Modelo de Búsqueda Web Basado en Información del Contexto del Usuario y Técnicas de Filtrado Colaborativo

Web Search Model Based on User Context Information and Collaborative Filtering Techniques

SARA DONNELLY GARCÉS AGREDO
Ingeniera de Sistemas, Magister en Computación (c)
Profesor Auxiliar, Departamento de Sistemas, Facultad de Ingeniería Electrónica y Telecomunicaciones
Miembro del Grupo de I+D en Tecnologías de la Información, Universidad del Cauca
sgarces@unicauca.edu.co
Popayán, Cauca, Colombia

CARLOS ALBERTO COBOS LOZADA
Ingeniero de Sistemas, Magíster en Informática, Ph.D. (c) en Ingeniería de Sistemas y Computación
Profesor Titular, Departamento de Sistemas, Facultad de Ingeniería Electrónica y Telecomunicaciones
Director del Grupo de I+D en Tecnologías de la Información, Universidad del Cauca
ccobos@unicauca.edu.co
Popayán, Cauca, Colombia

LUIS CARLOS GÓMEZ FLÓREZ
Ingeniero de Sistemas, Magíster en Informática
Profesor Titular, Escuela de Ingeniería de Sistemas e Informática, Facultad de Ingenierías Físico Mecánicas
Director del Grupo de I+D en Sistemas y Tecnologías de la Información, Universidad Industrial de Santander
lcgomezf@uis.edu.co
Bucaramanga, Santander, Colombia

Fecha recibido: 15/05/2012
Fecha de aprobación: 15/06/2012

RESUMEN

A pesar del continuo desarrollo que han tenido los buscadores Web modernos, estos aún no satisfacen a cabalidad las necesidades de los usuarios, siendo la relevancia de los documentos recuperados uno de los principales aspectos que afectan la calidad de búsqueda. En este artículo se propone un modelo de meta buscador Web que integra el filtrado colaborativo (basado en ítems) con la propuesta de Massimo Melucci, que se basa en proyectores sobre planos que se originan en la información del contexto del usuario. El modelo fue implementado en un meta buscador Web que recupera documentos de buscadores tradicionales como Google y Bing, donde se muestran los resultados por medio de una lista de documentos ordenados por relevancia, basado en la información del contexto del usuario y en la retroalimentación colaborativa de la comunidad. El modelo propuesto se constituye en un aporte para el área de recuperación de información, dado que muestra promisorios resultados en pruebas realizadas sobre colecciones cerradas y con usuarios.

PALABRAS CLAVE. Recuperación de Información, contexto de usuario, filtrado colaborativo, expansión de consulta, búsqueda web.

ABSTRACT

Despite the continuous development modern Web browsers have had, they have not fulfilled user needs, and the retrieved documents relevance is one of the main issues affecting the search quality. The proposed web search meta model engine integrates Web search collaborative filtering (based on items) to Massimo Melucci's proposal that is based on projectors on plans that came in the user context information. The obtained model was implemented in a meta search site that retrieves documents from traditional search engines like Google and Bing. It presents the results to the user through a list of documents sorted by relevance based on information from the user's context and the collaborative community feedback. The proposed model constitutes a contribution to the field of information retrieval, since it shows promising results in both closed collections and open collections tests.

KEYWORDS: Information retrieval, user's context, collaborative filtering, query expansion, Web search.

1. INTRODUCCIÓN

La recuperación de información (RI) es un área interdisciplinaria de estudio que busca las mejores formas de representar, almacenar, organizar y acceder ítems de información en forma automática [1], donde los ítems de información se consideran documentos (normalmente no estructurados) que están relacionados con las solicitudes de búsqueda de un usuario [2].

Un sistema de recuperación de información (SRI) está compuesto básicamente por: Documentos (almacenados en bases de datos o directorios), Usuarios, Consultas, Resultados/Respuestas (documentos relacionados y ordenados por relevancia), Re-alimentación (del usuario al sistema) y el Proceso (software y hardware que realiza el proceso de recuperación de información) [1–3].

Las características que describen al usuario, el tiempo, el lugar o cualquier otra información que emerge de la interacción entre el usuario y un SRI forma la noción del contexto [4]. La RI es intrínsecamente dependiente del contexto; lo que es relevante para un usuario en un lugar y en un momento determinado, puede no ser relevante para otro usuario, en otro lugar, u otro tiempo [5].

En la práctica, sistemas clásicos de RI como los motores de búsqueda (por ejemplo, Google, Yahoo!, Bing y Ask) son cada vez más populares [6–8] y de gran utilidad cuando se desea recuperar información en la Web [1], pero su funcionamiento interno aún presenta falencias en el filtrado, ordenamiento y manejo del contexto. En general, los problemas con la búsqueda Web pueden ser divididos en dos clases: de datos y de usuario. Entre los problemas relacionados con los datos están: el volumen, la velocidad de los cambios, la naturaleza contradictoria de la Web (spamming de metadatos, contenido y enlaces), la diversidad de lenguajes y de contenidos, el esquema de cooperación de servidores web para los buscadores, la recuperación de datos multimedia, entre otros. Desde el lado del usuario se necesitan mejores lenguajes de consulta, mejores interfaces de usuario y mejores esquemas de visualización de resultados. Además un problema de mucha importancia está relacionado con la relevancia de los resultados que se le presentan a los usuarios [9].

Por otro lado, en los motores de búsqueda tradicionales las características contextuales no son capturadas en el tiempo de indexación, ni son explotadas en el tiempo de recuperación [10]. Por lo tanto, la recuperación puede ser inexacta y con una alta posibilidad de equivocarse en el orden de presentación de los documentos [1, 3] lo que en general, conlleva a que el usuario lea una serie de páginas con resultados mixtos, gastando mayor tiempo en la lectura de las páginas recuperadas [6], por otro lado en ocasiones sólo las primeras páginas recuperadas son leídas sin tener en cuenta las demás, lo que puede generar que el usuario abandone la búsqueda sin obtener el resultado deseado [2].

Aunque como se mencionó anteriormente, existen muchas circunstancias que motivan la insatisfacción del usuario por la baja precisión de los resultados entregados por los buscadores tradicionales y sus tiempos de respuesta [9, 11] en el presente artículo se presenta un meta buscador Web, que usa explícitamente dos estrategias: la adecuada gestión del contexto del usuario basado en la propuesta "A Basis for Information Retrieval in Context" [5], y el filtrado colaborativo basado en la retroalimentación que el usuario puede registrar explícitamente en el SRI (o motor de búsqueda Web) con el fin de mejorar la relevancia de los resultados recuperados. La retroalimentación (feedback) del usuario es una de las estrategias más populares de reformulación de consultas en RI [1], y su combinación con técnicas de filtrado colaborativo le permite a los usuarios del SRI tomar provecho de la información que otros usuarios han generado, buscando obtener mejores resultados con un menor esfuerzo individual.

A continuación en la sección 2, se presenta un resumen de trabajos previos relacionados con el proceso de expansión de consulta. Luego, en la sección 3 se presenta la propuesta de Melucci [5] sobre la que se basa la presente investigación. En la sección 4 por su parte se muestra qué es el filtrado colaborativo, posteriormente en la sección 5 se detalla el modelo del Meta buscador Web propuesto, en la sección 6 se dan a conocer los resultados obtenidos en la experimentación y el análisis de los resultados. Finalmente en la sección 7 se presentan las conclusiones de la investigación y el trabajo futuro que el grupo de investigación espera desarrollar en el área.

2. EXPANSIÓN DE CONSULTAS EN SRI

Para la mayoría de usuarios es difícil formular consultas que estén bien diseñadas para propósitos de recuperación de información. Los documentos inicialmente obtenidos pueden ser revisados de acuerdo a la relevancia y luego mejorar la formulación de la consulta con el objetivo de encontrar documentos más relevantes. Tal reformulación involucra dos pasos básicos: expandir la consulta inicial con nuevos términos y re-ponderar términos en la consulta expandida [1].

Los enfoques para mejorar la consulta inicial a través de expansión de la consulta y de re-ponderación de términos, se agrupan en tres categorías (i) basados en el feedback del usuario, (ii) basados en información derivada del conjunto de documentos inicialmente recuperado, llamado conjunto local de documentos y (iii) basados en información global derivada de la colección de documentos [1].

La estrategia más popular de la categoría (i) feedback del usuario, es la retroalimentación de relevancia del usuario (URF por sus siglas en inglés, User Relevance Feedback), donde se requiere que el usuario marque los documentos como relevantes o no relevantes y luego, a cada nueva consulta del usuario se le agregan o quitan los términos que el sistema ha encontrado como relevantes o no en los documentos marcados [1, 2, 12]. La propuesta presentada en el presente artículo pertenece a esta categoría.

La idea básica de URF, en el modelo vectorial, es reformular la consulta de tal manera que se acerque más al espacio de los documentos relevantes, para lo cual Rocchio [2, 13] propone la fórmula (1) para generar la consulta expandida. Donde es la consulta inicialmente digitada por el usuario, R es el conjunto de documentos relevantes, R' es el conjunto de documentos no relevantes, α, β y γ son parámetros de afinación del algoritmo y es la consulta expandida.

$$\vec{q_e} = \alpha \times \vec{q_i} + \frac{\beta}{|R|}\sum_{d \in R} \vec{d} - \frac{\gamma}{|R'|}\sum_{d \in R'} \vec{d} \qquad (1)$$

Los parámetros α, β y γ controlan el equilibrio entre el conjunto de documentos evaluados versus la consulta, si se tiene una gran cantidad de documentos evaluados, se podría tener valores altos para β y γ. A partir de la nueva consulta se mueve alguna distancia hacia el centroide de los documentos relevantes y alguna distancia desde el centroide de los documentos no relevantes. La retroalimentación de relevancia puede mejorar tanto el recuerdo (Documentos relevantes recuperados/Documentos relevantes) como la precisión (Documentos Relevantes/Documentos recuperados). Por otro lado, la retroalimentación positiva resulta ser mucho más valiosa que la negativa, por lo que en la mayoría de sistemas de recuperación de información se fija $\gamma < \beta$. Valores razonables podrían ser $\alpha = 1$, $\beta = 0{,}75$, y $\gamma = 0{,}15$. De hecho, algunos sistemas como los de búsqueda de imágenes, sólo permiten la retroalimentación positiva, que es equivalente a establecer $\gamma = 0$.

Por otro lado, las categorías (ii) y (iii) expanden las consultas automáticamente, usando retroalimentación automática de relevancia (ARF por sus siglas en inglés, Automatic Relevance Feedback), también conocida como pseudo retroalimentación (Pseudo Feedback) [1, 2, 14]. Esta retroalimentación se realiza con documentos locales, categoría (ii) y documentos globales, categoría (iii).

En los métodos basados en documentos locales, categoría (ii), se envía originalmente la consulta al motor de búsqueda, con los resultados entregados, se selecciona un grupo de los documentos (los primeros resultados, considerados como más relevantes) y con ellos se reformula la consulta (formula de Rocchio con $\gamma=0$) y se re-envía al motor. Los resultados de la segunda consulta (o consulta expandida) son los que realmente se le presentan al usuario) [1, 2, 14]. Ejemplos de esta estrategia son los trabajos de Robertson y Sparck Jones [15, 16] que re-ponderan los términos de la consulta, o los de Dillon y Desper [17] que abandonan los términos del usuario y usan términos de los documentos inicialmente recuperados.

Otros enfoques de feedback local que expanden la consulta con términos correlacionados a los términos

de la consulta actual, usan clusters locales, donde se encuentran dichos términos correlacionados, construidos con el conjunto de documentos local. Los tres tipos de cluster comúnmente usados son: *clusters de asociación*, se basan en la idea que los términos que frecuentemente co-ocurren dentro de documentos tienen una asociación de sinonimia*;* los *cluster de métricas* se basan en la idea que dos términos que están juntos en la misma sentencia están más correlacionados que dos términos que ocurran separadamente en un documento; y el *cluster escalar* cuya idea básica es que dos términos con vecinos similares tienen algunas relaciones de sinonimia (la relación es indirecta o inducida por el vecino) [1].

En los métodos basados en documentos globales, categoría (iii), se analizan todos los documentos de la colección y se establecen relaciones entre los términos (palabras), por lo que, estos métodos normalmente se realizan basados en tesauros. La desventaja de este método es que necesita todos los documentos y el proceso de actualización del tesauro puede ser costoso y complejo [1, 2]. Otras estrategias dependientes del dominio (o de la colección) pueden estar basadas en clusters o agrupaciones de términos [18] y en similitud de términos [17]. Desafortunadamente, estos enfoques en aplicaciones específicas como la búsqueda Web, pueden promover la información publicitaria, por ejemplo, cuando las páginas incluyen repetidamente marcas o nombres de empresas o productos [17]. Otros enfoques que son independientes del dominio o corpus de la colección, consisten en usar diccionarios o tesauros globales, tales como WordNet.

Nuevos enfoques incluyen entre otros: técnicas morfológicas que procesan los términos de la consulta y técnicas semánticas que encuentran términos similares a los digitados por el usuario [19], el etiquetado social (social tagging) [20, 21] como una estrategia que aprovecha la creciente popularidad de las redes sociales y los sistemas de etiquetado colaborativo. Estos enfoques extienden la familia de las bien conocidas matrices de co-ocurrencia; el uso de conocimiento semántico representado en ontologías [22–24], a través del análisis de las relaciones de los conceptos y sus términos, las funciones, las instancias y los axiomas; y métodos que mezclan varias técnicas, por ejemplo el uso de ontologías con filtrado colaborativo y redes neuronales artificiales [25].

Por otro lado, no solamente se usa la expansión de consultas en la recuperación de texto, en [26] presentan un framework de recuperación de imágenes, que representa las imágenes como vectores de conceptos ponderados. Para generar el vocabulario de conceptos se construye un modelo estadístico utilizando técnicas de clasificación basadas en Support Vector Machine (SVM), la expansión de consulta se basa en análisis a nivel local y a nivel global. Para el análisis local se analiza la correlación entre los conceptos (patrones de co-ocurrencia) y las limitaciones métricas basadas en la proximidad de vecinos entre los conceptos codificados de las imágenes. En el análisis global, se analizan las similitudes de los conceptos de la colección como un todo en la forma de un tesauro de similitud. Los resultados experimentales de una colección de fotografías de escenas naturales y una base de datos biomédica de diferentes técnicas de imagen demuestran la eficacia del marco propuesto en términos de precisión y exhaustividad.

3. BASES PARA RECUPERAR INFORMACIÓN EN CONTEXTO

En [5], Melucci propone un modelo de manejo de la información del contexto, tomando cada propiedad o característica del contexto como una base no orto normal de un espacio vectorial que luego es usado para establecer una función probabilística denominada "probabilidad de relevancia o función de ranking", con la que se re-ordena la presentación de los resultados al usuario. Este modelo es general, independiente del medio y aplicable a varias tareas. Adicionalmente, usa múltiples fuentes de evidencia presentes en una descripción de contexto (las propiedades de contexto se utilizan para denotar una de las formas en que el contexto opera sobre la materialización de objetos de información, por ejemplo, tiempo de visualización, retención de documentos, el espacio, el contenido y el tipo de documento), funciones de rastreo, matrices de densidad (que incorporan información acerca de la ocurrencia de algunos factores contextuales en términos de preguntas cuyas respuestas están sujetas a medidas de probabilidad) y los proyectores (proyección de cualquier punto x del espacio vectorial a un punto del subespacio imagen de la transformación), para mejorar las estructuras de información de feedback implícito personalizado para cada usuario y para cada tarea de búsqueda.

El término factor contextual se usa para significar uno de los posibles valores de una propiedad contextual, por ejemplo, "introducción" y "matemáticas" son los posibles valores de la propiedad "tipo de documento". Si bien los factores de una propiedad contextual son mutuamente excluyentes, los factores de diferentes

propiedades no lo son. La idea entonces consiste en que cada factor puede ser instanciado como un vector de un espacio, es decir, como n-tuplas de números.

Una característica importante de este modelo es que todo se describe como un subespacio de un espacio vectorial complejo o como una combinación lineal de subespacios: Un rayo (semirecta generada por un vector) es un ejemplo de subespacio de este tipo. Esta característica busca describir los objetos como rayos, planos, o combinaciones de ellos. Además, los factores contextuales, también se describen como rayos, planos, o combinaciones de ellos.

En [10] Melucci también reporta la idea de modelar el contexto usando bases del espacio vectorial. La premisa básica es que un vector base modela un documento o una consulta, que la semántica del documento o la consulta depende del contexto y que un cambio en el contexto puede ser modelado por una transformación lineal de una base vectorial. En otras palabras, cada documento o consulta está asociado a una base vectorial distinta y la conexión de una base a otra está gobernada por una transformación lineal.

Por otro lado, como los términos (palabras claves) son representados como vectores base, diferentes contextos deberían reflejarse sobre bases diferentes. Hay más de una base para una colección de documentos. Similarmente no existe una única base para todas las consultas, hay tantas bases como contextos, aún para cada consulta única. El cambio de contexto puede ser modelado como una transformación de matrices [10]. Así, un cambio del contexto P_B a un nuevo contexto P_Q se logra transformando la correspondiente matriz P_B a la matriz P_Q .

A continuación se presenta un ejemplo numérico del proceso de proyección de vectores de un subespacio vectorial bidimensional a otro. (Ver Figura 1). Suponga que el primer subespacio se genera por P_B cuyos vectores columna base son t_1, t_2, que no son ortogonales sino independientes uno del otro, $P_B = \begin{bmatrix} 4 & 2 \\ 2 & 4 \end{bmatrix}$. El segundo subespacio se genera por P_Q cuyos vectores columna base son $b_1, b_2, p_Q = \begin{bmatrix} -2 & -2 \\ 4 & -2 \end{bmatrix}$. Asuma que un documento d está representado por el vector d cuyos coeficientes son a_1, a_2 definidos por: $a = \left[\frac{1}{3}, \frac{1}{3}\right]^T$ Si se desea representar el vector d en el subespacio generado por P_B se debe combinar linealmente los coeficientes a_1, a_2 con t_1, t_2, asi $d = P_B . a = [2, 2]^T$. De manera similar, considere que la consulta q se representa por el vector q, cuyos coeficientes son c_1, c_2 así: $c = \left[\frac{2}{4}, 2\right]^T$ pero se genera por la base P_Q, entonces $q = P_Q . c = [-5, -2]^T$.

Si otra consulta q' se genera por P_B pero con los mismos coeficientes de q, entonces $q' = P_B . c = [6, 9]^T$. Los vectores de las consultas se generan por los mismos coeficientes, pero q' está más cerca (medido por el ángulo entre los dos vectores) a d, ya que se generó utilizando su misma base, mientras que se generó por P_Q, que es "distante" de P_B.

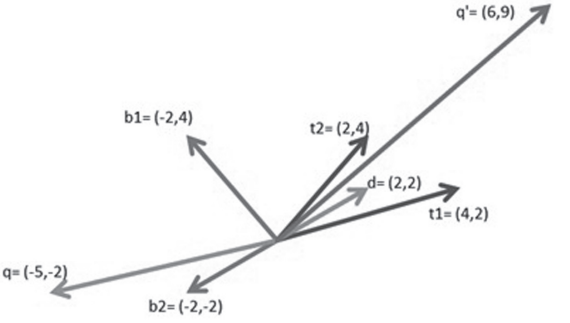


Figura 1. *Un ejemplo del modelado del contexto*

Formalmente, este modelo se expresa de la siguiente manera: Sea d un vector de un documento que puede ser generado por una base P_B y sea q un vector que representa una consulta (q no necesariamente se genera sobre la misma base) que puede ser generado por una base P_Q. Por lo tanto d está representado por $d = P_B \cdot a$; mientras que $q = P_Q \cdot b$ donde a y b son coeficientes usados para combinar los vectores base de P_B y P_Q respectivamente. Si la relevancia es estimada por el producto interno usual, los documentos son ordenados por $d^T . q = (P_B . a)^T . (P_Q . b) = a^T . (P_B^T . P_Q) . b$ considerada la función de ranking.

Así, en un escenario donde un usuario está accediendo a un SRI con el fin de recuperar documentos relevantes para su necesidad de información, el contexto influencia no solo la selección de las palabras clave de la consulta, sino su semántica y la forma en la que ésta se relaciona con otras palabras.

Para entender mejor lo expuesto anteriormente, a continuación se presenta un ejemplo con múltiples dimensiones, descriptores o características de contexto. Sean d_1, d_2, d_3, d_4 los siguientes cuatro documentos:

d_1= herramientas del lenguaje
d_2= lenguaje de modelado unificado
d_3= lenguajes de programación
d_4= lenguaje de modelado unificado.

Estos documentos son descritos por seis (6) términos (descriptores) t_1=lenguaje, t_2=modelado, t_3=relación,

t_4=clase, t_5=modernas, t_6=diagrama. Sean S_1 y S_2 las matrices de correlación correspondientes a los contextos de Ciencias de la Computación y de Lenguas Modernas respectivamente.

$S_1 =$

1	0,7	0,5	0,5	0,8	0,7
0,7	1	0,8	0,5	0,3	0,4
0,5	0,8	1	0,8	0,2	0,7
0,5	0,5	0,8	1	0,2	0,8
0,8	0,3	0,2	0,2	1	0,1
0,7	0,4	0,7	0,8	0,1	1

$S_2 =$

1	0,5	0,6	0,4	0,7	0,4
0,5	1	0,3	0,2	0,6	0,2
0,6	0,3	1	0,1	0,5	0,2
0,4	0,2	0,1	1	0,7	0,1
0,7	0,6	0,5	0,7	1	0,1
0,4	0,2	0,2	0,1	0,1	1

S_1 y S_2 son matrices simétricas con vectores columna independientes, su contenido representa la relación que tiene un término con cada uno de los otros términos almacenados en el contexto del usuario, así la correlación que tiene un término consigo mismo es uno (1) representado en la diagonal principal. Por otro lado, la correlación entre términos varía de acuerdo al contexto, por ejemplo los términos t_1=lenguaje y t_2=modelado tienen alta correlación (0,7) el contexto de Ciencias de la Computación y sólo (0,5) en el contexto de Lenguas Modernas.

Las matrices P_{B1} y P_{B2} que se muestran a continuación representan los proyectores de los contextos de Ciencias de la Computación y de Lenguas Modernas obtenidas a partir de las matrices de correlación S_1 y S_2 respectivamente, los cuales se calcularon usando la descomposición de Cholesky (que retorna una matriz triangular con columnas independientes P_B tal que S = $P_B^T \cdot P_B$):

$P_{B1} =$

1.00	0,70	0,50	0,50	0,80	0,70
0,00	0,71	0,63	0,21	-0,3	-0,1
0,00	0,00	0,59	0,70	0,05	0,72
0,00	0,00	0,00	0,46	-0,3	-0,06
0,00	0,00	0,00	0,00	0,32	-1,7
0,00	0,00	0,00	0,00	0,00	0,00

$P_{B2} =$

1,00	0,50	0,60	0,40	0,70	0,40
0,00	0,86	0,00	0,00	0,28	0,00
0,00	0,00	0,80	-0,1	0,10	-0,05
0,00	0,00	0,00	0,90	0,48	-0,07
0,00	0,00	0,00	0,00	0,42	-0,3
0,00	0,00	0,00	0,00	0,00	0,85

Sea q la consulta "modelado" representada por los coeficientes b = [0 1 0 0 0 0]. Si esta consulta se expresa en el contexto de "Ciencias de la Computación" el vector de consulta es $q_1 = P_{B1} \cdot b$ = [0,70 0,71 0 0 0 0] Por otro lado, si el contexto fuera "Lenguas Modernas" el vector consulta $q_2 = P_{B2} \cdot b$ sería $[0{,}50\ 0{,}86\ 0\ 0\ 0\ 0]^T$.

De manera similar, considere los coeficientes a_1, a_2, a_3 y a_4 de los documentos d_1, d_2, d_3 y d_4 respectivamente:

$$a_1 = [1\ 0\ 0\ 0\ 1\ 0]^T$$
$$a_2 = [1\ 1\ 0\ 0\ 0\ 0]^T$$
$$a_3 = [1\ 0\ 1\ 1\ 0\ 1]^T$$
$$a_4 = [1\ 1\ 1\ 1\ 0\ 1]^T$$

Al aplicar la función de ranking para la consulta "modelado", en el contexto de Ciencias de la Computación, el orden de la presentación de los documentos seria d_4, d_3, d_2, d_1:

Documento	Función de ranking	Resultado
d_4	$a_4^T\ P_{B1}^T\ P_{B1} \cdot b$	3,4
d_3	$a_3^T\ P_{B1}^T\ P_{B1} \cdot b$	2,4
d_2	$a_2^T\ P_{B1}^T\ P_{B1} \cdot b$	1,7
d_1	$a_1^T\ P_{B1}^T\ P_{B1} \cdot b$	1,0

Para la misma consulta en el contexto de Lenguas Modernas, el orden de la presentación de los documentos seria d_4, d_2, d_3, d_1:

Documento	Función de ranking	Resultado
d_4	$a_4^T\ P_{B2}^T\ P_{B2} \cdot b$	2,2
d_2	$a_2^T\ P_{B2}^T\ P_{B2} \cdot b$	1,5
d_3	$a_3^T\ P_{B2}^T\ P_{B2} \cdot b$	1,2
d_1	$a_1^T\ P_{B2}^T\ P_{B2} \cdot b$	1,1

La explicación de los diferentes resultados se deriva de la estrecha relación entre los términos, que son mayores en el contexto de Ciencias de la Computación que en Lenguas modernas, como se muestra en S_1 y S_2. Además, el orden de la presentación de los documentos es diferente dependiendo de cómo ellos mismos y la consulta se ven reflejados en cada uno de los contextos.

Este ejemplo explica cómo el conocimiento del contexto puede ser crucial ara poder obtener un "correcto" ordenamiento de los documentos, es decir para entregar resultados más relevantes a las necesidades de los usuarios es de vital importancia conocer su contexto.

4. FILTRADO COLABORATIVO

A mediados de los años 90, varios investigadores empezaron a trabajar en los sistemas de recomendación motivados por las limitaciones de los sistemas de búsqueda tradicionales. Estos sistemas de recomendación han sido diseñados para recolectar la experiencia de los usuarios con los ítems de información y hacer recomendaciones con base en esa experiencia, dichas recomendaciones se hacen usando diferentes técnicas, pero el filtrado colaborativo es la más común. Existen básicamente dos técnicas de filtrado colaborativo: basadas en usuario, que pretenden encontrar a otros usuarios que tengan gustos similares; y las basadas en ítem, donde un usuario estaría interesado en encontrar ítems que son similares a los que le gustaron a un usuario anterior [27]. Amazon [28] es tal vez el sistema de recomendación más conocido y usado en la actualidad, y es de especial interés porque maneja más de 29 millones de usuarios y varios millones de ítems en su catalogo; este sistema se basa en un algoritmo de filtrado colaborativo de ítem a ítem [29]. En general el filtrado colaborativo es el proceso de filtrar información o patrones usando técnicas que involucran la colaboración de múltiples agentes, puntos de vista, fuentes de datos, entre otros [30–32], para el caso de Amazon los ítems de información son los productos y las personas que los compran dan sus distintos puntos de vista, que sirven de recomendación a quienes compran productos similares.

Uno de los principales problemas que enfrentan la mayoría de técnicas de filtrado colaborativo es la falta de información, conocido en la literatura como el problema del ramp-up o cold start [33]. Este problema se refiere a dos situaciones: la primera, cuando un nuevo usuario ingresa al sistema, ya que no se tiene información sobre sus preferencias, la segunda, es análoga, pero cuando se crea un nuevo servicio, ya que no hay ninguna información (experiencias) de ningún usuario para ese nuevo servicio.

"Otra técnica de filtrado colaborativo busca un enfoque basado en la preferencia. En estos sistemas, al usuario se le encuesta sobre sus preferencias sobre los productos y basados en la teoría de la utilidad multi-atributo se encuentran los ítems (productos) preferidos, sin importar que el conjunto de alternativas sea extenso" (traducción libre) [34]. Estos sistemas, en muchos casos generan una sobrecarga cognitiva en el usuario, al momento de definir sus preferencias, y no son adecuados cuando las preferencias de los usuarios son extrañas (poco frecuentes). Otras investigaciones muestran diferentes perspectivas de usar el filtrado colaborativo, por ejemplo, el uso de la información que generan los usuarios (social bookmarking) para mejorar la búsqueda Web [34]; el uso de ontologías junto con el filtrado colaborativo ítem a ítem para superar el problema de falta de información [32]. Marlin en [31] además hace una descripción detallada de las técnicas y su relación directa con el aprendizaje de máquina.

5. MODELO DE BÚSQUEDA WEB BASADO EN INFORMACIÓN DEL CONTEXTO DEL USUARIO Y TÉCNICAS DE FILTRADO COLABORATIVO

El modelo del meta buscador Web propuesto integra el filtrado colaborativo (basado en ítems) con la propuesta de Melucci [5], basada en proyectores sobre planos que se originan en la información del contexto del usuario. Este modelo cuenta con los siguientes pasos y componentes generales para su funcionamiento (ver Figura 2):

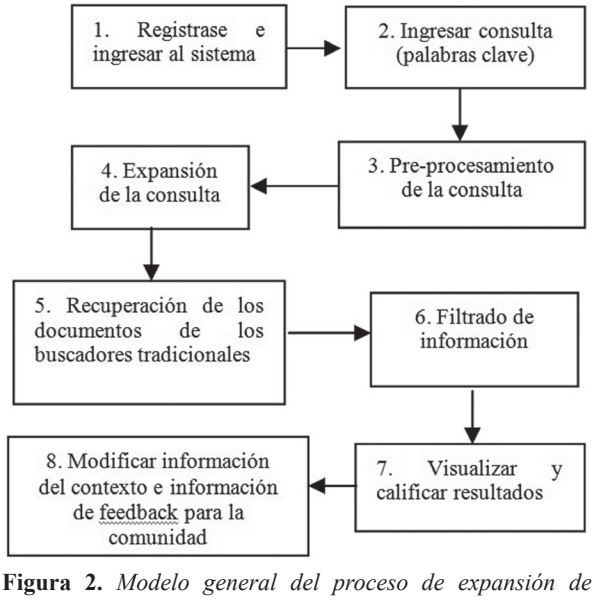


Figura 2. *Modelo general del proceso de expansión de consulta*

1. **Registrarse e Ingresar al sistema**: el usuario llena inicialmente un formulario para registrase. Luego cada vez que pretenda usar el meta buscador Web deberá realizar un proceso de login.

2. **El usuario ingresa la consulta basado en palabras clave**: El usuario digita la consulta de manera tradicional, basado en palabras claves. En principio lo que se busca es que el modelo se comporte (desde la perspectiva del usuario) de la misma manera como lo hacen los actuales buscadores web.

3. **Pre procesamiento de consulta**: En el pre procesamiento de la consulta se eliminan acentos y caracteres especiales, se organizan las palabras de la consulta, se eliminan las palabras vacías, se dejan todas las palabras en minúsculas y se realiza el proceso de stemming (llevar diferentes palabras a una raíz común, por ejemplo running y runner a su raíz común run) y luego se calcula la frecuencia de los términos en la colección de documentos [1][2].

4. **Expansión de la consulta**: de manera automática, el sistema realiza un proceso de expansión de consulta basado en la información de contexto disponible del usuario y de la comunidad y muestra una lista desplegable con los términos que se espera que complementen mejor los ya digitados por el usuario, para que éste los seleccione. Este primer proceso de expansión se hace explícitamente y con la aprobación del usuario (interacción del modelo con el usuario). Luego, cuando el usuario termina de digitar la consulta y solicita formalmente la búsqueda, el modelo realiza un proceso de expansión de consulta, que se hace de manera oculta al usuario, en el cual se le agregan términos a la consulta actual, que son obtenidos del contexto del usuario y que tienen mayor correlación con los términos que actualmente está consultando (basado en la matriz de correlación de términos, S_1 y S_2 del ejemplo final de la sección anterior). Esta última consulta se denomina consulta expandida.

5. **Recuperación de documentos de los buscadores tradicionales**: Se envía la petición de búsqueda (consulta expandida) a las APIs de los buscadores tradicionales (por ejemplo, Google y Bing). Los documentos se organizan en una tabla de términos por documentos estándar, comúnmente conocida como Matriz de Términos por Documentos [2], aplicando el pre procesamiento de cada documento y registrando solamente la frecuencia de los términos en los documentos.

6. **Filtrado de información**: basándose en la propuesta de Melucci [5] (información de contexto representada en un espacio vectorial no ortogonal), se toma provecho de la información de contexto del usuario y se complementa con la información de la comunidad, basado en técnicas de filtrado colaborativo [31], para filtrar y ordenar la información que se muestra al usuario en una lista ordenada (rankeada) de documentos tal y como lo presentan los buscadores tradicionales.

7. **Visualizar y calificar resultados**: a través de una interfaz gráfica de usuario, el usuario del modelo califica los documentos como relevantes o no relevantes a las necesidades de información inicialmente definidas en la consulta.

8. **Modificar información del contexto e información de feedback para la comunidad**: a partir de la calificación que el usuario realiza de los resultados obtenidos, el sistema realiza la gestión de la información de contexto del usuario y de la comunidad. Con esta información de contexto actualizada se afecta el proceso de expansión, filtrado y ordenado de las futuras consultas.

5.1 Modelo propuesto en detalle

Una de las estructuras básicas para entender el modelo en detalle, es la forma como se almacena el **contexto del usuario**. Este contexto se almacena en una matriz triangular superior que contiene los términos usados por cada usuario y la relación que tienen cada uno de los términos entre ellos, de ahora en adelante denominada la Matriz de co-ocurrencia S (ver Figura 3). De igual forma la comunidad de usuarios cuenta con una matriz de co-ocurrencia de términos, denominada S_c.

	t_1	t_2	…	t_K
t_1	1	0,6	0,7	0,5
t_2	0	1	0,8	0,93
…	0	0	1	0,3
t_K	0	0	0	1

Figura 3. *Matriz de co-ocurrencia S con información del contexto del usuario*

Los pasos 1 a 3 fueron descritos en el modelo general, en este apartado se profundiza en los pasos subsiguientes. En el proceso de **expansión de la consulta (**paso 4**)** se buscan los términos correlacionados utilizados por

el usuario y otros usuarios de la comunidad teniendo en cuenta el parámetro numero_Terminos, que indica la cantidad de términos que se desean obtener para mostrar en la lista de autocompletar, actualmente este parámetro se ha definido como 10 (términos), teniendo en cuenta que ésta es la cantidad comúnmente usada en la lista desplegable de autocompletar de los buscadores tradicionales, pero dicho valor puede ser modificado de acuerdo a los requisitos del usuario.

En la Figura 4 se muestra un esquema del proceso: Después de hacer el pre-procesamiento de la consulta se hace la revisión del contexto del usuario (paso 2 de la Figura 4), como se detalla a continuación:

1. La expansión inicial de la consulta se hace a partir de la matriz de co-ocurrencia S del usuario, donde se obtiene la *intersección* de los términos de mayor co-ocurrencia de la información contextual del *usuario* con los términos ingresados por el usuario en la consulta actual.
2. Si la intersección es vacía o si aún no se alcanza la cantidad de términos establecidos en el valor del parámetro numero_Terminos, se hace una segunda expansión a partir de la matriz de co-ocurrencia S_c de la comunidad, donde se obtiene la *intersección* de los términos de mayor co-ocurrencia de la *comunidad* con los términos ingresados por el usuario en la consulta actual. Se debe validar si aún no se alcanza el valor del parámetro numero_ Terminos.

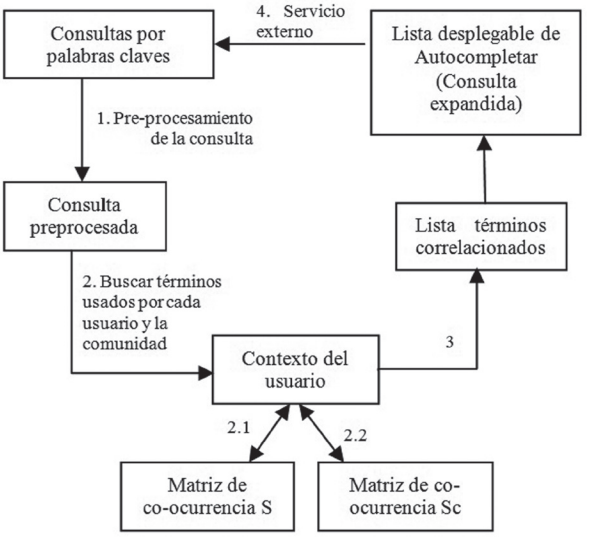


Figura 4. *Proceso de expansión de consulta*

3. Se procede a una tercera expansión a partir de la matriz de co-ocurrencia S del usuario, donde se obtiene la *unión* de los términos de mayor co-ocurrencia de la información contextual del *usuario* con los términos ingresados por el usuario en la consulta actual.
4. Si aún no se alcanza el valor del parámetro numero_ Terminos, se hace una cuarta expansión, a partir de la matriz de co-ocurrencia S_c de la comunidad, donde se obtiene la *unión* de los términos de mayor co-ocurrencia de la *comunidad* con los términos ingresados por el usuario en la consulta actual.
5. En el caso que no se obtengan términos después de aplicar los pasos 1 a 4. Por ejemplo, para las consultas realizadas antes de la primera calificación o cuando el usuario ingresa por primera vez y no tiene términos almacenados en su contexto, se procede a usar un servicio externo para realizar el proceso de expansión tradicional usado por motores como Google o Bing.

Cabe resaltar que en los procesos de intersección se debe actualizar la correlación, calculando el promedio de las co-ocurrencias de los términos comunes. De igual forma, cuando se realiza unión sobre las listas de términos correlacionados, se debe calcular el promedio de las correlaciones y luego se deben eliminar los términos repetidos. Posteriormente, se ordena la lista descendentemente (por correlación) y se concatena con los términos o palabras claves usadas originalmente por el usuario actual.

Las matrices S y S_c se registran en la medida en que los usuarios califican los documentos retornados por el sistema. Esta situación implica que para las consultas realizadas antes de la primera calificación, las matrices S y S_c pueden estar vacías ($S = \emptyset$, $S_c = \emptyset$), en este caso los pasos mencionados (1 a 4) anteriormente no hacen expansión de consulta, dependiendo de la matriz que se encuentre vacía.

Finalmente, considerando que el usuario puede ignorar las opciones sugeridas de consulta expandida, el modelo realiza un proceso de expansión adicional que es transparente al usuario, agregando la cantidad de términos con mayor correlación que sean necesarios hasta obtener la consulta expandida final, con la cantidad de términos establecido en el parámetro num_ TerminosEImplicita (actualmente fijado a un valor de 5 términos) o que tenga una longitud menor o igual a 70 caracteres, según las restricciones en la longitud de las llamadas a las API's de los buscadores tradicionales.

Después de realizar el proceso de expansión de la consulta, se continúa con la **Recuperación de**

documentos de los buscadores tradicionales (paso 5) teniendo en cuenta que la consulta ahora está compuesta de las palabras claves digitadas directamente por el usuario, las seleccionadas de la lista de autocompletar y además las obtenidas en el proceso oculto de expansión de consulta.

El proceso de adquisición realiza en paralelo la recolección de los resultados en los diferentes buscadores tradicionales, por ejemplo: Google, Yahoo! y Bing (ver Figura 5). A medida que los resultados son retornados por los buscadores tradicionales se realiza el pre-procesamiento de las entradas, este proceso incluye: Remoción de caracteres especiales, conversión del texto a minúsculas, remoción de palabras vacías y stemming del documento. Del proceso de adquisición de datos se obtiene una colección de documentos representados por {Snippet, URL, titulo, motores donde se encontró el documento}.

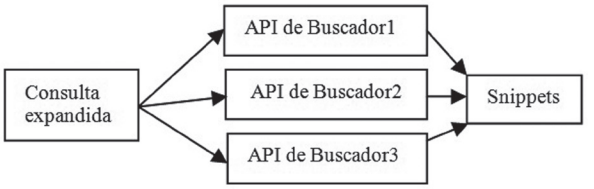


Figura 5. *Proceso de recuperación de documentos de los buscadores tradicionales*

Posteriormente se realiza el paso 6 correspondiente al **filtrado de información,** el sistema filtra los resultados, ubicando en las primeras posiciones los documentos de mayor relevancia para el usuario. Aquí los resultados se ordenan teniendo en cuenta el filtrado colaborativo basado en ítems, donde se buscan los términos correlacionados utilizados por el usuario y otros usuarios de la comunidad (Términos consultados anteriormente junto con los términos de consulta actual), además del contexto del usuario, basado en la propuesta [5] explicada en la sección 3 de este artículo.

En este paso se tiene en cuenta la creación de una matriz triangular superior auxiliar, denominadas Matriz Scruda del usuario (Figura 6), donde se almacena los términos usados por cada usuario y el valor que representa el número de veces que los termino y se encuentran juntos en los documentos. De manera similar, se cuenta con una Matriz Scruda para la comunidad que almacena los términos usados por todos los usuarios de la comunidad y el número de veces que los términos están juntos, éstas matrices son actualizadas en el momento que los usuarios realizan el proceso de calificación de documentos.

	t_1	t_2	…	t_K
t_1	1	7	2	5
t_2	0	1	3	9
…	0	0	1	6
t_K	0	0	0	1

Figura 6. *Matriz Scruda del usuario*

El proceso que se realiza para filtrar los documentos es el siguiente:

1. Se toman los documentos recuperados de los buscadores tradicionales y se indexan en memoria, se lee el snippet de cada documento recuperado y se llena una matriz auxiliar de Documentos donde se almacena: URL, título, texto, texto revisado, texto sin palabras vacías, términos, posición en Google, posición en Yahoo!, posición en Bing y orden (que se calcula en el siguiente paso).

2. Se calcula la similitud de cosenos frente a la consulta y se almacena en el campo orden de la matriz Documentos, esto con el fin de poder iniciar el manejo del contexto.

3. Se obtiene el contexto actual del usuario, consultando sus términos relevantes que se encuentran en la matriz de co-ocurrencia S.

4. Se calcula la función de ranking, basado en la propuesta [5] para lo cual se obtienen los vectores que representan a la consulta y a los documentos, se calculan los proyectores de la matriz de co-ocurrencia S, posteriormente, se realizan los cálculos necesarios (obtener las transpuestas de los vectores y de las matrices y se realizan los productos de acuerdo al algoritmo presentado en la Figura 7) para finalmente, mostrar los documentos de acuerdo la relevancia que tengan para el usuario de mayor a menor.

Sea a el vector que representa la consulta
Sea P_B los proyectores de la matriz S
Sea $l(i)$ el conjunto de documentos
Sea d un documento en $l(i)$ representado por el vector b
Para cada documento $d \in l(i)$

$$P_d = a^T.(P_B{}^T.\ P_B\ ).b$$

Fin para
Ordenar documentos por P_d de mayor a menor
Presentar resultados al usuario

Figura 7. *Algoritmo de ranking*

Cabe aclarar, que el modelo del meta buscador Web propuesto es *mono temático*, es decir un meta buscador donde los usuarios tratan sobre un único tema, persiguen objetivos comunes, y se considera que tienen un contexto similar. Si se quisiera tener un modelo multi-temático, la matriz de co-ocurrencia S_c de la comunidad debería remplazarse por varias matrices, una por cada tema o usar un enfoque más tradicional de filtrado colaborativo, por ejemplo, usando un enfoque centrado en usuarios, en el cual se calcula los usuarios más similares al que está actualmente haciendo la consulta y los términos que aporta la comunidad, se toman únicamente de las matrices S de cada uno de ese conjunto reducido de usuarios.

Por otro lado, el meta buscador es *mono lenguaje***,** actualmente funciona específicamente para el idioma inglés, si se desea que el buscador sea multilenguaje el modelo debe manejar diccionarios de términos por idioma y controlar la presentación de los términos almacenados en la matriz S basado en el lenguaje que es predominante para el usuario.

Para calcular la función de ranking se debe tener en cuenta que tanto el documento como la consulta se representan en el mismo contexto, de este modo la función de ranking se define como $P_d = a^T . (P_B^T . P_B) . b$, que obtiene la probabilidad de que un documento haya sido generado por el contexto de la consulta [5]; donde *a* y *b* son coeficientes usados para combinar los vectores base del documento y de la consulta respectivamente. El vector *a* de la consulta tiene un si la palabra clave o término i aparece en la consulta y 0 en caso contrario, de manera similar el vector *b* del documento tiene un si la palabra clave o término i aparece en el documento y 0 en otro caso. P_B es el proyector del subespacio extendido por un conjunto de documentos relevantes, calculado utilizando la Descomposición de Cholesky o Singular Value Decomposition (SVD).

Luego, se continúa con **la visualización y calificación de los resultados** (paso 7): A continuación el usuario califica como relevantes o no relevantes los documentos desplegados. En seguida, se actualizan la matriz S del usuario actual y la matriz S_C de la comunidad, basada en la información de dos matrices auxiliares denominadas S^+ y C.

La matriz S^+ (ver Figura 8) contiene el número de veces que un término aparece en los documentos relevantes y el número total de veces que aparece en todos los documentos, además el número total de documentos relevantes y el número total de documentos calificados.

	t_1	t_2	t_3	…	t_F	Número de doc.
Doc. Relevantes						
Total documentos						

Figura 8. *Matriz S^+*

C es una matriz de términos por documentos (ver Figura 9), donde el documento es sólo la URL (identificador) y cada elemento *(i, j)* de la matriz almacena el valor es calculado según (1).

$$cfi,j = \begin{cases} 1, \text{si el termino aparece en documento} \\ 0, \text{de otro modo} \end{cases} \quad (1)$$

		t_1	t_2	…	t_j	…	t_F
	doc_1						
	doc_2						
documentos relevantes	…						
	doc_i				*cfi,j*		
	….						
	doc_R						
	doc_{R+1}						
documentos no relevantes	doc_{R+2}						
	doc_{R+3}						
	….						
	doc_M						

Figura 9. *Matriz C de términos por documentos*

Es importante mencionar que para calcular las matrices de co-ocurrencia de los usuarios, los documentos se segmentan utilizando ventanas de texto basado en la propuesta [5], que se usan para colocar los términos en su propio contexto y para relacionar estos términos con otros en el mismo contexto. El tamaño de esta ventana se estableció en 13, el cual viene dado por el término objetivo junto con 6 términos a la izquierda y otros 6 términos a la derecha. Este valor puede ser configurado de acuerdo al tamaño de los resúmenes de las páginas web.

La matriz de co-ocurrencia S se calcula inicialmente (cuando no existe para el usuario actual como se muestra en la Figura 10) multiplicando cada valor *cfi,j* de la matriz C por la importancia relativa del término en los documentos consultados por el usuario.

Sea S la matriz k x k de co-ocurrencia de términos
Sea D el conjunto de documentos recuperados
Para cada documento $d \in D$
  Para cada termino $t_i \in d$
    Sea V_i una ventana centrada alrededor de t_i
    Para cada termino t_j que pertenece a V_i
      Si ($n_i = 0$ *or* $n_j = 0$ *or*
        ($n_i + n_j - n_{ij} = 0$)) entonces
          $S_{ij}=0$
      Si no
        $$S_{ij} = \frac{n_{ij}}{n_i + n_j - n_{ij}} * \frac{r_i}{n_i} * \frac{r_j}{n_j}$$
      Fin si
      $S_{j,i} \leftarrow S_{i,j}$
    Fin para
  $S_{i,i} \leftarrow 1$
  Fin para
Fin para
Se calculan los proyectores de S basado en SVD

Figura 10. *Algoritmo para calcular la matriz de co-ocurrencia y los proyectores*

La co-ocurrencia de los términos se calcula basada en la fórmula (2).

$$S_{ij} = \frac{n_{ij}}{n_i + n_j - n_{ij}} * \frac{r_i}{n_i} * \frac{r_j}{n_j} \qquad (2)$$

Donde:

n_{ij} es el número de veces que el término i y el término j están juntos en los documentos del usuario (se obtiene de la matriz Scruda),
n_i es un número de veces que termino i aparece en los documentos del usuario
r_i: Son las apariciones relevantes del término i para el usuario
n_j es un número de veces que termino j aparece en los documentos del usuario
r_j: Son las apariciones relevantes del término j para el usuario

Las matrices propuestas permiten que el proceso de actualización de la matriz S, cuando llega una nueva evaluación positiva (relevante) o negativa de un documento se pueda realizar rápidamente, actualizando sólo los términos del documento que están en la consulta actual y en la matriz S (ver Figura 11). La matriz S+ se amplía con los nuevos términos del documento que se acaba de calificar, el documento se registra en la matriz C y progresivamente se actualiza la matriz S.

Sea S la matriz k x k de co-ocurrencia de términos
Sea *d* el documento recientemente evaluado
Para cada termino $t_i \in D$
  Para cada termino $t_i \in d$
    Sea V_i una ventana centrada alrededor de t_i
    Para cada termino t_j que pertenece a V_i
      Si ($n_i = 0$ *or* $n_j = 0$ *or*
        ($n_i + n_j - n_{ij} = 0$)) entonces
          $S_{ij}=0$
      Si no
        $$S_{ij} = \frac{n_{ij}}{n_i + n_j - n_{ij}} * \frac{r_i}{n_i} * \frac{r_j}{n_j}$$
      Fin si
      $S_{j,i} \leftarrow S_{i,j}$
    Fin para
Fin para

Figura 11. *Actualización de la matriz S*

Posteriormente, la matriz S_c de la comunidad, se actualiza calculando la correlación de los términos existentes en dichas matrices con los nuevos términos relevantes usados en la consulta actual, basada en la fórmula (2) pero teniendo en cuenta los datos de la comunidad

Finalmente, cabe destacar que el modelo del meta buscador Web tiene como aportes los siguientes: la integración del filtrado colaborativo con la información del contexto propuesta por Massimo Melucci y la actualización dinámica que se hace sobre las matrices S+, C, S y Sc, sin embargo el costo computacional del modelo propuesto es alto (crece de acuerdo al tamaño de las matrices).

5.2 Implementación del modelo

Para el sistema se definió una arquitectura que consta de tres capas (ver Figura 12) lógica de presentación, lógica de negocio y lógica de servicios, al utilizar este tipo de arquitectura se tiene como principales ventajas: la alta escalabilidad, la flexibilidad, la facilidad de construcción y la facilidad del mantenimiento del sistema.

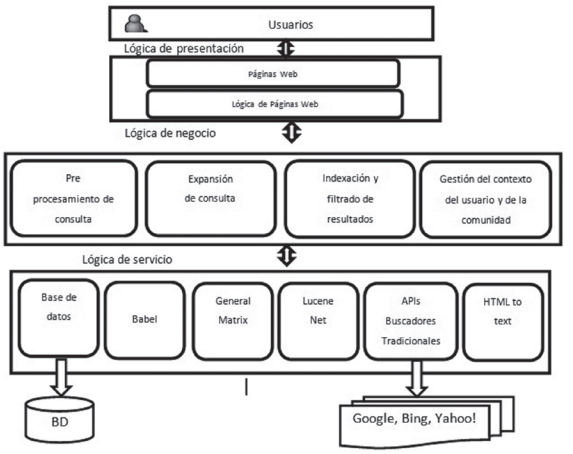


Figura 12. *Arquitectura del sistema*

La implementación del modelo se desarrolló utilizando una arquitectura multi-capa basada en servicios Web XML, en C# (C sharp) Visual Studio .NET 2010 y SQL Server Express como motor de base de datos. La Figura 13 y la Figura 14 muestran las vistas de clases implementadas en Visual Studio .Net 2010.

El modelo relacional de la base de datos se puede observar en la Figura 15 y una breve descripción de la misma en la Tabla 1. Todas las matrices del usuario y la comunidad son almacenadas en tablas, así:

- Matriz S+ = TotalDocUsuario + Termino_Usuario
- Matriz C = Documento_usuario + termino_ documento
- Matriz Scomunidad = TotalDocComunidad + Terminos
- Matriz Ccomunidad = Termino_Documentos + Documentos
-

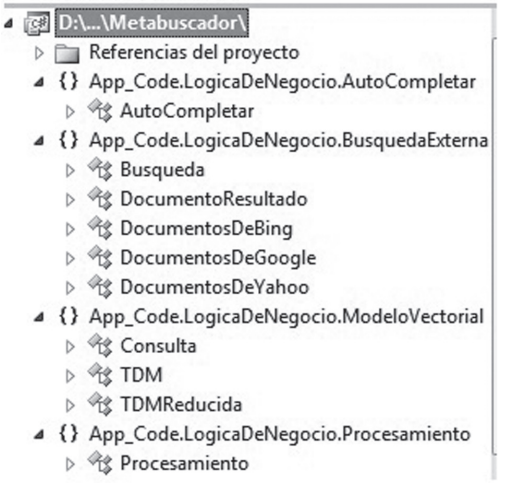


Figura 13. *Vista de clases de la lógica de Negocios en Visual Studio .NET*

Figura 14. *Vista de clases de la lógica de servicios en Visual Studio .NET*

En la Figura 16 a) se muestra la interfaz gráfica principal del meta buscador, después que el usuario ha realizado el proceso de registro e ingreso al sistema, se muestra la lista desplegable que contiene las diferentes opciones de consulta expandida, basada en las palabras claves digitadas inicialmente por el usuario. En Figura 16 b) se muestra la lista de resultados obtenidos que han sido filtrados de acuerdo al perfil del usuario y que están disponibles para el proceso de calificación por parte del mismo.

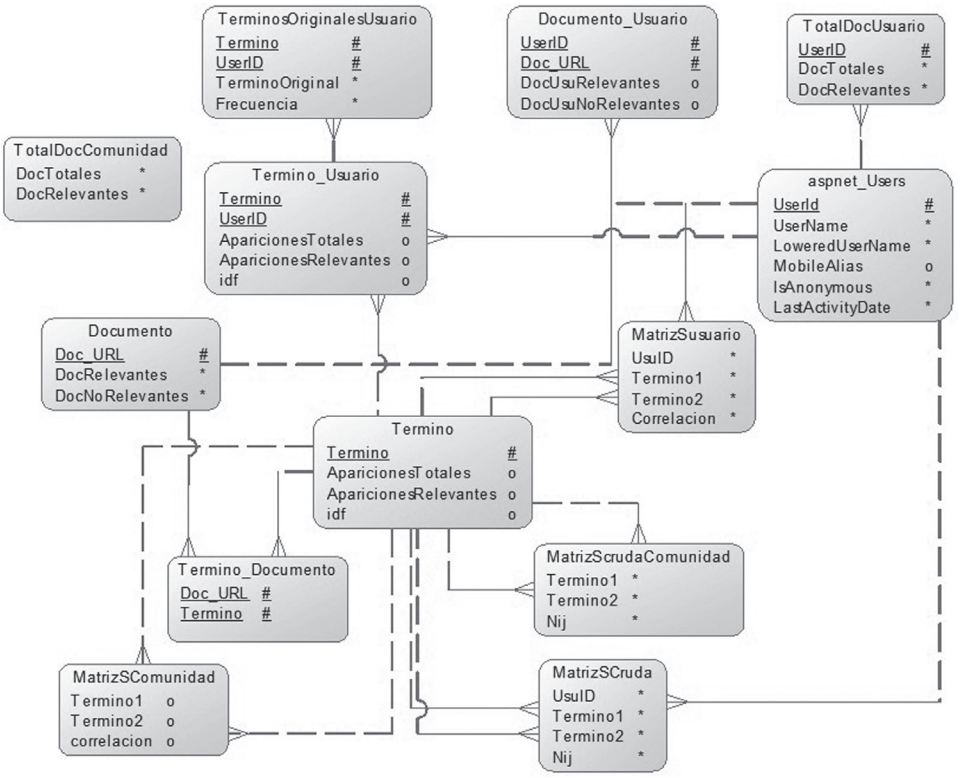
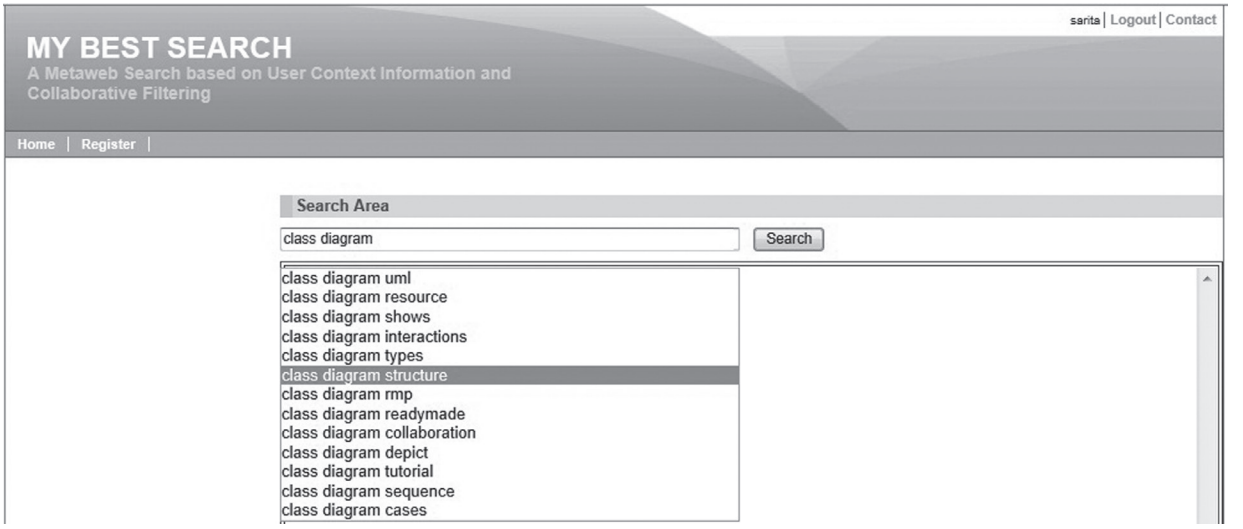


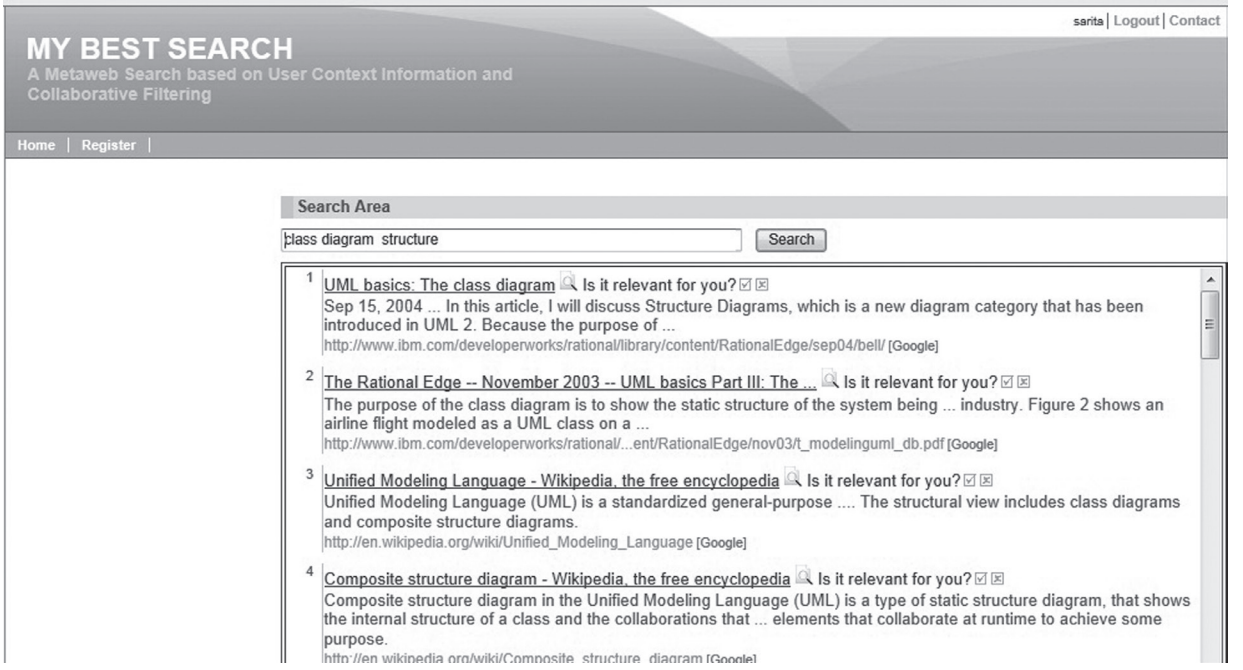
Figura 15. *Modelo de base de datos*

Tabla 1. *Descripción de las tablas de la base de datos*

TABLA	DESCRIPCIÓN
Documento	Almacena la información del documento para la comunidad
Documento_usuario	Almacena la información de los documentos evaluados por el usuario
TotalDocUsuario	Almacena la cantidad de documentos evaluados por el usuario y la cantidad de documentos que han sido relevantes para él
TotalDocComunidad	Almacena la cantidad de documentos evaluados por toda la comunidad de usuarios y la cantidad de documentos que se evaluaron como relevantes.
Termino	Almacena la información de los términos para la comunidad
Termino_documento	Almacena la relación de los términos que aparecen en cada documento de la comunidad
Termino_usuario	Almacena la información de los términos evaluados por un usuario
matrizScruda	Almacena el número de veces que un par de términos evaluados por un usuario se encuentran juntos
MatrizScrudaComunidad	Almacena el número de veces que un par de términos evaluados por la comunidad de usuarios se encuentran juntos
MatrizSusuario	Almacena el contexto del usuario, teniendo en cuenta que para controlar el tamaño de esta tabla sólo se guarda la relación de dos términos que superen un umbral de correlación, que se establece en un parámetro de la aplicación
MatrizScomunidad	Similar a la MatrizSusuario pero teniendo en cuenta los términos de toda la comunidad
Aspnet_User	Almacena la información básica de un usuario del sistema



a) Expansión de la consulta utilizando una lista desplegable



b) Lista de resultados obtenidos a partir de la consulta basada en palabras clave

Figura 16. *Interfaz grafica del meta buscador web*

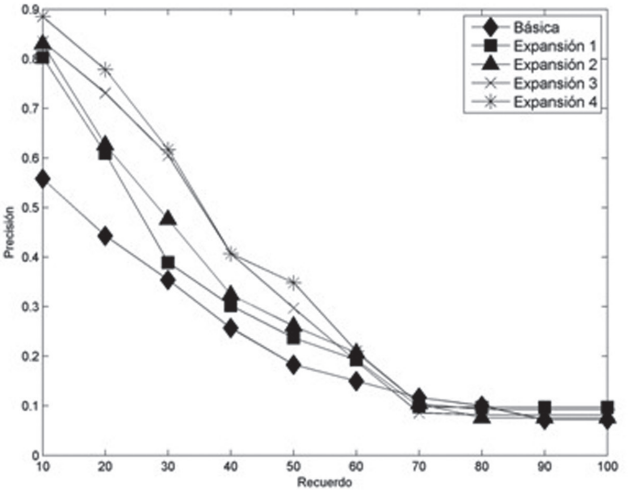


Figura 17. *Curva precisión-recuerdo Experimento 4*

Tabla 2. *Valores de precisión-recuerdo para el experimento 4.*

	Precisión				
Recuerdo	**Básica**	**Expansión 1**	**Expansión 2**	**Expansión 3**	**Expansión 4**
0,1	0,558	0,803	0,830	0,835	0,885
0,2	0,443	0,609	0,626	0,731	0,778
0,3	0,354	0,389	0,476	0,605	0,617
0,4	0,257	0,302	0,324	0,407	0,407
0,5	0,183	0,237	0,261	0,297	0,348
0,6	0,150	0,193	0,207	0,190	0,210
0,7	0,117	0,098	0,104	0,086	0,104
0,8	0,100	0,097	0,076	0,082	0,093
0,9	0,072	0,097	0,076	0,082	0,093
1	0,072	0,097	0,076	0,082	0,093

6. EVALUACIÓN Y RESULTADOS

En la evaluación del meta buscador web propuesto se aplicaron medidas clásicas del área de la recuperación de la información a saber: la curva de precisión-recuerdo, Mean Average Precision (MAP), precisión en K resultados ordenados y el estadístico Kappa (este último para evaluar la concordancia de los evaluadores). El proceso de evaluación se dividió en dos partes, la primera con el objetivo de obtener mejoras sobre el modelo del meta buscador, en relación a la precisión utilizando una colección cerrada de textos denominada CACM; y la segunda parte en relación con la precisión del modelo y su comparación con los resultados de búsqueda obtenidos por los buscadores tradicionales Google y Bing.

CACM es una colección de títulos y resúmenes de artículos publicados en la revista "Communications of the ACM". En la colección se encuentran 3204 documentos y 64 consultas. Para cada consulta, asesores humanos leyeron todos los documentos y evaluaron cuáles de ellos son relevantes. En la presente investigación se tomaron las 52 consultas que tenían completos los juicios de relevancia en la colección. Se realizaron un total de cuatro (4) experimentos, los cuales se hicieron con memoria de consulta, es decir, se simuló la ejecución de una consulta cinco veces, guardando el feedback de los resultados en el contexto del usuario.

Como resultado del análisis de cada experimento realizado con la colección cerrada CACM, se afinaron

algunos parámetros y se modificaron algunas rutinas propuestas inicialmente. A continuación se describe cómo se realizó el cuarto experimento: La primera ejecución denominada "Básica" usa la similitud de Lucene (una variante de la similitud de cosenos); la segunda una expansión de la consulta basada en la versión definitiva del modelo del meta buscador denominada "expansión 1"; luego se realizó una "expansión 2" con los juicios de relevancia de expansión 1 y de la misma forma se realizó una expansión 3 y una expansión 4. La Tabla 2 muestra los valores de precisión-recuerdo obtenidos en este experimento y la Figura 17 la curva precisión-recuerdo asociada.

Los resultados muestran mejoras consistentes con respecto a la consulta básica, el resultado de dicha consulta usando Lucene, inicia en un 55,8% de precisión para un nivel de recuerdo de 10%, y decrece hasta un 7,2% cuando el nivel de recuerdo es del 100%. Luego en la expansión 1, se muestra una mejora apreciable que comienza en 80,3% de precisión para un nivel de recuerdo de 10%, y decrece hasta un 9,7% cuando el nivel de recuerdo es del 100%. La Figura 17 muestra que las curvas de precisión-recuerdo en cada iteración, en general van mejorando o se conservan, en todos los niveles de recuerdo con respecto a la iteración anterior.

Por otra parte, se realizaron once (11) evaluaciones con usuarios finales (grupos de estudiantes de diferentes semestres del programa de Ingeniería de Sistemas de la Universidad del Cauca) donde participaron como mínimo 14 estudiantes en cada prueba, quienes fueron divididos en dos subgrupos con igual cantidad de personas. El primer subgrupo realizó evaluación de los buscadores tradicionales Google y Bing y el segundo subgrupo la evaluación del meta buscador propuesto disponible en (http://www.mybestmetawebsearch.com), a continuación se muestran los resultados de las dos últimas pruebas, teniendo en cuenta que todas se hicieron de la misma forma.

Para cada prueba se definieron 3 consultas, los estudiantes evaluaron los primeros 8 documentos recuperados con la primer consulta, como relevante (R), no relevante (N), e inaccesible (X) cuando el documento web no pudo ser visto por el usuario. Este mismo proceso se realizó para la segunda y tercer consulta. Las personas que evaluaron el meta buscador propuesto realizaron 3 iteraciones de búsqueda y para cada iteración se evaluaron los primeros 8 documentos, es decir, se realizó la primera consulta (primera iteración) y se evaluaron los primeros 8 documentos, luego se repitió la consulta (segunda iteración) y se realizó nuevamente la evaluación de los primeros 8 documentos, esa misma consulta se digitó por tercera vez (tercera iteración) y se realizó nuevamente la evaluación de los primeros 8 documentos, lo mismo se hizo para la segunda y tercera consulta. En la Tabla 3 se muestran los resultados obtenidos en la evaluación 1.

Tabla 3. *Evaluación 1- Comparación de la precisión media frente a buscadores tradicionales*

Precisión media (MAP)			
Consulta	**Google**	**Bing**	**MyBest Search**
1	69,7%	57,9%	67,2%
2	56,0%	71,7%	63,0%
3	59,3%	63,9%	64,5%

De los resultados obtenidos en la evaluación 1 se puede notar que MybestMetaWebSearch reporta para la primera consulta una precisión media menor 2,5% que Google, pero 9,3% por encima de Bing. Para la segunda consulta la precisión media de MybestMetaWebSearch es un 7% mejor que Google, y 8,7% menor que Bing y finalmente, para la tercera consulta la precisión media de MybestMetaWebSearch está por encima un 5,2 % contra Google y 0,6% contra Bing. Destacándose que los resultados de MybestMetaWebSearch tienen una menor varianza.

Tabla 4. *Evaluación 2- Comparación de la precisión media frente a buscadores tradicionales*

Precisión media (MAP)			
Consulta	**Google**	**Bing**	**MyBest Search**
1	91,6%	37,9%	74,2%
2	76,2%	78,4%	64,9%
3	83,2%	39,4%	69,1%

De los resultados obtenidos en la evaluación 2 se puede notar que MybestMetaWebSearch reporta resultados menos relevantes que los reportados por Google en todos los casos. Por otro lado, los resultados obtenidos en la primera y en la tercera consulta son más relevantes que los reportados por Bing como se puede observar en la Tabla 4. Después de analizar los resultados obtenidos se logró observar una fuerte incidencia en los resultados del modelo propuesto y la calidad de los snippets. Si los snipets son de mala calidad la información de contexto que se almacena no es la apropiada y de esta forma no se puede mejor la relevancia de los resultados que se presentan al usuario.

7. CONCLUSIONES Y TRABAJO FUTURO

El modelo del meta buscador Web propuesto representa una posible solución al problema planteado inicialmente de sobrecarga de información y la baja relevancia de los resultados obtenidos por los buscadores web tradicionales. Este modelo usa explícitamente dos estrategias: la adecuada gestión del contexto del usuario basado en la propuesta "A Basis for Information Retrieval in Context" [5], y la retroalimentación que el usuario puede registrar explícitamente al Sistema de Recuperación de Información (o búsqueda Web) usando técnicas de filtrado colaborativo basada en ítems.

La aplicación Web denominada MyBestMetaWebSearch construida con base en el modelo del meta buscador propuesto, permite fltrar y re-ordenar los resultados entregados por los buscadores tradicionales (Google y Bing) de acuerdo al contexto del usuario y de la comunidad, presentando en algunos casos resultados más precisos y relevantes a las necesidades de información de los usuario.

La evaluación del modelo propuesto se realizó utilizando medidas clásicas del área de la recuperación de la información, satisfacción del usuario y relevancia, a través de la Curva de Precision-Recuerdo, Mean Average Precision (MAP) y Precisión en K resultados ordenados. Los resultados de la evaluación son prometedores pero el costo computacional para mantener actualizado el contexto del usuario es de orden $O(n^2)$.

El proceso de evaluación del modelo se dividió en dos partes:

En la primera se utilizó una colección cerrada de textos denominada CACM, con el fin de calcular la Curva de Precision-Recuerdo. Los 4 experimentos realizados se hicieron con memoria de consulta, es decir, se simuló la ejecución de una consulta cinco veces, guardando el feedback de los resultados en el contexto del usuario. Los resultados obtenidos en cada experimento permitieron realizar la afinación de algunos parámetros y de algunas rutinas hasta obtener el modelo finalmente expuesto.

En la segunda parte, se calcularon las medidas Mean Average Precision (MAP), Precisión en K resultados ordenados y el estadístico Kappa, para lo cual se realizaron once (11) evaluaciones con estudiantes de diferentes semestres del Programa de Ingeniería de Sistemas de la Universidad del Cauca, en este documento sólo se muestran las dos últimas evaluaciones realizadas sobre MyBestMetaWebSearch, teniendo en cuenta que todas se hicieron de la misma forma. Las evaluaciones mostraron que los 8 primeros resultados entregados por el modelo propuesto en algunas ocasiones son mejores que los entregados por los buscadores Web tradicionales más usados hoy en día, Google y Bing, aunque dicha mejora no es significativa.

Por otro lado, la calidad de los snippets obtenidos (resumen de un documento que le permite al usuario entender si dicho documento es relevante sin acceder a él) no es la apropiada, esto tiene dos implicaciones sobre el modelo propuesto: (i) los términos que se ponderan no necesariamente están relacionados con el verdadero contenido de los documentos y por ello el contexto no refleja la verdadera relación entre los documentos relevantes o no, y el usuario o la comunidad, (ii) los procesos de expansión de consulta se basan en los términos almacenados en el contexto del usuario y de la comunidad, por lo tanto al no tener los términos adecuados, se puede desviar el objetivo de la consulta original y obtener resultados que nada tienen que ver con la necesidad de información del usuario.

Como trabajo futuro se plantea utilizar servicios externos de generación automática de resúmenes, con el fin de obtener snippets más confiables o de mejor calidad que los reportados actualmente por Google y Bing; de esta manera se tendrían los términos que realmente representan el contexto del usuario y se podrían manejar matrices de correlación reducidas, con el fin de mejorar el rendimiento de los procesos de creación y actualización de las matrices S^+, C, S y S_c, lo que a su vez se vería reflejado en mejoras en los procesos de expansión de las consultas y ranking.

Por otro lado se espera que la implementación del modelo del meta buscador sea multilenguaje (extender su uso otros lenguajes) y que se pueda manejar múltiples ejes temáticos o contextos de la comunidad diferentes. Finalmente, realizar una comparación del modelo propuesto contra el propuesto en [35].

8. AGRADECIMIENTOS

El trabajo en este artículo fue soportado por el Grupo de I+D en Tecnologías de la Información de la Universidad del Cauca bajo el proyecto VRI-2560 y el Grupo de I+D en Sistemas y Tecnologías de la Información (STI) de la Universidad Industrial de Santander.

9. BIBLIOGRAFÍA

[1] R. Baeza-Yates and B. Ribeiro-Neto, *Modern information retrieval*. Addison-Wesley Longman Publishing Co., Inc., 1999, p. 513.

[2] C. Manning, P. Raghavan, and H. Schütze, "An Introduction to Information Retrieval." Cambridge University Press, Cambridge, England, 2007.

[3] C. J. V. Rijsbergen, *Information Retrieval*. Butterworth-Heinemann, 1979, p.208.

[4] M. Melucci, "Exploring a mechanics for context-aware information retrieval," *In Proceedings of the AAAI Spring Symposium on Quantum Interaction. AAAI Press*, 2007.

[5] M. Melucci, "A basis for information retrieval in context," *ACM Transactions on Information Systems (TOIS)*, vol. 26, no. 3, pp. 1–41, 2008.

[6] J. Nielsen, "When search engines become answer engines," *Jakob Nielsen's Alertbox*, pp. 1–5, 2004.

[7] K. O'hara and N. Shabdolt, "Knowledge Technologies and the semantic web," 2004. [Online]. Available: http://eprints.ecs.soton.ac.uk/12469/.

[8] D. Sullivan, "Nielsen NetRatings search engine ratings," *Search Engine Watch*, 2006.

[9] R. Baeza-Yates, C. Castillo, and B. Keith, "Web Searching," in *Encyclopedia of Language & Linguistics*, Oxford: Elsevier, 2006, pp. 527–538.

[10] M. Melucci, "Context modeling and discovery using vector space bases," *In Proceedings of the AAAI Spring Symposium on Quantum Interaction. AAAI Press*, pp. 808–815, 2005.

[11] S. Liaw and H. Huang, "Information retrieval from the World Wide Web: a user-focused approach based on individual experience with search engines," *Computers in human behavior*, vol. 22, no. 3, pp. 501–517, 2006.

[12] Y. Liu and C. Li, "A query expansion algorithm based on phrases semantic similarity," *Information Processing (ISIP), ...*, 2008.

[13] J. Rocchio, "Relevance feedback in information retrieval," *Englewood Cliffs, NJ: Prentice Hall.*, pp. 313–323, 1971.

[14] Y. Liu, C. Li, P. Zhang, and Z. Xiong, "A query expansion algorithm based on phrases semantic similarity," *Proceedings of the 2008 International Symposiums on Information Processing*, 2008.

[15] S. Robertson and K. Jones, "Relevance weighting of search terms," in *Document retrieval systems*, Taylor Graham Publishing, 1988, pp. 143–160.

[16] E. Garcia, "RSJ-PM Tutorial: A Tutorial on the Robertson-Sparck Jones Probabilistic Model for Information Retrieval," 2009.

[17] E. N. Efthimiadis, "Query Expansion," *In: Martha E. Williams (ed.), Annual Review of Information Systems and Technology (ARIST)*, vol. 31, pp. 121–187.

[18] I. G. Kalmanovich and O. Kurland, "Cluster-based query expansion," *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pp. 646–647, 2009.

[19] A. Abdelali, J. Cowie, and H. S. Soliman, "Improving query precision using semantic expansion," *Inf. Process. Manage.*, vol. 43, no. 3, pp. 705–716, 2007.

[20] Claudio Biancalana and Alessandro Micarelli, "Social tagging in query expansion: A new way for personalized web search," *Proceedings of the 2009 International Conference on Computational Science and Engineering - Volume 04*, 2009.

[21] M. Bertier and R. Guerraoui, "Toward personalized query expansion," *Proceedings of the Second ACM EuroSys Workshop on Social Network Systems*, 2009.

[22] Z. D. and W. Liqing, "Study on Key Techniques of Query Expansion based on Ontology and Its Application," in *Computational Intelligence and Software Engineering, 2009. CiSE 2009. International Conference on*, 2009.

[23] T. Nguyen and T. Phan, "An ontology-based approach of query expansion," *Proceedings of the 9th International ...*, 2007.

[24] N. A. Segura, "An empirical analysis of ontology-based query expansion for learning resource searches using MERLOT and the Gene ontology," *Knowledge-Based Systems*, vol. 24, no. 1, pp. 119–133, Feb. 2011.

[25] L. Han and G. Chen, "HQE: A hybrid method for query expansion," *Expert Systems with Applications*, vol. 36, pp. 7985–7991, 2009.

[26] M. Rahman, "A query expansion framework in image retrieval domain based on local and global analysis," *Information Processing & Management*, vol. 47, no. 5, pp. 676–691, 2011.

[27] L. Jong-Seok and O. Sigurdur, "Two-way cooperative prediction for collaborative filtering recommendations," *Expert Systems with Applications*, vol. 36, no. 3, pp. 5353–5361, 2009.

[28] Amazon, "Sitio web de Amazon." [Online]. Available: http://www.amazon.com/.

[29] G. Linden, "Amazon. com recommendations: Item-to-item collaborative filtering," *Internet Computing, IEEE*, vol. 7, no. 1, pp. 76–80, 2003.

[30] B. Sarwar and G. Karypis, "Item-based collaborative filtering recommendation algorithms," *Proceedings of the 10th international conference on World Wide Web*, 2001.

[31] B. Marlin, "Collaborative filtering: A machine learning perspective," University of Toronto, 2004.

[32]V. Schickel-Zuber, "Ontology filtering," ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE, Suisse, 2007.

[33] J. A. Konstan, J. Riedl, A. Borchers, and J. L. Herlocker, "Recommender systems: A grouplens perspective," in *Recommender Systems: Papers from the 1998 Workshop (AAAI Technical Report WS-00-04)*, 1998, pp. 60–64.

[34] P. Heymann, G. Koutrika, and H. Garcia-Molina, "Can social bookmarking improve web search?," *Proceedings of the international conference on Web search and web data mining*, 2008.

[35] C. Cobos, E. Estevez, M. Mendoza, L. Gomez, and E. León, "Algoritmos de expansión de consulta basados en una nueva función discreta de relevancia," *Revista UIS Ingenierías*, vol. 10, no. 1, pp. 9–22, 2012.