset R)$. Simplificando, tenemos:

$$\prod_{t_i \in d_n^T \cup q_k^T} \frac{p_{ik}}{q_{ik}} \prod_{t_i \in q_k^T \setminus d_m^T} \frac{1 - p_{ik}}{1 - q_{ik}}, \tag{9}$$

$$\prod_{t_i \in d_n^T \cup q_k^T} \frac{p_{ik}(1 - q_{ik})}{q_{ik}(1 - p_{ik})} \prod_{t_i \in q_k^T} \frac{1 - p_{ik}}{1 - q_{ik}}. \tag{10}$$

Si solo estamos interesados en ranquear los documentos, más que en ver el valor de probabilidad de relevancia, entonces consideramos el primer producto, pues el primero es constante. Si tomamos el logaritmo, el “valor de estado de recuperación” (VER) del documento d_m para la consulta q_k se calcula por la suma

$$\sum_{t_i \in d_n^T \cup q_k^T} c_{ik}, \quad \text{con } c_{ik} = \frac{p_{ik}(1 - q_{ik})}{q_{ik}(1 - p_{ik})}.$$

Así, los documentos son ranqueados por su VER.

Para aplicar el modelo tenemos que estimar los parámetros p_{ik} y q_{ik} para los términos $t_i \in q_k^T$. Esto puede ser hecho mediante la retroalimentación de relevancia. Sea f el número de documentos presentados al usuario, de los cuales r han sido juzgados relevantes. Para un término t_i , f_i es el número entre los f documentos en los cuales t_i ocurre, y r_i es el número de documentos relevantes que contienen a t_i . Podemos estimar

$$p_{ik} = \frac{r_i}{r} \quad \text{y} \quad q_{ik} = \frac{(f_i - r_i)}{(f - r)}.$$

Ilustremos este modelo mediante un ejemplo, suponiendo que la consulta q contiene dos términos $q^T = \{t_1, t_2\}$. La siguiente tabla asigna relevancias de 20 documentos junto con la distribución de los términos dentro de los documentos:

d_i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
x_i	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
x_2	1	1	1	1	1	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0
$r(q, d_i)$	R	R	R	R	\bar{R}	R	R	R	R	\bar{R}	\bar{R}	R	R	\bar{R}	\bar{R}	\bar{R}	R	\bar{R}	\bar{R}	\bar{R}

Cuadro 3.

Así, $p_1 = 8/12$, $q_1 = 3/8$, $p_2 = 7/2$, $q_2 = 4/8$, $c_1 = \log 10/3$ y $c_2 = \log 7/5$. Basado en estos pesos, los documentos son ranqueados de acuerdo con sus vectores binarios (1, 1), (1, 0), (0, 1) y (0, 0).

Referencias

- [1] R. BAEZA-YATES and B. RIBEIRO-NETO. *Modern Information Retrieval*, Addison-Wesley, Wokingham, UK, 1999.
- [2] R. KORFHAGE. *Information Storage and Retrieval*, New York: John Wiley, 1997.
- [3] G. SALTON and M. J. MCGILL. *Introduction to Modern Information Retrieval*, New York: McGraw-Hill, 1983.
- [4] R. BAEZA-YATES, EDS. *Information Retrieval: Data Structures and Algorithms's*, Englewood Cliffs (NJ): Prentice-Hall, 1992, 241–263.
- [5] WEIGO FAN, LINDA WALLACE, STEPHANIE RICH and ZHONGJU ZHANG. “Tapping into the Power of Text Mining”, *Journal the Communications of ACM* 2205.
- [6] RADA MIHALCEA. *IR models: Vectorial Model*.
<http://www.cs.unt.edu/rada/CSCE5300>.
 Some slides in this set were adapted from an IR course taught by Ray Mooney at UT Austin, 2004.
- [7] K. SPARK, S. WALKER and S. E. ROBERTSON. *A probabilistic model information retrieval: development and status*, Computer Laboratory, University of Cambridge and Microsoft Research Limited, 1998.
- [8] NORBERT FUHR and CHRIS BUCKLEY. *A probabilistic Learning Approach for Document Indexing*, TH Darmstadt Germany and Cornell University, NY USA, 1994.
- [9] NORBERT FUHR and ULRICH PFEIFER. *Probabilistic Information Retrieval as Combination of Abstraction, Inductive Learning and Probabilistic Assumptions*, University of Dortmund, Dortmund, Germany.
- [10] C. J. RIJSBERGEN VAN. “A theoretical basis for the use of co-occurrence data in information retrieval”, *Journal of Documentation*, **33** (1977), 106–119.

LUIS GABRIEL JAIMES
 Universidad de Puerto Rico en Humacao
 Departamento de Matemáticas Computacionales
 Humacao, Puerto Rico, 00749
e-mail: luisgj@mate.uprh.edu, *url*: <http://mate.uprh.edu/~luisgj>

FERNANDO VEGA RIVEROS
 Universidad de Puerto Rico en Mayagüez
 Departamento de Ingeniería Eléctrica y Computadoras
 Mayagüez, Puerto Rico, 00680
e-mail: fvega@ece.uprm.edu, *url*: <http://ece.uprm.edu/~fvega>