Un Motor de Búsqueda Implementado sobre el Índice XM-Tree

A WEB SEARCH ENGINE BASED ON XM-TREE INDEX

Claudia Deco, Cristina Bender
Departamento de Investigación Institucional
Facultad de Química e Ingeniería
Universidad Católica Argentina
{cdeco, cbender}@uca.edu.ar

Guillermo Pierángeli
Facultad de Ciencias Exactas, Ingeniería y Agrimensura
Universidad Nacional de Rosario
guillepier@yahoo.com

Abstract

Web Information Retrieval is a problem related with finding a set of elements closest to a given query under a similarity criterion. Metric spaces can be used to solve effectively and efficiently this problem. This article presents *XM-Search*, a search index that uses *XM-tree* with extensions to handle Boolean logic. With an adequate representation of web documents, this system achieves a better efficiency compared with other indexes. Experimentation shows good results achieved.

Keywords: Web Information Retrieval, Similarity Search, Metric Spaces.

Resumen

La Recuperación de Información de la Web es un problema relacionado con buscar los elementos más cercanos a una consulta dada bajo un cierto criterio de similitud. Es de interés aprovechar las cualidades de los espacios métricos con el objeto de resolver una consulta de manera efectiva y eficiente. En este artículo se presenta el sistema XM-Search, que es un motor de búsqueda que utiliza el índice XM-tree con extensiones para el manejo de lógica booleana. Con una adecuada representación de los documentos web, el sistema logra efectividad y eficiencia en comparación con otros índices. Las experimentaciones muestran los buenos resultados alcanzados.

Palabras claves: Recuperación de Información Web, Búsqueda por Similitud, Espacios Métricos.

1. INTRODUCCIÓN

Con la evolución de las tecnologías de información y comunicación han surgido repositorios de información no estructurada (texto libre, imágenes, video y audio). En un entorno de este tipo surge el concepto de búsqueda por similitud o búsqueda por proximidad, que consiste en recuperar los elementos más similares al elemento dado en la consulta. En el modelo de Espacios Métricos la similitud se modela con una función distancia, y el conjunto de objetos es llamado espacio métrico.

El objetivo de muchos trabajos sobre espacios métricos es disminuir el costo de calcular las distancias entre los objetos y reducir las entradas/salidas. En estos problemas, algunos avances importantes se destacan en el concepto de construir un índice. La velocidad es de vital importancia en un sistema de recuperación de información en la web que debe responder consultas en tiempo real, ya que un posiblemente largo período de espera no es aceptable para los usuarios comunes. La principal ventaja de los índices en general es que permiten una rápida recuperación de datos relevantes en las búsquedas por similitud, dado que permiten reducir el número de evaluaciones de distancias en el momento de la consulta.

La web es un repositorio de millones de datos, por lo tanto propone un desafío constante a la creación de índices veloces. En general, los motores de búsqueda en la web, cuando resuelven consultas de varias palabras, primero recuperan los documentos correspondientes a cada una de estas palabras y luego realizan un proceso de mezcla e intersección de las páginas resultantes. Para agilizar la velocidad de respuestas estos buscadores devuelven no sólo los documentos que satisfacen la estrategia de búsqueda, sino también aquellos que posean alguno de los términos. En este trabajo se presenta el motor de búsqueda *XM-Search* que aprovecha las propiedades de los índices. Para mejorar los resultados, se propone un posprocesamiento de la lista de documentos resultantes, y para la construcción de los índices se utiliza el algoritmo *XM-Tree* [1].

El resto del trabajo se organiza de la siguiente forma: en la Sección 2 se presentan conceptos básicos. En la Sección 3 se presentan trabajos relacionados. En la Sección 4 se presenta el sistema propuesto. La Sección 5 presenta la experimentación. Finalmente se presentan las conclusiones.

2. CONCEPTOS BÁSICOS

2.1. Recuperación de Información en la Web

La web es una colección de información no estructurada en rápida expansión, conformada por documentos enlazados. Dado un conjunto de documentos y una consulta, la Recuperación de Información determina el subconjunto de documentos relevante a la consulta. El proceso de búsqueda es realizado por un motor de búsqueda, el cual maneja un cuerpo de documentos a través de dos pasos: indexado y recuperación.

El *indexado* produce un *índice invertido*, que se construye a partir del cuerpo de documentos realizando un mapeo entre cada término presente en el cuerpo, hacia los documentos que contienen dicho término. La ocurrencia de un término en un documento se llama *posting*. El conjunto de postings asociados a un término se almacena en una *posting list*.

En la *recuperación*, el motor de búsqueda recibe como entrada una consulta constituida por una o más palabras. El motor utiliza el índice invertido para ver qué documentos contienen las palabras ingresadas según el índice, y obtiene las posting lists de cada una. Estas listas se procesan para obtener el conjunto de resultados final. Este procesamiento consiste en realizar uniones ó intersecciones según la semántica de la consulta manejada por cada tipo de motor. Por ejemplo, los motores de búsqueda booleanos exigen que la consulta sea formulada utilizando operadores booleanos (AND, OR, NOT) entre las palabras ingresadas. El conjunto resultado se obtiene aplicando las operaciones de conjuntos intersección, unión y diferencia apropiadamente según la lógica booleana de la consulta.

El desempeño de un motor de búsqueda se puede evaluar a través de los indicadores *Precisión* y *Recall*. La Precisión es el número de documentos recuperados relevantes a la consulta dada dividido el total de documentos recuperados. El Recall es el cociente entre la cantidad de documentos relevantes recuperados y el total de documentos relevantes de la colección. Un motor de búsqueda

alcanza un buen rendimiento cuando maximiza ambos valores; es decir, recupera la mayoría de los documentos relevantes de la colección con la menor cantidad de documentos irrelevantes.

Una forma de representar los documentos en la recuperación de información es el *modelo espacio vectorial*, en el cual los documentos son representados con vectores de palabras [2]. En estos vectores se descartan los términos muy frecuentes, artículos y pronombres. Los verbos conjugados, sustantivos y adjetivos, se reducen a una forma canónica con un proceso de *stemming*. El conjunto de palabras resultante representa al documento como un vector en el espacio euclídeo, donde cada término canónico representa un eje en este espacio. Si el término t_i ocurre en el documento d, la i-ésima componente del vector de d vale 1, si no, vale 0. Las consultas se representan de la misma manera. La similitud entre una consulta q y un documento d se cuantifica calculando el producto interno entre los vectores de q y d: $sim(q,d)=\sum q_i \times d_i$. Esto permite retornar una secuencia de documentos ordenados por su relevancia a la consulta, ofreciendo un mecanismo de *ranking*.

Esta representación con componentes en 0 ó 1 no captura la importancia de un término frente a otro en un documento. La solución es que cada componente represente el peso de cada término en el documento, por ejemplo considerando la cantidad de ocurrencias del término. Pero esta representación favorece a los documentos más grandes. Esto se resuelve con el *esquema de frecuencia de términos TF,* en el cual se normalizan los vectores dividiendo cada componente por la longitud de los mismos. De esta manera, cada componente d_i del vector vale tf_i, o sea, la frecuencia del término t_i en el documento d. El esquema TF no tiene en cuenta la distribución del término en la colección entera. Por esto no modela la ocurrencia de términos en pocos documentos, lo que ayudaría a discriminar esos documentos del resto de la colección. Tampoco modela la menor importancia de aquellos términos que ocurren frecuentemente en toda la colección. La distribución de un término en la colección entera puede ser capturada con la *frecuencia de documento inversa*: $idf_i=log(N/n_i)$, donde N es el tamaño de la colección y n_i es el número de documentos donde ocurre el término t_i. En la práctica, se combinan las medidas *tf* e *idf* en el denominado *esquema TF/IDF*: $w_{ij}=tf_{ij} \times idf_i$, donde w_{ij} es el valor que se le asigna a la componente i del documento j.

2.2. Espacios Métricos

Un espacio métrico (X,d) está formado por un universo de objetos válidos X y una función distancia $d:X \times X \rightarrow R^+$ que mide la semejanza o proximidad entre dos objetos cualesquiera. La función distancia tiene las siguientes propiedades: no negatividad ($d(x,y) \geq 0$), simetría ($d(x,y)=d(y,x)$), reflexividad ($d(x,x)=0$), positividad estricta ($\forall x,y \in X, x \neq y \Rightarrow d(x,y)>0$), y satisface la desigualdad triangular ($d(x,z) \leq d(x,y)+d(y,z)$). El subconjunto finito U de X, de tamaño $n=|U|$, es el conjunto donde se realizan las búsquedas. Los tipos de consultas de interés en los espacios métricos son la *Consulta por rango* $(q,r)_d$ que recupera todos los elementos que están dentro de la distancia r de q, *o* sea $\{u \in U \mid d(q,u) \leq r\}$ y la *Consulta por k vecinos más cercanos* $NN_k(q)$ que recupera los k elementos más cercanos a q en U, obteniendo un conjunto $A \subseteq U$ tal que $|A|=k$ y $\forall u \in A$, $v \in U-A$, $d(q,u) \leq d(q,v)$. En este trabajo se utiliza la *Consulta por rango estricta* $(q,r)_d$ para recuperar todos los elementos que están dentro de la distancia r de q y no la alcanzan, o sea $\{u \in U \mid d(q,u)<r\}$.

Un *espacio vectorial k*-dimensional es un espacio métrico particular donde los objetos se identifican con k coordenadas de valores reales $(x_1, ..., x_k)$. Las funciones distancia más utilizadas en los espacios vectoriales son la familia L_s, definida como: $L_s((x_1,...,x_k), (y_1,...,y_k)) = (\sum |x_i-y_i|^s)^{1/s}$. La distancia L_1 ($s=1$), también llamada *de bloques*, es la suma de las diferencias entre las coordenadas. La distancia L_2, o *euclídea*, corresponde a la noción de distancia espacial. Otra muy utilizada es L_∞, que equivale al límite de L_s cuando s tiende a infinito, y se define como la máxima diferencia entre dos coordenadas de los dos objetos dados: $L_\infty((x_1,...,x_k), (y_1,...,y_k)) = \max| x_i-y_i |$.

Los índices sobre espacios métricos permiten la recuperación efectiva y eficiente de objetos. Efectiva porque los resultados tienen un alto grado de exactitud por propiedades del espacio y del índice; y eficiente porque los índices se construyen para reducir el número de cálculos y objetos revisados. Estas dos propiedades son muy deseadas en un motor de búsqueda y hacen de la utilización de espacios métricos una alternativa interesante en los procesos de indexado y búsqueda.

3. TRABAJOS RELACIONADOS

El continuo crecimiento de la web, propone un desafío constante a la creación de índices veloces que puedan utilizar los motores de búsqueda. Una solución actual es la utilización de índices invertidos, empleados por buscadores reconocidos como *Google* y *CiteSeer*.

Google es un motor de búsqueda que tiene dos propiedades que le permiten recuperar resultados con alta precisión [3]. Primero utiliza la estructura de enlaces de la web para calcular un ranking de calidad de cada página mediante el algoritmo *PageRank* [4]. Luego utiliza la información provista por el texto de los enlaces para mejorar los resultados. Utiliza dos índices invertidos: uno para los textos de los enlaces y de los títulos de las páginas, y otro para los textos completos de las páginas. En ambos índices se almacena información sobre la posición en el documento, tipo de fuente y capitalización de cada palabra.

CiteSeer es un sistema automático de indexado de citas de literatura académica [5]. Recupera literatura a partir de los enlaces con citas a otros artículos, realiza evaluación y ranking de artículos, autores, publicaciones periódicas, etc. según el número de citas, e identifica tendencias de investigación. Se indexa el texto completo del documento y las citas. CiteSeer mantiene un índice invertido el cual contiene documentos y posición donde ocurre cada palabra. Utiliza el esquema TF/IDF para elaborar una medida de similitud entre documentos y poder recuperar documentos relacionados. Permite recuperación por lógica booleana, frases o proximidad.

Los índices invertidos logran velocidad pero tienen la desventaja de indexar de a una palabra. Cuando se resuelven consultas de varias palabras, primero se recuperan los documentos correspondientes a cada una de éstas, y luego se realizan las operaciones lógicas entre estos conjuntos de documentos. Para agilizar la velocidad de respuesta los buscadores devuelven no sólo los documentos que satisfacen la estrategia de búsqueda, sino que devuelven también aquellos que posean alguno de los términos. En este trabajo se propone mejorar estos resultados mediante un posprocesamiento de la lista de documentos resultantes.

4. EL SISTEMA XM-SEARCH

El modelo espacio vectorial tiene la ventaja de informar las ocurrencias de los términos de un documento en un solo vector representativo. Pero, a diferencia de los índices invertidos, no ofrece un acceso directo a los documentos que contienen un término dado, sino que hay que recorrer los vectores de la colección completa. Esto último se puede solucionar agrupando los vectores según algún criterio de similitud. En [1] se presenta el índice *XM-tree*, que extiende un *M-tree* [6] sobre espacios vectoriales, adaptando sus algoritmos a la búsqueda en la web. En el artículo citado se propone al XM-tree como un índice sobre páginas web, el cual utiliza la distancia L_2 para el indexado y la distancia L_∞ para la consulta. El XM-tree tiene la posibilidad de acceder por diferentes recorridos a distintos subconjuntos de datos agrupados por su cercanía. Al ejecutar una búsqueda, elige los recorridos analizando la consulta ingresada con información que contiene la misma estructura de este índice. Logra velocidad porque en este proceso va descartando grupos de documentos que no tienen datos próximos a la consulta.

Para lograr velocidad y calidad de resultados, se debe utilizar una adecuada representación de las

páginas web y un criterio conveniente de similitud entre las mismas que posibiliten el empleo de un índice XM-tree. Las decisiones tomadas para esto se presentan a continuación.

Generación del Diccionario: El *diccionario* es un archivo de texto donde cada entrada corresponde a un término. Se descartan palabras no significativas (artículos, pronombres, preposiciones, etc.), o palabras muy comunes ("uso", "encontrar", "normal", "otro"). Se confecciona manualmente con la ayuda de expertos por tratar directamente con la semántica de las palabras. Por ejemplo, para trabajar con documentos que tratan sobre Medicina se deben incluir en el diccionario términos como: "cáncer", "célula", "tejido", "organismo". Si hay documentos que tratan sobre Astronomía se insertan palabras como: "estrella", "constelación", "galaxia". Si el tema es Astrología, entonces se agregan: "signo", "horóscopo", "Aries". Notar que hay términos que se refieren a varios temas, por ejemplo "constelación" puede aparecer en documentos sobre Astronomía y en documentos sobre Astrología; y "cáncer", puede aparecer en documentos sobre Medicina, Astronomía, Astrología o Geografía. Los términos pueden aparecer en los documentos con distintas variantes morfológicas. En el caso de un verbo puede figurar cualquiera de sus conjugaciones. Los adjetivos y sustantivos aparecen con distintos género y número. Así, cada entrada del diccionario se representa con su correspondiente término y sus variantes morfológicas. Una muestra del diccionario es:

abdomen|abdominal|abdominales|abdomenes
abril
acotar|acotada|acota|acotó ...
acuario|acuariana|acuarianas|acuariano|acuarianos ...
anemia|anemias|anemica|anemicas|anemico|anemicos

A partir del diccionario se define un espacio vectorial de dimensión T_DIC, la cual corresponde a la cantidad de entradas del diccionario, y donde cada eje representa una entrada del diccionario.

Representación de los Documentos: Los documentos se representan como puntos en el espacio vectorial definido por el diccionario. La componente *i*-ésima del documento *d* es el número de veces que aparecen en el documento las variantes morfológicas que figuran en la entrada *i* del diccionario. Para mejorar la comparación entre documentos de distintos tamaños, es decir con distintas cantidades de términos, se considera la frecuencia de los términos. Para ello, se normalizan los vectores dividiendo cada componente por el número total de palabras del documento. Esta representación corresponde al esquema TF. Se descartó utilizar el esquema TF/IDF porque requiere para el cálculo de cada componente el número de ocurrencias del término en el resto de los documentos. En este caso, al ingresar un nuevo documento, habría que recalcular los vectores de todos los documentos. Por esto, el índice no sería dinámico y no sería apto para la web. El esquema TF es eficaz porque los documentos que comparten palabras muy frecuentes que estén en el diccionario, quedan representados por vectores muy cercanos en cualquier distancia L_s. Entonces, el XM-tree que utiliza la distancia L_2, agrupa estos documentos en el mismo grupo. Por lo tanto, acelera la resolución de las consultas conformadas por los términos compartidos.

A modo de ejemplo, en la Tabla 1 se describen posibles representaciones de documentos que contienen el término "cáncer" y tratan sobre cuatro áreas distintas del conocimiento: Medicina, Astronomía, Astrología y Geografía. Las "X" denotan la ocurrencia de los términos en cada área. Los documentos que contienen las palabras "cáncer", "células" y "tejidos" (o sus variantes morfológicas) de manera muy frecuente están codificados como muestra la columna Medicina. Por lo tanto, las representaciones similares a estas cuatro columnas figuran agrupadas en distintos subárboles. Para resolver la consulta "trópico cáncer", el algoritmo de búsqueda del XM-tree poda los subárboles Medicina, Astronomía y Astrología descartándolos aunque contengan el término "cáncer", y revisa directamente el subárbol Geografía. Si la consulta es "constelación cáncer", los subárboles podados son Medicina y Geografía, y los revisados son Astronomía y Astrología. Como

se observa, con esta representación el XM-tree agrupa por similitud los vectores del esquema TF, y además resuelve la dificultad inherente de los índices invertidos que ante las consultas mencionadas revisaría documentos de los cuatro temas por contener la palabra "cáncer".

Tabla 1: Posibles representaciones de documentos con el término "cáncer"

	Medicina	Astronomía	Astrología	Geografía
Cáncer	X	X	X	X
Células	X			
Tejidos	X			
Estrella		X		
Constelación		X	X	
Galaxia		X		
Signo			X	
Horóscopo			X	
Aries		X	X	
Trópico				X
Ecuador				X
Polo				X

Indexado: El índice utilizado es un XM-tree cuya distancia de indexado es L_2, o distancia euclídea, la cual permite a este índice agrupar satisfactoriamente los documentos similares en el esquema TF.

Búsquedas: Dada una consulta q, el algoritmo de búsqueda recupera los documentos d tal que $L_\infty(q,d)<1$. La consulta se representa con un vector q de dimensión T_DIC de valores reales, donde los términos de la consulta que figuran en la entrada i del diccionario se indican con valor 1 en la componente i-ésima de q, y el resto de las componentes se valúan con 0. Con esta representación de q, el XM-tree recupera todos los documentos que contienen todos los términos ingresados en la consulta. Por ejemplo, consideremos tres páginas que tratan sobre cáncer: la enfermedad, la constelación, y el signo; y una página que trata sobre el hierro en la dieta. Sus títulos y direcciones URL son:

Doc1: Medline Plus - Cánceres -www.nlm.nih.gov/medlineplus/spanish/cancers.html
Doc2: Cancer (constelación) - Wikipedia - es.wikipedia.org/wiki/Cáncer_(constelación)
Doc3: Yahoo! Astrología – Cáncer - ar.astrology.yahoo.com/daily/cancer.html
Doc4: MedlinePlus Enciclopedia Médico – Hierro en la dieta - www.nlm.nih.gov/medlineplus/spanish/ency/article/002422.htm

Supongamos que las ocurrencias de algunos términos de estas páginas son las indicadas en la Tabla 2 en las columnas Doc1 a Doc4, bajo el título Documentos. En esta tabla también se muestran, bajo el título Consultas, las respectivas codificaciones q_1, q_2, q_3 y q_4 de las siguientes cuatro consultas: c_1="cáncer", c_2="cáncer cerebro", c_3="cáncer constelación" y c_4="hierro".

Tabla 2: Ocurrencia de términos en documentos y codificaciones de consultas

	Documentos					Consultas			
	Doc1	Doc2	Doc3	Doc4		q_1	q_2	q_3	q_4
Cáncer	0.45	0.11	0.06	0	Cáncer	1	1	1	0
Cerebro	0.01	0	0	0	Cerebro	0	1	0	0
Constelación	0	0.10	0	0	Constelación	0	0	1	0
Horóscopo	0	0	0.03	0	Horóscopo	0	0	0	0
Hierro	0	0	0	0.15	Hierro	0	0	0	1

Los valores de L_∞ entre consulta y documento se muestran en la Tabla 3. Como se observa, la consulta c_1 es satisfecha por Doc1, Doc2 y Doc3, lo que es cierto porque contienen al único término ingresado. Las consultas de dos palabras c_2 y c_3 son verificadas por Doc1 y Doc2 respectivamente, y c_4, con el término "hierro" es satisfecha por el único documento que lo contiene: Doc4.

Tabla 3: Distancias entre cada consulta y cada documento usando L_∞

	Doc1	Doc2	Doc3	Doc4
q_1	<1	<1	<1	1
q_2	<1	1	1	1
q_3	1	<1	1	1
q_4	1	1	1	<1

Lógica Booleana: El algoritmo de búsqueda sobre la representación TF propuesta resuelve consultas utilizando el operador booleano AND. Es decir, recupera los documentos donde ocurren de manera conjunta todos los términos ingresados. Una consulta se codifica en un vector q para recorrer el XM-tree. Según dicho esquema, un vector con tres de sus componentes (por ejemplo, 1, 2 y 3) valuadas en 1 y el resto nulas, representa a la consulta: "$c = t_1$ AND t_2 AND t_3", donde t_1, t_2 y t_3 pertenecen a las entradas del diccionario. La búsqueda sobre el XM-tree devuelve los documentos d tal que $L_\infty(q,d)<1$, es decir, donde ocurren términos de las tres entradas.

Para modelar los otros operadores booleanos (OR y NOT) se extiende la estructura de datos que modela las consultas y se modifica el algoritmo de búsqueda. La primera extensión corresponde a modelar el operador booleano NOT. Para esto, se propone recuperar tanto los documentos que contengan o no el término negado, y utilizar un posprocesamiento para descartar los que tienen este término negado. Si la frecuencia v_i de una entrada i del diccionario es nula indica que dicho término no está en el documento. Si su valor es distinto de cero indica que por lo menos un término de la entrada correspondiente del diccionario ocurre en el documento. Para conservar el criterio de similitud L_∞ se representan los términos t que están precedidos por NOT en la consulta asignando a su componente del vector q el valor 0.5. De esta manera, $| q_i - v_i |<1$ vale siempre, sin importar que la frecuencia v_i sea nula o no, y el posprocesamiento podrá descartar aquellos con v_i distinto de 0.

Para modelar el operador OR se suponen las consultas en forma de *estrategia disyuntiva.* Sean t_{ij} los términos ingresados en la consulta, la consulta $c_i=t_{i1}$ AND t_{i2} AND … AND t_{ij}, y el vector q_i que la representa. La estrategia disyuntiva $c^*=c_1$ OR c_2 OR...OR c_n está representada por $e=\{q_i \mid i=1,...,n\}$, es decir, un conjunto de n vectores q (con valores reales 0, 0.5 y 1). Entonces, un documento d satisface la consulta c^* codificada en e si satisface por lo menos una de las consultas c_i, o sea existe $q_i \in e$ tal que $L_\infty(q_i,d)<1$. De esta manera, un sistema de recuperación de información basado en XM-tree puede resolver consultas con los operadores AND, OR y NOT.

4.1. Prototipo

El prototipo del sistema *XM-Search* ha sido implementado en C^{++} utilizando el compilador g^{++}-4.1 de la colección GCC[1]. Además, se desarrolló una versión en Java, con *Oracle JDeveloper 10.1.2*[2], para mostrar al sistema en un entorno web real, y facilitar el ingreso de las consultas y la exploración de los resultados vía enlaces, siendo así más amigable para el usuario común.

La búsqueda se realiza a través de procesos de indexado y recuperación sobre el cuerpo de documentos, y un posprocesamiento de los resultados obtenidos. En el indexado y recuperación utiliza un diccionario, que puede tomar el formato de un índice invertido para acelerar las consultas. El programa *generador.c* realiza el indexado, calculando los vectores TF de las páginas a indexar. Para considerar solamente el texto visible de las páginas se utiliza *HTMLAsText v1.05*[3] que convierte documentos HTML en archivos de texto. El programa *generador.c* recibe como entrada un archivo con las páginas a indexar, el directorio con las versiones txt de dichas páginas y el índice

[1] GNU Project – Free Software Foundation (FSF), gcc.gnu.org
[2] www.oracle.com/technology/products/jdev/index.html
[3] www.nirsoft.net/utils/htmlastext.html

invertido. El archivo con las páginas a indexar contiene la dirección URL de cada una y la ubicación de su respaldo. El respaldo es una versión de la página almacenada en el servidor donde corre el sistema. La salida es un archivo con un número identificador (docID), el vector de frecuencias, el título, la dirección URL y la ubicación del respaldo para cada página. El programa *buscador.c* realiza la recuperación, con los procedimientos que modelan los algoritmos de inserción, separación y búsqueda del XM-tree. Este programa lee el archivo de datos producido por el generador, construye el árbol dinámicamente y solicita el ingreso de consultas. La aridad *M* del árbol es un valor fijo del programa. Esta implementación le permite al usuario utilizar los operadores booleanos AND, OR y NOT en la consulta. El operador por defecto es AND y el orden de precedencia es NOT, OR y AND. Las consultas ingresadas se transforman a su equivalente en formato de estrategia disyuntiva. Los elementos que componen la estrategia obtenida se codifican a un conjunto de vectores *q* con el que se invoca el procedimiento de búsqueda. El posprocesamiento se aplica sobre los resultados obtenidos del XM-tree, obteniendo las páginas que satisfacen la consulta del usuario. Esto incluye la verificación de la no ocurrencia de los términos, y sus variantes morfológicas, precedidos por NOT, que son ignorados en los vectores *q*.Las palabras de la consulta que no figuran en el diccionario son informadas al usuario dado que su existencia afecta al óptimo rendimiento del sistema porque no se consideran en el recorrido del índice y por esto se obtienen resultados irrelevantes. Para citar el caso más extremo, una consulta sin términos del diccionario se codifica en vectores *q* nulos, los cuales no ocasionan ninguna poda y la búsqueda culmina con un chequeo exhaustivo sobre todas las páginas indexadas. Por esto, el posprocesamiento también incluye considerar términos que no figuran en el diccionario y por ende, no están en el XM-tree.

Con el fin de ofrecer un mecanismo simple de ranking, los resultados se ordenan de acuerdo al máximo producto interno con los vectores *q*, de mayor a menor.

Una vez procesada la consulta, el sistema presenta datos de las siguientes tres tipos de búsquedas: la búsqueda del XM-Search, es decir en el XM-tree y con el posprocesamiento; una búsqueda exhaustiva, con la consulta original sobre todas las páginas; y una búsqueda que emula un sistema de índices invertidos. La búsqueda exhaustiva corresponde a la aplicación de la consulta ingresada al texto de cada página. La última búsqueda encuentra los datos que recupera en una primera instancia un sistema con índices invertidos, o sea aquellas páginas que figuran en las posting lists de los términos ingresados. En una segunda instancia, el sistema resuelve la estrategia de consulta sobre tales páginas quedándose solamente con aquellas que verifican la lógica booleana de la consulta, y las retorna al usuario.

Los resultados de las tres búsquedas, con información de cada página, se muestran en pantalla y se vuelcan en un archivo HTML. El árbol construido se puede visualizar en otro archivo HTML.

5. EXPERIMENTACIÓN

El objetivo de la experimentación es mostrar la performance del sistema XM-Search en un ambiente similar al entorno web real. Para simular de la mejor manera posible la heterogeneidad temática que caracteriza a la Web en una muestra reducida, se recolectaron páginas web reales que traten varios temas distintos pero que comparten algunos términos significativos. Los documentos que componen la muestra se obtuvieron mediante consultas realizadas en Google. Un primer subconjunto de documentos fue recuperado con la consulta “cáncer”, por tener varias acepciones. Otro subconjunto de páginas se recuperó con la consulta “hierro”. El cuerpo final de documentos contiene poco más de 200 páginas, es suficientemente representativo porque se obtuvieron de la web real, la gran mayoría concierne a 8 tópicos, y a su vez, algunos de estos temas pueden estar relacionados entre sí por compartir términos. El diccionario es un archivo de texto donde cada línea contiene una entrada con sus variantes morfológicas. Se construyó un diccionario de 1000 entradas, donde la mayoría

pertenece a las áreas Medicina, Astronomía, Astrología y Geografía. También se incluyeron algunas entradas sobre Biología, Química, Turismo, Literatura y Cinematografía. Considerando las variantes morfológicas de cada entrada, el diccionario agrupa unos 7000 términos.

5.1. Funcionamiento del XM-Search

Supongamos que un usuario desea recuperar información sobre el cáncer de pulmón. Podría hacerlo a través de la consulta "cáncer pulmón". De esta forma, el usuario indica que desea recuperar todos los documentos que contengan la palabra "cáncer" y la palabra "pulmón". Esta consulta se expande utilizando el refinamiento semántico propuesto en [7] agregando términos relacionados y variantes morfológicas a cada uno de los términos ingresados, a fin de mejorar la recuperación. Se obtiene entonces la siguiente estrategia de búsqueda:

(cáncer OR tumor) AND (pulmón OR pulmones OR pulmonar)

Esta estrategia satisface el formato de estrategia conjuntiva aceptado por el prototipo. En la Figura 1 se muestra la pantalla con la información sobre el procesamiento de la consulta.

```
1   consulta recibida : ( cancer OR tumor )  AND ( pulmon OR pulmones OR pulmonar )
2   consulta enviada al XM-tree: ( cancer AND pulmon ) OR ( cancer AND pulmones ) OR
3                                ( cancer AND pulmonar ) OR ( tumor AND pulmon ) OR
4                                ( tumor AND pulmones ) OR ( tumor AND pulmonar )
5                        (*) terminos que no figuran en el diccionario
6   resultados en     : cancer OR tumor AND pulmon OR pulmones OR pulmonar.htm
7   === BUSQUEDA EN XM-SEARCH ========================================
8   estructura              : 19 ints, 59 hojas, 224 datos, total 302
9   revisados               : 10 ints (52.63 %) 16 hojas (27.12 %) 86 datos (38.39 %)
10  podados                 : 30              errores    : 0
11  Resultados XM-tree      : 31 (13.84 %)    Precision : 1.00   Recall : 1.00
12  Resultados XM-Search : 31 (13.84 %)       Precision : 1.00   Recall : 1.00
13  === COMPARACION CON BUSQUEDA CON INDICE INVERTIDO ================
14  Resultados                          : 176 (78.57 %)      Precision: 0.18
15  no recuperados por el XM-tree       : 145
16  no recuperados por el indice invertido  : 0
17  === COMPARACION CON BUSQUEDA EXHAUSTIVA ========================
18  Resultados                          : 31 (13.84 %)
19  no recuperados por el XM-tree       : 0
20  no recuperados por la busqueda exhaustiva  : 0
```

Figura 1: Información del procesamiento de la consulta

En la línea 1 se muestra la consulta recibida por el XM-Search. En la línea 2 se muestra la consulta enviada al XM-tree. Los términos que no hubiesen figurado en el diccionario se habrían señalado con '*' (línea 5), algo que no se da en este caso. En la línea 6 se informa el archivo HTML donde se envía toda la información de las tres búsquedas, incluyendo los listados de páginas obtenidos por cada una. En las líneas siguientes se presentan datos de la búsqueda con el XM-Search, la búsqueda que emula un sistema de índices invertidos, y la búsqueda exhaustiva.

En las líneas 7 a 12 se muestra la información referida a la búsqueda con XM-Search. En *estructura* se informa la cantidad de nodos internos, nodos hoja y datos del árbol construido. La cantidad de datos coincide con el total de documentos indexados, en este caso 224. Para analizar la eficiencia de la búsqueda en el árbol se muestran: cantidad y porcentaje de nodos *revisados* y cantidad de nodos *podados* (líneas 9 y 10). La revisión de un nodo implica el cálculo de distancias. Un bajo porcentaje de nodos revisados indica un buen agrupamiento de los datos similares en las hojas del árbol, por lo que el algoritmo de búsqueda encamina la consulta hacia pocas hojas y poda muchos nodos. La

cantidad de nodos podados corresponde al número de subárboles descartados por el algoritmo siguiendo los criterios de poda, o sea, nodos internos y hojas sin datos relevantes. En este caso es bastante alto, 30 podas sobre 78 nodos, por la alta especificidad de la consulta en la colección. Aquellas páginas que verifican el criterio y no fueron retornadas en la búsqueda, o que no lo verifican y sí fueron retornadas, son falencias del proceso de construcción del índice. Se indica con el título *errores* (línea 10). Aquí, el número de errores es nulo porque la búsqueda en el árbol recuperó correctamente todos los datos que verifican el criterio de similitud L_∞ con la consulta. Los indicadores Precisión y Recall de la recuperación XM-tree en sus valores óptimos reflejan la alta efectividad de la búsqueda en dicho índice: obtuvo todos y sólo los documentos relevantes, como se verifica en la comparación con la búsqueda exhaustiva. Por esta razón, el posprocesamiento no tuvo que descartar datos y la búsqueda principal del XM-Search mantiene exactamente los mismos resultados en Precisión y Recall. La búsqueda que emula un sistema de índices invertidos (líneas 13 a 16) obtiene 176 datos correspondientes a las posting lists de los términos ingresados, los cuales posteriormente se someten a las operaciones de conjuntos, quedando solamente aquellas páginas que verifican la lógica booleana de la consulta. Este valor se compara con el número de resultados XM-tree previos al posprocesamiento, en este caso 31. La intención de esta comparación es verificar la mayor eficiencia del XM-tree sobre los índices invertidos, ya que indexa sobre varios términos a la vez y recupera menos datos irrelevantes. Como el resultado obtenido es mucho menor verifica el mejor desempeño del XM-tree para la consulta ingresada.

La búsqueda exhaustiva (líneas 18 a 20) corresponde a la aplicación de la consulta al texto de cada página. El objetivo es mostrar la efectividad del XM-Search como sistema de recuperación de información en la web. Esto permite también corroborar las aptitudes del esquema TF como mecanismo de representación de páginas, del criterio de similitud L_∞ entre página y consulta, y el XM-tree como método de indexado. Se muestran la cantidad de resultados diferentes entre esta búsqueda y la principal. El número de resultados aquí obtenidos se considera número total de documentos relevantes en la colección y se utiliza para calcular los indicadores Precisión y Recall.

En la Figura 2 se muestran los primeros 10 resultados, según el ranking proporcionado por el programa. Para cada resultado se muestra: docID, título y URL.

69	MedlinePlus: Cánceres - www.nlm.nih.gov/medlineplus/spanish/cancers.html
7	Resources:Health Fact Sheets::Cáncer del pulmón www.hispanichealth.org/lung_cancer_span.lasso
162	Clinica Universitaria de Navarra: Cáncer - www.cun.es/areadesalud/enfermedades/cancer/
60	Tipos de cáncer - www.noah-health.org/es/cancer/types/
145	El Cáncer - www.noah-health.org/es/cancer
135	Inicio Cartilade - www.Cartilade.com
115	OncoLink | El primo recurso de cáncer de la Web - es.oncolink.org/
36	ABC del cáncer - www.arsys.es/usuarios/mariano/index.htm
24	Al sitio Web en Español del Centro de Investigaciones del Cáncer Fred Hutchinson - www.fhcrc.org/portal/es/
114	ACS :: ¿Qué es? www.cancer.org/docroot/ESP/content/ESP_1_1X_Que_es_general.asp

Figura 2: Primeros 10 resultados de la consulta

5.2. Análisis de la performance del XM-Search

Para la comparación de la performance del XM-Search propuesto y un sistema de índices invertidos, se realizaron las siguientes consultas:

c_1 = (cáncer OR tumor) AND (pulmón OR pulmones OR pulmonar OR pulmonares)
c_2 = horóscopo cáncer
c_3 = (constelación OR constelaciones) AND cáncer

c_4 = isla hierro
c_5 = (elemento OR elementos) AND hierro
c_6 = (cáncer OR tumor) AND (cerebro OR cerebral OR cerebrales)
c_7 = trópico cáncer
c_8 = metal hierro
c_9 = josé hierro
c_{10} = película hierro
c_{11} = turismo isla hierro
c_{12} = playas isla hierro
c_{13} = (poesías OR poemas) josé hierro
c_{14} = predicciones amor trabajo cáncer horóscopo
c_{15} = (estrella OR estrellas) constelación cáncer

Estas consultas atañen a los principales temas de la colección, incluso algunas son muy específicas dentro de un mismo tema. En la Figura 3, estas consultas están numeradas de 1 a 15 en las abscisas. Para cada una se muestran los siguientes valores en las ordenadas:

- el porcentaje de datos recuperados por un sistema de índices invertidos (*busq ii*),
- el porcentaje de datos revisados por el algoritmo de búsqueda del XM-tree (*busq xmt*),
- el porcentaje de datos recuperados por el algoritmo de búsqueda del XM-tree (*resul xmt*),
- el porcentaje de datos recuperados por el XM-Search (*resul xms*),
- el número de nodos podados por la búsqueda en el árbol (*podas*).

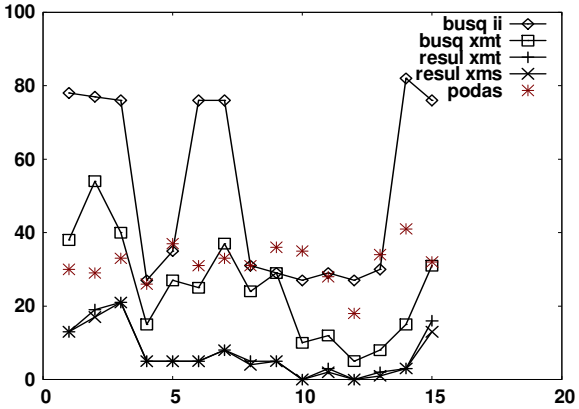

Figura 3. Porcentajes de datos revisados, resultados y podas por consulta.

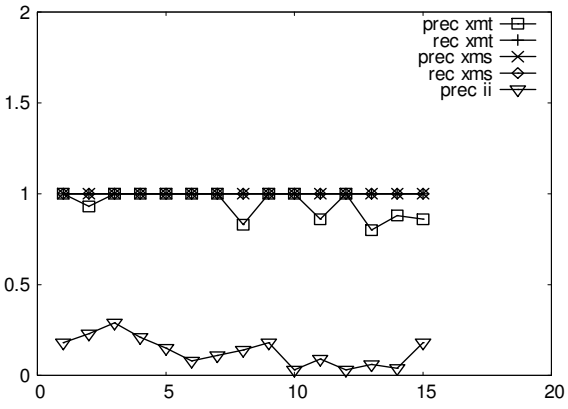

Figura 4. Precisión y Recall del XM-tree, XM-Search e índices invertidos por consulta.

En dicha figura se observa que la cantidad de datos revisados por las búsquedas del XM-tree y del XM-Search es proporcional a la cantidad de resultados relevantes en cada consulta. Esto se debe al agrupamiento de documentos similares en las hojas del XM-tree, con lo cual solamente se revisan las hojas con documentos próximos a la consulta. Esto no ocurre en los índices invertidos, donde la cantidad de datos revisados depende directamente del término ingresado con más ocurrencias en la colección, y que, por tal motivo, contiene posting lists de gran tamaño. Por ejemplo, notar que la consulta c_3 es mucho menos específica que c_{14}, por lo que los porcentajes de documentos en la colección que las satisfacen son muy distintos (21,88% y 3,12% respectivamente). Esto es reflejado por el XM-tree, que revisa el 40,18% y 15,18% en cada caso. Mientras que un sistema con índices invertidos debe procesar el 76,34% y 82,59% de la colección, para c_3 y c_{14} respectivamente, sin importar el grado de especificidad de ambas, sino por el solo hecho de poseer un término muy frecuente: "cáncer". Más aún, revisa más datos para c_{14} siendo más específica, pero posee más palabras. La menor revisión de datos irrelevantes en XM-Search también es verificada por el número de nodos podados, las consultas con mayores cantidades de podas, por ejemplo c_5, c_9, c_{10} y c_{14}, son muy específicas en la colección. Resumiendo, en cuanto a la revisión de datos de la

colección durante una búsqueda, el sistema propuesto efectúa un gasto proporcional a la cantidad de documentos relevantes a la consulta dada.

En la Figura 4 para cada consulta se muestran: Precisión y Recall de la búsqueda en el árbol, Precisión y Recall finales del sistema, y Precisión de las posting lists obtenidas del índice invertido. Los resultados verifican la eficacia del sistema propuesto. Todos los indicadores alcanzan el valor óptimo salvo pocos casos de Precisión de la búsqueda XM-tree. Estos casos se deben a que se recuperan del árbol documentos sin algunos de los términos ingresados, pero que poseen sus variantes morfológicas, y tales documentos son removidos por el proceso de filtrado. La baja Precisión de las posting lists corrobora nuevamente el mejor desempeño del XM-Search respecto a los índices invertidos.

6. CONCLUSIONES

El sistema XM-Search presentado en este trabajo consiste en un motor de búsqueda implementado sobre un XM-tree. Esta estructura indexa documentos web representados en el esquema TF, emplea como distancia de indexado L_2 y resuelve las búsquedas aplicando L_∞ como criterio de similitud entre consulta y documento. El algoritmo de búsqueda utilizado también resuelve consultas formuladas en lógica booleana.

El XM-Search logra un alto rendimiento en cuanto a calidad de los resultados en la recuperación de información en la web, alcanzando buenos valores de Precisión y Recall. Además, en cuanto a la eficiencia de las búsquedas, ofrece importantes mejoras sobre los métodos de búsqueda en espacios métricos y sobre los motores de búsqueda con índices invertidos. Respecto a los espacios métricos, el índice del XM-Search agrupa de una manera adecuada los documentos lo que permite recorrer sólo los próximos a la consulta. Y en relación a los índices invertidos, revisa una fracción de la colección proporcional al conjunto de documentos relevantes.

Referencias

[1] Deco, C., Pierángeli, G., Bender, C. Reyes, N. XM-Tree: A new index for web information retrieval. Journal of Computer Science and Technology (JCS&T). Vol. 8, Nro 2, pp 78-84, 2008

[2] Baeza-Yates, R., Ribeiro-Neto, B. (eds.), Modern Information Retrieval. ACM Press, New York, 1999

[3] Brin, S., Page, L. The Anatomy of a Large-Scale Hypertextual Web Search Engine. In Proc. Seventh International World Wide Web Conference, Vol. 30 of Computer Networks and ISDN Systems, pp. 107-117, 1998

[4] Page, L., Brin, S., Motwani, R., Winograd, T. The PageRank Citation Ranking: Bringing Order to the Web. Technical report, Stanford Digital Libraries, 1998

[5] Giles, C., Bollacker, K., Lawrence, S. CiteSeer: An Automatic Citation Indexing System. In Ian Witten, Rob Akscyn, and Frank M. Shipman III, editors, Digital Libraries 98 - The Third ACM Conference on Digital Libraries, pp. 89-98, Pittsburgh, PA, ACM Press, 1998

[6] Ciaccia, P., Patella, M., Zezula, P. M-tree: An Efficient Access Method for Similarity Search in Metric Spaces. In Proc. 23rd Int. Conf. on Very Large Data Bases, Greece, pp. 426-435,1997

[7] Deco, C., Bender, C., Pierángeli, G., et al. Una interfaz para mejorar la búsqueda de información en la web mediante la expansión semiautomática de la consulta. 34º Jornadas Argentinas de Informática e Investigación Operativa, Argentina, 2005