# Algoritmos de expansión de consulta basados en una nueva función discreta de relevancia

***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*

***EDUARDO ESTEVEZ MENDOZA***
*Estudiante de Ingeniería de Sistemas*
*Programa de Ingeniería de Sistemas, Escuela de Ingeniería de Sistemas e Informática*
*Miembro del Grupo de I+D en Sistemas y Tecnologías de la Información, Universidad Industrial de Santander*
*eestevez25@hotmail.com*
*Bucaramanga, Santander, Colombia*

***MARTHA ELIANA MENDOZA BECERRA***
*Ingeniera de Sistemas, Magíster en Informática, Estudiante de Doctorado en Ingeniería de Sistemas y Computación*
*Profesora Titular, Departamento de Sistemas, Facultad de Ingeniería Electrónica y Telecomunicaciones*
*Miembro del Grupo de I+D en Tecnologías de la Información, Universidad del Cauca*
*mmendoza@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*

***ELIZABETH LEÓN GUZMÁN***
*Ingeniera de Sistemas, Magíster en Ingeniería de Sistemas, Ph.D. in Computer Science and Computer Engineering*
*Profesora Asistente, Departamento de Ingeniería de Sistemas e Industrial, Facultad de Ingeniería*
*Directora del Grupo de Investigación en Minería de Datos, Universidad Nacional de Colombia*
*eleonguz@unal.edu.co*
*Bogotá D.C., Colombia*

*Fecha de recibido:* 31/03/2011
*Fecha de aprobación:* 15/06/2011

UIS Ingenierías
REVISTA DE LA FACULTAD DE INGENIERÍAS FISICOMECÁNICAS



**RESUMEN**

Se ha demostrado que el proceso de expansión de las consultas en el modelo espacio vectorial de representación de documentos en un sistema de recuperación de información, es una técnica útil para mejorar la relevancia medida por la precisión de los resultados entregados a los usuarios. En este artículo se presenta un nuevo algoritmo y una variación del mismo para realizar expansión de consultas en un sistema de recuperación de información. Estos algoritmos se basan en una nueva función discreta que define la importancia relativa de un término en una colección de documentos. El algoritmo y su variación se evalúan frente a la búsqueda por similitud de cosenos y el algoritmo de expansión propuesto por Rocchio, obteniendo excelentes resultados sobre la colección de datos CACM (artículos publicados en la revista Communications of the ACM).

**PALABRAS CLAVE**: Expansión de consulta, Rocchio, Término relevante, IDF, Frecuencia invertida de documento.

**ABSTRACT**

It has been shown that the query expansion process in the vector space model of document's representation in a retrieval system, it is a useful technique for improving the relevance measured by precision of the results delivered to users. This paper presents a new algorithm and a variation of itself used to perform query expansion in information retrieval systems. These algorithms are based on a new discrete function that defines the relative importance of a term in a document collection. The algorithm and its variation were evaluated against the cosine similarity search and the query expansion algorithm proposed by Rocchio, with excellent results on data collection CACM (articles published in the Communications of the ACM journal).

**KEYWORDS:** Query expansion, Rocchio, Relevant Term, IDF, Inverse document frequency.

## 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]. Para entender mejor esta definición, es necesario pensar en ítems de información como documentos (normalmente no estructurados) que están relacionados con las solicitudes de búsqueda de un usuario [2].

La recuperación de información ha tomado gran importancia desde 1940, y con el creciente uso de las computadoras se creó la posibilidad de manejar automáticamente grandes volúmenes de información, como por ejemplo las librerías digitales y los buscadores web. Un sistema de recuperación de información (SRI) está compuesto básicamente por: Documentos (almacenados en bases de datos o directorios), Usuarios, Consultas (solicitudes), 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]. La Figura 1 muestra los componentes de un SRI.


**Figura 1.** *Componentes de un SRI. Adaptado de [1]*

Los SRI ofrecen al usuario la posibilidad de realizar búsquedas sobre grandes cantidades de documentos teniendo en cuenta: concordancias parciales o las mejores concordancias frente a una solicitud de información, un mecanismo de inferencia basado en la inducción, un modelo de búsqueda probabilístico, la posibilidad de clasificar los documentos en múltiples temas, el uso de un lenguaje de consulta similar al natural implicando condiciones de consultas que son incompletas (es decir, un conjunto términos que no necesariamente representan en su totalidad la necesidad de información del usuario), y un despliegue de documentos ordenados por relevancia y con una alta posibilidad de equivocarse en el orden de presentación de dichos documentos [1, 3].

En la actualidad existen varios modelos de representación de los documentos en un SRI, los más destacados [1, 3] son: el modelo booleano, el modelo de espacio vectorial y el modelo probabilístico. Existen algunas variaciones de estos tres modelos, entre las más importantes están: el modelo de conjuntos difusos, el modelo booleano extendido, el modelo del espacio vectorial generalizado, el modelo de indexado de semántica latente, el modelo de redes neuronales, el modelo de las redes bayesianas, el modelo de las redes de inferencia, el modelo de red de creencias, entre otros.

El modelo Espacio Vectorial [1, 4] (VSM por sus siglas en inglés, Vector Space Model) es el que en general reporta mejores niveles de relevancia en los resultados que se le presentan a los usuarios. En este modelo se conciben los documentos como bolsas de palabras y la colección de documentos se representa con una matriz de M-términos por N-documentos. Cada documento se representa como un vector fila d en el espacio de términos tal que d= $\{w_1, w_2, \ldots, w_M\}$ (ver Figura 2), donde $w_{i,j}$ es igual a la frecuencia del término (conocido como TF) normalizado en la colección multiplicado por la inversa de la frecuencia del documento (conocido como IDF) para ese término, en lo que se conoce como el valor TF-IDF que se resume en la formula (1) o una variación de la misma.

$$w_{i,j} = \frac{frecuencia_{i,j}}{\max\left(frequecia_i\right)} \times \log\left(\frac{N}{1+n_j}\right) \quad \textbf{(1)}$$

| | $t_1$ | $t_2$ | ... | $t_j$ | ... | $t_F$ |
|---|---|---|---|---|---|---|
| $d_1$ | | | | | | |
| $d_2$ | | | | | | |
| ... | | | | | | |
| $d_i$ | | | | $W_{i,j}$ | | |
| ... | | | | | | |
| $d_N$ | | | | | | |

**Figura 2.** *Matriz de Términos por Documentos*

En este modelo de representación de documentos, se usa la distancia de cósenos para medir el grado de similitud entre dos documentos o entre un documento y la consulta del usuario, calculado por la formula (2) e ilustrado gráficamente en la Figura 3.

$$Sim(d,q) = Cos(\theta) = \frac{\sum_{i=1}^{M} W_{i,d} \times W_{i,q}}{\sqrt{\sum_{i=1}^{M} W^2{}_{i,d}} \sqrt{\sum_{i=1}^{M} W^2{}_{i,q}}} \quad \textbf{(2)}$$

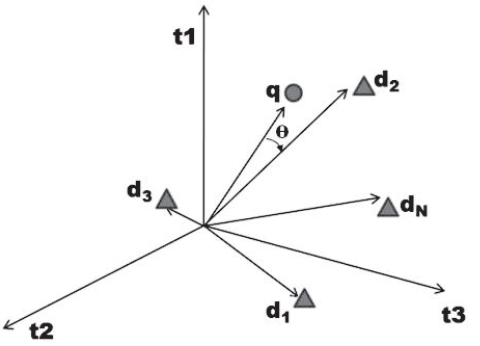



**Figura 3.** *Similitud entre documentos y consultas*

Adicionalmente, en este modelo, se ha demostrado que el proceso de expansión de las consultas es una técnica útil para mejorar la relevancia (medida por la precisión) de los resultados entregados a los usuarios [1, 2, 5]. En este artículo se presentan un algoritmo y una variación del mismo para la expansión de las consultas en un SRI, estos algoritmos obtienen mejores resultados que el algoritmo propuesto por Rocchio [1, 2, 5]. Los algoritmos, al igual que la propuesta de Rocchio se basan en el juicio de relevancia o no (re-alimentación) que da el usuario a los documentos que le entrega el SRI, cuando realiza una consulta.

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 en detalle el algoritmo de expansión de consulta basado en re-alimentación de relevancia del usuario, propuesto por Rocchio, que es la línea base para comparar los resultados de los algoritmos propuestos en esta investigación. Después, en la sección 4 se presentan en detalle los algoritmos de expansión de consulta propuestos, iniciando con la definición de la función de la importancia relativa de un término para un usuario y mostrando luego el detalle del algoritmo de expansión y su variación. En la sección 5 se presentan los resultados de la experimentación y finalmente en la sección 6 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

La expansión de la consulta en un sistema de búsqueda web normalmente se hace usando diferentes técnicas, entre ellas: Re-alimentación de relevancia del usuario (URF por sus siglas en inglés, User Relevance Feedback), re-alimentación automática de relevancia (ARF por sus siglas en inglés, Automatic Relevance

Feedback) [1, 2, 5], técnicas morfológicas que procesen los términos de la consulta y técnicas semánticas que encuentran términos similares a los digitados por el usuario.

URF 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, 5]. Rocchio propone la fórmula (3) para generar la consulta expandida. Donde $\vec{q}_i$ es la consulta inicialmente digitada por el usuario, R es un conjunto de documentos relevantes, R' es un conjunto de documentos no relevantes, α, β y γ son parámetros de afinación del algoritmo y $\vec{q}_e$ es la consulta expandida [1, 2, 5].

$$\vec{q}_e = \alpha \times \vec{q}_i + \frac{\beta}{|R|}\sum_{d \in R}\vec{d} - \frac{\gamma}{|R'|}\sum_{d \in R'}\vec{d} \quad \textbf{(3)}$$

En contraste con URF, la retroalimentación automática de relevancia, también conocida como pseudo retroalimentación (Pseudo Feedback) expande las consultas automáticamente basado en dos métodos, principalmente: documentos globales y documentos parciales [1, 2, 5]. En los métodos basados en documentos globales 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, 5]. Otras estrategias dependientes del dominio (o de la colección) pueden estar basadas en clusters o agrupaciones de términos [6] y en similitud de términos [7]. 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 [7]. Otros enfoques que son independientes del dominio o corpus de la colección, consisten en usar diccionarios o tesauros globales, tales como WordNet.

En los métodos basados en documentos parciales, 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 γ=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, 5]. Trabajos como los de Robertson y Sparck Jones [8, 9] que re-ponderan los términos de la consulta, o los de Dillon y Desper [7] que abandonan los términos del usuario y usan términos de los documentos inicialmente recuperados, son ejemplos de esta estrategia.

Nuevos enfoques incluyen entre otros: el etiquetado social (social tagging) [10, 11], 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 [12, 13], 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 [14].

En este artículo se hace una propuesta desde el enfoque de re-alimentación de relevancia del usuario, que es libre de parámetros, contrario a la propuesta de Rocchio. Además los algoritmos propuestos son computacionalmente menos costosos que Rocchio, haciéndolos una opción viable para las aplicaciones donde se recupera información usando el proceso de expansión de consulta.

## 3. ALGORITMO DE ROCCHIO

El algoritmo de Rocchio que se resume en la fórmula (3), se destacan varios elementos, el primero de ellos un conjunto de documentos evaluados como relevantes, un conjunto de documentos evaluados como no relevantes y una consulta expresada como vector de términos con pesos (no como la cadena de texto que usualmente digitan los usuarios en los sistemas de recuperación de información). Los conjuntos de documentos evaluados como relevantes y no relevantes también se deben expresar como vectores de términos con pesos.

En la práctica, el algoritmo no registra todos los documentos que han sido relevantes y no relevantes para cada usuario del sistema, en lugar de ello, el perfil almacena: un vector de términos representativo del documento relevante promedio, el total de documentos que han sido relevantes (|R|), un vector de términos representativo del documento no relevante promedio y el número total de documentos que han sido evaluados como no relevantes (|R|). En cada celda de los vectores se almacena el valor TF-IDF promedio de todos los documentos del conjunto, según

la fórmula (1) o una de sus variaciones. Por ejemplo, en Lucene (framework vectorial para el desarrollo de aplicaciones de recuperación de información) el valor que se almacena es la frecuencia observada de cada término en el documento y el valor IDF (inverse document frecuency) es igual a $1 + \log(\frac{N}{ni+1})$, donde N es el número de documentos en la colección y ni es el número de documentos en los que aparece el término.

Cuando se va a expandir una consulta, el texto inicial digitado por el usuario se convierte en un vector de términos similar a los vectores que representan a los documentos en el espacio multidimensional de términos y cada celda almacena el valor TF-IDF para los términos que digitó el usuario.

Luego, el vector de términos que representa la consulta del usuario se multiplica celda a celda por el valor α. Después, al resultado se le suma celda a celda (suma de vectores) el vector de términos representativos del documento relevante promedio previamente multiplicado por el parámetro β. Finalmente, al resultado se le suma celda a celda (suma de vectores) el vector de términos representativos del documento no relevante promedio previamente multiplicado por el parámetro γ. De esta forma la consulta textual digitada por el usuario se expande y el resultado es un vector de términos que representa los términos con sus pesos en el espacio multidimensional, el cual es comparado con los documentos mediante la similitud de cosenos de la formula (2), o una variación de la misma, para obtener el ranking de los documentos. Por ejemplo, en [15] se puede apreciar la medida de similitud usada por Lucene.

## 4. ALGORITMOS PROPUESTOS

Con base en un estudio detallado de los resultados de un algoritmo de recuperación de información tradicional (basado en TF-IDF y similitud de cosenos), el algoritmo de expansión de consultas propuesto por Rocchio y el análisis de las frecuencias de los términos en los diferentes conjuntos de documentos (relevantes y no relevantes) entregados al usuario, se definió una nueva función para el valor IDF de cada término en el perfil de un usuario de un SRI o búsqueda web.

### 4. 1 Función IDF

Esta función IDF, ver fórmula (4), define la importancia de un término en relación con el número de documentos evaluados por el usuario (N), el número de documentos relevantes para el usuario (R), el número de documentos en los que aparece el termino i ($n_i$) y el número de documentos relevantes en los que aparece el termino i ($r_i$).

$$idf_i = \begin{cases} \frac{r_i}{N} \cdots & Si \quad n_i \le R \\ \frac{r_i * R}{n_i * N} \cdots & Si \quad n_i > R \end{cases} \qquad \textbf{(4)}$$

La función IDF propuesta en esta investigación, ver Figura 4, tiene un rango de valores discreto entre cero y uno [0,1]. Cero cuando el término no es relevante en absoluto y uno cuando es totalmente relevante. El grado de relevancia está en relación con el radio de documentos relevantes, es decir, si existen muchos documentos evaluados (por ejemplo en la gráfica, la serie de datos con marcador en forma de rectángulo, N=50) y de ellos el término aparece en sólo unos documentos (por ejemplo 10) y todos son relevantes, la función IDF alcanza un valor de 0.2, en contraste con un número menor de documentos (por ejemplo en la gráfica, la serie de datos con marcador en forma circular, N=10), donde obtendría un valor de 1.0 (valor máximo).

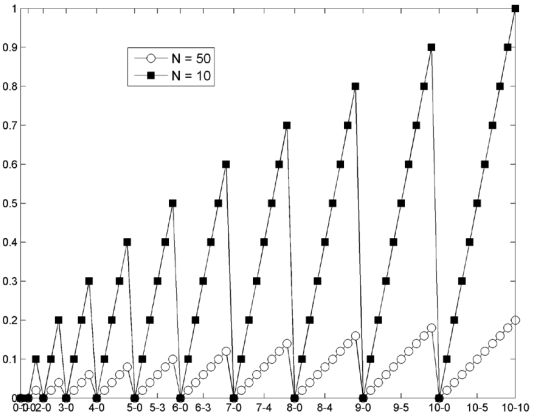



**Figura 4.** *Función IDF propuesta en la investigación*

En la Figura 4 el eje X muestra distintos valores de $n_i$ y $r_i$, iniciando con (0-0), pasando por ejemplo por (7,3) y finalizando con (10-10). En esta gráfica se muestran valores de $n_i$ y de $r_i$ entre 0 y 10. Para las dos series de datos, se logra el máximo cuando $n_i = r_i$, en este caso (10,10) y el mínimo cuando $r_i = 0$, sin importar el valor de $n_i$.

### 4.2 Algoritmo con vector ponderado (VP-IDF)

Con base en esta función IDF se planteó un algoritmo de expansión de consulta. Este algoritmo recibe como entrada la consulta del usuario y entrega como resultado una consulta expandida con pesos para cada uno de los términos contenidos en dicha consulta. Se parte del hecho, de que cada documento evaluado (relevante o no) por el usuario, modifica la función IDF y otros datos en el perfil del usuario.

El perfil del usuario está compuesto por los elementos que se muestran en la Figura 5, a saber: El número (N) de documentos que el usuario ha evaluado, el número (R) de documentos evaluados como relevantes y una ***lista de términos del usuario***, en la cual se registra por cada término que ha aparecido en los documentos que ha evaluado el usuario, el número ($n_i$) de veces que el término específico ha aparecido en los documentos evaluados, el número ($r_i$) de veces que el término ha aparecido en los documentos relevantes y el valor IDF para cada término.

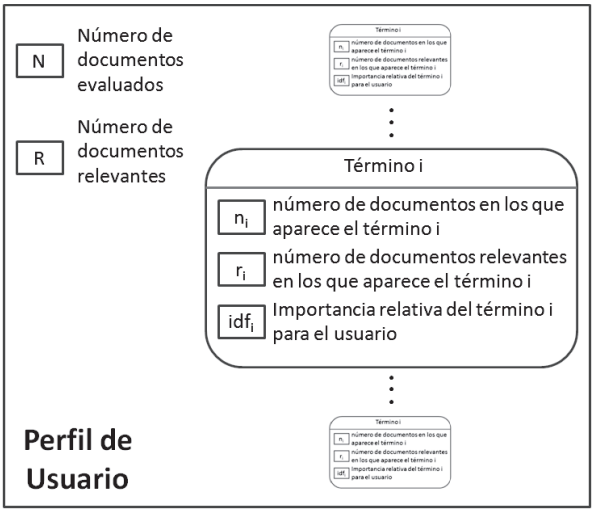



**Figura 5.** *Perfil de Usuario*

Cuando el usuario realiza una consulta ($q_{inicial}$) por palabras claves se realiza un primer paso de pre-procesamiento de dicha consulta, lo que implica: Tokenización (división de la cadena de consulta en términos individuales), eliminación de acentos y caracteres especiales, paso a minúsculas, eliminación de palabras vacías (basado en una lista de palabras vacías) y lematización (extracción de la raíz léxica del término) con un algoritmo como el propuesto por Porter [16]. Como resultado se obtiene la consulta inicial procesada. Por ejemplo, si la consulta inicial $q_{inicial}$ es igual a "What articles exist which deal with TSS (Time Sharing System), an operating system for IBM computers?", el resultado, $q_{inicial\text{-}procesada}$ es igual a "articl exist deal tss time share system oper system ibm comput".

La consulta inicial procesada, se expande con el algoritmo que se ha denominado, vector ponderado según IDF (VP-IDF) y que se explica a continuación:

1. Se consulta de la ***lista de términos del usuario*** en el perfil de usuario, todos los términos que cumplan con la siguiente condición: $r_i > 0.5 * n_i$. Es decir, todos los términos que hayan aparecido más número de veces en los documentos relevantes que en los documentos no relevantes. Una parte de la lista de términos que cumplen con este criterio para un usuario puede ser, LTC = {{"satisfactori", 1, 1, 0.25},{"tss", 1, 1, 0.25},{"sequenc", 1, 1, 0.111112}, …}. La lista de términos que cumplen con la condición previa se denomina ***lista de términos candidatos*** (LTC).

2. Se divide la cadena de la consulta inicial procesada ($q_{inicial\text{-}procesada}$) en tokens o términos independientes y por cada uno de ellos se hace lo siguiente:

   2.1. Si el término se encuentra en la LTC se modifican los valor del término en dicha lista de la siguiente forma: $n_i = n_i + 1$, $r_i = r_i + 1$ y se recalcula el valor IDF de ese término con los nuevos valores de $n_i$ y $r_i$. Estas operaciones aumentan la relevancia del término en la consulta expandida final, ya que el término está en el perfil y además está en la consulta inicial.

   2.2. De otro modo (es decir, si el término NO se encuentra en la LTC), se inserta un nuevo nodo en la LTC con el texto del término, un valor de uno (1) para $n_i$ y $r_i$, y se calcula el valor de IDF (con la función propuesta) para este nuevo término. Si el valor IDF es igual a cero (0), es decir no ha aparecido en ningún documento evaluado como relevante, se asigna el valor IDF general de la colección de datos. Si este nuevo IDF sigue siendo cero, el término

no se adiciona, ya que, es un término que no existe en la colección de documentos y por esta razón no sirve como término de búsqueda. De esta forma, la lista de términos candidatos consultada del perfil, se complementa con términos de consulta nuevos digitados por el usuario en la consulta específica.

3. Se recorre la lista de términos candidatos y se genera la consulta expandida, teniendo en cuenta la opción de boosting definida en Lucene, es decir, representando la consulta como un vector en el espacio multidimensional de términos. Cada término se anexa a una cadena de texto de salida usando el formato "termino^peso", donde término es cada uno de los términos de la lista de candidatos que supera el valor IDF y el peso es igual al valor IDF del término multiplicado por ni (el número de documentos en los que aparece el término). A continuación se presenta un ejemplo de una consulta expandida final o $q_{expandida}$ = "satisfactori^0.25… tss^0.2… sequenc^0.1111111 paper^0.3157895… articl^0.05 exist^0.05 deal^0.05 time^0.05 share^0.05 system^0.2 oper^0.05 ibm^0.05 comput^0.05".

VP-IDF al igual que Rocchio entrega como resultado un vector de términos ponderados, donde cada término tiene un peso en el espacio multidimensional de términos por documentos. A diferencia de Rocchio que usa todos los términos de la colección de documentos, en VP-IDF se tienen en cuenta sólo los que son más relevantes.

### 4.3 Algoritmo con cadena expandida (CE-IDF)

Teniendo en cuenta las características de la función de IDF previamente definida y buscando generar una consulta expandida compuesta solamente de términos (evitando el boosting) se diseñó una variante del algoritmo VP-IDF, el cual se ha denominado cadena expandida según IDF (CE-IDF), que opera sobre la consulta inicial procesada como sigue:

1. Se consulta de la ***lista de términos del usuario*** en el perfil de usuario, todos los términos que cumplan con la siguiente condición: $r_i > 0.5 * n_i$, formando la LTC. Este paso es igual al primer paso de VP-IDF.

2. La cadena de la consulta final expandida $q_{expandida}$ es igual a la cadena de la consulta inicial, $q_{inicial\text{-}procesada}$ concatenada con los términos de la LTC. En este algoritmo el resultado de la consulta del ejemplo sería igual a "articl exist deal tss time share system oper system ibm comput satisfactory… tss… sequenc… paper…". Nótese que los términos se repiten como en un documento de texto cualquiera.

## 5. EVALUACIÓN

Los SRI, al igual que cualquier sistema software, deben ser evaluados antes de iniciar su funcionamiento en el ambiente real de producción. Dicha evaluación contempla aspectos como: análisis de funcionalidad, de unidad, de integridad, de tolerancia a fallos y de rendimiento (tiempo de respuesta al usuario, espacio requerido para almacenamiento adicional de índices de búsqueda, la velocidad de los canales de comunicación, entre otros). Pero uno de los aspectos más importantes en la evaluación de los SRI es la precisión de la respuesta del sistema, conocida como la evaluación del rendimiento de la recuperación. Las medidas más conocidas y ampliamente usadas para realizar esta evaluación son la precisión y el recuerdo (exhaustividad) [1] y otras medidas derivadas de ellas.

La precisión corresponde a la fracción de los documentos recuperados por el sistema que realmente son relevantes para el usuario, formalmente definida por (5).

El recuerdo (exhaustividad) corresponde a la fracción de los documentos relevantes que han sido recuperados por el sistema, del total de documentos relevantes, formalmente definida por (6).

$$P = \frac{\left|\{doc\_relevantes\} \cap \{doc\_recuperados\}\right|}{\left|\{doc\_recuperados\}\right|} \quad \textbf{(5)}$$

$$R = \frac{\left|\{doc\_relevantes\} \cap \{doc\_recuperados\}\right|}{\left|\{doc\_relevantes\}\right|} \quad \textbf{(6)}$$

Como se aprecia, las medidas de precisión y recuerdo están basadas en conjuntos; con el paso del tiempo ellas fueron adaptadas para evaluar sistemas que muestran resultados en una lista ordenada de documentos (como la mayoría de buscadores web de hoy en día), donde se espera que los primeros documentos estén más relacionados (sean más relevantes) con las necesidades del usuario. Una de estas adecuaciones es la curva de precisión-recuerdo, que en forma grafica representa el valor de la precisión a diferentes niveles de recuerdo [1, 2, 17]. Esta curva permite comparar visualmente

el rendimiento de dos o más SRI. La Figura 6 muestra un ejemplo de un gráfico de una curva de precisión recuerdo. En ella se puede ver que el sistema presenta una precisión aproximada de 50% cuando obtiene el 10% de recuerdo (cuando ha recuperado el 10% del total de los documentos relevantes a la consulta del usuario). Además muestra que esta curva, en general, es descendente; es decir a mayor valor de recuerdo se obtienen menores valores de precisión.

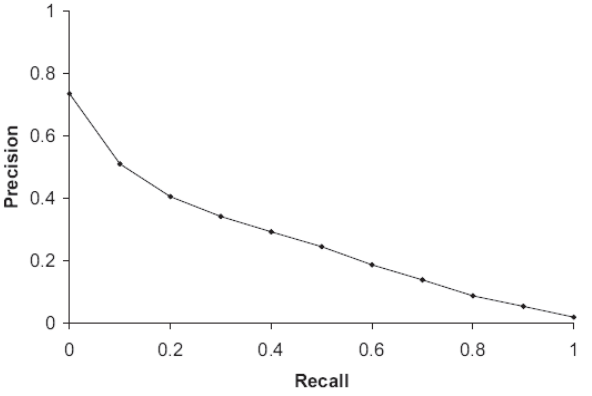



**Figura 6.** *Curva de Precisión-Recuerdo (Tomada de [2])*

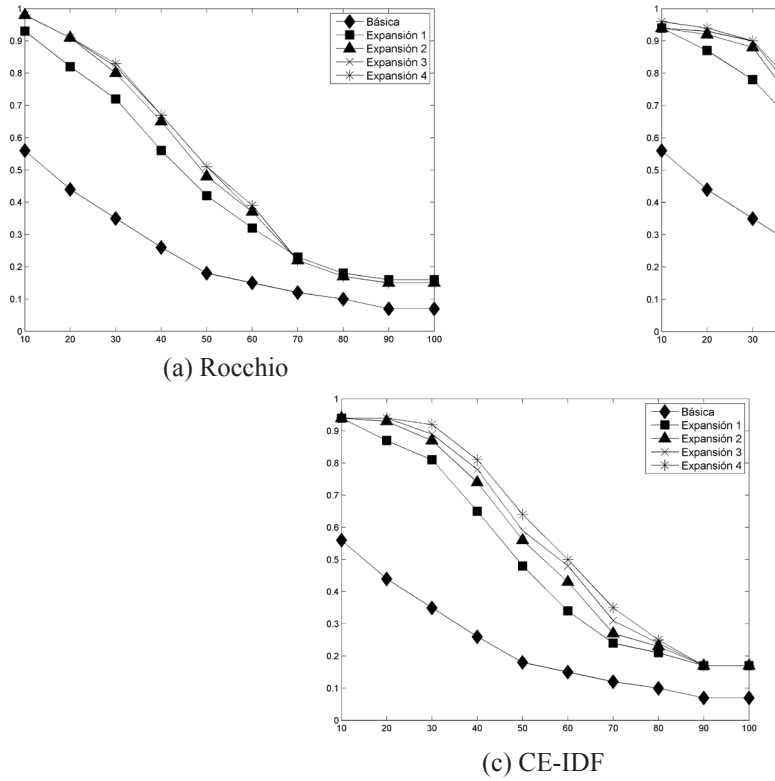
Con el objetivo de verificar el rendimiento de los algoritmos propuestos en el presente trabajo, se compararon los resultados de los mismos frente a la medida básica de ranking usada por Lucene (basada en similitud de cosenos) y el algoritmo de re-alimentación de relevancia del usuario propuesto por Rocchio [1, 2, 17]. Para este último se toman los siguientes valores para los parámetros de este algoritmo: $\alpha = 50\%$, $\beta = 50\%$ y $\gamma = 0\%$. Valores que reportaron los mejores resultados en dos de los tres experimentos realizados.

El conjunto de datos (dataset) usado para los experimentos fue la colección de textos CACM disponible en forma gratuita en [18] (Colecciones de prueba del Grupo de I+D en Recuperación de Información de la Universidad de Glasgow en Escocia, Reino Unido). Este conjunto de datos 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.

Con estos datos se simuló la ejecución de cada consulta cinco veces, la primera, denominada "Básica" que usa la similitud de Lucene (una variante de la similitud de cosenos); la segunda una expansión de la consulta basada en los documentos relevantes o no, que se presentaron en la consulta básica, a esta expansión se le denomina "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. Lo anterior con el objetivo de simular el proceso de refinación de las búsquedas que realiza un usuario cuando está buscando repetidamente sobre un tema específico. Es de notar que la memoria del perfil del usuario en este caso sólo dura de una solicitud de consulta a otra (en adelante se denomina, sin memoria del perfil de usuario). Los resultados de este experimento se presentan en la Figura 7.

En las tres gráficas de la Figura 7 (a, b y c) se muestra el resultado de la consulta básica usando Lucene, que inicia en un 56% de precisión para un nivel de recuerdo de 10%, y decrece hasta un 7% cuando el nivel de recuerdo es del 100%. Luego en las líneas con marcador rectangular se muestra el resultado de la expansión 1, mostrando una mejora apreciable en los tres algoritmos, llegando a un promedio de 94% de precisión en el primer nivel de recuerdo y cayendo a un promedio de 16% en el último nivel de recuerdo. Este primer proceso de expansión, muestra una curva de precisión-recuerdo que es muy superior en todos los niveles de recuerdo en la consulta básica. Además muestra como los tres algoritmos siguen mejorando poco a poco en la expansión 2, 3 y 4.

En la Tabla 1 se muestran en detalle los valores de la Figura 7. Se muestra como VT-IDF logra desde la expansión 1 una precisión de 94% en el 10% de recuerdo y como en la expansión 4 alcanza un 96%. Mientras que Rocchio logra un 93% en la primer expansión y un máximo de 98% en la expansión 4. Finalmente, muestra que CE-IDF logra un valor inicial y final de 94% en las 4 expansiones, pero logrando mejores resultados en los niveles de recuerdo del 20%, 30% y 40%.




(a) Rocchio (b) VT-IDF

(c) CE-IDF

**Figura 7.** *Curva de precisión-recuerdo para Rocchio (a), VT-IDF (b) y CE-IDF (c) sobre CACM sin memoria del perfil*

**Tabla 1.** *Valores de precisión -recuerdo para Rocchio, VT-IDF y CE-IDF sobre CACM sin memoria del perfil*

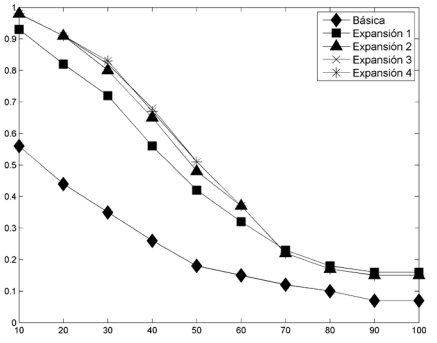
| | Recuerdo | 10% | 20% | 30% | 40% | 50% | 60% | 70% | 80% | 90% | 100% |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Básica** | Lucene | 0,56 | 0,44 | 0,35 | 0,26 | 0,18 | 0,15 | 0,12 | 0,10 | 0,07 | 0,07 |
| **Expansión 1** | CE-IDF | 0,94 | 0,87 | 0,81 | 0,65 | 0,48 | 0,34 | 0,24 | 0,21 | 0,17 | 0,17 |
| | VT-IDF | 0,94 | 0,87 | 0,78 | 0,63 | 0,50 | 0,34 | 0,23 | 0,18 | 0,15 | 0,15 |
| | Rocchio | 0,93 | 0,82 | 0,72 | 0,56 | 0,42 | 0,32 | 0,23 | 0,18 | 0,16 | 0,16 |
| **Expansión 2** | CE-IDF | 0,94 | 0,93 | 0,87 | 0,74 | 0,56 | 0,43 | 0,27 | 0,23 | 0,17 | 0,17 |
| | VT-IDF | 0,94 | 0,92 | 0,88 | 0,69 | 0,58 | 0,44 | 0,29 | 0,19 | 0,15 | 0,15 |
| | Rocchio | 0,98 | 0,91 | 0,80 | 0,65 | 0,48 | 0,37 | 0,22 | 0,17 | 0,15 | 0,15 |
| **Expansión 3** | CE-IDF | 0,94 | 0,94 | 0,89 | 0,78 | 0,59 | 0,48 | 0,31 | 0,24 | 0,17 | 0,17 |
| | VT-IDF | 0,94 | 0,93 | 0,90 | 0,72 | 0,62 | 0,49 | 0,32 | 0,19 | 0,16 | 0,16 |
| | Rocchio | 0,98 | 0,91 | 0,82 | 0,67 | 0,51 | 0,37 | 0,22 | 0,17 | 0,15 | 0,15 |
| **Expansión 4** | CE-IDF | 0,94 | 0,94 | 0,92 | 0,81 | 0,64 | 0,50 | 0,35 | 0,25 | 0,17 | 0,17 |
| | VT-IDF | 0,96 | 0,94 | 0,90 | 0,75 | 0,66 | 0,54 | 0,33 | 0,20 | 0,16 | 0,16 |
| | Rocchio | 0,98 | 0,91 | 0,83 | 0,67 | 0,51 | 0,39 | 0,22 | 0,17 | 0,15 | 0,15 |

En este primer experimento queda demostrado como el uso de la expansión de una consulta con base en la relevancia de los resultados previamente presentados al usuario, puede mejorar ostensiblemente los resultados del sistema. También muestra que para la colección de datos seleccionada el algoritmo de Rocchio obtiene mejores resultados en los primeros niveles de recuerdo, pero que VT-IDF y CE-IDF obtienen resultados muy similares.

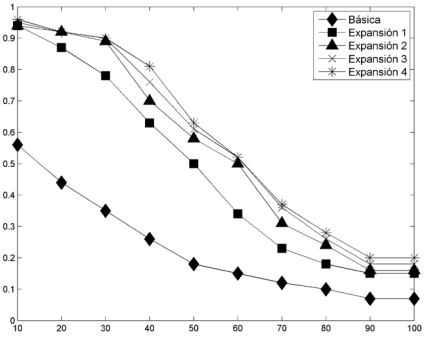
Un segundo experimento se llevó a cabo sobre el mismo conjunto de datos. El proceso seguido fue el mismo del experimento anterior, pero en este caso el perfil del usuario mantuvo memoria en las cinco ejecuciones de la misma consulta. Este proceso simula el almacenamiento del perfil de un usuario durante una sesión de consulta de un tema. Los resultados de este experimento se presentan en la Figura 8.

En las tres gráficas de la Figura 8 (a, b y c), se muestra el resultado de la consulta básica usando Lucene, luego en las líneas con marcador rectangular se muestra el resultado de la expansión 1, mostrando una mejora apreciable en los tres algoritmos, llegando a un promedio de 94% de precisión en el primer nivel de recuerdo y cayendo a un promedio de 16% en el último nivel de recuerdo. Este primer proceso de expansión muestra una curva de precisión-recuerdo que es evidentemente muy superior en todos los niveles de recuerdo a la consulta básica. Además se muestra como Rocchio, VT-IDF y CE-IDF aprovechan la mayor información del perfil para mejorar la precisión de los resultados, expansión tras expansión, en los diferentes niveles de recuerdo.

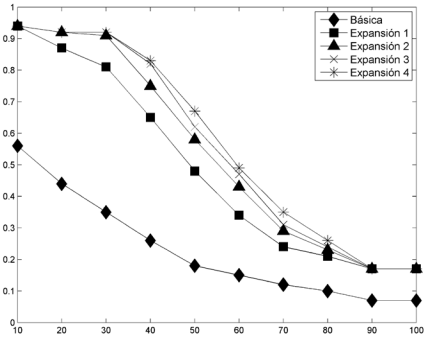
En la Tabla 2 se muestran en detalle los valores de la Figura 8. Se muestra como VT-IDF logra desde la expansión 1 una precisión de 94% en el 10% de recuerdo y como en la expansión 4 alcanza un 96%. Mientras que Rocchio logra un 93% en la primer expansión y un máximo de 98% en la expansión 4. Finalmente, muestra que CE-IDF logra un 94% en el primer nivel de recuerdo en todas las expansiones. De igual forma que en el experimento anterior, este algoritmo obtiene consistentemente mejores niveles de precisión en los niveles 20%, 30% y 40% de recuerdo.




(a) Rocchio




(b) VT-IDF




(c) CE-IDF

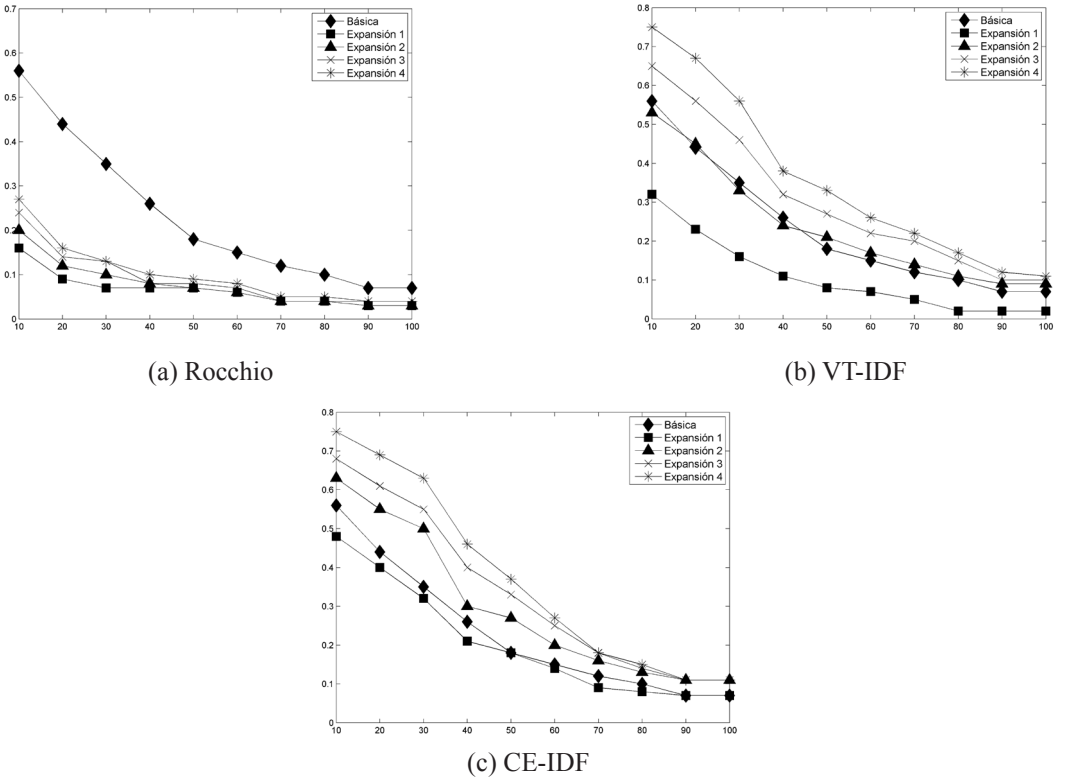
**Figura 8.** *Curva de precisión-recuerdo para Rocchio (a), VT-IDF (b) y CE-IDF (c) sobre CACM con memoria de sesión*

En general los resultados no son muy diferentes a los obtenidos en el experimento anterior.

**Tabla 2.** *Valores de precisión -recuerdo para Rocchio, VT-IDF y CE-IDF sobre CACM con memoria de sesión*

| | Recuerdo | 10% | 20% | 30% | 40% | 50% | 60% | 70% | 80% | 90% | 100% |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Básica** | Lucene | 0,56 | 0,44 | 0,35 | 0,26 | 0,18 | 0,15 | 0,12 | 0,10 | 0,07 | 0,07 |
| **Expansión 1** | CE-IDF | 0,94 | 0,87 | 0,81 | 0,65 | 0,48 | 0,34 | 0,24 | 0,21 | 0,17 | 0,17 |
| | VT-IDF | 0,94 | 0,87 | 0,78 | 0,63 | 0,50 | 0,34 | 0,23 | 0,18 | 0,15 | 0,15 |
| | Rocchio | 0,93 | 0,82 | 0,72 | 0,56 | 0,42 | 0,32 | 0,23 | 0,18 | 0,16 | 0,16 |
| **Expansión 2** | CE-IDF | 0,94 | 0,92 | 0,91 | 0,75 | 0,58 | 0,43 | 0,29 | 0,23 | 0,17 | 0,17 |
| | VT-IDF | 0,94 | 0,92 | 0,89 | 0,70 | 0,58 | 0,50 | 0,31 | 0,24 | 0,16 | 0,16 |
| | Rocchio | 0,98 | 0,91 | 0,80 | 0,65 | 0,48 | 0,37 | 0,22 | 0,17 | 0,15 | 0,15 |
| **Expansión 3** | CE-IDF | 0,94 | 0,92 | 0,92 | 0,82 | 0,62 | 0,47 | 0,31 | 0,24 | 0,17 | 0,17 |
| | VT-IDF | 0,95 | 0,92 | 0,90 | 0,76 | 0,61 | 0,52 | 0,36 | 0,26 | 0,18 | 0,18 |
| | Rocchio | 0,98 | 0,91 | 0,82 | 0,68 | 0,51 | 0,37 | 0,22 | 0,17 | 0,15 | 0,15 |
| **Expansión 4** | CE-IDF | 0,94 | 0,92 | 0,92 | 0,83 | 0,67 | 0,49 | 0,35 | 0,26 | 0,17 | 0,17 |
| | VT-IDF | 0,96 | 0,92 | 0,90 | 0,81 | 0,63 | 0,52 | 0,37 | 0,28 | 0,20 | 0,20 |
| | Rocchio | 0,98 | 0,91 | 0,83 | 0,67 | 0,51 | 0,37 | 0,22 | 0,17 | 0,15 | 0,15 |

Finalmente, se realizó un tercer experimento sobre el mismo conjunto de datos. El proceso seguido fue el mismo del experimento anterior, pero en este caso el perfil del usuario se mantuvo durante todas las consultas. Este proceso simula el almacenamiento del perfil de un usuario durante toda su vida en el sistema. Se considera el experimento más importante, debido a que en general, los SRI o búsqueda web deben mantener un perfil del usuario durante todo el tiempo que el usuario use el sistema y que este perfil se adapte a las cambiantes necesidades de búsqueda de los usuarios. Los resultados de este experimento se presentan en la Figura 9.




(a) Rocchio

(b) VT-IDF

(c) CE-IDF

**Figura 9.** *Curva de precisión-recuerdo para Rocchio (a), VT-IDF (b) y CE-IDF (c) con memoria de largo plazo*

En las tres gráficas de la Figura 9 (a, b y c), se muestra el resultado de la consulta básica usando Lucene (serie de datos con marcador en forma de rombo), luego en las líneas con marcador rectangular se muestra el resultado de la expansión 1. En este caso los tres algoritmos obtienen precisiones más bajas que las logradas con la expansión básica, esto debido al peso del perfil del usuario (historia de las consultas pasadas) sobre la consulta que se está realizando. Pero en este caso el algoritmo CE-IDF obtiene un mayor valor de precisión, mostrando que este algoritmo es menos sensible a la historia del usuario o dicho de otro modo, que CE-IDF se adapta más rápidamente a los cambios en los requerimientos de las consultas del usuario.

En la expansión 2 (serie de datos con marcador triangular), se muestra como los tres algoritmos mejoran la precisión, pero sólo CE-IDF mejora la consulta básica. Para la expansión 3 y 4 todos los algoritmos mejoran sus resultados progresivamente, pero sólo CE-IDF y VT-IDF obtienen mejores resultados a la consulta básica, llegando a una diferencia hasta de 20% en el primer nivel de recuerdo. En todos los casos CE-IDF obtiene mejores resultados, reafirmando con esto, la idea de que es un método que se adapta más rápidamente a las nuevas necesidades del usuario.

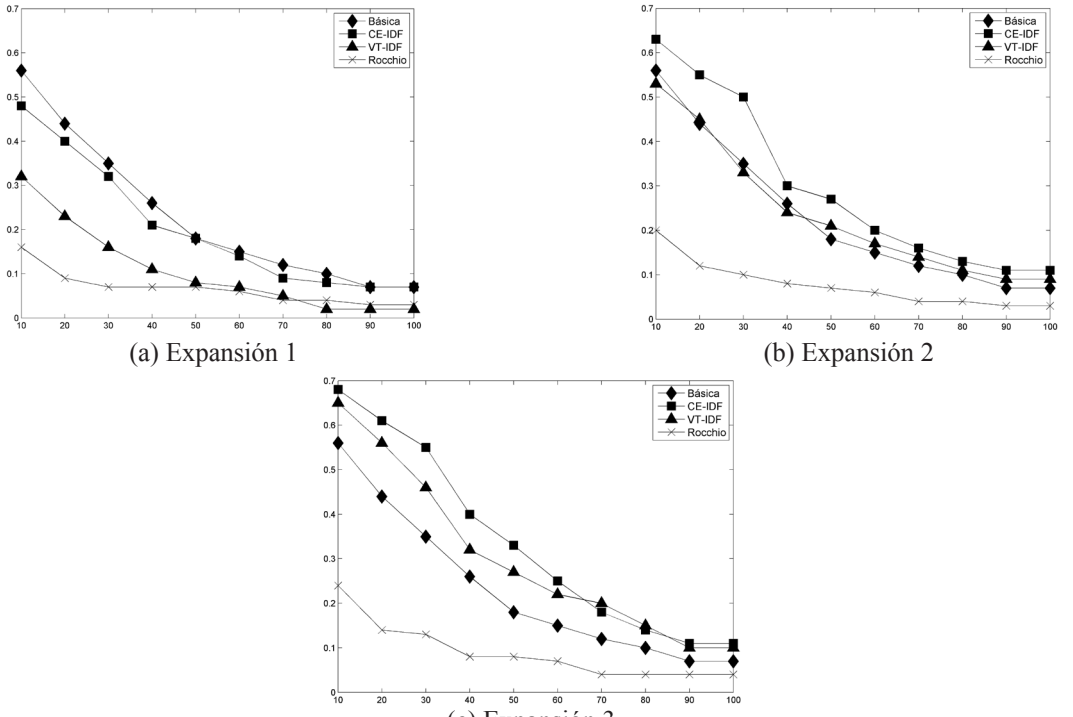
En la grafica de Rocchio de la Figura 9, además se observa que el proceso de mejora es más lento que el obtenido con los otros dos algoritmos. Evaluaciones adicionales, mostraron que Rocchio puede obtener mejores resultados de precisión en este tercer experimento cuando $\alpha = 90\%$, $\beta = 10\%$ y $\gamma = 0\%$. En este caso la precisión oscila entre 55% y 62% en el primer nivel de recuerdo durante las cuatro expansiones. Desafortunadamente, con estos parámetros los valores de precisión para los dos primeros experimentos disminuyen a 91% y 94% en el primer nivel de recuerdo en las cuatros expansiones. Con estos nuevos valores para los parámetros se logra disminuir el peso del historial sobre la consulta inicial del usuario en el algoritmo de Rocchio. Además con esto se evidencia la dificultad que puede presentar la definición apropiada de estos valores en este algoritmo.

En la Tabla 3 se muestran en detalle los valores de la Figura 9. Se muestra como CE-IDF logra desde la expansión 1 una precisión de 48% en el 10% de recuerdo y como en la expansión 4 alcanza un 75%. Mientras que Rocchio logra tan sólo un 16% en la expansión 1 y un máximo de 27% en la expansión 4. Finalmente, muestra que VT-IDF a pesar de empezar con un 32% en la primera expansión, alcanza a igualar a CE-IDF en la expansión 4 con un 75% de precisión

En la Figura 10 se muestra la curva de precisión-recuerdo de tres expansiones y permite comparar visualmente los resultados obtenidos con los tres algoritmos. En general los resultados muestran que CE-IDF es un mejor algoritmo cuando se tiene en cuenta un perfil de largo plazo, seguido de VT-IDF y por último de Rocchio.

**Tabla 3.** *Valores de precisión -recuerdo para Rocchio, VT-IDF y CE-IDF con memoria de largo plazo*

| | Recuerdo | 10% | 20% | 30% | 40% | 50% | 60% | 70% | 80% | 90% | 100% |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Básica** | Lucene | 0,56 | 0,44 | 0,35 | 0,26 | 0,18 | 0,15 | 0,12 | 0,10 | 0,07 | 0,07 |
| **Expansión 1** | CE-IDF | 0,48 | 0,40 | 0,32 | 0,21 | 0,18 | 0,14 | 0,09 | 0,08 | 0,07 | 0,07 |
| | VT-IDF | 0,32 | 0,23 | 0,16 | 0,11 | 0,08 | 0,07 | 0,05 | 0,02 | 0,02 | 0,02 |
| | Rocchio | 0,16 | 0,09 | 0,07 | 0,07 | 0,07 | 0,06 | 0,04 | 0,04 | 0,03 | 0,03 |
| **Expansión 2** | CE-IDF | 0,63 | 0,55 | 0,50 | 0,30 | 0,27 | 0,20 | 0,16 | 0,13 | 0,11 | 0,11 |
| | VT-IDF | 0,53 | 0,45 | 0,33 | 0,24 | 0,21 | 0,17 | 0,14 | 0,11 | 0,09 | 0,09 |
| | Rocchio | 0,20 | 0,12 | 0,10 | 0,08 | 0,07 | 0,06 | 0,04 | 0,04 | 0,03 | 0,03 |
| **Expansión 3** | CE-IDF | 0,68 | 0,61 | 0,55 | 0,40 | 0,33 | 0,25 | 0,18 | 0,14 | 0,11 | 0,11 |
| | VT-IDF | 0,65 | 0,56 | 0,46 | 0,32 | 0,27 | 0,22 | 0,20 | 0,15 | 0,10 | 0,10 |
| | Rocchio | 0,24 | 0,14 | 0,13 | 0,08 | 0,08 | 0,07 | 0,04 | 0,04 | 0,04 | 0,04 |
| **Expansión 4** | CE-IDF | 0,75 | 0,69 | 0,63 | 0,46 | 0,37 | 0,27 | 0,18 | 0,15 | 0,11 | 0,11 |
| | VT-IDF | 0,75 | 0,67 | 0,56 | 0,38 | 0,33 | 0,26 | 0,22 | 0,17 | 0,12 | 0,11 |
| | Rocchio | 0,27 | 0,16 | 0,13 | 0,10 | 0,09 | 0,08 | 0,05 | 0,05 | 0,04 | 0,04 |




(a) Expansión 1 (b) Expansión 2

(c) Expansión 3

**Figura 10.** *Comparación de Rocchio, VT-IDF y CE-IDF en tres expansiones*

## 6. CONCLUSIONES Y TRABAJO FUTURO

En este artículo se presentó una nueva función de la importancia relativa de un término en una colección de documentos. Esta función es discreta, está en el rango de 0 a 1 incluidos y refleja la prevalencia de los términos que aparecen mayoritariamente en documentos relevantes.

Con base en la nueva función para el cálculo del valor IDF de un término, se presentaron dos algoritmos de expansión de consulta. El primero denominado VT-IDF que tiene la misma orientación de Rocchio, en el sentido de obtener una consulta expresada como un vector de términos con sus pesos para ubicar en el espacio multidimensional de términos por documentos. El segundo denominado CE-IDF que complementa la consulta del usuario con los términos más relevantes del perfil y entrega como resultado una lista de términos en una cadena de texto similar a la digitada por el usuario.

La evaluación de VT-IDF y CE-IDF se realizó frente a una consulta "básica", una búsqueda por similitud de cosenos sin proceso de expansión, y el algoritmo de Rocchio de expansión de consulta. Los experimentos se realizaron en tres escenarios: sin memoria, con memoria de sesión y con memoria a largo plazo. Los resultados en los dos primeros escenarios favorecieron a VT-IDF y dejaron a CE-IDF a una corta distancia de los mejores resultados. Pero en el tercer escenario, el más importante, el algoritmo CE-IDF obtuvo los mejores resultados, mostrándose como un algoritmo que se adapta más rápidamente a los cambios en los requerimientos de los usuarios de búsqueda, una situación muy común en los usuarios de los buscadores de Internet y de los sistemas de recuperación de información en general.

Como trabajo futuro, el grupo de investigación espera evaluar los algoritmos propuestos con otras colecciones comúnmente usadas en recuperación de información, como por ejemplo: TREC, LISA, ISI, NPL, TIME, MED [1]. Realizar una propuesta de los algoritmos, incluyendo la capacidad de analizar semánticamente los términos, con el objetivo de gestionar conceptos en lugar de términos, a través de ontologías (por ejemplo: WordNet), diccionarios o tesauros de dominio general.

## 7. AGRADECIMIENTOS

El trabajo en este artículo fue soportado por la Universidad del Cauca bajo el proyecto VRI-2560, la Universidad Industrial de Santander y la Universidad Nacional de Colombia sede Bogotá.

## 8. BIBLIOGRAFÍA

[1] Baeza-Yates, R., A. and B. Ribeiro-Neto, *Modern Information Retrieval.* 1999: Addison-Wesley Longman Publishing Co., Inc. 513.

[2] Manning, C., P. Raghavan, and H. Schütze, *Introduction to Information Retrieval.* 2008, Cambridge University Press: Cambridge, England.

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

[4] Hammouda, K., *Web Mining: Clustering Web Documents A Preliminary Review.* 2001. p. 1-13.

[5] Yongli, L., et al., *A Query Expansion Algorithm Based on Phrases Semantic Similarity, in Proceedings of the 2008 International Symposiums on Information Processing.* 2008, IEEE Computer Society.

[6] Inna Gelfer, K. and K. Oren, *Cluster-based query expansion, in Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval.* 2009, ACM: Boston, MA, USA.

[7] Efthimiadis, E.N. *Query Expansion.* 1996 [cited 2011; in: Williams, Martha E., ed. Annual Review of Information Systems and Technology (ARIST), v31, pp 121-187, 1996]. Available from: http://faculty.washington.edu/efthimis/pubs/Pubs/qe-arist/QE-arist.html.

[8] Robertson, S.E. and K. Sparck-Jones, *Relevance weighting of search terms, in Document retrieval systems.* 1988, Taylor Graham Publishing. p. 143-160.

[9] Garcia, E. RSJ-PM *Tutorial: A Tutorial on the Robertson-Sparck Jones Probabilistic Model for Information Retrieval.* 2009; Available from: http://www.miislita.com/information-retrieval-tutorial/information-retrieval-probabilistic-model-tutorial.pdf.

[10] Biancalana, C. and A. Micarelli. *Social Tagging in Query Expansion: A New Way for Personalized Web Search. in SocialCom-09 the 2009 IEEE International Conference on Social Computing.* 2009. Vancouver, Canada.

[11] Marin, B., et al., *Toward personalized query expansion, in Proceedings of the Second ACM EuroSys Workshop on Social Network Systems.* 2009, ACM: Nuremberg, Germany.

[12] Dongsheng, Z. 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.

[13] Nguyen, T.C. and T.T. Phan. *An Ontology-Based Approach of Query Expansion. in iiWAS'2007 - The Ninth International Conference on Information Integration and Web-based Applications Services.* 2007. Jakarta, Indonesia.

[14] Han, L. and G. Chen, HQE: *A hybrid method for query expansion.* Expert Systems with Applications, 2009. 36(4): p. 7985-7991.

[15] ASF. *Class Similarity.* [cited 2011 January 10, 2011]; Available from: http://lucene.apache.org/java/2_9_0/api/core/org/apache/lucene/search/Similarity.html.

[16] Porter, M.F., *An algorithm for suffix stripping.* Program, 1980. 14(3): p. 130-137.

[17] Dominich, S., *The Modern Algebra of Information Retrieval.* The Information Retrieval Series, ed. W.B. Croft. 2008: Springer-Verlag Berlin Heidelberg.

[18] IRG. *Test collections.* [cited 2011 January 15, 2011]; Available from: http://ir.dcs.gla.ac.uk/resources/test_collections/.