Departamento de Lenguajes y Ciencias de la Computación

UNIVERSIDAD DE MÁLAGA

Recomendación personalizada de documentos en sistemas de recuperación de la información basada en objetivos

Tesis doctoral presentada por D. David Bueno Vallejo para optar al grado de Doctor Ingeniero en Informática

Dirigida por el Dr. D. Ricardo Conejo Muñoz, Profesor Titular de Universidad de Área de Lenguajes y Sistemas Informáticos, y el Dr. D. Amos A. David, Maître de Conférences de la Universidad Nancy 2

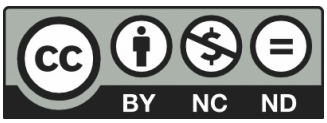
Málaga, Mayo de 2002

UNIVERSIDAD DE MÁLAGA

AUTOR: David Bueno Vallejo

http://orcid.org/0000-0003-3799-6692

EDITA: Publicaciones y Divulgación Científica. Universidad de Málaga



DEPARTAMENTO DE LENGUAJES Y CIENCIAS DE LA COMPUTACIÓN
UNIVERSIDAD DE MALAGA

D. Ricardo Conejo Muñoz, Profesor Titular de Universidad de Área de Lenguajes y Sistemas Informáticos, y **D. Amos A. David**, Maître de Conférences de la Universidad Nancy 2

CERTIFICAN:

Que D. David Bueno Vallejo, Ingeniero en Informática, ha realizado en el departamento de Lenguajes y Ciencias de la Computación de la Universidad de Málaga, bajo su dirección, el trabajo de investigación correspondiente a su Tesis Doctoral titulada:

Recomendación personalizada de documentos en sistemas de recuperación de la información basada en objetivos

Revisado el presente trabajo, estiman que puede ser presentado al tribunal que ha de juzgarlo.

Y para que conste a efecto de lo establecido en el artículo 8 del Real Decreto 778/1998, autorizan la presentación de esta tesis en la Universidad de Málaga.

Málaga, 25 de marzo de 2003

Fdo.: Ricardo Conejo Muñoz

Fdo.: Amos A. David

Dedicado a Cristina y a mis padres

ÍNDICE

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

ÍNDICE DE ECUACIONES

AGRADECIMIENTOS

Esta parte de la tesis es el lugar más indicado para expresar a las personas que siempre han estado a tu lado cosas que no suelen decirse. En primer lugar quisiera agradecer a Ricardo Conejo, no sólo por haberme dirigido esta tesis, sino por haber sido mi mentor desde que comencé en la Universidad con mi primera beca. Es alguien que me ha dado la confianza suficiente para poder hablarle de cualquier tema pudiendo expresarle con libertad mis ideas. A mi otro director de la tesis Amos David, le agradezco los esfuerzos para que pudiera realizar mis estancias en Nancy, donde me ha tratado como a alguien de su familia. Con él siempre he sentido ante cualquier problema que todo estaba controlado, lo que siempre me inspiró confianza y tranquilidad.

A Beatriz Barros y Maite Urretavizcaya por sus revisiones detalladas de la memoria que contribuyeron a mejorar la calidad de la misma.

A José Luis Pérez de la Cruz que me ha apoyado en los momentos más críticos de la tesis depositando su confianza en mí. También expreso mis agradecimientos a Paco Triguero quién a pesar de todas sus ocupaciones siempre se ha preocupado por mi bienestar.

A mis compañeros de grupo Rafael Morales, Marlon Núñez, Mónica Trella, Eduardo Guzman, Eva Millán y Lawrence Mandow por su apoyo.

A mis compañeros de docencia José Luis Pastrana, José Jerez, Antonio Maña, José Mª Álvarez, Juan Falgueras, Llanos Mora y Mª del Mar Gallardo que me han ayudado con las asignaturas y exámenes cuando he tenido que ausentarme por la tesis.

A mis amigos de Nancy, Mónica Hombreiro y Dimitri Samborski con los que año tras año creció nuestra amistad que me gustaría que durara para siempre. También a Philippe Kislin quién se rompió la cabeza en más de una ocasión probando METIORE.

Por último y no menos importante a mi familia. Empezando por Cristina que me ha demostrado su cariño y ser la mejor compañera que pueda tener, quien se ha quedado sola muchos veranos por mis estancias en Nancy, que siempre me ha apoyado e incluso ha desarrollado la interfaz Web de la aplicación.

También quisiera agradecer a mi padre su fe en mí. El ha sido durante toda mi vida un ejemplo a seguir como persona, fiel a sus ideas, un músico y deportista al que siempre he admirado y nunca superado. A mi madre, cuya ternura y cariño han sido algo especial que guardo muy adentro. A mi hermana Elena, la bohemia de la casa, de la que me siento muy orgulloso como pintora. A mis sobrinos Vicente que siente devoción por mí y a Laurita que espero lo sienta pronto. Por último a mi hermano pequeño que siempre viene a recibirme con mucha alegría, a quién me encanta acariciar, tirarle muñecos y darle salchichas (Guau!!!).

RESUMEN

Los sistemas de recuperación de la información surgen de la necesidad del hombre de organizar la información contenida en bibliotecas para poder localizar todos los documentos contenidos en ellas. En la actualidad todo el que se conecta a la Web está en contacto con sistemas como Altavista, Google o Yahoo. El problema de estos buscadores es que a veces devuelven cientos o miles de resultados de los que el usuario sólo va a considerar los 20 o 30 primeros. La ordenación que realizan devuelve los documentos Web que mejor corresponden a la consulta, pero eso no quiere decir que sean los más interesantes para el usuario. Muchas veces el usuario no experto encuentra dificultades en expresar su necesidad de información en el lenguaje de consulta que le ofrece el sistema, con lo que no encuentra los documentos aunque éstos estén en el sistema.

Una posible solución al problema de recuperación de la información viene dada por la aplicación de un modelo del usuario que permita ofrecer una respuesta personalizada de acuerdo a las preferencias y necesidades de cada individuo. Los aspectos más importantes a adaptar estarán relacionados con la ordenación de los resultados de acuerdo a las necesidades del usuario y con las facilidades ofrecidas para realizar las búsquedas.

Esta tesis ofrece resultados relativos al estudio de la personalización en los sistemas de recuperación de la información. Se ofrece una nueva visión de la personalización en estos sistemas orientada a los objetivos del usuario, frente al refinamiento de consultas utilizado en otros sistemas. La mayoría de los sistemas que ofrecen algún tipo de personalización suelen tener una visión general de los intereses de los usuarios y tienen un único modelo para cada usuario. En el enfoque propuesto, un usuario puede tener diferentes intereses que no tienen porque guardar relación. De esta forma, cuando esté interesado en un objetivo, sólo se le ofrecerán documentos relevantes para éste.

En esta tesis se ofrecen algoritmos probabilísticos que permiten aplicar personalización basada en objetivos a distintas bases de datos multimedia o documentales, teniendo en cuenta múltiples parámetros de los datos. Además, se ofrecen mecanismos que permitan hacer análisis complejos de la base de datos sin necesidad de conocer su contenido. En

el caso en que el usuario no sea capaz de encontrar documentos que le satisfagan, se ha desarrollado una arquitectura para que el usuario encuentre un apoyo de otros usuarios del sistema a través de Internet.

Por último se ha comprobado la validez de la aportación mediante la implementación de un sistema, METIORE, que incluye las propuestas anteriores y que ha sido evaluado en entornos de aplicaciones convencionales y en entornos webs aplicados a diferentes bases de datos de referencias bibliográficas.

I. INTRODUCCIÓN

En esta introducción se mostrará el objetivo que se ha perseguido con el desarrollo de esta tesis y las motivaciones que han hecho disfrutar al autor realizando este trabajo. Todo se podría reducir en una sola palabra: "Personalización". Algo con lo que se vive a diario pero fuera de los ordenadores. A continuación se ilustrará esta afirmación con varios ejemplos.

Cuando un paciente llega al médico con síntomas de un resfriado, no puede mandarle de forma automática tomarse unas aspirinas para el mal cuerpo. Antes de recetar nada, el médico consulta el historial del paciente donde pueden ver sus enfermedades anteriores; por ejemplo no se le debería dar aspirinas si tiene alguna enfermedad hepática, o es alérgico al ácido acetilsalicílico, componente principal de las aspirinas. Por lo tanto, para recetarle un medicamento, tendrá en cuenta factores exclusivos de esa persona.

En el buzón se encuentra todos los días mucha publicidad sobre temas que no importan demasiado, lo que hace que el coste de esa publicidad haya sido para nada. Las empresas han empezado a pensar en la publicidad personalizada, es decir, darle al usuario lo que quiere ver y lo que potencialmente podría comprar. Las tarjetas de fidelidad por comprar dan puntos que posteriormente se podrán canjear por algún regalo, pero al pasar esa tarjeta, queda registrado con detalle todas las fechas, artículos y cantidades de la compra. El análisis de esos datos podría utilizarse para generar publicidad sobre artículos que realmente interesan a cada usuario en concreto. Este ejemplo de alimentos puede aplicarse en otros tipos de ventas como libros, ordenadores, etc.

Cuando se accede a una biblioteca para buscar un libro, si es la primera vez que se hace, el bibliotecario necesitará alguna información sobre el cliente para poder ayudarle en lo que busca y recomendarle algunos. Si ese mismo cliente vuelve de forma regular y comenta con el bibliotecario si las recomendaciones que éste le hizo le resultaron interesantes o no, podrá hacerse una idea de las preferencias y necesidades concretas de ese cliente, por lo que en futuras recomendaciones tendrá en cuenta el *modelo* que se ha hecho de esa persona.

Otro ejemplo interesante es el de los periódicos en la Web, y en general cualquier servidor Web que ofrezca gran variedad de contenidos. Cada persona está interesada en algún tipo de noticias, a algunos les gusta el fútbol, otros están interesados en noticias económicas, etc. Una misma persona puede estar interesada en diversos temas. ¿Cómo podría la personalización influir para estas personas? Una posible idea sería que cuando una persona accede al periódico, en función de las noticias del día, las más importantes para este usuario apareciesen en primera página.

Cuando se habla de personalización y de obtener información de un individuo para analizarla y obtener conclusiones sobre algún aspecto de esa persona, se crítica que se está invadiendo la intimidad de la persona. Es cierto que para ámbitos comerciales la información de la persona puede ser utilizada con diversos motivos, por eso hay tanta polémica con la publicación de datos, aunque a veces el usuario puede ser el interesado en ofrecer información sobre lo que necesita para poder ser mejor atendido. Cumpliendo unas mínimas normas éticas no debería ser un problema que mediante interacciones con él, se extraigan algunas conclusiones sobre sus preferencias o intereses que además van a ser utilizadas en su beneficio.

El campo de la personalización es muy amplio y puede abarcar todo ámbito relacionado con la interacción con personas. Se centrará el estudio de esta tesis en la personalización asociada a los sistemas de recuperación de la información.

Según la definición de [Kowalski2001]:

> "*Un sistema de recuperación de la información es aquel capaz de almacenar, recuperar y mantener la información. La información en este contexto puede estar compuesta de texto, imágenes, audio, video y otros objetos multimedia*".

Este tipo de sistema surge de la necesidad del hombre de organizar la información contenida en bibliotecas para poder localizar todos los documentos contenidos en ellas. Para desarrollarlos es necesario realizar tareas de cierta complejidad como indexar los documentos, extrayendo elementos clave para su localización o crear estructuras de almacenamiento que permitan realizar búsquedas eficientes sobre la base de datos.

En la actualidad todo el que se conecta a la Web esta en contacto con sistemas de recuperación de la información mediante los buscadores, como Altavista, Google o Yahoo. El problema de estos buscadores es que a veces devuelven cientos o miles de resultados de los que el usuario solo va a mirar los 20 o 30 primeros. La ordenación que realizan devuelve los documentos Web que mejor corresponden a la consulta pero eso no quiere decir que sean los mejores para el usuario. Muchas veces el usuario no experto, encuentra dificultades en expresar su necesidad de información en el lenguaje de consulta que le ofrece el sistema, con lo que no encuentra los resultados aunque estos estén en el sistema.

La solución a esos problemas en los sistemas de recuperación de la información, la dificultad en el interfaz de búsqueda o la gran cantidad de soluciones, viene dada por el modelado del usuario. Según Judy Kay [Kay1995b] *"Como cada individuo tiene diferente conocimiento, preferencias y necesidades, hay muchas situaciones donde personalizar las interacciones con un usuario basándose en la información de un modelo de ese usuario debería ofrecer ventajas"*. Continúa definiendo informalmente

el concepto de modelo de usuario como *"un conjunto de creencias sobre el usuario"*. Una definición más formal puede encontrarse en el prefacio de la conferencia de modelado del usuario del 2001 [Bauer2001]:

> *"Un modelo de usuario es una representación explícita de las propiedades de un usuario particular, las cuales permiten a los sistemas adaptar diversos aspectos de su funcionamiento a las necesidades individuales de los usuarios"*

En concreto para el problema de recuperación de la información los aspectos más importantes a adaptar estarán relacionados con la ordenación de los resultados de acuerdo a las necesidades del usuario y con las facilidades ofrecidas para realizar las búsquedas.

I.1. OBJETIVO DE LA TESIS

El objetivo fundamental de esta tesis es el de resolver problemas que se han observado en los sistemas de recuperación de la información. Estos problemas son, por un lado, la necesidad de una respuesta personalizada para evitar la impotencia que siente un usuario cuando realiza una búsqueda y recibe cientos de soluciones que aunque se correspondan con su consulta. Por ejemplo, por coincidencia de palabras clave, son demasiadas para poder examinarlas todas, y muchas de ellas no se ajustan con lo que el usuario busca. Por otro lado, aunque hay sistemas que ofrecen personalización, ésta es demasiado general y con un único modelo de usuario se hacen todas las recomendaciones de documentos, pensando que el usuario siempre tiene las mismas necesidades. Esto lleva a que el usuario reciba recomendaciones de documentos que no les interesan en ese momento. Dados esos problemas se podría definir el objetivo de la tesis como:

> *"Ofrecer una arquitectura y los algoritmos necesarios que permitan implementar sistemas de recuperación de la información que ofrezcan respuesta personalizadas a los usuarios en función de sus necesidades/objetivos en cada momento"*

Para lograr dicho objetivo, se ofrecen resultados relativos al estudio de la personalización en los sistemas de recuperación de la información. Entre las contribuciones de la tesis cabe destacar:

i. Realización de un estudio detallado de las diferentes técnicas utilizadas en los sistemas de recuperación de la información y de la aplicación del modelado del usuario a las mismas para comprender las carencias del área.
ii. Ofrecer una nueva visión de la personalización en los sistemas de recuperación de la información orientada a los objetivos del usuario y no a refinamiento de consultas como en otros sistemas.
iii. Algoritmo que permita aplicar ese tipo de personalización basada en objetivos a distintas bases de datos multimedia teniendo en cuenta múltiples parámetros de los datos.
iv. Mecanismos que permitan hacer análisis complejos de la base de datos sin necesidad de conocer su contenido.
v. Arquitectura necesaria para que el usuario, en caso de no encontrar soluciones que le satisfagan encuentre un apoyo de otros usuarios del sistema a través de Internet.

vi. Comprobar la validez de la aportación mediante la implementación de un sistema que incluya las propuestas anteriores y analizar los resultados de las interacciones de los usuarios.

I.2. ESTRUCTURA DE LA MEMORIA

Esta memoria se organiza en siete capítulos, tres apéndices y un conjunto de referencias. La línea argumental se basa en exponer las dificultades de los sistemas de recuperación de la información para ofrecer los resultados que los usuarios necesitan. Para ello se analizan las características de estos sistemas y se continúa estudiando el campo de modelado del usuario como una posible solución para satisfacer al usuario mediante la personalización de las respuestas. En este marco, se ofrece una posible solución al problema utilizando algoritmos probabilísticos.

En el capítulo segundo se introducen los conceptos y características de los sistemas de recuperación de la información. Abordando las distintas aproximaciones: Booleana, Vectorial, Probabilística y Lógica. Se mostrarán las técnicas utilizadas en cada caso para devolver los documentos más pertinentes a cada consulta. Además, se realizará un estudio comparativo de diferentes sistemas reales.

El capítulo tercero se dedica al modelado del usuario. Se mostrarán las técnicas clásicas y generales de esta área. A continuación se profundizará en los sistemas recomendadores, que en general podrían considerarse la aplicación de la personalización a los sistemas de recuperación de la información. Se analizarán diversos trabajos destacando sus características más representativas.

El capítulo cuarto marca el comienzo de una serie de capítulos donde se recogen las contribuciones de la tesis. En este punto, se presentan las aportaciones teóricas a la personalización para la recuperación de la información realizadas alrededor del concepto de objetivo del usuario y no de consultas como se realiza de forma tradicional.

En el capítulo quinto se muestra la aplicación de los conceptos teóricos en el sistema METIORE desarrollado para esta tesis. Este es un sistema de recuperación de la información que ofrece respuestas personalizadas. Se detallará su arquitectura, sus modos de funcionamiento y las estructuras internas utilizadas para conseguir personalizar respuestas.

En el capítulo sexto se resumen los experimentos realizados con METIORE. Se compara el algoritmo con otros similares y se muestran los resultados de pruebas con usuarios para dos bases de datos e interfaces diferentes (aplicación y Web).

El último capítulo resume las conclusiones de este trabajo y las posibles líneas futuras.

Finalmente, el apéndice I resume algunos aspectos de la notación y terminología empleadas. El apéndice II muestra la utilidad de METIORE para realizar análisis bibliométricos y el apéndice III muestra las tablas utilizadas para comparar diferentes versiones del algoritmo Naïve Bayes.

II. RECUPERACIÓN DE LA INFORMACIÓN

II.1. INTRODUCCIÓN

Este primer capítulo de antecedentes trata sobre los sistemas de recuperación de la información útiles para tareas como obtener información sobre los libros de una biblioteca, o para buscar información en la Web. En los siguientes apartados se introducirán algunos conceptos necesarios para comprender cómo funcionan estos sistemas. Se ilustrarán las técnicas más utilizadas para indexar y almacenar los documentos, además se comentarán distintas técnicas para realizar una correspondencia entre la consulta que un usuario realiza para buscar información, con los documentos que el sistema contiene. Posteriormente se detallarán las bases para la recuperación de la información probabilística y vectorial. Se ilustrarán los métodos clásicos para evaluar sistemas de recuperación. A continuación se hablará de los sistemas de filtrado de la información que comparten muchas de las características con los de recuperación. Para terminar se mostrarán diferentes ejemplos reales de sistemas de este tipo incluyendo una tabla resumen y las conclusiones del capítulo. Las ideas aquí expuestas se completarán en el capítulo siguiente en el que se mostrará cómo personalizar la respuesta en estos sistemas mediante el modelado del usuario.

II.2. CONCEPTOS GENERALES

Un Sistema de Recuperación de la información es aquel capaz de almacenar, recuperar y mantener información. Esta información puede estar compuesta de texto, imágenes, audio, video y otros elementos multimedia.

Con objetivo de unificar la nomenclatura, en esta tesis se entenderá por *documento* cualquier objeto que puede utilizarse como elemento principal en un sistema de recuperación de la información. Por ejemplo, un documento textual (articulo, libro, revista, correo electrónico, noticia...) o un documento multimedia (imagen, video, sonido,...). Para describir un documento se pueden utilizar diferentes *parámetros*. Un parámetro representa alguna característica del documento. Por ejemplo, en un artículo, posibles parámetros son el autor, titulo, palabras clave, etc. Se llamará *término* al valor concreto de un parámetro. La mayoría de los sistemas de recuperación de la información

utilizan como parámetro exclusivamente las palabras clave del documento que podrán haber sido obtenidas de forma manual o automática.

El proceso de recuperación de la información (*Information Retrieval-en adelante IR*) consiste en devolver documentos relevantes para el usuario como respuesta a una consulta. Para ello habrá que clasificar los documentos a buscar para posteriormente calcular la similitud entre una consulta y un documento.

La investigación en IR se ha realizado durante mucho tiempo por una pequeña comunidad, básicamente en centros de documentación, con poco impacto en la industria. La mayoría de las aplicaciones de recuperación de documentos estaba enfocada a las bases de datos bibliográficas. Estos sistemas estaban basados en aproximaciones de la lógica booleana estándar sin prestar mucha atención a los resultados de investigación como modelos de recuperación, proceso de las consultas, valoración de los términos (*weighting*) y evaluación pertinente (*relevance feedback-en adelante RF) [Croft1995]*.

El crecimiento de Internet y la gran cantidad de datos que existen actualmente en la red, han hecho que parte importante de la investigación sobre recuperación de la información se centre en este ámbito. Uno de los principales problemas es que la información no está muy bien organizada, y que hay muchos documentos que se recuperan con los sistemas de búsqueda actuales que no son relevantes para los usuarios. En la actualidad, es mucho más frecuente hacer búsquedas en este entorno que en una biblioteca, por lo tanto las técnicas de recuperación de la información deben reorientarse en este sentido.

Es importante destacar las diferencias existentes en IR dependiendo del punto de vista académico o comercial. La primera no se preocupa de tiempos de respuestas, recursos necesitados, costes de mantenimiento, etc, aunque por otro lado estudia todos los aspectos de los sistemas de información, investiga teorías, algoritmos y heurísticos. Por otro lado las empresas no se preocupan tanto de la aproximación teórica sino que tratan de minimizar costes e incrementar las ventas de sus productos.

Un aspecto que generalmente es motivo de confusión es comprender las diferencias entre un Sistema Gestor de Bases de Datos (SGBD) y un Sistema de Recuperación de la Información (IRS). La principal diferencia está en que un SGBD es capaz de manejar eficientemente datos estructurados y un IRS ofrece funciones para procesar información. Los datos estructurados son datos bien definidos, normalmente representados por tablas. Cada dato simple (campo) tiene una definición semántica asociada, y difícilmente pueden confundirse distintos campos de una tabla. Por otro lado, para realizar búsquedas sobre un documento se puede utilizar un resumen. Uno de los problemas que se pueden encontrar consiste en que para la misma descripción distintas personas utilicen diferente vocabulario o hagan hincapié en aspectos diferentes. Esta diversidad y ambigüedad del lenguaje es la que se asocia a elementos de información que se procesarán con un IRS. Esta diferencia en el tipo de datos es lo que hace necesario la existencia de esos dos tipos de sistemas, aunque los SGBDs comerciales recientes empiezan a incluir funciones para trabajar con información no estructurada.

Cuando se trabaja con datos estructurados el usuario introduce una consulta específica y los resultados son devueltos con la información deseada. Normalmente la información aparece tabulada, y todos los documentos que satisfacen la consulta son devueltos. Por otro lado, cuando el usuario está buscando información, por ejemplo en un buscador Web, se le ofrecen una serie de documentos ordenados en función de la probabilidad de que correspondan con la consulta, aunque debido a las características de esa información, y dependiendo de cómo funcione el IRS es posible que haya muchos documentos que aunque sean relevantes para el usuario no se les muestren. Por lo tanto, será necesario realizar nuevas búsquedas, refinando la consulta hasta encontrar documentos relevantes.

La correspondencia es una operación fundamental en la recuperación de la información. Cuando un usuario realiza una consulta, la función de correspondencia es la encargada de relacionar esta consulta con los documentos del sistema y de seleccionar aquellos que cumplan dicha función. Algunas de estas funciones devolverán una lista de documentos sin más información, por ejemplo, la correspondencia booleana que para cada documento decide si satisface la consulta o no, sin hacer distinción entre los documentos seleccionados, al igual que ocurre en las bases de datos. La otra opción consiste en devolver la lista de documentos ordenados por similitud con la consulta, como ocurre en los buscadores Web. Estas funciones dependerán del *modelo* de recuperación utilizado.

Los modelos están relacionados con la forma en la que se van a procesar los datos cuando el usuario realice una consulta para ofrecerle las soluciones que sean relevantes para él. Básicamente hay cuatro modelos según [Rijsbergen2000]: Vectorial, Probabilístico, Lógico y Redes Bayesianas. Además, está el modelo Booleano, que aparece complementando en muchos casos los anteriores. El modelo booleano utiliza el álgebra de Boole para seleccionar los documentos de acuerdo a una expresión booleana. Su único objetivo es decidir si un documento se corresponde o no con la expresión pero no realiza ningún tipo de ordenación de los resultados (*ranking*). Esa ordenación vendrá dada por alguno de los modelos que se muestran a continuación. El sistema SIFT [Yan1995] permite aplicar este modelo además del vectorial.

El modelo Vectorial representa un documento como un vector en el que aparecen los términos que lo representan. Este modelo ha sido muy utilizado y muchos tipos de medidas de similitud entre documentos se han desarrollado para él. Entre ellas la más popular es la correlación del coseno (Ver II.4), y también los coeficientes de Jaccard y Dice [Rijsbergen1979]. El sistema que podría considerarse representante de este modelo es SMART [Salton1971] (detalles en II.9.3).

El modelo Probabilístico hace estimaciones de probabilidades para calcular la relevancia de un documento. En general se suele calcular la probabilidad de que un término aparezca en un documento relevante a una consulta y la probabilidad de que aparezcan en un documento que no sea relevante (son dos espacios probabilísticos diferentes). Para decidir la relevancia o no del documento se tendrán en cuenta todos los términos que lo componen (Ver II.5). Este modelo se aplica en Okapi [Robertson1997] (ver II.9.5).

El modelo lógico ha sido menos desarrollado que los dos anteriores, y asume que un IRS puede verse como una inferencia incierta que relaciona consultas con documentos.

Utiliza la lógica asociada a la certeza/incetidumbre de hechos para manejar implicaciones. Para ello se utiliza lógica de predicados, existiendo dos modelos fundamentales llamados terminológico y Datalog [Fuhr2000]. El reciente sistema HySpirit [Rölleke1999] (comentado en II.9.6) es una muestra de la aplicación del modelo Datalog.

La última aproximación se basa en Redes Bayesianas que utilizan el teorema de Bayes para definir la relación entre nodos y la propagación de certeza. El sistema INQUERY [Callan1992] es el más importante que aplica este modelo.

II.3. DESCRIPCIÓN DE DOCUMENTOS

Para que un documento pueda ser tratado por un IRS debe ser analizado para obtener una descripción que permita indexarlo en el sistema. El proceso de indexación consiste en transformar el documento para obtener la semántica de los temas que se tratan en él. Esa semántica no se limita a conocer el tema del documento, sino también, si el sistema utiliza pesos, a saber con qué profundidad se trata. La indexación puede realizarse utilizando el texto completo, para generar de forma manual o automática un conjunto de términos que representen dicho elemento. El resultado de dicha indexación se suele almacenar en estructuras de datos invertidas que permiten partiendo de un término saber qué documentos lo contienen. Este tipo de estructuras se verá con detalle en II.3.1.

En muchos sistemas, la descripción del contenido de un documento se hace mediante los términos que se extraen del texto. Estos términos se almacenan en un campo para palabras clave, que puede incluir pesos en función de su importancia. Si los documentos están estructurados, se podrán obtener diferentes propiedades para indexarlos. Por ejemplo, para representar libros se pueden utilizar propiedades como titulo, autores, fecha de edición, etc.

Durante mucho tiempo, en bibliotecas, la técnica de indexación de documentos ha sido manual. Esto consiste en que un bibliotecario se encarga de elegir los descriptores para los valores de las propiedades que representan un documento. El principal problema es que dos personas pueden describir el mismo documento con diferentes valores dependiendo, por ejemplo, de la importancia que encuentren a distintas partes del documento. Por otro lado, la indexación automática consiste en analizar el documento mediante un programa que como resultado dirá cuales son los términos que lo representan. Probablemente la mejor solución es una combinación de ambas técnicas, en las que primero se haga una indexación automática que posteriormente sea revisada por un experto humano. La estrategia de indexación no está limitada sólo al nivel de detalle de la descripción. También está relacionada con lo que podría llamarse *cobertura semántica* de descriptores. Donde no sólo se buscan términos utilizados en el documento, sino también aquellos que pueden deducirse por relaciones semánticas a partir de los términos utilizados en el documento. Estos también aparecen de forma explícita en algunos sistemas mediante la utilización de ontologías.

Para la realización de una indexación automática se suelen seguir varios pasos que darán como resultado estructuras de datos que faciliten la búsqueda de la información [Kowalski2001]:

1. *Estandarizar la entrada.*- El primer paso a realizar consiste en transformar los documentos origen a un formato que pueda ser manipulado para la generación de los índices

2. *Zoning*.- Es el proceso de dividir un documento en distintas zonas lógicas que tengan significado para el usuario, por ejemplo: titulo, autor, resumen, texto principal y referencias.
3. *Identificación de términos*.- Para cada zona será necesario obtener la información que va a utilizarse para el proceso de búsqueda. Para obtener estos términos, primero hay que encontrar las palabras y los separadores de palabras. Estas palabras serán los términos iniciales que en pasos posteriores podrán ser refinados.
4. *Algoritmos de parada*.- Una vez que se tienen los términos iniciales, es necesario aplicar este tipo de algoritmos que van a reducir el número de términos válidos. Se van a eliminar todos los términos cuyas frecuencias o semánticas hagan que no sea un elemento interesante para la búsqueda. Se suelen utilizar listas de parada, con elementos comunes del lenguaje como artículos, preposiciones, pronombres, etc. También se eliminan elementos que aparecen sólo 1 o 2 veces en todos los documentos, por considerarse muy probable que no formarán parte del lenguaje del usuario. Esto hace que el almacenamiento de términos sea menor y que el acceso a éstas estructuras en las búsquedas pueda ser más rápido.
5. *Caracterización de los términos*.- Algunos sistemas intentan evitar problemas de ambigüedad haciendo un análisis morfológico. Pues hay palabras que tienen distinto significado dependiendo que sea un nombre, adjetivo o verbo.
6. *Algoritmos de stemming*.- Estos algoritmos normalizan los términos a una representación semántica estándar. Básicamente lo que hacen es guardar las raíces de las palabras, para almacenar como uno sólo, las distintas posibilidades de singular/plural, tiempos verbales, posesivos, etc. Por ejemplo, las palabras pensar, pensando, pensativo, pensamiento, pensamientos podrían almacenarse como elementos con una misma raíz *pens*- esto aumentará su frecuencia dentro de un documento además de minimizar espacio de almacenamiento. De estos algoritmos el más extendido es el algoritmo de Porter [Porter1980] del que existen implementaciones en varios lenguajes e idiomas.
7. *Estructura de datos para la búsqueda*.- Cuando ya se tienen los terminos finales (términos) que van a utilizarse se deben almacenar en alguna estructura que permita utilizar las consultas del usuario para buscar los documentos relevantes con cierta facilidad.

II.3.1 ESTRUCTURAS DE ALMACENAMIENTO

La estructura más utilizada tanto en gestión de bases de datos como en IRS es la estructura de fichero invertido. Esta estructura se compone de tres ficheros básicos: El fichero de documentos, la lista invertida y el diccionario. A cada documento se le da un identificador único en el sistema. Para cada término del sistema, se crea una lista que indica cuales son los documentos en los que éste aparece. El acceso a esta lista se hace a través del diccionario, que básicamente es una lista ordenada de términos en el sistema con un puntero a su lista invertida. Los diccionarios pueden almacenar otra información utilizada para optimizar las consultas, como la longitud de las listas. Si se utilizan zonas, puede haber un diccionario y un conjunto de listas invertidas para cada una de las zonas. Si se quiere soportar proximidad, frases con palabras contiguas o algoritmos que utilicen pesos de los términos dentro de documentos, se pueden almacenar en la lista invertida todas las ocurrencias de un término indicando el documento en el que aparece y su posición en él.

Cuando se realiza una búsqueda se localizan los términos de la consulta en las listas invertidas. El resultado de la consulta será una lista de documentos, formada por

aquellos que aparecen en las listas invertidas de los términos de la consulta. Los identificadores de documentos que aparecen en la lista invertida se utilizan para recuperarlos del fichero de documentos donde se encuentras en su formato original. Como ejemplo, se puede suponer que en el sistema hay 5 documentos. En la Tabla 1 se muestran las palabras clave que contienen y el número de apariciones en cada documento. En la Fig. 1 se muestra un ejemplo de diccionario y lista invertida, asociadas a estos documentos.

	Adaptive Hypermedia	*Information Retrieval*	*Relevant Feedback*	*Tutorial Systems*	*User Modeling*
Document1	3				
Document2		4	2		
Document3	1	5			
Document4				3	4
Document5	2			2	1

Tabla 1. Valores de los términos en varios documentos

Para este ejemplo, en el diccionario se muestra la lista de términos y el número de documentos que lo contienen, y en la lista invertida asociada se muestra para cada término, el identificador de documento y entre paréntesis el número de veces que este término aparece en ese documento.

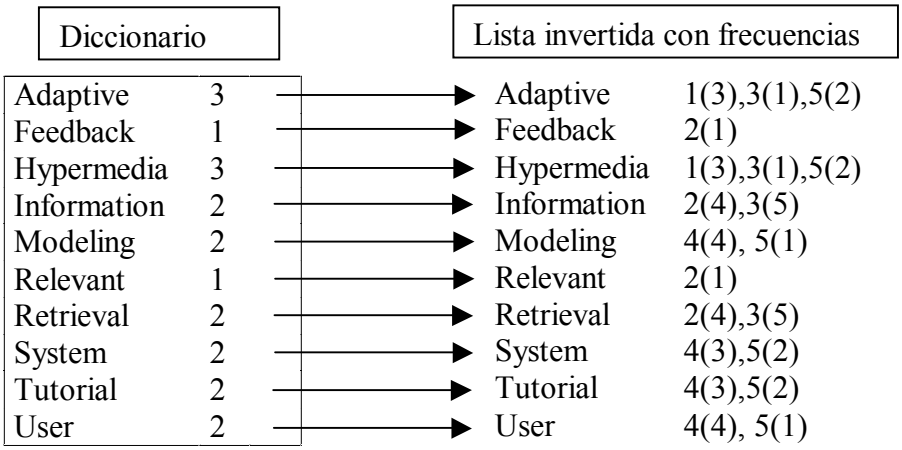


Fig. 1 Ejemplo de Diccionario y lista invertida que indica documento(nºapariciones)

Si se realizara una consulta booleana con las palabras *modeling AND adaptive*, el sistema buscará en las dos listas invertidas los elementos comunes. *modeling* aparece en los documentos 4,5 y *adaptive* en 1,3,5. Por lo tanto una AND de los dos devolverá el documento 5. Cuando se va a devolver más de un documento se aplicará algún algoritmo para ordenarlos por importancia dependiendo del modelo utilizado.

II.3.2 TIPOS DE INDEXACIÓN AUTOMÁTICA

El proceso de indexado es aquel que transforma un documento en una estructura de datos que permita hacer búsquedas eficientes. En este apartado se mostrará cómo se extrae del documento principal la información necesaria para obtener los términos que representaran a un documento. Por ejemplo, cómo se ha llegado a obtener los datos de la Tabla 1.

Hay varios algoritmos que permiten obtener palabras claves de un documento textual. El primero de ellos es el algoritmo de *frecuencia simple de términos*. En este método estadístico, se calcula el número de apariciones de un término dentro de un documento (*Term Frequency*, en adelante TF). Se llama peso de un término en un documento a la importancia que ese término tiene en ese documento. La aproximación más simple para obtener los términos más relevantes de un documento consiste en que el peso de cada término sea su frecuencia dentro del documento (TF). El problema de esta técnica es la normalización entre los distintos documentos de la base de datos, ya que cuanto mayor sea el documento, mayor será el valor de TF para un término. La normalización más sencilla utilizada consiste en obtener el valor de TF(t,d)=TF(t,d)/MAX(TF(x,d)), donde *t* es el término, *d* el *documento* y *MAX(TF(x,d))* es la frecuencia máxima de un término en el documento. Otras normalizaciones más complejas se muestran en [Kowalski2001].

Una mejora de TF se consigue cuando se tiene en cuenta la frecuencia de aparición del término en la base de datos. No deberá tener la misma importancia un término que aparece en pocos documentos de la base de datos, que un término que aparece en casi todos los documentos, pues este último será menos discriminante en una consulta. Esto lleva a la conclusión de que el peso asignado a un término debería ser inversamente proporcional a la frecuencia de aparición de ese término en la base de datos. A esto se le llama *frecuencia inversa de documento (Inverse Document Frequency- IDF)*. Como resumen, en la Ecuación 1 se muestran las ecuaciones que según [Fuhr2000] son las más utilizadas actualmente para IDF, TF y TFIDF que es la combinación de ambos y lo más recomendable para utilizar como peso de un término, ya que tiene en cuenta la importancia del término dentro de su documento (TF) y dentro de la base de datos completa (IDF).

$$idf(t) = \frac{\log \frac{N_d}{df(t)}}{N_d + 1} \qquad \textbf{a)}$$

$$ntf(t,d) = \frac{tf(t,d)}{tf(t,d) + 0{,}5 + 1{,}5\frac{l(d)}{al}} \qquad \textbf{b)}$$

$$tfidf(t,d) = ntf(t,d) \cdot idf(t) \qquad \textbf{c)}$$

Ecuación 1.a) IDF b) TF normalizado c) TFIDF

Donde

al = Longitud media de los documentos de la colección
df(t) = Número de documentos que contienen el término *t*
l(d) = Número de términos del documento *d*
N_d = Número de documentos de la colección
tf(t,d) = Número de apariciones del término *t* en el documento *f*

II.4. TECNICAS DE CORRESPONDENCIA (MEDIDAS DE SIMILITUD)

En este apartado se comentarán diferentes medidas de similitud que pueden utilizarse para relacionar un documento con la consulta de búsqueda. Una función de similitud deberá devolver 0 cuando los dos elementos no tengan nada en común e incrementar su valor conforme aumente su parecido. Se supone que un documento y una consulta se pueden descomponer en una serie de términos (ver Ecuación 2). Los valores de ω_{ij} representan el peso del término $term_j$ en el documento D_i.

$$\vec{D}_i = \omega_{i1} term_1, \omega_{i2} term_2, ..., \omega_{in} term_n$$
$$\vec{Q}_j = a_{j1} term_1, a_{j2} term_2, ..., a_{jn} term_n$$

Ecuación 2. Representación de un documento (D_i) y una consulta (Q_j)

La ecuación de similitud más sencilla es la *suma de productos* que se muestra en la Ecuación 3. Como puede verse puede utilizarse para comparar dos documentos y ver su similitud, pero si se cambia D_j por Q_j, la misma ecuación sirve para comparar un documento con una consulta. El problema de esta medida es que no esta normalizada y sus posibles valores son todos los reales mayores que 0.

$$SIM(\vec{D}_i, \vec{D}_j) = \sum_{k=1}^{n} \omega_{i,k} \cdot \omega_{j,k}$$

Ecuación 3. Medida básica de similitud basada en la suma de productos

Sería muy recomendable que una función de similitud diera unos valores normalizados, por ejemplo entre 0 y 1. Una de las funciones más populares para realizar esta misión es la fórmula del coseno utilizada por Salton [Salton1971] en el sistema SMART. Dicha fórmula calcula el coseno del ángulo entre los dos vectores. Cuando el coseno se aproxima a 1, los dos vectores son coincidentes. Si los dos vectores no tienen ninguna relación, entonces son ortogonales y el valor del coseno es 0. En la Ecuación 4 se muestra dicha ecuación en la que $a_{j,k}$ representa el peso del término *k* dentro de la consulta *j*.

$$Sim(\vec{D}_i, \vec{Q}_j) = \frac{\sum_{k=1}^{n} \omega_{i,k} \cdot a_{j,k}}{\sqrt{\sum_{k=1}^{n} \omega_{i,k}^{\ 2} \cdot \sum_{k=1}^{n} a_{i,k}^{\ 2}}}$$

Ecuación 4. Fórmula del coseno para calcular la similitud

El valor que reciben los pesos $\omega_{i,j}$ puede ser, en el caso mas sencillo 0 si el término *j* no aparece en el documento *i* y 1 si aparece. También puede utilizarse TF, nTF o TFIDF para darle valores a los pesos (vistos en II.3.2).

El problema del uso de los algoritmos de similitud es que devuelven la base de datos completa como resultado de una búsqueda. Habrá muchos documentos en el resultado que tengan un valor de similitud muy cercano a cero. Para evitar este problema se suele utilizar un umbral para marcar un mínimo que deban cumplir los documentos a mostrar como solución de la consulta. Otra opción consiste en hacer una recuperación booleana que reducirá el número de documentos y posteriormente aplicarle el algoritmo que corresponda.

II.5. RECUPERACIÓN PROBABILISTICA

Un problema típico de IR es que una consulta no expresa exactamente la necesidad de información del usuario. Como resultado, dos usuarios pueden juzgar de forma diferente la relevancia de un documento para una consulta. Por lo tanto, se puede buscar la probabilidad de que un documento *d* sea juzgado como relevante por un usuario cualquiera para una consulta *q*. Para esto es necesario utilizar las siguientes variables estocásticas:

- R que indica el conjunto de posibles evaluaciones. Lo más sencillo es {relevante, no relevante}
- D representa el valor de un documento
- Q representa la consulta

Uno de los modelos fundamentales y más representativos de los modelos probabilisticos es el BIR (Binary Independence Retrieval Model) [Robertson1976]. Para representar un documento se utiliza una indexación binaria de cada término. En la Ecuación 5 se muestra la representación de un documento, donde los α_i pueden tener los valores 1 ó 0 si aparecen o no en el documento y $t_1...t_n$ son los posibles términos.

$$\vec{d} = \alpha_1 t_1, \alpha_2 t_2, ..., \alpha_n t_n$$

Ecuación 5. Representación binaria de un documento (Caso particular de la Ecuación 2)

Se asume la independencia de los términos para facilitar el cálculo de las probabilidades. Dada una consulta, el sistema debe decidir sobre la relevancia de un documento y una consulta, es decir se debe estimar la probabilidad de que dado un documento y una consulta la evaluación sea relevante: P(rel|d,q). El significado de P(q|d) es la probabilidad condicionada de que la consulta q sea cierta dado el documento d. P(q) hace referencia a la probabilidad de que la consulta q sea cierta. Es decir, si de 100 documentos 40 satisfacen la consulta q, entonces P(q)=0,4. En este método, se calcula el valor de relevancia de un documento para una consulta como una transformación de P(q|d), conocida como *logg-odds*:

$$Lo(q,d) = \log\frac{P(q \mid d)}{1 - P(q \mid d)} = \log\frac{P(q \mid d)}{P(\overline{q} \mid d)} = \log\frac{P(d \mid q)\dfrac{P(q)}{P(d)}}{P(d \mid \overline{q})\dfrac{P(\overline{q})}{P(d)}} =$$

$$\log\frac{P(d \mid q)}{P(d \mid \overline{q})} \cdot \frac{P(q)}{P(\overline{q})} = \log\frac{P(\alpha_1 t_{1,} \ldots, \alpha_n t_n \mid q)}{P(\alpha_1 t_{1,} \ldots, \alpha_n t_n \mid \overline{q})} + \log\frac{P(q)}{P(\overline{q})}$$

Ecuación 6. Logg-odds

La asunción de independencia hace que se pueda representar las probabilidades conjuntas de los términos que forman el documento, como el producto de sus probabilidades, quedando la ecuación anterior como:

$$Lo(q,d) = \log\frac{P(q \mid d)}{1 - P(q \mid d)} = \log\frac{\prod_{i=1}^{n} P(\alpha_i t_i \mid q)}{\prod_{i=1}^{n} P(\alpha_i t_i \mid \overline{q})} + \log\frac{P(q)}{P(\overline{q})} = \sum_{i=1}^{n} \log\frac{P(\alpha_i t_i \mid q)}{P(\alpha_i t_i \mid \overline{q})} + \log\frac{P(q)}{P(\overline{q})}$$

Ecuación 7. Logg-odds suponiendo independencia entre los términos

Para poder calcular la ecuación anterior se necesita estimar los valores $P(\alpha_i t_i|q)$, $P(\alpha_i t_i|\neg q)$ para cada término, además de las P(q) y P(¬q). Para simplificar la ecuación se define $u_i=P(t_i|q)$ y $v_i=P(t_i|\neg q)$ y:

$$P(\alpha_i t_i \mid q) = u_i^{\alpha_i}(1-u_i)^{1-\alpha_i} \quad y \quad P(\alpha_i t_i \mid \overline{q}) = v_i^{\alpha_i}(1-v_i)^{1-\alpha_i}$$

Ecuación 8. Estimación de probabilidades condicionadas para calcular logg-odds

Sustituyendo y desarrollando los exponentes en la Ecuación 7 :

$$Lo(q,d) = \log\frac{P(q \mid d)}{1-P(q \mid d)} = \sum_{i=1}^{n}\log\frac{u_i^{\alpha_i}(1-u_i)^{1-\alpha_i}}{v_i^{\alpha_i}(1-v_i)^{1-\alpha_i}} + \log\frac{P(q)}{P(\overline{q})} =$$

$$= \sum_{i=1}^{n}\alpha_i \log\frac{u_i\ (1-v_i)}{v_i\ (1-u_i)} + \sum_{i=1}^{n}\log\frac{1-u_i}{1-v_i} + \log\frac{P(q)}{P(\overline{q})}$$

Ecuación 9. Desarrollo de loggs-odds utilizando las estimaciones de la Ecuación 8

Retomando el origen de este desarrollo, lo que se pretendía era obtener algún valor que indicase si un documento podía ser relevante o no para una consulta. Esta ecuación va a devolver un valor para cada documento que, además, servirá para ordenar los documentos que son resultado. Los más relevantes serán los que tengan el valor mayor. Como puede verse en la Ecuación 9 sólo el primer sumatorio depende específicamente de un documento (por el coeficiente α_i que indica si el término está o no en el documento). El resto son constantes. Por lo tanto, si lo único que interesa es ordenar los documentos, se puede prescindir de la parte constante quedando:

$$\log\frac{P(q \mid d)}{1-P(q \mid d)} \approx \sum_{i=1}^{n}\alpha_i \log\frac{u_i\ (1-v_i)}{v_i\ (1-u_i)}$$

Ecuación 10. Valor para calcular la relación entre un documento y una consulta en BIR

Para aclarar el uso de la fórmula se mostrará un ejemplo, en el que se realiza una consulta y se evalúan 12 documentos. Además se suponen 2 términos *t1* y *t2* que podrán estar o no en alguno de esos documentos. En la Tabla 2 se muestran los 12 documentos. Para cada uno se indica si los términos *t1* y *t2* aparecen en él mediante un 1 o con un 0 en caso contrario. La línea *Rel(q,d_i)* indica con una '*S*' si el documento d_i se ha evaluado como relevante para la consulta *q*.

d_i	1 2 3 4	5 6 7	8 9 10	11 12
t_1	1 1 1 1	1 1 1	0 0 0	0 0
t_2	1 1 1 1	0 0 0	1 1 1	0 0
Rel(q,d_i)	S S N S	N N S	S S N	N S
BIR	$0,70_{(2)}$	$-0,39_{(4)}$	$0,73_{(1)}$	$0,42_{(3)}$
P(q|d)	$0,75_{(1)}$	$0,33_{(4)}$	$0,66_{(2)}$	$0,5_{(3)}$

Tabla 2. Ejemplo del uso del modelo probabilístico BIR

De esas líneas de la tabla se obtienen los valores de u_i y v_i para cada uno de los términos. La forma de obtenerlos es contando los valores que correspondan en la Tabla 2. Por ejemplo, para calcular u_1 es necesario mirar cuantas veces el término t_1 se ha evaluado como correcto (4) y dividirlo por el total de evaluaciones correctas para esa consulta (7)

$$u_1 = P(t_1 \mid q) = 4/7 \quad u_2 = P(t_2 \mid q) = 5/7$$
$$v_1 = P(t_1 \mid \overline{q}) = 3/5 \quad v_2 = P(t_2 \mid \overline{q}) = 2/5$$

A continuación se calcula el BIR para cada documento. Si se quiere utilizar la Ecuación 9 será necesario calcular la parte constante:

$$\sum_{i=1}^{n} \log \frac{1-u_i}{1-v_i} + \log \frac{P(q)}{P(\overline{q})} = \log \frac{1-4/7}{1-3/5} + \log \frac{1-5/7}{1-2/5} + \log \frac{7/12}{5/12} = -0{,}1461$$

Ecuación 11. Valor del BIR constante para todos los documentos

Por lo tanto, para aquellos documentos que no contengan el término t_1 ni el término t_2, la probabilidad que calcula BIR de que el documento se corresponda a la consulta, se obtiene de:

$$\log \frac{P(q \mid d)}{1-P(q \mid d)} = \log x \Rightarrow P(q \mid d) = \frac{x}{x+1}$$

$$BIR = P(q \mid d) = \frac{10^{-0{,}1461}}{10^{-0{,}1461}+1} = 0{,}42$$

Para completar el ejemplo se va a calcular la probabilidad de que la consulta sea relevante para los documentos que sólo tienen el término t_2. Estos son aquellos documentos en los que para la Ecuación 9 $\alpha_1=0$ y $\alpha_2=1$. A continuación se calculará el valor depediente de este término y se le sumará la constante calculada en la Ecuación 11:

$$\sum_{i=1}^{n} \alpha_i \log \frac{u_i\ (1-v_i)}{v_i\ (1-u_i)} + cte = \log \frac{5/7 \cdot (1-2/5)}{2/5 \cdot (1-5/7)} - 0{,}1461 = \log \frac{15}{4} = 0{,}4279$$

$$BIR = P(q \mid d) = \frac{10^{0{,}4279}}{10^{0{,}4279}+1} = 0{,}73$$

Ecuación 12. Ejemplo del calculo de BIR para un documento que contiene el término t_2

En la Tabla 2 se muestra también el resultado de aplicar directamente la P(q|d) que se calcula dividiendo el número de casos favorables por el de casos posibles para cada grupo de documentos. Por ejemplo, para los documentos que contienen t_1 y t_2 se han evaluado 3 como interesantes de 4 evaluaciones, lo que da P(q|d)=3/4.

Como puede verse tanto BIR como la aproximación directa dan resultados similares a la hora de ordenar documentos aunque los valores sean diferentes, ya que el cálculo de BIR es una aproximación que supone que el interés del usuario se basa exclusivamente en los valores de sus términos. La ventaja de BIR es que cuando el número de términos (n) es mayor, sólo requiere calcular 2n parámetros (u_i y v_i) en cambio para la aproximación directa se requieren 2^n subconjuntos diferentes, haciendo impracticable esta aproximación.

II.6. BÚSQUEDAS ITERATIVAS Y LA EVALUACIÓN PERTINENTE

Con bastante frecuencia, el resultado de una búsqueda devuelve un número de posibles soluciones mucho mayor de las que el usuario está dispuesto a analizar. Normalmente, después de una búsqueda en la Web, se miran unas 6 o 7 soluciones y se vuelve a escribir una nueva consulta. Algunos sistemas permiten refinar la búsqueda anterior añadiéndole nuevos términos. Esta técnica recibe el nombre de búsqueda iterativa [Kowalski2001].

Otras veces ocurre que el principal problema para encontrar elementos relevantes es la diferencia en el vocabulario de los autores de los documentos y el usuario. Esto puede solucionarse en parte con tesauros (*Thesaurus*) y redes semánticas (*Semantic Network*) u ontologías que suelen contener relaciones entre términos como:

- A es sinónimo de B
- A es antónimo de B
- A esta relacionado con B
- A es parte de B
- A generaliza B
- A es un B

La utilización de estos elementos expande la consulta del usuario para encontrar posibles términos relacionados. Aunque la intención sea buena, estas posibles mejoras pueden introducir demasiado ruido, efecto no deseado. Por ejemplo, si se está haciendo la búsqueda: "Pentium IV", podrían estar definidas las relaciones (Pentium IV es un procesador, y AMD es un procesador), con lo que se podría ampliar la búsqueda a: "*Pentium IV OR procesador OR AMD"*. En este caso los resultados no serían los deseados por el usuario. Además, la construcción de una ontología o tesauro completo puede ser muy complicada y específica para un sistema muy concreto.

Algo que suele interesar a los usuarios es que cuando han encontrado una solución que para ellos es relevante les gustaría encontrar más soluciones relacionadas con ella. O indicar de alguna forma al sistema que en futuras búsquedas se utilicen los documentos relevantes para modificar la consulta. Esta técnica de aumentar la consulta con información sobre documentos importantes para dicha consulta se llama evaluación pertinente (*relevant feedback.* En adelante *RF*) [Rocchio1971].

El RF consiste en crear una nueva consulta utilizando la consulta anterior, incrementando los pesos de los términos evaluados en documentos interesantes y decrementándolos si aparecen en documentos no interesantes.

$$Q_n = \alpha Q_0 + \beta \sum_{i=1}^{r} DR_i - \gamma \sum_{j=1}^{nr} DNR_j$$

Ecuación 13. Fórmula para modificar la consulta con *relevant feedback*

Donde

Q_n = Consulta revisada
Q_0 = Consulta original
r = Número de documentos relevantes
DR_i = Vector para los documentos relevantes
nr = Número de documentos no relevantes
DNR_j = Vector para los no documentos relevantes
α = Factor de importancia de la consulta original
β = Factor de importancia de los términos que aparecen en documentos relevantes
γ = Factor de importancia de los términos que aparecen en documentos no relevantes

Normalmente β es una constante asociada al factor 1/r, y γ asociada al factor 1/nr, pues se utilizan para dar peso a los términos relevantes y no relevantes respectivamente. Si por ejemplo se quiere dar más importancia a los documentos evaluados como relevantes los valores de β y γ se pueden calcular como $\beta=1/(2\cdot r)$ y $\gamma=1/(4\cdot nr)$. Este mecanismo se entenderá mejor con un ejemplo:

Suponiendo los términos posibles: *adaptive hypermedia*(1), *information retrieval*(2), *relevant feedback*(3), *tutorial systems*(4), *user modeling*(5) y los documentos de la Tabla 1 (utilizada anteriormente) que contiene algunos de estos términos.

Si se realiza una consulta con el término *Adaptive Hypermedia* utilizando la fórmula básica de similitud (Ecuación 3), se mostrará al usuario los documentos 1,5 y 3 en ese orden, pues el vector de la consulta sería $Q_0=(1,0,0,0,0)$. Si el usuario evalúa el documento 5 como interesante y el 3 como no interesante. La nueva consulta que se genera, utilizará como factor $\alpha=1$, $\beta=1/(2\cdot r)$ y $\gamma=1/(4\cdot nr)$. Como se ha evaluado un documento como interesante y otro como no, los valores finales son: $\alpha=1$, $\beta=1/2$ y $\gamma=1/4$, ponderando algo más las evaluaciones positivas. La consulta modificada por el *RF* seria:

$$Q_1=(1,0,0,0,0)+1/2(2,0,0,2,1)-1/4(1,5,0,0,0)=(1.75,-1.25,0,1,0.5)$$

La modificación de la consulta hace que la similitud con los documentos del sistema sea mayor en los documentos que contengan términos que han sido bien evaluados. Así para los 5 documentos del ejemplo, su similitud con ambas consultas sería la que se muestra en la Tabla 3:

	Document1	*Document2*	*Document3*	*Document4*	*Document5*
Q_o	3	0	1	0	2
Q_1	5.25	-5	-4.5	5	5

Tabla 3. Similitud entre los documentos y la consulta original (Q_o) y la consulta mejorada (Q_1)

Puede verse que se ha aumentado el número de documentos a recomendar debido a la ampliación de la consulta inicial. Por ejemplo, el documento 4, que no contiene el término de la consulta original (*Adaptive Hypermedia*) ha salido bien clasificado. El porqué puede verse en que los términos que lo definen aparecen en los documentos (doc5) que han sido evaluados correctamente.

II.7. EVALUACIÓN

Para poder conocer la eficiencia de un sistema éste debe poder evaluarse. El problema de los IRS es que no están muy bien definidas cuales son las tareas que debe realizar, y es bastante difícil saber si los resultados que devuelve a una consulta son correctos o no, especialmente, si se trata de aspectos subjetivos al usuario. Dos usuarios diferentes, utilizando la misma consulta pueden decir que un documento se adapta o no a dicha consulta dependiendo de sus criterios personales. Aunque es bastante difícil hacer esas comparaciones se han diseñado algunos componentes para hacer estas evaluaciones de forma experimental. Los componentes para los experimentos son: el sistema, una colección de documentos, una colección de solicitudes de información, un criterio de evaluación y un diseño del experimento.

Los sistemas normalmente están formados por varios componentes, las pruebas generales de recuperación de la información se asocian más a esos componentes que al sistema completo. El objetivo de la evaluación debería ser encontrar un criterio para conocer la eficiencia de la recuperación de la información [Robertson2000].

Los documentos con los que trata el sistema pueden ser sólo textos, contenido multimedia, y algunos añadidos como la categoría a la que pertenece o palabras clave. Muchos sistemas generan esos añadidos de forma automática para indexar el documento.

Por otro lado se necesita una colección de solicitudes de información, desde el punto de vista de necesidad de un tema de interés más que el de una simple consulta. Esa necesidad de información tendrá que ser codificada en consultas al sistema, que puede modificarse como resultado de la interacción con el sistema.

También es importante tener un criterio de evaluación, es decir, saber que es un documento relevante para esa necesidad de información. Este es un punto difícil de aclarar cuando se trata de relevancia para un usuario en concreto. Para sistemas experimentales, se supone que un experto ha indicado qué documentos son los relevantes para cada tema de interés. La relevancia de un documento, suele tomarse como un valor binario (relevante/no relevante). Esta suposición no es totalmente correcta ya que puede haber distintos grados de importancia. Para los experimentos también se supone que la importancia de un documento es independiente de otros de la colección.

Hay dos medidas tradicionales para calcular la eficiencia de un sistema que son *precisión* y *recall* (Ecuación 14) [Rijsbergen1971] donde la primera indica la proporción de los documentos recuperados que son relevantes, y la segunda indica la proporción entre los documentos recuperados relevantes con el total de documentos relevantes.

$$recall = \frac{N^{o}\ de\ documentos\ relevantes\ recuperados}{N^{o}\ de\ documentos\ relevantes\ en\ la\ colección}$$

$$precision = \frac{N^{o}\ de\ documentos\ relevantes\ recuperados}{N^{o}\ de\ documentos\ recuperados}$$

Ecuación 14. Ecuaciones para el cálculo de *precisión* y *recall*

Cuando un usuario decide realizar una búsqueda en un sistema sobre un tema, la base de datos se divide lógicamente en cuatro segmentos.

	Relevante	***No Relevante***
Recuperado	Éxito	Ruido
No Recuperado	Faltan	

Tabla 4. Relación entre documentos relevantes y los recuperados en una consulta

La precisión representa un aspecto de la sobrecarga de información asociada a una búsqueda en particular. El valor ideal será cuando todos los documentos recuperados sean los relevantes, es decir, cuando valga 1. El *recall* mide la calidad del sistema para procesar una consulta y recuperar los documentos interesantes para el usuario. Su valor ideal también es 1. Aunque el concepto de *recall* es muy útil, existe el problema de que

en los sistemas reales no se puede calcular directamente el valor del denominador, pues si se conocieran todos los documentos relevantes, el sistema los habría devuelto.

Aunque *precisión/recall* son las bases para medir la eficiencia de los sistemas de recuperación de la información aparecen problemas, por ejemplo cuando no hay elementos relevantes que recuperar el valor de *recall* es 0/0 (0 recuperados/0 posibles relevantes) y cuando no se ha recuperado nada la precisión es 0/0 (0 recuperados relevantes/0 recuperados). Para ello se utiliza otra medida relacionada directamente con la recuperación de elementos no relevantes, que también puede dar información de la efectividad del sistema. Esta medida se llama *fallout* y se muestra en la Ecuación 15:

$$fallout = \frac{N^{o}\ de\ documentos\ recuperados\ no\ relevantes}{N^{o}\ de\ documentos\ no\ relevantes\ en\ la\ colección}$$

Ecuación 15. Ecuación para el cálculo de *fallout*

Para la evaluación de IRS hay una conferencia anual TREC (*Text REtrieval Conference*) [TREC2001], en la que se ofrece material con el que se pueden probar los algoritmos, para hacer experimentos en laboratorio, aunque hay muchos apartados en el diseño de IRS que quedan fuera del alcance de las pruebas de laboratorio. Especialmente sistemas muy interactivos, en los que es importante el estudio del proceso de búsqueda por parte del usuario para encontrar una solución.

II.8. FILTRADO DE LA INFORMACIÓN

Para tener una visión completa del área parece necesario hablar de filtrado de la información (*Information Filtering-IF*). Según [Belkin1992] podría considerarse que ambas líneas de investigación (IR e IF) son las dos caras de una misma moneda. Aunque cada una tiene sus características propias, ambas comparten la mayoría de las técnicas y modelos. Ambas tienen el objetivo de recuperar aquellos elementos que son relevantes para el usuario, minimizando el número de elementos irrelevantes. Lo que diferencia a ambos campos es la relación entre las necesidades del usuario y las características de los datos. Un sistema de recuperación de la información suele tener una base de datos más o menos estable a la que el usuario accede mediante consultas que suelen ser diferentes, como puede ser una biblioteca o buscar información en la Web. En un sistema de filtrado, los usuarios tienen una consulta que se mantiene a lo largo del tiempo (llamada generalmente perfil), y los datos se actualizan con frecuencia, como por ejemplo un sistema para clasificar el correo electrónico. En algunos sistemas puede no quedar claro si su objetivo es de filtrar o de recuperar información. En el apartado siguiente se mostrarán algunos ejemplos y se verán las características comunes y diferencias en sistemas reales.

II.9. EJEMPLOS

En este apartado se mostrarán algunos de los sistemas de recuperación y filtrado de información más relevantes. Para cada uno de ellos se indicará para qué sirve, que técnicas utiliza y de los que se dispone de información suficiente algún comentario sobre ventajas e inconvenientes. Al final se hará una tabla comparativa resumiendo las características de cada uno.

II.9.1 INQUERY

El sistema INQUERY [Callan1992] es un IRS orientado a hacer búsquedas en bases de datos grandes y heterogéneas. Está basado en un modelo probabilístico con un sistema que permite consultas complejas. INQUERY se basa en un tipo de red bayesiana llamada *red de inferencia para recuperación de documentos*. Esta red esta formada por dos subredes, la red de documentos y la red de consultas. Una red de documentos puede representar un conjunto de documentos con diferentes técnicas de representación y niveles de abstracción. Parte del tratamiento que hace para indexar los documentos consiste en aplicar una lista de parada y la supresión de terminaciones de las palabras (*uso de raices: stemming)*, como participios o gerundios, además convierte todo el texto a minúsculas. Utiliza una tabla hash para almacenar el diccionario y listas invertidas para gestionar los términos.

II.9.2 SIFT

Este sistema se utiliza para filtrar noticias [Yan1995]. Podría considerarse un sistema de recuperación, ya que cada cierto tiempo realiza una consulta a las noticias para ver cuales se adecuan al perfil. El usuario introduce una lista de las palabras clave de su perfil y selecciona si quiere hacer una correspondencia booleana o vectorial. El perfil de usuario se limita a una lista de palabras. Para la correspondencia booleana una noticia se mostrará al usuario en el caso en que todas las palabras clave del perfil aparezcan en el documento. Para la correspondencia vectorial, se utiliza la fórmula del coseno para decidir si se selecciona o no una noticia.

Quizás su principal limitación sea la de utilizar sólo una lista de palabras, podría mejorarse incluyendo más parámetros o alguna red semántica para evitar ambigüedades en los distintos significados de las palabras.

II.9.3 SMART

SMART [Salton1971] es un sistema de recuperación de la información que puede aplicarse a una gran variedad de documentos que pueden ir desde correos electrónicos, manuales de programación o artículos de revistas. Implementa el modelo vectorial. Es uno de los más antiguos pero bastante utilizado debido a las posibilidades de modificar las funciones de correspondencia para poder comparar resultados. El usuario introduce una lista de palabras para su consulta y como resultado se le muestra una lista de documentos ordenados en función de la función de correspondencia activa. El sistema también dispone de RF y se crea una nueva consulta tras la evaluación del usuario sobre los documentos recuperados.

La mayor ventaja de este sistema, a pesar de sus modelos simples, es que puede ser una buena referencia para comparar resultados con otros sistemas debido a que se puede utilizar con diferentes bases de datos.

II.9.4 GHOSTS

Es un sistema para filtrar correos y noticias. El sistema filtra las noticias utilizando reglas que permitirán clasificar los mensajes en carpetas, descartarlos, reenviarlos, etc. Este sistema sólo utiliza el texto del título del mensaje y las palabras clave si están disponibles.

El hecho de utilizar sólo el título y de no analizar el cuerpo del mensaje, aunque puede dar resultados rápidos, no parece suficiente para representar su contenido.

II.9.5 OKAPI

Se utiliza para recuperación de textos científicos (biblioteca universitaria y resúmenes de revistas) [Robertson1993; Robertson1994]. No utilizan expresiones booleanas para las búsquedas sino sólo una lista de palabras. El modelo probabilístico que utiliza es una propuesta del propio autor, Robertson, tomada como referencia de este modelo. Para los términos de la búsqueda calcula el IDF para darle un peso en la consulta. Utiliza una lista invertida para almacenar los términos. Permite RF aunque sólo de evaluaciones positivas. Este sistema da muy buenos resultados en las comparativas de las conferencias TREC[1][Robertson1997].

El sistema podría dar aun mejores resultados si para el cálculo de RF utilizase también las evaluaciones negativas de los usuarios.

II.9.6 HYSPIRIT

Este sistema [Rölleke1999] se aplica para recuperación de la información de documentos estructurados mediante XML y para televisión interactiva (MPEG-7). Además se utiliza en entornos comerciales donde diferentes necesidades de información ocurren en entornos heterogéneos y complejos. Esta formado por un conjunto de módulos que se implementan en diferentes capas de abstracción, utilizando orientación a objetos y Datalog. Esto le permite explotar la estructura lógica de los documentos, aplicar estrategias de recuperación basadas en relaciones (espaciales, temporales, semánticas, etc.) y la aportación más importante es la de extender modelos de bases de datos con teoría de la probabilidad aprovechando así la expresividad de los modelos de bases de datos y la incertidumbre de la teoría de probabilidad.

II.9.7 CITESEER[2]

Es uno de los sistemas de recuperación de la información más importantes para la localización de publicaciones científicas. Su organización de índices es mediante tabla hash de palabras, es decir, listas invertidas en las que cada entrada contiene una versión comprimida de una palabra (término) y la lista de documentos en los que aparece, junto con la posición en la que lo hacen. Permite consultas de tipo booleano, o una lista de palabras. En cualquier caso utiliza un modelo booleano. La ordenación de los resultados viene dada de acuerdo al número de citas que tienen o por fecha y no por relevancia con la consulta. No utiliza lista de parada, según los autores para permitir una búsqueda de mayor precisión. Una de las características que lo hacen especial es que crea un sistema de referencias autónomo. Para ello analiza las referencias de los documentos que cataloga.

Su mayor defecto es que el sistema de búsqueda que utiliza es muy limitado y restrictivo, resultando incomodo a veces. Este sistema mejoraría enormemente si incluyese características de personalización, pues las respuestas serían más acordes con las necesidades del usuario. Estás características se verán en el capítulo siguiente.

[1] Artículos sobre resultados de Okapi en TREC: http://research.microsoft.com/users/robertson/papers/trec_papers.htm

[2] Página de Citeseer: http://www.citeseer.com

II.9.8 RESUMEN DE SISTEMAS

Se muestra a continuación la Tabla 5, que resume las características de los sistemas comentados en apartados anteriores. Para cada sistema se indica el modelo de recuperación utilizado, si el sistema esta orientado al filtrado de datos o recuperación, una información sobre el dominio de aplicación y por último algunas características adicionales como el uso de listas de parada, evaluación relevante (RF) o el uso de raíces para procesar los términos.

Sistema	*Modelo/Método de clasificación*	*Filtrado/ Recuperación*	*Otras Características*	*Dominio/Notas*
Citeseer	Booleano	Recuperación	Listas Invertidas No Lista de Parada	Publicaciones científicas
FAQFinder	Vectorial		Lista de Parada RF	Utiliza SMART
HySpirit	Lógico/Datalog	Recuperación		XML y Televisión Interactiva
Inquery	Redes Bayesianas	Recuperación	Lista de Parada Uso de Raíces	Grandes bases de datos documentales
Ghosts	Basado en Reglas	Filtrado	RF	E-mail, Noticias de Internet
Okapi	Probabilístico	Recuperación	Lista de Parada Listas Invertidas Uso de Raíces RF	Libros y artículos de Bibliotecas Universitarias
SIFT	Vectorial-Coseno Booleano	Filtrado		Noticias de Internet
Smart	Vectorial	Recuperación	Lista de Parada RF	Publicaciones, E-Mail

Tabla 5. Comparativa de diferentes sistemas de Recuperación de la información

II.10. CONCLUSIONES

En este capítulo se han visto las diferentes técnicas utilizadas para la recuperación de la información. Cada una de ellas utiliza un método de correspondencia para encontrar aquellas soluciones que son relevantes para la consulta del usuario. La más simple, aunque utilizada en muchos sistemas es la técnica booleana que devuelve como resultado de una consulta aquellos documentos que cumplen la expresión booleana de la consulta sin ordenarlos de ninguna forma. Un poco más eficiente es la correspondencia Vectorial que representa un documento como un vector de términos a los que da un peso que servirá para ordenar los resultados de acuerdo a su similitud con la consulta. Por otro lado están los modelos probabilísticos que utilizan la teoría de la probabilidad para seleccionar los documentos. También utilizan probabilidades los modelos basados en redes Bayesianas que quizás, por su complejidad, necesiten más tiempo para dar una solución que otros modelos más sencillos, aunque en los sistemas analizados las redes no son demasiado complejas. Por último, están modelos basados en la lógica que están empezando a aplicarse.

Los IRS necesitan realizar una búsqueda iterativa para dar soluciones al usuario. La mayoría sólo se concentra en comparar la consulta con los índices que disponen sobre sus documentos. El elemento más importante visto en este sentido es la mejora de la

consulta mediante *evaluación pertinente*. Dicho método realimenta una nueva consulta en función de los documentos evaluados.

Como se ha indicado anteriormente, la evaluación en IRS está muy orientada a experimentos de laboratorio, donde para una consulta se sabe cuales son los valores que tienen que devolverse, y dependiendo de cómo lo haga el sistema, se puede calcular los valores de los factores "*precision"* y *"recal"l* sin ningún problema para poder compararlo con otros sistemas. Además después de una consulta se sabe exactamente cuales son todos los documentos que corresponden y los que no. Lo que no queda claro es cómo realizar la evaluación cuando se trata de la satisfacción del usuario. No se puede pedir a un usuario que si el sistema le devuelve 100 documentos, evalúe todos con una precisión absoluta sobre su satisfacción o no. Probablemente el usuario evaluará 3 ó 4.

Parece claro que independientemente del método de búsqueda utilizado por un IRS sería muy conveniente un elemento adicional que tuviese en cuenta características de la persona, pues la consulta que se hace al sistema es una forma que tiene el usuario de expresar su necesidad de información y en la mayoría de las ocasiones, los usuarios no saben transformar esa necesidad en una consulta en el lenguaje del buscador que están utilizando. Los usuarios agradecerían que dicho buscador tuviese cierta inteligencia y pudiese ayudarle en su tarea de búsqueda.

En esta tesis se propone un IRS que además de disponer de los métodos clásicos de búsqueda de información, permitirá un análisis global de las informaciones del sistema y que aprenderá las necesidades del usuario a medida que este interactúe con él. Se propondrá una mejora a la *evaluación pertinente*, más potente en el sentido de que el usuario no deberá depender en una consulta de las consultas anteriores, pero las consultas anteriores servirán para generar un modelo del usuario con los datos de las evaluaciones positivas y negativas, que se utilizarán para reordenar los documentos en función de ese modelo. Así el usuario puede realizar diferentes consultas independientes en busca de la misma información y todas esas consultas irán perfeccionando su modelo y dando soluciones más exactas, además de permitirle en cualquier momento, que con la información que el sistema tiene sobre el usuario, se pueda crear una consulta automáticamente con todos los datos del usuario hasta el momento.

En el capítulo siguiente se introducirán las técnicas de modelado del usuario que permitirán crear sistemas de recuperación de la información personalizados.

III. RECUPERACIÓN DE LA INFORMACIÓN PERSONALIZADA

III.1. INTRODUCCIÓN

En el capítulo anterior se veían las técnicas clásicas de recuperación de la información. La característica común de todas ellas era que la respuesta que devolvía al usuario dependía exclusivamente de su consulta. Por lo tanto, si dos usuarios diferentes realizan la misma consulta obtendrán como resultado los mismos datos. Si el objetivo de un sistema es dar respuestas que se adecuen a cada usuario será necesario incluir técnicas de personalización a los sistemas de recuperación de la información. Esto podrá realizarse guardando los datos del usuario en un modelo, que se utilizará para seleccionar y reordenar los posibles resultados de su consulta en función de sus preferencias, conocimiento o necesidades. En este capítulo se analiza brevemente el área de modelado de usuario en general y su aplicación a la recuperación de la información. Se verán conceptos importantes como los modelos de usuarios genéricos o los estereotipos. Posteriormente se comentarán algunos sistemas que realizan recuperación de la información personalizada. Se analizarán las ventajas e inconvenientes que servirán para justificar la propuesta de esta tesis.

III.2. MODELADO DEL USUARIO

Es importante comprender qué es el modelado del usuario. Para ello se recuerda la definición de [Bauer2001]: "*Un modelo de usuario es una representación explícita de las propiedades de un usuario particular, las cuales permiten a los sistemas adaptar diversos aspectos de su funcionamiento a las necesidades individuales de los usuarios*". Los primeros trabajos sobre modelado del usuario (*User Modeling*- en adelante *UM*) son los realizados por E. Rich [Rich1979,1983]. Alfred Kobsa realiza uno de las primeras recopilaciones sobre UM en [Kobsa1993]. Actualmente se aplica en áreas diferentes relacionadas siempre con el uso de ordenadores por poblaciones heterogéneas de usuarios. Entre estos campos están interacción hombre-maquina, interfaces inteligentes, interfaces adaptativos, ingeniería del conocimiento, recuperación de la información inteligente, tutoriales inteligentes, sistemas de ayuda activos y pasivos, sistemas de guía, hipertextos y sistemas expertos entre otros.

Los modelos de usuario son importantes porque representan información sobre el usuario de forma que el sistema puede operar de una forma más eficiente. Debido a que los individuos tienen diferente conocimiento, preferencias y objetivos hay muchas situaciones donde el tratamiento individualizado del usuario basada en la información de un modelo de usuario puede ofrecer ventajas. Por eso podría también decirse de manera informal que un modelo de usuario es un conjunto de creencias sobre el usuario.

III.2.1 MODELOS DE USUARIO GENÉRICOS

Una parte de la comunidad de UM está dedicada al estudio de modelos de usuarios genéricos. Estos modelos intentan recopilar las características más utilizadas en sistemas individuales para poder ser utilizados en múltiples sistemas. Entre los sistemas genéricos más relevantes destacan UMT [Brajnik1994], BGP-MS [Kobsa1995] o UM [Kay1995a] Su análisis puede dar una idea de las aplicaciones más comunes del modelado del usuario. En una reciente revisión [Kobsa2001] se muestran las características que frecuentemente aparecen en este tipo de sistemas:

- Representación para cada usuario de suposiciones sobre uno o más tipos de características como: nivel de conocimiento, conceptos erróneos, objetivos, planes, preferencias, tareas y habilidades
- Representación de características compartidas con otros usuarios formando subgrupos dentro de la aplicación (estereotipos)
- Clasificación de los usuarios en algunos de esos subgrupos e integración de las características de los subgrupos a los que pertenezcan dentro de su modelo de usuario
- Almacenamiento la información sobre el comportamiento del usuario en sus interacciones con el sistema
- Obtención de conclusiones sobre los usuarios basadas en sus interacciones pasadas
- Generalización de las interacciones de muchos usuarios para crear estereotipos
- Mantenenimiento la consistencia del modelo de usuario
- Posibilidad de mostrar las suposiciones del usuario y justificación de esas posibilidades

Esta lista de características da idea de algunas de las aproximaciones que se utilizan para ofrecer respuestas personalizadas, destacando el uso de estereotipos o reconocimiento de planes que se detallan en los apartados siguientes.

III.2.2 ESTEREOTIPOS

Los estereotipos son la representación de características comunes de usuarios que pertenecen a subgrupos específicos de una aplicación. Son uno de los elementos más comunes en el trabajo de modelado del usuario y capturan información sobre grupos de personas. Un usuario puede clasificarse como perteneciente a varias comunidades donde el modelo del usuario no tiene información explícita sobre algunos aspectos para ese usuario en concreto.

En realidad las personas razonan en la vida diaria utilizando estereotipos, pues se hacen inferencias por defecto sobre las personas basándose en poca información y se asume

mucho hasta que se reconoce posteriormente que se necesita alterar las suposiciones pues se ha profundizado en el conocimiento de esa persona. Este sistema puede ser bueno para establecer creencias rápidas sobre el usuario mientras que no se tengan disponibles otros hechos más fiables.

Es importante identificar la utilidad de los estereotipos para refinar la comprensión sobre qué son, qué no son, y cómo se relacionan con otros elementos del modelado del usuario. Para ilustrar esta necesidad, en [Kay1994] se proponen varios escenarios donde los estereotipos pueden ser útiles:

- En un sistema que recomienda películas, pueden hacerse las siguientes suposiciones: A los usuarios que les gustó la película X también les gustará la película Y, o los que no les gustó la película W tampoco les gustará la película Z.
- Si se quiere aprender a programar en C, los usuarios que sepan Pascal, podrían aprenderlo de forma sencilla viendo una comparación de cómo son las estructuras de C que equivalen a las de Pascal. Es decir, para ese tipo de usuario, en un sistema tutorial se les puede enseñar el lenguaje por comparación.

Normalmente se necesita obtener un rápido reconocimiento de algunas características fundamentales del usuario, para ello se requieren tres tareas:

1. *Identificar al usuario dentro de un grupo.* Se intenta encontrar poblaciones de personas cuyos miembros tengan ciertas características que lo hagan un grupo homogéneo.
2. *Identificación de características clave.* Son pequeñas pistas que llevan al sistema a situar al usuario dentro de un grupo u otro: Tipo de preguntas, terminología, tipo de ayuda requerida, etc.
3. *Representación de los estereotipos.* Muchos autores proponen una representación lineal de los estereotipos que se corresponden con principiante, intermedio y experto.

Los modelos de usuario que utilizan estereotipos los almacenan en una base de datos. Estos serán la base del razonamiento basado en estereotipos. Antes de empezar a utilizarlos es necesario que estos se activen mediante algún tipo de evento. Una vez que un estereotipo se activa, se pueden inferir conclusiones sobre él para aplicarlas a cada usuario. Las conclusiones que se tomen utilizando estereotipos deben ser temporales hasta que se haya desarrollado un modelo del usuario completo.

Uno de los sistemas más representativos que utilizan personalización basada en estereotipos es KNOME [Chin1989], que es la componente que realiza el modelado del usuario en el sistema UNIX CONSULTANT (UC) que es un sistema de ayuda del sistema operativo UNIX. La ayuda que ofrece al usuario se adapta al conocimiento que éste posee. En la Fig. 2 se muestran las interacciones de dos usuario. En función de las preguntas el sistema es capaz de inferir el conocimiento de los usuarios y por lo tanto ofrece una respuesta más avanzada para el segundo usuario. KNOME tiene cuatro estereotipos en los que clasifica al usuario: *Novato, principiante, intermedio y experto*. Además clasifica los conceptos en *simple, usual, complejo y avanzado*. El sistema utiliza reglas difusas para predecir el conocimiento del usuario y adaptarlo dentro de alguno de los estereotipos.

Usuario 1	**Usuario 2**
Hola, ¿Cómo puedo ayudarte?	Hola, ¿Cómo puedo ayudarte?
#¿Cómo puedo borrar un fichero?	*#¿Cómo puedo encontrar el inodo de un fichero?*
Usa rm. Por ejemplo para borrar el fichero llamado foo, escribe 'rm foo'	Usa ls -i
#¿Qué hace ruptime?	*#¿Qué hace ruptime?*
Ruptime se utiliza para sacar un listado de todas las maquinas de la red, sus usuarios y la carga media	Ruptime es como uptime, pero funciona para todas las máquinas de la red

Fig. 2. Sesiones con el sistema UC para dos usuarios

III.2.3 RECONOCIMIENTO DE PLANES

Un plan es una secuencia de acciones del usuario que buscan un cierto objetivo. El reconocimiento de planes observa las acciones del usuario e intenta determinar todos los posibles planes del usuario que pueden completar a las acciones llevadas a cabo por él. Un campo importante de aplicación es el de los sistemas de ayuda inteligente. Se encuentran muchos problemas cuando se implementa en la práctica estas técnicas, porque muchas veces no se sabe si el usuario ha comenzado un nuevo plan o si una serie de acciones pertenecen a más de un plan, o qué pasa si un usuario interrumpe la ejecución de sus planes actuales. Se suelen utilizar dos técnicas para el reconocimiento de los planes del usuario.

1. *Las librerías de planes.* Tienen todos los posibles procedimientos previamente almacenados.
2. *Construcción de planes.* En esta aproximación el sistema contiene una librería de todas las posibles acciones del usuario junto a sus efectos y precondiciones. Los efectos de algunas acciones serán las precondiciones de otras. Por lo tanto se podrán prever todas las posibles acciones del usuario. El problema de las librerías es que se requiere que todos los planes posibles del usuario deben ser previstos de antemano. Esto puede ser perfectamente válido para dominios con objetivos limitados. La construcción del plan tiene el problema desde el punto de vista de la complejidad, ya que a veces las posibles acciones que puede realizar un usuario son muchas.

Como ejemplo de sistemas de reconocimiento de planes, están los sistemas sistema PHI [Bauer1995] y Collagen [Lesh1997] que intentan ayudar a un usuario a trabajar con su correo electrónico. En función de la experiencia con el usuario intentan predecir las acciones que en cada momento el usuario haría, y en caso de ambigüedad o duda solicitan una elección que utilizarán para mejorar el modelo del usuario. Por ejemplo, en el sistema *Collagen* si el usuario para responder un correo de *A* elige la dirección de *B,* le pregunta si quiere escribir un correo nuevo o si quiere responder al correo de *A*. Cuando elige la segunda opción, esa información se guarda en el modelo del usuario y puede utilizarse si en el futuro el usuario se plantea por ejemplo, si respondió o no ese correo y como lo hizo.

III.2.4 MODELOS A CORTO/LARGO PLAZO

Una posible clasificación de los modelos de usuario estaría asociada a la cantidad de interacción con el usuario que es necesaria para poder tomar decisiones. Por un lado, están los modelos a corto plazo, que utilizan poca información para empezar a

personalizar la interacción con el sistema, la información que almacenan del usuario puede variar rápidamente tras cada nueva interacción. Por otro lado, están los modelos a largo plazo que guardan información a través de diferentes sesiones y es una información más estable. Normalmente, no se utilizan de forma aislada, sino que se combinan de forma que en el modelo a corto plazo se toman decisiones rápidas, destinadas a usuarios que trabajan con el sistema por primera vez y en el modelo a largo plazo para usuarios que utilizan el sistema con frecuencias y de los que se tiene una información más completa.

Un ejemplo de sistema que utiliza ambos modelos [Bornscheuer2001] es una tienda virtual que ofrece información personalizada. Utiliza un modelo a largo plazo para usuarios que han accedido varias veces al sistema, comprando o mostrando interés en algunos productos. Pero para otros usuarios, que no quieren que se guarde información personal sobre ellos o que acceden por primera vez, tiene un modelo a corto plazo en el que sólo se utiliza la información de la sesión actual para recomendarle productos.

Otro ejemplo es el recomendador de noticias de [Billsus1999] que utiliza un modelo a corto plazo para las observaciones recientes. Si una noticia no puede clasificarse con este modelo se utiliza el modelo a largo plazo. Este modelo híbrido permite adaptar los posibles cambios de intereses del usuario utilizando el modelo a corto plazo, conservando los intereses generales en el modelo a largo plazo.

III.2.5 MODELOS OBSERVABLES/REVISABLES

La información que los modelos almacenan para adaptar la interacción a usuario a veces es demasiado compleja para poder mostrársela. En ciertos casos puede ser interesante que un usuario pueda de visualizar lo que el sistema sabe o cree saber sobre él (modelos observables). Algunos sistemas permiten además que el usuario modifique su modelo (modelos revisables). Esto permitiría que la información del sistema fuera más precisa, pero también sería desaconsejable para usuarios inexpertos. Un sistema que permite a los usuarios observar y modificar su modelo de usuario es el propuesto en [Mizzaro2002].

III.2.6 APRENDIZAJE DEL MODELO DE USUARIO

El aprendizaje de los modelos de usuario en los que se representa su conocimiento o preferencias es una tarea necesaria en muchos tipos de sistema. Conseguir que un sistema tenga la capacidad de dar respuestas de una forma única y personalizada es un objetivo importante. Dependiendo del tipo de sistema y de la interacción que tienen con el usuario será necesario personalizar unos u otros elementos del usuario.

En el caso de los sistemas de búsqueda de información, la personalización consiste en que el listado de las soluciones que se ofrecen al usuario esté organizado en orden de preferencias. La forma óptima será aquella en la que el sistema muestra como primer elemento aquel que seguro va a interesar más que ninguna otra solución y los siguientes elementos estén organizados de mayor a menor importancia.

Para realizar esta operación de ordenación es necesario poder comparar las características del objeto posible solución y el modelo del usuario. Esa comparación se realizará utilizando un clasificador. Pero para poder comparar una solución con el modelo primero hay que construir el modelo.

Los IRS que obtienen información sobre documentos como [Pazzani1996] suelen utilizar un conjunto de palabras claves que se extraen del documento. Esas palabras claves se situarán en una clase del modelo de usuario que estará asociada a una evaluación. Por lo tanto, tras cada evaluación del usuario, se añade nueva información al modelo del usuario de forma que se tiene en una clase una lista de palabras claves con cierta puntuación. Éstas pueden a su vez estar en otras clases con otra puntuación, debido a que el usuario ha realizado una evaluación diferente de un documento que también la contenía.

Con los documentos divididos de esa forma es necesario utilizar alguna técnica para saber si un nuevo documento se asociará a una u otra clase. Esa operación es realizada por el clasificador.

Una de las tareas más importantes que debe realizar un sistema que trabaje con modelos de usuario en sistemas de búsqueda de información es la clasificación. En el modelo del usuario aparece información sobre documentos y las evaluaciones que el usuario hizo de esos documentos asociada al objetivo actual. Cuando llega un nuevo documento el sistema debe ser capaz de clasificar ese nuevo objeto dentro de la clase con la que tenga más relación [Billsus1997; Kononenko1990; Pazzani1996]

III.3. RECOMENDACIÓN PERSONALIZADA

En este apartado se introduce una clasificación de IRS orientados a ayudar a los usuarios en su tarea de recuperar información. La mayoría de los trabajos realizados sobre recomendaciones de documentos aparecen para solucionar las dificultades de los usuarios al encontrar documentos en la Web aunque en general no tienen por que ser exclusiva de ésta. Por ejemplo *GroupLens* [Resnick1994], *URN* [Brewer1994] realizan recomendaciones sobre *Usenet* o *News Dude* [Billsus1999] que informa al usuario sobre las nuevas noticias del mundo que más le interesan. Los sistemas recomendadores intentan facilitar la tarea del usuario en su búsqueda de información y para ello se encuentran dos técnicas principales: las recomendaciones basadas en el contenido (*content-based recommendations*) y las recomendaciones colaborativas (*collaborative recommendation*) [Balabanovic1997]. En los siguientes apartados se detalla en que consisten estas técnicas.

III.3.1 RECOMENDACIONES BASADAS EN EL CONTENIDO

En algunos sistemas se realizan recomendaciones basadas exclusivamente en el contenido de los documentos, es decir, cuando un usuario evalúa un documento como interesante, de este documento se extraen un conjunto de palabras claves que en cierto modo identifican el documento. Si la evaluación de ese documento es positiva, se utilizarán estos parámetros para buscar documentos similares. Por lo tanto, en este tipo de sistemas todas las recomendaciones que se hacen están basadas única y exclusivamente en la información que el sistema tenga sobre el usuario y sobre sus evaluaciones.

III.3.2 RECOMENDACIONES COLABORATIVAS

En muchas ocasiones las actividades que un usuario realiza con un sistema de búsqueda de información son muy parecidas a las realizadas por otros usuarios con intereses

similares. Por lo tanto, podemos plantearnos aprovechar las búsquedas realizadas por algún usuario para facilitar la tarea a otros 'afines' a él en un momento dado o con un objetivo de búsqueda parecido.

Un sistema de colaboración puro recomienda documentos a un usuario, no por su contenido, sino porque hay un usuario 'similar' al primero, que evaluó ese documento como interesante. Es decir, en este caso no se analiza la similitud entre documentos sino la similitud entre personas, que en algún momento anterior evaluaron de la misma forma algunos documentos. El problema de este tipo de sistemas únicamente colaborativos aparece cuando un nuevo documento llega al sistema. Hasta que un usuario no lo evalúe no se dispone de ninguna información para poder recomendarlo a nadie, pues no se hace un análisis de su contenido. Otro problema importante aparece en un sistema con pocos usuarios, donde es poco probable que dos de ellos evalúen un mismo documento y si lo hacen, que la evaluación sea igual. En este caso, el sistema sería bastante inútil.

III.3.3 SOLUCIÓN HÍBRIDA

Aunque el método basado en el contenido ha sido bastante utilizado en el área de IR con buenos resultados, la idea de completar las recomendaciones basadas en documentos anteriores con los documentos evaluados por usuarios con cierta similitud parece bastante atractiva. Es por tanto esta opción intermedia, más potente que cualquiera de las dos que la componen de forma individual y sin los problemas característicos de cada una de ellas, la que se ha utilizado en algunos sistemas 'recomendadores', como se verá a continuación.

III.4. TRABAJOS RELACIONADOS

En esta sección se analizarán los sistemas más representativos que aplican alguna de las soluciones vistas en el apartado anterior. Esta lista aunque no pretende ser exhaustiva puede servir como referencia para introducirse en este tema de los sistemas que realizan recomendaciones personalizadas.

III.4.1 EL SISTEMA FAB

FAB es un recomendador adaptativo de páginas Web [Balabanovic1995; Balabanovic1997] que divide el proceso de recomendación en dos partes: búsqueda de elementos para formar una base de datos o índice y selección de elementos de esa base de datos para usuarios concretos. En la fase de búsqueda se obtienen páginas que son relevantes para un número pequeño de temas. Un usuario puede estar interesado en varios temas y un tema puede interesar a varios usuarios.
La implementación se hace mediante agentes de *búsqueda* y de *selección*. Cada agente de *búsqueda* tiene un perfil basado en las palabras claves de las páginas que ha evaluado y está asociado a un tema. Por otro lado, cada agente de *selección* está asociado a un usuario. Cuando los agentes de búsqueda obtienen nuevas páginas, estas son enviadas a un *router* central que envía las páginas a los usuarios con un perfil similar al contenido de la página. Las páginas que ya han sido vistas por un usuario, se descartan a la hora de hacer una nueva selección.

FAB envía al usuario cada cierto tiempo una lista de nuevas páginas que debe evaluar en una escala de 7 puntos. Esas evaluaciones se utilizan para actualizar su agente de

selección y también los agentes de búsqueda. Sólo las páginas mejor evaluadas se envían a usuarios con perfiles similares.

El método de selección de palabras claves dentro de un documento es el TFIDF [Joachims1997]. Así obtienen las palabras más descriptivas de un documento que como máximo será de 100. Para calcular la similitud entre un documento y el perfil de un usuario utilizan la medida del coseno (agente de selección).

Cuando un usuario nuevo llega al sistema se le ofrece una serie de páginas aleatorias dentro de un conjunto que contiene las páginas que varios agentes creen como las que más interesan a la población actual que usa el sistema. Así el usuario empieza con mucha más información disponible que si empezase con un perfil vacío. Cada evaluación de los usuarios se almacena en un perfil global (*amalgamated profile)* que representa la media de la opinión de los usuarios.

Hay varios tipos de agentes buscadores disponibles (*collection agents*), los cuales, a partir de una página buscan documentos relacionados por enlaces en la propia página o los agentes de *índices* que utilizan buscadores como Altavista, Inktomi o Excite. Utilizan un número de palabras clave para realizar la consulta en cada buscador que varía entre 10 y 20.

Una posible crítica a este sistema es que si en un momento dado un usuario está interesado en buscar alguna información en concreto, no desea que el sistema empiece dándole la información general que a otros usuarios les interese.

III.4.2 PTV

El sistema PTV [Cotter2000] recomienda programas de televisión a un usuario que está consultando la programación. Esta programación es personalizada utilizando recomendaciones basadas en el contenido, para ello el perfil del usuario se parametriza con los elementos: canales, palabras clave, género favorito, programas, etc.... El modelo puede actualizarlo el usuario, aunque no suelen hacerlo de forma exhaustiva, o automáticamente utilizando la evaluación pertinente (RF). Para este sistema las evaluaciones de un programa van entre –2 y 2.

Este sistema también permite recomendaciones colaborativas utilizando los *k* usuarios más parecidos al usuario actual y se genera una lista con los *r* programas mejor evaluados. Para calcular la similitud entre usuarios utilizan una métrica propia aunque comentan que se podrían obtener resultados parecidos con la correlación de Pearson. A la hora de mostrar una lista de programas interesantes se seleccionan algunos elementos obtenidos utilizando las evaluaciones del usuario y otros entre los *r* elegidos de los usuarios parecidos. Este sistema funciona tanto para clientes WWW como WAP.

III.4.3 MOVIELENS[3]

Movielens [Good1999] es un sistema que recomienda películas utilizando las informaciones de otros usuarios con gustos similares al actual además de recomendaciones basadas solamente en el perfil del usuario. Utiliza diferentes agentes para recolectar información de diferentes fuentes y combinarlas para ofrecer los mejores

[3] MovieLens http://movielens.umn.edu/

resultados. En sus experimentaciones comparan los resultados; cuando sólo se usan las opiniones del usuario en películas anteriores; cuando sólo se utiliza un agente para realizar los filtros; o cuando se combinan múltiples agentes contando o no con la opinión del usuario. En este último caso, en el que se combinan resultados de agentes y opinión de usuarios han conseguido resultados interesantes.

III.4.4 WEBWATCHER[4]

Dentro de los sistemas que realizan recomendaciones basándose exclusivamente en los contenidos se encuentra WebWatcher [Armstrong1995]. El usuario introduce información sobre lo que busca en forma de palabras clave y ese será su objetivo (*goal*). Estos objetivos están restringidos a búsquedas de papeles técnicos donde se introducen datos como el autor, título, etc. El usuario empieza a navegar por la web bajo la supervisión de este sistema que le asiste para seguir los hiperenlaces que más se adaptan a las palabras claves introducidas. En sus experimentos utilizan para calcular la similitud entre un enlace y la página objetivo las técnicas de Winnon, Wordstat, TFIDF con la medida de similitud del coseno y aleatorio.

III.4.5 LETIZIA

El sistema Letizia [Lieberman1995] ayuda al usuario a localizar documentos interesantes para él. En este caso, no es necesario que el usuario introduzca palabras claves ni un objetivo de búsqueda. El agente Letizia acompaña al usuario en su búsqueda y aprende del comportamiento del usuario, es decir, realiza inferencias basándose en las acciones del usuario, como los enlaces que ha seguido. Letizia realiza búsquedas de documentos interesantes utilizando esa información durante el periodo inactivo en el que el usuario esta leyendo un documento para indicarle posibles enlaces a seguir, pero siempre permitiendo al usuario navegación libre. El conocimiento para realizar inferencias lo realiza siguiendo algunos heurísticos de los cuáles cabe destacar: i) Si un usuario añade una página a su lista de *favoritos* está indicando explícitamente que ese documento le interesa; ii) Si un usuario analiza una página y sale de ella es porque esa página no le interesa, pero si sigue sus enlaces entonces sí se considera como interesante.

El modelo que tiene del documento es una lista de palabras clave. Cuando Letizia encuentra un documento como interesante para recomendar, indica el motivo por el que lo ha elegido, que puede ser por ejemplo una palabra clave importante tanto en el documento actual como en los ya visitados.

III.4.6 SYSKILL & WEBERT

En este sistema [Pazzani1996] se recomiendan páginas a un usuario utilizando solamente razonamientos basados en el contenido. Realiza un perfil de usuario para cada tema que interesa a ese usuario. Estos temas (*topics*) están predefinidos y tienen una página índice construida manualmente con enlaces interesantes sobre cada uno. El usuario va evaluando las páginas de ese índice y esa información es utilizada para recomendar o no el resto de las páginas de ese índice. Aunque también añade la posibilidad de sugerir enlaces formando consultas en LYCOS. El algoritmo que utiliza para clasificar las páginas como interesantes es un clasificador bayesiano. Para poder

[4] WebWatcher http://www.cs.cmu.edu:8001/afs/cs.cmu.edu/project/theo-6/web-agent/www/project-home.html

crear el perfil, al igual que en otras aplicaciones, es necesario un conjunto mínimo de ejemplos positivos y negativos.

III.4.7 SITEIF

La propuesta de [Magnini2001] es un agente personal para leer noticias multilingües. Supervisa las acciones del usuario (noticias solicitadas) para actualizar su modelo. Intenta anticipar qué documentos podrían ser interesantes para el usuario. La característica destacable de este sistema es que su modelo no se basa exclusivamente en palabras clave sino en significados, para ello utiliza una red semántica multilingüe de libre acceso *MultiWordNet* que le permite recuperar documentos en italiano e inglés usando el mismo modelo de usuario.

III.4.8 IFWEB

Este sistema [Asnicar1997] es un asistente para los usuarios de la web. El sistema parte de un documento seleccionado por el usuario y recopila documentos en la web, analizándolos y clasificándolos. Como resultado muestra una serie de enlaces que pueden ser interesantes ordenados por su relevancia. Las estrategias que utiliza están basadas en un modelo de usuario basado en el contenido de los documentos. Dicho modelo incluye conceptos que representan los elementos interesantes y no interesantes de los usuarios. Utiliza una red semántica de nodos correspondientes a conceptos encontrados en los documentos con arcos a los términos que aparecen como coocurencias en el mismo documento. Esto soluciona algunos problemas como la polisemia. El usuario evalúa los documentos indicando interés positivo o negativo.

III.4.9 OTROS SISTEMAS

El sistema GASs [Barra2000] pretende que un grupo de personas con el mismo objetivo busque información en la Web y la compartan entre ellos, para lo que sería necesario tener un modelo de grupo además de un modelo de usuario. Como curiosidad, el desarrollo del sistema se basa en la programación de Proxies que se sitúan como intermediarios entre el cliente y el servidor Web.

En WebCobra [Vel1997] un usuario evalúa un conjunto de documentos y de ellos se obtiene un vector de palabras claves que identifican a ese usuario. Ese vector se envía a un servidor que utilizando un cálculo simple de similitud mediante el método del coseno, asigna al usuario a un grupo, este grupo es el considerado más afín al usuario actual. Cuando el usuario evalúa diferentes documentos selecciona aquellos que cree pueden interesar a los miembros de su grupo. Los temas de cada grupo se centran en dominios muy concretos para permitir la formación de los grupos fácilmente. El usuario puede pedir recomendaciones y recibirá todos los documentos marcados como importantes por otros usuarios del grupo. Este los evaluará entre 1 y 7, y eso se utilizará para modificar los pesos asociados a las palabras clave de su vector.

El sistema Casper/Jobfinder [Bradley2000] recomienda puestos trabajos utilizando razonamiento basado en casos, donde cada caso es un trabajo que ha sido anteriormente evaluado por el usuario. Para ver si un nuevo trabajo interesaría o no, se compararía con los casos similares anteriores. Con eso consiguen que esa selección sea un problema de clasificación. La técnica de similitud utilizada es una 'media estándar ponderada' utilizando como parámetros el tipo de trabajo, salario, experiencia mínima, etc... Por

otro lado también es colaborativo pues hace recomendaciones de usuarios similares, donde la similitud se calcula en función del número de trabajos que hayan evaluado en común.

III.4.10 RESUMEN

Con objetivo de tener una visión general de las características de los sistemas recomendadores, se muestra a continuación una tabla resumen con los sistemas analizados.

Por un lado se indica su clasificación como recomendador basado en contenido, híbrido o colaborativa pura, aunque ninguno de los sistemas analizados entra en esta categoría. Además se indica cómo estos sistemas modifican el modelo del usuario. Esta actualización puede ser: a) explícita, es decir, cuando el usuario dé una valoración para un documento, y b) mediante heurísticos, es decir, cuando el sistema no pide de forma explícita una evaluación. Los heurísticos más utilizados están basados en la navegación (nav) o en actividades como añadir un documento a la lista de favoritos (fav)

En la cuarta columna se clasifica el sistema como sistema de filtrado de información (IF) o de recuperación de la información (IR), aunque en algunos casos los sistemas tienen características de ambos. En la siguiente columna se indican algunas características relevantes del sistema, como si para empezar a recomendar necesita un entrenamiento inicial, si esta basado en Agentes, si utiliza palabras claves o significados (redes semánticas), etc. En la última columna se indica el dominio de aplicación del sistema.

Sistema	*Recomend. basadas en*	*Tipo Evaluación*	*IF/IR*	*Otras Características*	*Dominio/Notas*
Casper Jobfinder	Híbrido	Supervisión (Nav)	IR	Razonamiento Basado en Casos	Búsqueda Empleo
FAB	Híbrido	Explicitas [1,7]	IR	Entrenamiento/TFIDF/Agentes	Páginas Web
Gas	Híbrido	Supervisión (Nav)	IR	Actua como proxy	Páginas Web
IfWeb	Contenido	Explícita	IF/IR	Red semantica (coocurencias)	Páginas Web
Letizia	Contenido	Supervisión (Nav/Fav)	IR	Heurísticos para aprender	Páginas Web
MOVIELENS	Híbrido	Explícita [1,5]	IR	Entrenamiento/Agentes	Películas
PTV	Híbrido	Explicitas [-2,2]	IF	RF/correlación entre usuarios	Programas de Televisión
SiteIf	Contenido	Supervisión (Nav)	IF/IR	Multilingüe	Noticias Web
Syskill & Webert	Contenido	Explícita	IR	Clasificador Bayesiano	Páginas Web
WebCobra	Híbrido	Explicitas [1,7]	IR	Palabras clave iniciales/Coseno	Páginas Web
WebWatcher	Contenido	Supervisión (Nav)	IR	Palabras clave iniciales/TFIDF	Publicaciones Científicas

Tabla 6. Comparativa de los sistemas recomendadores

La mayoría de los sistemas analizados suelen tener una visión general de los intereses de los usuarios. Por ejemplo: Letizia [Lieberman1995], FAB [Balabanovic1997], PTV [Cotter2000], MOVIELENS [Good1999] o WebCobra [Vel1997] no tiene en cuenta los objetivos del usuario cuando realiza la búsqueda, es decir, tienen un modelo de usuario único que se aplica en cualquier sesión. El concepto de objetivo, fundamental en este

trabajo como se verá en el capítulo IV, no parece aplicarse en los sistemas actuales de recuperación de la información. La solución más sencilla para adaptar las recomendaciones al usuario consiste en tener varios modelos de usuario, cada uno asociado a un objetivo, con lo que el sistema aprenderá de acuerdo al contexto de la búsqueda. Algunos sistemas definen conceptos similares pero muy limitados. Por ejemplo, WebWatcher [Armstrong1995] permite indicar un objetivo, restringido a informes técnicos (que en realidad es una lista de palabras claves que se utilizan para hacer la búsqueda). Syskill & Webert [Pazzani1996] permite seleccionar entre un conjunto de temas para los cuales se ha construido un índice manualmente. Y GAS [Barra2000] supone que varios usuarios tienen un único objetivo común.

III.5. CONCLUSIONES

En este capítulo se ha introducido la necesidad de aplicar modelos de usuario a los sistemas de recuperación de la información para que estos ofrezcan los mejores resultados a los usuarios en sus búsquedas. Para ello se han mostrado diferentes conceptos generales sobre modelado de usuario. Posteriormente, se ha reducido el ámbito a los sistemas de recuperación de la información personalizada mostrando distintos métodos (basados en contenido o en evaluaciones de otros usuarios) y por últimos trabajos relacionados con el tema.

Entre los conceptos generales, destaca la utilización de estereotipos que puede ser muy interesante y útil en algunos sistemas para dar respuestas con algo de personalización cuando no se tienen muchos datos sobre el usuario. Un problema a tener en cuenta es que se clasifique erróneamente al principio al usuario dentro de un estereotipo y luego sea difícil desligarlo; eso implicará que muchas de las recomendaciones o decisiones basadas en este estereotipo no serían correctas. A veces puede ser mejor no dar respuestas personalizadas hasta que no se tenga algo más de seguridad sobre las características de esa persona que pueden ser relevantes para el sistema. Para el tipo de aplicación que se busca en este trabajo, éstos pueden no ser demasiado útiles. Como se dice en [Vassileva1994] los estereotipos no son muy utilizados para modelar las preferencias en IR. La principal dificultad es que no es fácil dar una clasificación sistemática de los usuarios, pues hay factores que influyen en la necesidad de información de un momento dado, como la tarea a realizar, el lugar, su profesión, etc. Según Vassileva, la única solución general será representar de forma explícita todos los factores y sus posibles valores para asegurar que se cubren todas las posibles combinaciones.

Los sistemas de recuperación de la información vistos en el apartado III.4, a pesar de todas sus virtudes carecen de una visión realista de la situación y de los intereses de una persona en un momento dado. En la mayoría de ellos se supone que una persona quiere siempre lo mismo y realizan recomendaciones centradas en lo que puede gustar o no pero en general. No tienen en cuenta en el mismo usuario puede tener necesidades diferentes a lo largo del tiempo. Por lo tanto parece claro que hay que añadir un elemento adicional que es la orientación de las recomendaciones por **objetivos**.

Otro elemento que consideramos importante es la posibilidad de revisar su historial, es decir los documentos ya visualizados, y de utilizarlo ya sea para modificar evaluaciones anteriores o simplemente para acceder a ese enlace. Esta característica no esta disponible prácticamente en ninguno de ellos, y algunos, como por ejemplo en Fab [Balabanovic1997] nunca recomienda documentos ya evaluados.

IV. MODELO TEÓRICO

IV.1. INTRODUCCIÓN

En los capítulos anteriores se han introducido las técnicas utilizadas en los sistemas de recuperación de la información (Capítulo **II**). Se ha visto la necesidad de incluir dentro del sistema características propias de cada individuo, que permitan que las búsquedas realizadas por los usuarios se adaptaran a sus necesidades. Para ello, en el Capítulo **III** se han mostrado las técnicas de modelado del usuario, y en concreto aquellas orientadas a la personalización de los sistemas de recuperación de la información.

En este capítulo se introducen las contribuciones teóricas de esta tesis a la personalización en los sistemas de recuperación de la información. Las propuestas aquí detalladas se han implementado en un sistema llamado METIORE[5] que se mostrará en el capítulo siguiente.

A continuación se describe la propuesta de la tesis en la que se aplican los conceptos de capítulos anteriores mostrando aportaciones que pretenden dar una nueva visión para los sistemas recomendadores centradas en objetivos. Para ello se introducirán una serie de definiciones de conceptos relevantes en este trabajo. A continuación se mostrarán las hipótesis de partida de la tesis y se continuará analizando las técnicas de búsqueda y análisis de datos utilizados en esta propuesta. Además se introducen las características que debe tener un modelo de usuario, así como los algoritmos de personalización desarrollados. Este es un punto importante pues se proponen mejoras sobre algoritmos existentes como por ejemplo *Naïve Bayes* (en adelante NB). En concreto se propone un algoritmo llamado NBM (*Naïve Bayes Metiore*) que dado un documento permite calcular, para un objetivo del usuario, cual será su evaluación más probable. También se mostrará su extensión WNBM (Weighted *Naïve Bayes Metiore*) que permite ponderar la importancia de los distintos parámetros del documento.

Aunque NBM y WNBM pueden aplicarse a distintas bases de datos, para los ejemplos principales de este capítulo se utilizará una base de datos bibliográfica en la que el

[5] METIORE (***M**ultimedia coop**E**rative Informa**TIO**n **R**etrieval Syst**E**m*)

elemento principal es un artículo o libro con los parámetros: *título, autor, editorial, palabras clave, año, lugar, volumen, páginas, publicado_en, etc.*

IV.2. DEFINICIONES

En este apartado se definen algunos conceptos fundamentales para el desarrollo de la propuesta de esta tesis.

Objetivo

> *El **objetivo** de un usuario es la expresión en lenguaje natural que representa la necesidad de información del usuario cuando utiliza el sistema de búsqueda de información*

Sesión

> *Se llama **sesión** al espacio de tiempo continuo que el usuario utiliza el sistema para satisfacer una necesidad concreta de información. Durante una sesión el usuario puede realizar múltiples consultas, aunque todas ellas están dirigidas a resolver esa necesidad.*

Actividad

> *Dentro de esta propuesta, se define como **actividad** a cualquier interacción del usuario con el sistema. Hay diferentes tipos de actividades que se detallarán posteriormente, como búsqueda, evaluación, pedir recomendación, ver también o explotación del historial.*

Evaluación

> *La acción de **evaluación** consiste en proporcionar al sistema una valoración sobre un documento que el sistema ha mostrado al usuario como respuesta a una consulta. Esta actividad es fundamental para poder ayudar al usuario y crear un perfil que represente sus preferencias.*

En algunos sistemas, el usuario puede expresar su necesidad de información a través de un conjunto de consultas, pero estas consultas se deben escribir usando un lenguaje específico. Esto limita su capacidad de expresión y en muchas ocasiones no sabe exactamente qué desea encontrar ni cómo interactuar con el sistema para obtener resultados interesantes. Algunas de estas dificultades se presentan en [Vassileva1994].

Cabe preguntarse ¿cómo podría el concepto de objetivo ayudar a obtener una respuesta más relevante y personalizada? El objetivo se utiliza como un identificador que agrupa un conjunto de consultas, conceptos y decisiones que hace el usuario teniendo ese objetivo en mente. La aproximación de personalización está muy ligada al concepto de objetivo. Muy pocos sistemas utilizan elementos similares, y donde los hay no se les da la misma importancia que en este trabajo. La mayoría de los sistemas que ofrecen algún tipo de personalización suelen tener una visión general de los intereses de los usuarios como se mostró en III.4. Un ejemplo, muy sencillo para comprender la necesidad de los objetivos puede ser el siguiente: "*A veces, si estoy trabajando sobre la tesis, estaré muy interesado en que un buscador que tiene mi perfil de usuario, me ayude a encontrar documentos relacionados con mis intereses sobre modelado del usuario o recuperación de la información. Pero en otro momento, a lo mejor estoy buscando información sobre programación de juegos para preparar las clases. En ese caso, mi objetivo de búsqueda es diferente y las conclusiones del sistema sobre mi modelo cuando busco modelado del*

usuario, no deben modificarse porque haya evaluado como interesante una página de juegos, ni estoy interesado en que en ese momento me recomiende documentos sobre recuperación de la información". Ese concepto fundamental en este trabajo, no parece aplicarse en los sistemas actuales de recuperación de la información. La solución más sencilla para ese ejemplo es tener dos modelos de usuario, uno asociado a cada objetivo, con lo que el sistema aprenderá un modelo de usuario en función de las necesidades en cada momento del usuario.

IV.3. HIPÓTESIS

En este apartado se exponen las hipótesis que se habían planteado con esta tesis y que mediante cuestionarios a los usuarios y otras evaluaciones en su mayoría se han probado válidas. El análisis de resultados se realiza en el capítulo de Experimentación (VI). Las hipótesis, organizadas por conceptos son las siguientes:

Objetivos del usuario y Actividades

1. Agrupar las interacciones de los usuarios en objetivos ayuda al usuario a encontrar información relevante.
2. La integración de las actividades del usuario y su asociación con el objetivo para el cálculo de soluciones debería proponer más soluciones que el usuario evaluará como relevantes, que si esta integración no se hiciese.

Historial

3. La exploración del historial activo debería facilitar la operación de recuperación de la información y ayudar a ofrecer soluciones relevantes

Evaluación

4. La posibilidad del usuario de dar evaluaciones debería ayudar a comprender mejor sus necesidades de información.

Ordenación de resultados y códigos de colores

5. Clasificar las soluciones por tipo de evaluación y la asociación de códigos de colores debería facilitar la recuperación de soluciones.

Cooperación

6. La posibilidad de recuperación cooperativa debería acelerar el proceso de encontrar soluciones relevantes.
7. Cuando el usuario no tiene historial, el sistema puede ofrecer ayuda utilizando historiales de otros usuarios.
8. La explotación del historial de otros usuarios debería acelerar el acceso a soluciones relevantes.

IV.4. MÉTODOS DE RECUPERACIÓN DE LA INFORMACIÓN

A continuación se proponen las características fundamentales del modelo de recuperación propuesto. Este modelo utiliza métodos clásicos de búsqueda de información, completados con un método de análisis global de la información de la base de datos. Este análisis global aporta informaciones sobre el dominio. A esta técnica se le llama en este trabajo *análisis cruzado con restricciones*. Por lo tanto, se dispone de un modelo híbrido para la recuperación de datos, compuesto de:

- Un **modelo booleano.-** Se indexan los documentos por diferentes descriptores, y se pueden realizar consultas booleanas utilizando conectores AND/OR y comparaciones (termino op valor).
- Un **modelo vectorial.-** Que permite análisis multiatributos, construyendo clusters, y devuelve resultados en función de la frecuencia de aparición y de la similitud de los resultados con la consulta.
- Un **modelo probabilístico.-** Que tiene en cuenta las evaluaciones del usuario de forma que se pueda proponer una respuesta personalizada al usuario, ofreciendo en primer lugar los documentos que probablemente interesarán más al usuario.
- Un **modelo hipertexto dinámico.-** Que permite al usuario acceder a la información de múltiples formas siguiendo enlaces, mediante consultas intra e inter-campo o buscando documentos relacionados con uno dado.

IV.4.1 BÚSQUEDA CLÁSICA CON RESTRICCIONES

En nuestra aproximación mixta, las restricciones se corresponden con los criterios de las consultas booleanas clásicas. Dichas restricciones pueden expresarse en notación BNF de la forma:

$$< \textit{consulta} >::= \{NO\} < \exp > \{(Y \mid O)\{NO < \exp >\}\}$$
$$< \exp >::= \textit{campo} < \textit{oprel} > \textit{valor}$$
$$< \textit{oprel} >::= (=|<|<=|>|>=|!=|*=)$$

Ecuación 16. Restricciones booleanas

Como ejemplo, una posible consulta a la base de datos bibliográfica del ejemplo sería:

```
(editorial='Springer') Y (año>1998)
```

En un sistema booleano clásico, esto daría como resultado un listado de los libros de la editorial *Springer* publicados con posterioridad al año 1998, pero no daría ninguna distribución por años. Esa es una de las razones por las que se ha completado la posibilidad de hacer consultas con el análisis de datos.

IV.4.2 ANÁLISIS DE DATOS

Los tipos de documentos que se tratan en los IRS suelen estar estructurados, mediante algún preprocesamiento para realizar búsquedas. Por ejemplo, si se trabaja con películas de vídeo, se tienen parámetros como director, título, resumen, actores, etc. Si se trabaja con publicaciones: autores, título, palabras clave, año, etc. Prácticamente cualquier tipo de datos puede descomponerse en diferentes parámetros y podrá ser utilizado en este modelo.

Se pueden hacer varios tipos de análisis [David1999] dependiendo del número de campos/parametros involucrados. A dichos análisis les llamamos:

a. **Análisis de Frecuencia**.- Muestra la distribución de cada valor posible para ese campo.
b. **Análisis cruzado intra-campo**.- Para analizar la co-ocurrencia de dos o más valores del mismo campo

c. **Análisis cruzado inter-campo**.- Para analizar co-ocurencias de valores de dos o más campos diferentes.
d. **Análisis + restricciones.-** Permite los análisis anteriores limitándolos a los datos que cumplan alguna condición.

Estas técnicas de análisis ofrecen un valor añadido con relación a los buscadores tradicionales ya que permitirán al usuario tener un mayor conocimiento del contenido de la base de datos, sin necesidad de tener ninguna idea inicial de mismo. Esto también ayuda a aclarar el objetivo exacto de su interacción, pues a veces el usuario tiene una idea aproximada de lo que quiere, pero no sabe como expresarlo, o no sabe que datos están disponibles en el sistema. Los ejemplos prácticos de la aplicación se muestran en V.2.4 y V.5

a) Análisis de frecuencia

El análisis de frecuencia ofrece una visión global de la distribución de valores de un campo. Siendo este campo cualquier atributo de los documentos de la base de datos. Representamos este análisis como:

$$f_j = \sum_{i=1}^{n} \left| \vec{D}_i \cap v_j \right|$$

donde

$$\left| \vec{D}_i \cap v_j \right| \in \{0,1\} \quad y \quad v_j \in V$$

Ecuación 17. Análisis de frecuencia

En la Ecuación 17, D_i representa el conjunto de términos utilizados para describir el documento *i*, *V* es el conjunto de posibles valores del campo analizado, *n* es el conjunto de documentos del sistema y f_j contiene el número de documentos donde aparece el término v_j. También puede representarse el mismo análisis de forma matricial:

Valor del campo	**v_1 v_2 ... v_m**
Frecuencia	f_1 f_2 ... f_m

Por ejemplo, para una base de datos de publicaciones científicas se podría hacer el análisis de frecuencia para el campo autor y se obtendría algo como:

Autor	**Kobsa**	**Brusilovsky**	**Milosavijevic …**
Frecuencia	14	12	9

Lo que indica el número de documentos contenidos en la base de datos en la que cada autor ha publicado. La interpretación de esta información adicional de la base de datos dependerá del usuario, y para ciertos dominios, será necesaria la contribución de expertos que determinen qué campos son de interés para este tipo de análisis. Por ejemplo, la aplicación de este sistema a una empresa, con los datos de las ventas de ordenadores, puede ayudar a mostrar qué componente, por ejemplo, qué procesador ha sido el más vendido en el último año, disponiendo en el sistema exclusivamente información de los ordenadores vendidos (con sus componentes).

b) Análisis cruzado intra-campo

El objetivo principal de éste tipo de análisis es estudiar la distribución de co-ocurrencias de términos pertenecientes a un solo campo o criterio. Permite ver qué valores de un campo aparecen en el mismo elemento de la base de datos. Representamos el análisis intra-campo por:

$$f_{jk} = \sum_{i=1}^{n} \left| \vec{D}_i \cap \{v_j, v_k\} \right|$$

donde

$$\left| \vec{D}_i \cap \{v_j, v_k\} \right| \in \{0,1\} \quad y \quad v_j, v_k \in V$$

Ecuación 18. Análisis cruzado intra-campo

En la Ecuación 18, D_i es el conjunto de términos utilizados para describir el documento i, V es el conjunto de valores del campo analizado, n es el número de documentos del sistema y f_{jk} contiene el número de documentos en los que aparece el término v_k y el término v_j. La representación matricial de dicho análisis se muestra a continuación:

Valor del campo	**v_1 v_2 … v_m**
v_1	f_{11} f_{12} … f_{1m}
v_2	f_{21} f_{22} … f_{2m}
…	…
v_m	f_{m1} f_{m2} … f_{mm}

En la siguiente tabla se muestra un ejemplo de análisis intra campo en una base de datos de publicaciones. El campo analizado es el de *palabras_clave*, y permite estudiar que palabras claves suelen aparecer juntas, o están relacionadas desde el punto de vista del tema de investigación:

Valor del campo	**Artificial_Intelligence**	**KB_systems**	**Natural_Language**
Artificial_Intelligence		13	7
KB_systems	13		1
Natural_Language	7	1	

c) Análisis cruzado inter-campo

El análisis cruzado inter-campo se utiliza cuando los documentos de la base de datos están organizados en diferentes campos. Consiste en ver qué documentos contiene co-ocurencias de valores que pertenecen a campos distintos del documento. El análisis intra-campo puede considerarse como mono criterio, pues sólo analiza un campo. El análisis inter-campo es un análisis multicriterio, ya que se analizan conjuntamente distintas características de los documentos. Lo representaremos como:

$$f_{jk} = \sum_{i=1}^{n} \left| \vec{D}_i \cap \{v_j, w_k\} \right|$$

donde

$$\left| \vec{D}_i \cap \{v_j, w_k\} \right| \in \{0,1\} \quad y \quad v_j \in V, w_k \in W$$

Ecuación 19. Análisis cruzado inter-campo

En la Ecuación 19, D_i es el conjunto de términos utilizados para describir el documento *i*, *V* es el conjunto de valores del 1er campo analizado y W es el conjunto de valores del 2º campo utilizado, *n* es el número de documentos del sistema y f_{jk} contiene el número de documentos en los que aparece el término v_j y el término w_k. La representación matricial:

Valor del campo 1\valor del campo 2	**w_1**	**w_2**	**…**	**w_m**
v_1	f_{11}	f_{12}	…	f_{1m}
v_2	f_{21}	f_{22}	…	f_{2m}
…	…			
v_k	f_{k1}	f_{k2}	…	f_{km}

En el siguiente ejemplo se muestra el resultado de una consulta autor/año, que permite ver la cantidad de publicaciones y evolución temporal de los autores. Un posible resultado sería:

Autor\Año	**1996**	**1997**	**1998**	**1999**
Brusilovsky, Peter	3	3	2	1
Kobsa, Alfred	2	3	2	3

d) Análisis con restricciones

En los tres apartados anteriores se muestran tres tipos de análisis de datos de forma general. Su potencia se incrementa de forma importante cuando se complementan con restricciones. Es decir, se pueden hacer análisis de datos, pero no necesariamente de forma global a la base de datos, sino restringidos a las condiciones que el usuario desee. Por ejemplo, se puede estar interesado en analizar qué autores trabajan juntos, pero refiriéndose a algún autor en concreto.

En principio no hay limitación teórica en el número de campos a añadir. Cada campo que se añada abre nuevas posibilidades de consultas. Por ejemplo, si se añade un tercer campo, la consulta: *autor, palabra clave, año;* permite ver la evolución de un autor en sus investigaciones, a través de las palabras claves utilizadas en cada época. A continuación se muestra el esquema general para análisis con tres campos y un ejemplo de análisis con restricciones con tres campos en el que se han seleccionado los atributos autor, palabras clave y año, y como restricción *autor= Bueno, D.* Esta restricción hace que se muestren todas las tripletas pertenecientes a los documentos con el autor *Bueno, D.* Eso explica que aparezcan otros autores que aunque no estén en la restricción, están en los documentos donde aparece dicha restricción.

Frecuencia	**Valor campo1**	**campo2**	**campo3**
f_{111}	**v_1**	**w_1**	**z_1**
f_{112}	**v_1**	**w_1**	**z_2**
…	**v_i**	**w_j**	**z_k**
v_k			

Frecuencia	Autor	palabras clave	año
2	Bueno, D.	El Castillo	1997
2	Bueno, D.	ITS	1999
3	Bueno, D.	METIORE	1999
1	Bueno, D.	METIORE	2002
1	Conejo, R.	METIORE	2002

En el Apéndice II se muestran con detalle las posibilidades de estas herramientas para hacer análisis bibliométricos. En el capítulo siguiente se verá cómo se genera esa información en una aplicación concreta para optimizar las consultas.

IV.5. MODELO DE USUARIO

Cuando se plantea la idea de personalizar la interacción con el usuario, para que sus respuestas sean diferentes para cada persona y adaptadas a sus necesidades, se plantean algunas preguntas que hay que resolver:

- ¿Qué representar?
- ¿Cómo obtener la información del usuario?
- Formas de explotar el modelo del usuario

En los siguientes subapartados se darán las líneas más importantes de los pasos seguidos en esta tesis para resolver estos problemas.

IV.5.1 ¿QUÉ REPRESENTAR?

Para poder personalizar la respuesta a un usuario, el primer elemento necesario es tener información sobre él. Elegir cuidadosamente qué información va a almacenarse hará que el resto de los pasos sea menos complicado. Por un lado hay que almacenar información para identificar al usuario, por ejemplo: nombre, apellidos, e-mail, nacionalidad, profesión, página Web, etc. Esta información puede no ser demasiado relevante para la personalización, aunque de ahí se podría obtener, por ejemplo, utilizando la profesión, una estimación del nivel de conocimiento. Otro tipo de información puede ser la relacionada con opciones de configuración como el idioma o preferencias de la interfaz. Estos elementos harán que el usuario se sienta cómodo cuando interactúe con el sistema.

La información almacenada hasta el momento carece de interés si se desea conseguir respuesta personalizada. Los datos útiles se obtendrán observando al usuario mientras interactúa con el sistema. Para obtener esos datos se van a almacenar todas las interacciones agrupándolas de una forma lógica en distintas categorías. Cada una de esas interacciones se denominan *actividades* y se detallan en el apartado IV.5.4.

Dentro de las actividades cabe destacar la *evaluación*, que permitirá conocer los gustos del usuario de forma explícita. Cuando éste evalúa un objeto de los propuestos por el sistema como respuesta a alguna consulta, su pronunciamiento sobre la corrección de la respuesta con su objetivo puede ser una información muy importante pues ayuda a conocer algo más sobre sus preferencias. Por lo tanto, será necesario almacenar cuales

son los objetos que el usuario ha evaluado, y cuál ha sido su valoración para cada objeto.

Podría ser difícil extraer conclusiones sobre las preferencias de un usuario si únicamente se utilizaran las evaluaciones asociadas a los objetos como entidades únicas. Pues sin conocer su composición, no podría relacionarse con otros objetos. Por ejemplo, si el sistema trabaja con una base de datos de discos, no sería suficiente conocer qué disco ha evaluado como interesante, para poder recomendarle otros discos similares, sino que sería necesario conocer el cantante/grupo, genero, año, casa discográfica, etc, para poder hacer una recomendación precisa. Por lo tanto hay que almacenar también información obtenida al realizar un análisis de los objetos y sus parámetros. Sobre cómo se generan y se almacenan estos datos se habla con detalle en IV.6.

Por último, y con el objetivo de evaluar la eficiencia, se deben almacenar las predicciones realizadas. Para cada objeto que el usuario evalúe, se hará una predicción de cual será su evaluación, además se guardarán las evaluaciones del usuario para cada predicción. Con esto se podrán generar estadísticas sobre la precisión del sistema como recomendador.

IV.5.2 ¿CÓMO OBTENER LA INFORMACIÓN DEL USUARIO?

Existen al menos tres formas diferentes de obtenerse los datos de un usuario. En primer lugar estaría la recogida automática de la información tras cada acción que el usuario realiza, por ejemplo, almacenando los datos y características de sus búsquedas. De forma también automática, podría realizarse una síntesis de los datos obtenidos del usuario para mejorar el modelo. Por otro lado, mediante interacciones conscientes del usuario, podría mejorarse el modelo, ya sea mediante formularios en los que el usuario indique sus objetivos, preferencias, o incluso modificaciones de algunos parámetros del modelo. Un tercer modo puede ser mediante el análisis del historial de un experto humano en un contexto de recuperación de la información cooperativa. En este último caso, el experto puede deducir en función de las interacciones del usuario cuál es su verdadera necesidad de información y utilizar esto para mejorar el modelo.

IV.5.3 FORMAS DE EXPLOTAR EL MODELO DEL USUARIO

Una vez construido el modelo del usuario se pueden ver tres maneras de explotar los datos. En primer lugar será el propio usuario el que pueda acceder a la información que el sistema tiene sobre sus interacciones para trabajar por ejemplo con su historial (Ver IV.5.7).

En segundo lugar, y quizás el más importante para este estudio, es la utilización del modelo realizada por el sistema. Éste utiliza las evaluaciones anteriores para determinar las soluciones que mejor se adapten al objetivo actual. Para cada resultado posible, el sistema debe ver si el objeto ha sido ya evaluado por el usuario. Si no es así, hay que calcular su ‘grado de pertinencia’ para el objetivo actual y cuál sería la posible evaluación que el usuario realizaría para este documento. Además el sistema utilizará los criterios de la búsqueda y las evaluaciones para calcular la pertinencia de las soluciones. Eso permitirá recomendar, ya sea de forma absoluta (sin que el usuario realice ninguna consulta), o como resultado a una búsqueda, cuáles son los objetos que podrían interesar más al usuario.

El tercer caso consiste en la ayuda de un experto/colega en un contexto de recuperación cooperativa. Este compañero humano, que conoce el sistema, puede guiar al usuario a encontrar la información que necesita utilizando las técnicas que se mostrarán en V.4. Esto permitirá sacar al usuario de un apuro si no es capaz de encontrar la solución ni por si mismo ni con la ayuda del sistema. Esta tercera opción también es muy interesante para realizar trabajos a distancia que necesiten la colaboración de varias personas para encontrar algún tipo de información.

IV.5.4 ACTIVIDADES DEL USUARIO

Como se definió en el apartado IV.2, a cada interacción del usuario con el sistema se le llama *actividad*. En la Fig. 3 se muestran los tipos de actividades contempladas en esta tesis. En función de las actividades que realice el usuario, se podrán obtener algunas conclusiones sobre su conocimiento sobre el sistema, sobre los datos que contiene y sus necesidades y preferencias. Debido a la orientación de esta propuesta a la recuperación de la información y a la gran variedad de posibilidades diferentes que tiene el usuario para buscar información, existen distintos tipos de actividades de búsqueda en función de la modalidad elegida.

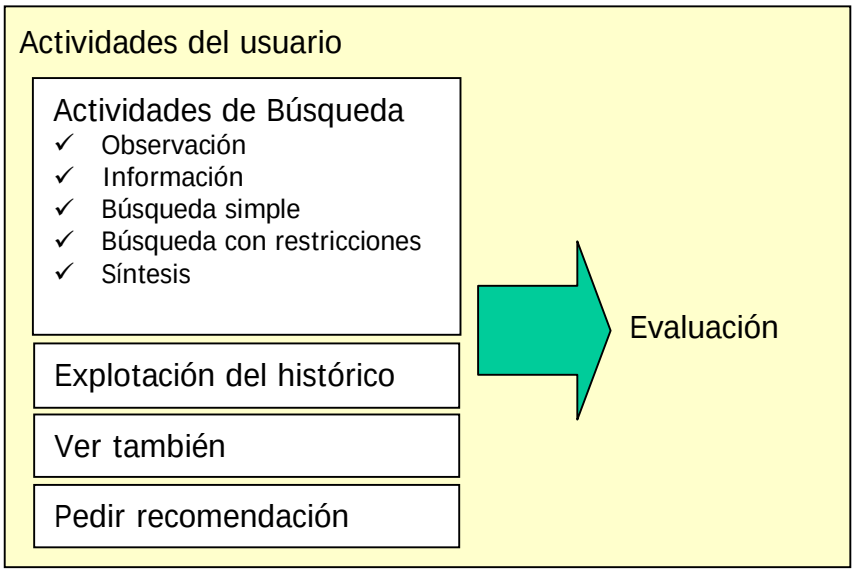


Fig. 3. Tipos de actividades

Las *actividades de búsqueda* se diferencian entre sí por el número de atributos utilizados, el uso o no de restricciones, su número, etc (Ver IV.4.2) . A continuación se detallan estos tipos en los que, además, se hace un análisis sobre el usuario:

- *Actividad de Observación.*- La consulta que realiza el usuario contiene solamente un atributo. Como respuesta se da sólo la lista de valores de este atributo agrupados por frecuencia de aparición. Esto quiere decir que el usuario tiene un medio de visualizar los diferentes valores y su importancia en la base de datos. Cuando un usuario utiliza este tipo de búsqueda, es posible que desconozca el tipo de datos contiene el sistema, y necesita tener una 'panorámica' de los distintos elementos almacenados utilizando algún atributo en concreto. (Ver IV.4.2.a)
- *Actividad de Información.*- En este caso el usuario utiliza para su búsqueda una sola restricción, es decir, un único criterio de búsqueda, sin seleccionar de forma explícita la lista de valores de ningún atributo. Esta forma de trabajar indica que el

usuario posee algunas ideas sobre lo que quiere, pero esas ideas están muy limitadas, asociadas a alguna restricción/característica en concreto.

- *Actividad de Búsqueda Simple*.- Es la más conocida, en contra de lo que sería deseable en un sistema con muchas otras posibilidades. Quizás sea así porque los usuarios están más acostumbrados a este tipo de búsqueda. Es el modelo clásico de cualquier buscador Web. En él se pone una serie de palabras y se pulsa un botón para empezar la búsqueda. Este modelo no requiere mucho conocimiento del sistema.
- *Actividad de Búsqueda con Restricciones*.- Esta debería ser el tipo de actividad más común realizada por usuarios con cierto dominio del contenido de la base de datos. En ella se utiliza un atributo y una o más restricciones, con lo que se puede reducir de forma importante el número de respuestas posibles con la selección del atributo adecuado.
- *Actividad de Síntesis*.- Está relacionada con la clasificación de datos relacionados con dos o más atributos y opcionalmente con uso de restricciones. Las técnicas utilizadas son de análisis cruzado de datos inter-campo e intra-campo (Ver IV.4.2.b, c y d). Este tipo de consultas están realizadas por usuarios que conocen bien el sistema y que quieren hacer algún tipo de estudio sobre las distribuciones de información contenidas en la base de datos.
- *Ver También*. Esta actividad consiste en la búsqueda de otros documentos relacionados en algún aspecto con el documento actual. Esto puede hacerse de forma general, es decir, buscar documentos donde aparezca alguno de los valores de los atributos del documento original, o filtrarlos utilizando varios valores de atributos que aparecen en el documento. Por ejemplo, si fuera una base de datos bibliográfica, seleccionando uno de los autores y dos palabras clave del documento actual. Se buscarían todos los documentos con esos elementos. Este tipo de actividad permite al usuario, una vez que ha encontrado un documento que le interesa, seguir profundizando con otros relacionados, pero no sólo de una forma general y descontrolada, como la que hacen algunos buscadores Web, sino indicando en qué van a parecerse con el documento actual aquellos que se busquen.
- *Explotación del historial.* Esta es otra actividad importante en la que el usuario puede trabajar con documentos vistos anteriormente. Este tipo de actividad se ve con detalle en IV.5.7.
- *Pedir Recomendación.* Algunos usuarios prefieren ver documentos que evaluar sin tener que hacer muchas búsquedas. Para ese caso, está disponible esta actividad. Cuando se pide una recomendación, se compara el modelo del usuario con todos los documentos que tengan algún atributo en común con el modelo y se clasifican. Esta clasificación da como resultado un listado, ordenado por importancia para el usuario. El algoritmo de similitud entre documentos y el modelo del usuario es el mismo que se utiliza para cualquier otro tipo de búsqueda y se desarrolla en IV.6. La actividad de recomendación sólo debería utilizarse después de que el usuario haya encontrado al menos un documento que le interese, pues en otro caso no tendría sentido ya que si todos los documentos evaluados por el usuario no le interesan, el sistema no puede predecir qué es lo que le gusta, sin tener ningún ejemplo.

Para cualquiera de las actividades anteriores, al final se llega a un documento, en este punto se debería realizar una nueva actividad de *Evaluación*, que hará que el modelo del usuario crezca y se refuerce. La falta de actividades de evaluación es indicativa de que el usuario no tiene demasiado interés por las recomendaciones del sistema.

IV.5.5 ANÁLISIS DE LAS ACTIVIDADES

Para estudiar la forma de uso de los usuarios del sistema, se puede analizar la información que se registra sobre las actividades para obtener algunas conclusiones, que den los distintos perfiles de posibles usuarios (En el apartado **V.7** se muestra la herramienta desarrollada para hacer este tipo de análisis). Para este tipo de estudios se puede utilizar el análisis de frecuencias para observar distintos parámetros:

- Si se analiza el campo *tipo de actividad*, se pueden observar sus distribuciones, ofreciendo una visión global sobre las preferencias de un usuario. A continuación se muestran algunos ejemplos con el posible análisis.

  En la Tabla 7 se muestran los análisis de frecuencia para cuatro posibles usuarios (U1,U2,U3,U4). El uso dominante de algún tipo de actividad de búsqueda en concreto puede dar pistas sobre los objetivos o estrategias de los distintos usuarios.

Tipo Actividad	Frecuencia U1	Frecuencia U2	Frecuencia U3	Frecuencia U4
Observación	10	10	2	2
Información	1	8	2	1
Búsqueda	2	2	8	3
Búsqueda Simple	2	2	2	9
Síntesis	0	1	1	0

Tabla 7. Ejemplos de tipos de actividades para cuatro hipotéticos usuarios

  a) Si las actividades de observación y de información son mayoria (U1 y U2), esto puede indicar que el usuario no posee demasiado conocimiento sobre el dominio.
  b) La mayoría en las actividades de búsqueda simple puede indicar conocimiento de los contenidos de la base de datos, o una idea muy concreta de lo que se busca sin saber exactamente a qué parámetro corresponde. Por ejemplo, para referencias bibliográficas se puede buscar: "*springer 2001 user modeling*", consulta que si se realizase con una búsqueda compleja requeriría saber que *Springer* es la editorial, *2001* el año y *user modeling* un título o palabra clave.(U4)
  c) Si la actividad de búsqueda es prioritaria junto con el uso de evaluaciones puede interpretarse como un usuario que posee un buen conocimiento del dominio (U3)
  d) Por otro lado, si la actividad mayoritaria es de análisis, probablemente el usuario tenga un buen conocimiento del dominio de los datos y no esté buscando una solución concreta, sino más bien un análisis de la información del sistema.

- Además de los tipos de actividades, puede ser interesante observar la frecuencia de los términos utilizados para las búsquedas simples, o para las restricciones de las consultas. Éstos son indicadores de los dominios de interés del usuario. Además, estos términos pueden dar una idea del nivel de conocimiento del usuario en el dominio.
- Un análisis similar al anterior puede realizarse para los términos que forman parte del objetivo, pues en la formulación del objetivo en lenguaje natural se puede

encontrar el fundamento de las necesidades de ese usuario (aunque no siempre ha de ser así).

IV.5.6 EVALUACIONES DE LOS USUARIOS

Uno de los elementos más importantes de cara a la personalización es la evaluación de documentos por parte de los usuarios. En los estudios realizados para el desarrollo de esta tesis se han analizado distintos tipos de evaluaciones. Prácticamente todos utilizan lo que se llama evaluaciones positivas [Schwab1999] es decir, para el modelo del usuario sólo tienen en cuenta las evaluaciones en las que el usuario está interesado con el documento que se le muestra. También en la mayoría de los casos la evaluación se limita a un par (interesante, no interesante), actuando sobre el modelo del usuario sólo en el caso en que la evaluación sea *interesante*. En la aproximación de esta tesis se ha querido ir más allá. Cuando un usuario analiza un documento, puede haber mucha información más relevante que sí/no. Por ejemplo, ¿Por qué sí o por qué no?. Es decir, cuáles son las razones que tiene el usuario para evaluar de esa forma. Conocer esas razones podría ayudar a entender mejor al usuario y poder recomendarle documentos con más precisión. Los algoritmos que se proponen en IV.6 se han preparado para poder reconocer diferentes niveles y causas de satisfacción, que pueden variarse para diferentes bases de datos o tipos de usuarios. Una primera aproximación, consiste en tener cuatro posibles evaluaciones:

- *La solución es relevante (ok).* Esto quiere decir que la solución propuesta por el sistema interesa al usuario y está en la línea de lo que busca.
- *La solución no es relevante porque el usuario ya la conoce (known).* Si el usuario evalúa de esta forma quiere decir que la solución es interesante para este objetivo, pero que ya la conoce. Esa información puede utilizarse para reforzar evaluaciones positivas aunque de forma diferente a si se hubiera evaluado como *ok*. En sistemas con respuesta interesante/no interesante, es probable que este documento se hubiera evaluado como no interesante (pues en ese momento no es lo que el usuario busca) perdiendo una información que puede ser útil.
- *El usuario no opina sobre la relevancia del documento para su objetivo (*?). Esta evaluación se aplica en el caso en que el usuario no fuera capaz de dar otro tipo de juicio para esa solución, quizás porque la información actual que se le ofrece, no le permite ubicarse sin más datos. Cuando tenga clara su postura sobre ese documento, podrá cambiar su evaluación.
- *La solución no es relevante porque no se corresponde al objetivo del usuario (wrong)*. El usuario juzga la solución como irrelevante para el objetivo actual.

Además de esas posibles evaluaciones, se tiene:

- *Soluciones no evaluadas (normal).* Esta es la evaluación por defecto, que indica en las soluciones propuestas al usuario que no han sido evaluadas.

Aunque desde el punto de vista de la investigación resulta más interesante conocer las distintas razones del usuario para comprenderlo mejor, el modelo propuesto en esta tesis, puede adaptarse a otro tipo de evaluaciones más simples de comprender para el usuario. Otro conjunto de evaluaciones aplicables, aunque más tradicional es el siguiente:

- *Muy interesante/Interesante (ok).* El usuario tiene dos posibles evaluaciones para indicar su satisfacción con el documento. En realidad cualquiera de las dos hace una evaluación positiva, aunque internamente cambia la ponderación que se da a los términos del documento.

- *Poco interesante/Nada interesante (wrong)*. De la misma forma, puede mostrar su desacuerdo con el documento en dos niveles.
- *No lo sé (?)*. Igual que en el conjunto de evaluaciones anteriores indica que el usuario no está seguro de cómo pronunciarse sobre el documento que se le presenta y toma ésta que es la opción más conservadora.

Un tercer conjunto que ofrecería alguna información adicional sobre por qué evalúa el usuario un objeto de una u otra forma y la relación entre el objeto que se analiza y las necesidades del usuario se muestran a continuación:

- Se corresponde con mi dominio
- No se corresponde con mi dominio
- El tema del documento es nuevo para mí
- El tema del documento NO es nuevo para mí
- Las propuestas son útiles para mi trabajo
- Las propuestas NO son útiles para mi trabajo

Este tipo de propuesta ofrece mucha más información que juzgar algo como bueno o muy bueno, pues no se sabe con relación a qué es bueno o no. Este último tipo expresa razones explícitas del criterio de evaluación del usuario que podría ser completado si fuera necesario dependiendo del tipo de objetos que se evalúan. Desde el punto de vista de la investigación, este tipo de evaluaciones sería prácticamente ideal. En la práctica, contrastada tras los experimentos realizados, los usuarios prefieren algo muy sencillo de evaluar. En principio los usuarios están buscando resultados y ese gasto cognitivo de tener que pensar cuál es la evaluación a realizar, perjudica la interacción con el usuario.

Es importante destacar que independientemente del conjunto de evaluaciones que se desee, cualquiera de ellas puede integrarse pues los algoritmos propuestos en IV.6 son capaces de tratarlas de forma similar.

Los sistemas analizados que utilizan evaluaciones positivas o aceptar/rechazar no suelen guardar información histórica sobre las evaluaciones de los documentos. Es decir, si un documento se evalúa como interesante y posteriormente se evalúa el mismo documento como no interesante se tienen en cuenta como dos evaluaciones diferentes. Esto facilita la implementación pero hace que el modelo no sea totalmente consistente. Esta filosofía es utilizada en sistemas como en [Tasso2002]. Esa simplicidad podría ser irrelevante si los usuarios no tienen la tendencia de modificar sus evaluaciones y si el número de documentos evaluados por el mismo usuario es muy alto, de forma que la imprecisión que introduce es poca. Desde el punto de vista de esta tesis, sí es interesante almacenar esta información y en el caso en que el usuario cambie la evaluación de alguno de los documentos analizados, se actualiza todo el modelo eliminando la evaluación negativa y añadiendo la positiva. Si no se hiciera esto, cuando hay pocos documentos analizados y el usuario no está muy convencido de su decisión, las propuestas serían inconsistentes.

IV.5.7 HISTORIAL

Otros de los elementos considerados importantes en esta tesis es la posibilidad de que los usuarios puedan explotar sus búsquedas anteriores y aprovechar los resultados obtenidos. La importancia del concepto de historial se expone por ejemplo en [Kobsa1996]. En esta tesis el concepto de historial aparece de una forma original debido

a la organización basada en objetivos. La idea es que el usuario pueda acceder a todos los documentos evaluados organizados en primer lugar por categorías(objetivos) y dentro de cada objetivo ordenados por la importancia de la evaluación que el usuario diera, es decir, los documentos evaluados como más interesantes se presentarán los primeros. Como se vio en el apartado IV.5 el concepto de historial está presente en el modelo del usuario y su explotación constituye otra categoría de actividad. Esta actividad puede utilizarse por distintas razones que se resumen a continuación:

a) Recuperar soluciones anteriores
b) Modificar las evaluaciones de soluciones anteriores
c) Buscar objetos ya presentados para buscar soluciones similares
d) Analizar soluciones pasadas para otros objetivos aplicándolas al actual

IV.6. ALGORITMO DE PERSONALIZACIÓN NBM

IV.6.1 EL ALGORITMO NBM

El objetivo de este algoritmo es predecir la evaluación del usuario para cada documento de acuerdo a su objetivo actual. El NBM (Naïve Bayes Metiore) [Bueno2000b; Bueno2001] se basa en el objetivo del usuario y no en sus consultas como en el trabajo de [Zukerman1999] donde se utilizan probabilidades y modelos de Markov para predecir las preferencias del usuario. En los sistemas que no integran ningún tipo de personalización, la lista de soluciones se presenta en el mismo orden, asociado normalmente a la similitud entre los documentos y las palabras de la consulta. El algoritmo NBM se combina con los algoritmos de reconocimiento de patrones para obtener documentos relacionados con la consulta y posteriormente se ordenan combinando primero la relevancia con el modelo del usuario de acuerdo a nuestro algoritmo y segundo, si hubiera documentos con relevancia similar para el modelo del usuario, estos últimos se ordenan siguiendo el criterio de similitud con la consulta. Ideas similares se proponen en [Schwab1999]. En su aproximación, los posibles tipos de evaluación se limitan a interesante/no interesante. Además el cálculo de relevancia para las soluciones sólo tiene en cuenta las evaluaciones positivas. En nuestra propuesta todos los tipos de evaluaciones presentados en IV.5.6 se tienen en cuenta para calcular la relevancia de un documento. A continuación se muestra el funcionamiento del algoritmo y de los elementos que se utilizan para predecir la posible evaluación del usuario para un documento dado en función del modelo del usuario. Entre los posibles tipos de evaluaciones de IV.5.6 se elige para la explicación del algoritmo la primera (*ok, known, bof, wrong*) por ser el caso intermedio de dificultad (este es uno de los esquemas utilizados en los experimentos que se muestran en el capítulo VI).

A continuación se justifica porqué es recomendable que los objetos con los que se trabaja sean parametrizables y no hay que quedarse con un único tipo de elementos para representar un documento. Para ilustrar el algoritmo se utilizarán sólo dos atributos de la base de datos del ejemplo: *palabras_clave* y *año*. Si el usuario evalúa una referencia como *ok* se incrementará el número de evaluaciones como *ok* para el año del documento y también para las palabras clave que formen parte de ese documento.

No debe utilizarse un sólo atributo para calcular el grado de relevancia de un objeto (la referencia bibliográfica en este caso) para un objetivo dado. Suponiendo que un usuario dice en su objetivo que está buscando referencias sobre Sistemas Tutoriales Inteligentes (ITS), aunque lo que él quiere en realidad son publicaciones recientes sobre ese tema,

muchas referencias que contengan la palabra clave ITS pueden ser rechazadas si el sistema solamente utiliza el atributo *palabra_clave* para calcular el grado de relevancia de ese objeto para el objetivo de este usuario. Estas evaluaciones negativas pueden llevar al sistema a la conclusión de que el usuario no está interesado en ITS. Por supuesto, éste no es el caso. El usuario esta interesado en ITS pero hay otros parámetros asociados que deben tenerse en cuenta para calcular la relevancia de este documento, como por ejemplo la fecha de publicación o los autores. Probablemente el problema venga por parte de usuario que no sepa formular el concepto de referencia reciente. De todas formas, el concepto de 'reciente' depende mucho del individuo y del dominio de estudio.

Este problema podría plantearse como un problema de clasificación, pues lo que se quiere es dado un documento ver entre las posibles evaluaciones del usuario cual será la más probable dependiendo de su modelo de usuario y dentro de los pertenecientes a una clase también se quiere saber como se ordenarían por importancia para el usuario.

Se han hecho muchos estudios para comparar los diferentes métodos de clasificación: Redes neuronales, ID3 y Bayesianos. La mayoría de ellos tienen una gran complejidad de cálculo que limita la posibilidad de usarlo en sistemas que necesitan una respuesta rápida. Sin embargo, el algoritmo *Naïve Bayes,* a pesar de su simplicidad, debido a la suposición de independencia entre los parámetros, es capaz de dar resultados similares y muchas veces mejores que los otros algoritmos, con un cálculo mucho más simple. Esas comparaciones se han hecho en [Keogh1999; Kononenko1990; Mitchell1997; Singh1996] y [Versteegen2000].

El algoritmo que se propone se inspira en la teoría probabilística de Bayes, usando las evaluaciones anteriores de los usuarios. El resultado del algoritmo será el grado de relevancia de un objeto para el objetivo actual de un usuario. En otras palabras, se calcula la probabilidad de que el usuario de una evaluación concreta para un documento en el contexto del objetivo actual. Con esta idea en mente se utiliza una adaptación del algoritmo Naïve Bayes [Kononenko1990] para ese objetivo concreto. La fórmula original se muestra en (1)

$$P(C/V_{1,J_1},...,V_{n,J_n}) = P(C)\prod_{i=1}^{n} Q_i(C,J_i) \qquad (1)$$

donde

$$Q_i(C,J_i) = \frac{P(V_{i,J_i} \mid C)}{P(V_{i,J_i})} \qquad (2)$$

Ecuación 20. Fórmula original del algoritmo Naïve Bayes

En la Ecuación 20.1 y la Ecuación 20.2 C es una de las posibles clases de evaluación (*ok, known, bof, wrong* para el ejemplo).$V_{i,Ji}$ es una variable booleana con valor 1 si la instancia actual tiene valor J_i y P(C) es la probabilidad de la clase C. La aplicación de la Ecuación 20.1 consistirá en dado un documento, que contiene una serie de valores que lo describen (J_i), obtener la probabilidad de que el usuario evalúe ese documento dentro de cada clase. La clase que obtenga la mayor probabilidad será la elegida.

La ecuación Ecuación 20.2 es sólo válida si los posibles atributos son independientes entre sí, como se asume en este trabajo al igual que la mayoría de los sistemas de búsqueda de información bayesianos. Esa suposición simplifica los cálculos dando resultados similares a otros algoritmos como se ha indicado antes.

La modificación de la Ecuación 20.1 que se propone da resultados similares (Ver justficación en 0), pero es menos restrictiva porque usa la media de los pesos de cada atributo, como se muestra en la Ecuación 21. Eso permite empezar a dar resultados con pocos datos.

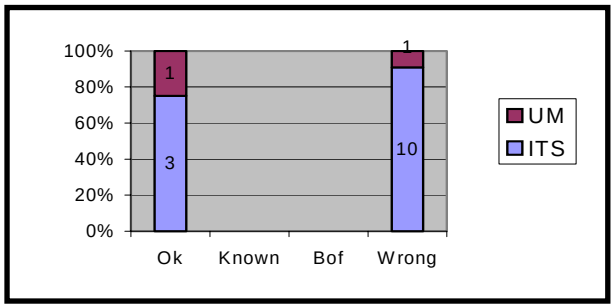
$$P'(C/V_{1,J_1},...,V_{n,J_n}) = P(C)\frac{\sum_{i=1}^{n} Q_i(C,J_i)}{n}$$

Ecuación 21. Adaptación para NBM de Naïve Bayes

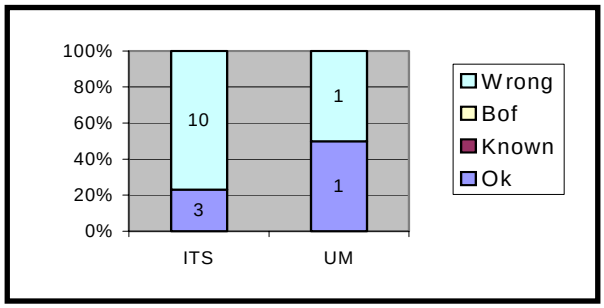
IV.6.2 EJEMPLO

Para aclarar la utilidad de esta ecuación se va a mostrar un ejemplo donde se decide como clasificar un documento. Esto quiere decir que analizando el modelo del usuario intentará predecir como éste evaluaría cada documento, y en consecuencia darle el tipo de recomendación que corresponda.

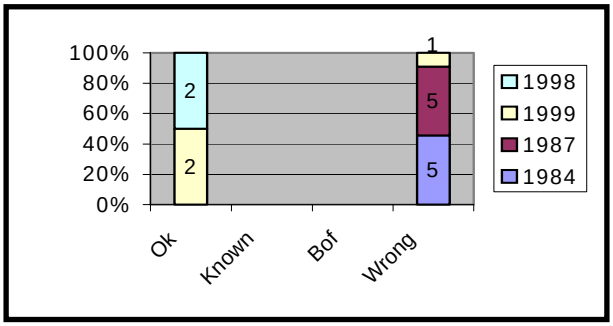
En la Fig. 4 se presenta una representación gráfica de 15 evaluaciones que componen la parte del modelo del usuario que se utiliza en este algoritmo. En el modelo del usuario simplificado del ejemplo se almacenan solamente los datos referentes al año y palabras clave. En la Fig. 4a se indica cuántas veces está asociada cada palabra clave con cada tipo de evaluación. Por ejemplo ITS fue evaluada 3 veces como correcta y 10 veces como incorrecta. La Fig. 4b indica la distribución de los tipos de evaluación sobre cada palabra clave. Por ejemplo, el tipo de evaluación *wrong* se asocia con ITS 10 veces. Las Fig. 4a y Fig. 4b representan la misma información pero vista de forma diferente.



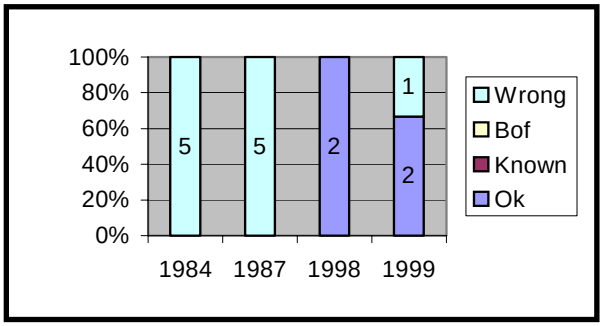
(a) Distribución de las palabras clave sobre los tipos de evaluación



(b) Distribuc. de los tipos de evaluación sobre las pal. clave



(c) Distribución de años sobre tipos de evaluación



(d) Distribución de tipos de evaluación sobre años

Fig. 4. Representación gráfica de las 15 evaluaciones

Fig. 4c representa la distribución de años sobre los tipos de evaluación mientras la Fig. 4d es otra gráfica que representa la misma información agrupando en este caso los tipos de evaluación sobre los años. Por ejemplo, los 2 documentos que el usuario evaluó de 1998 los evaluó como *ok.*

La información de las gráficas puede organizarse como se muestra en la Fig. 5. Donde para cada objetivo se tiene una lista de los parámetros que se utilizan para personalizar las respuestas. Cada parámetro tiene a su vez una lista de los valores que alguna vez han sido evaluados por el usuario, junto con las distintas evaluaciones.

```
objetivo1:
        parámetro1
          valor ev_wrong ev_bof ev_known ev_ok
          valor ...
        parámetron
          ...
objetivom:parámetro1
          valor ev_wrong ev_bof ev_known ev_ok
          ...
          parámetron
          ...
```

Fig. 5. Historial de evaluaciones en modo lista

En la Fig. 6 se muestran los valores que se almacenarían para guardar la información del ejemplo.

```
"Artículos Recientes de Tutoriales Inteligentes":
        palabra_clave
          ITS 10 0 0 3
          UM 1  0 0 1
        año
          1984 5 0 0 0
          1987 5 0 0 0
          1998 0 0 0 2
          1999 1 0 0 2
 "otro objetivo"…
```

Fig. 6. Valores del modelo para el ejemplo

Puesto en escena el modelo simplificado del usuario para el análisis del algoritmo, se supone que el usuario pide una recomendación al sistema y éste encuentra un documento que contiene la palabra clave **ITS** y que fue publicado en **1998**. ¿Qué le dirá el sistema al usuario?. El algoritmo debe calcular el grado de relevancia del documento para el usuario, es decir, el tipo de evaluación más probable que el usuario daría a este libro si lo evaluase.

A continuación se muestra información relacionada con el documento, extraída de las gráficas que representan el modelo del usuario:

- De 15 palabras evaluadas, 13 están relacionadas con ITS. Esto podría interpretarse como un concepto importante en la necesidad del usuario. Sin embargo, la misma palabra clave fue evaluada por el usuario 10 veces como errónea. Esto puede sugerir que hay otros atributos de los objetos que pueden contribuir a este 'ruido'.
- Las dos referencias publicadas en 1998 se evaluaron como *ok.*

El algoritmo usará la información de los dos atributos para calcular el nivel de relevancia asociado a cada posible evaluación y el mayor será el elegido. A continuación se calcularán los valores necesarios en la Ecuación 21.

Primero se va a calcular la probabilidad de que el usuario evalúe *ok* en un documento que contenga la palabra clave ITS y el año 1998.

$$P'(ok/ITS,1998) = p(ok)\frac{Q(ok,ITS)+Q(ok,1998)\cdot}{2} =$$

$$= p(ok)\frac{\frac{p(ITS/ok)}{P(ITS)}+\frac{P(1998/ok)}{P(1998)}}{2} = \frac{4}{15}\frac{\frac{3/4}{13/15}+\frac{2/4}{2/15}}{2} = 0.625$$

Por otro lado se calculará la probabilidad de que el usuario evalúe como erróneo un documento con esas mismas características:

$$P'(error/ITS,1998) = p(error)\frac{Q(error,ITS)+Q(error,1998)\cdot}{2} =$$

$$= p(error)\frac{\frac{p(ITS/error)}{P(ITS)}+\frac{P(1998/error)}{P(1998)}}{2} = \frac{11}{15}\frac{\frac{10/11}{13/15}+\frac{0/2}{2/15}}{2} = 0.385$$

Si se calculase *P'(known/ITS,1998)* y *P'(bof/ITS,1998*) se obtendría 0 ya que ningún documento ha sido evaluado para esas categorías en este ejemplo. Por lo tanto, si llamamos *prox(X)* a *P'(X/ITS,1998)* se obtiene finalmente el nivel de relevancia de ese documento para cada tipo de evaluación:

prox(ok)=0,615	*prox(error)=0,385*	*prox(known)=prox(bof)=0*

La evaluación más probable será aquella en la que el nivel de relevancia sea mayor. En este caso, el sistema recomendaría el documento al usuario con la evaluación *ok* y se lo mostraría delante de otros con *prox(ok)* menor. Cuantos más documentos evalúe el usuario, los niveles de relevancia serán actualizados y la respuesta será más fiable. Cuando se va a realizar una recomendación al usuario, los documentos serán sistemáticamente clasificados en el tipo de clasificación más probable. Además, se ordenan en forma decreciente de acuerdo a su nivel de relevancia dentro de su clase. En la aplicación real se tiene en cuenta tantos parámetros como sean necesarios y no sólo el *año* y las *palabras clave*.

IV.6.3 PROPIEDADES

En este apartado se exponen dos propiedades del algoritmo NBM. Por un lado, se demostrará que cumple la Ecuación 22 y por otro lado, se mostrará que NBM es mejor en ciertos aspectos que NB.

$$\sum_i P'(C_i / V_{1,J_1},...,V_{n,J_n}) = 1$$

Ecuación 22. Propiedad de NBM

Demostración

Puesto que la ecuación propuesta por el autor es diferente a la conocida como Naïve Bayes, a continuación se demuestra que Ecuación 22 es cierta para NBM. La demostración utiliza dos ecuaciones bien conocidas de la teoría de la probabilidad:

$$P(a/b) = \frac{P(b/a)P(a)}{P(b)} \quad (Teorema\ de\ bayes) \qquad (1)$$

$$\sum_{i=1}^{m} P(C_i \mid b) = 1 \quad \text{si } C_i \text{ son todas las clases posibles donde clasificar a } b \qquad (2)$$

Ecuación 23. Ecuaciones básicas de probabilidad

Desarrollando la Ecuación 21 se obtiene:

$$P'(C/V_{1,J_1},...,V_{n,J_n}) = P(C)\frac{\sum_{i=1}^{n} Qi(C,J_i)}{n} =$$

Según la Ecuación 20.2:

$$= P(C)\frac{\sum_{i=1}^{n} \frac{P(V_{i,J_i} \mid C)}{P(V_{i,J_i})}}{n} = \frac{\sum_{i=1}^{n} P(C)\frac{P(V_{i,J_i} \mid C)}{P(V_{i,J_i})}}{n} =$$

Según el Teorema de Bayes:

$$= \frac{\sum_{i=1}^{n} P(C \mid V_{i,J_i})}{n}$$

Como $\sum_{j=1}^{m} P(C_j \mid V_{i,J_i}) = 1$ es cierto según Ecuación 23.2 entonces:

$$\sum_j P'(C_j/V_{1,J_1},...,V_{n,J_n}) = \sum_{j=1}^{m} \frac{\sum_{i=1}^{n} P(C_j \mid V_{i,J_i})}{n} = \frac{\sum_{i=1}^{n}\sum_{j=1}^{m} P(C_j \mid V_{i,J_i})}{n} = \frac{\sum_{i=1}^{n} 1}{n} = \frac{n}{n} = 1 \quad \text{(c.q.d.)}$$

Ventajas del algoritmo sobre Naïve Bayes

En este apartado se intentará justificar porqué se ha elegido la modificación (Ecuación 21) del algoritmo Naïve Bayes (NB) en lugar del original (Ecuación 20).

El algoritmo Naïve Bayes clásico adolece de un problema importante cuando el número de muestras tomadas es bajo. En el caso que nos ocupa, se puede ver que cuando alguno de los atributos que aparecen en un documento no había aparecido antes, la probabilidad de que este pertenezca a alguna clase es cero. Como NB realiza un producto de probabilidades, el valor conjunto de *n* atributos para una clase será cero con que alguno de sus componentes sea cero. Por ejemplo, si en un documento aparecen los términos: *adaptive hypermedia, user modeling, tutorial systems, y* para una clase de evaluación (*ok),* las probabilidades son *adaptive hypermedia (0,95), user modeling(0,7), tutorial systems(0)*, donde el último tiene 0 simplemente porque no apareció anteriormente. El resultado que daría NB al evaluar este documento es 0. Es decir, no lo recomendaría a un usuario que está muy interesado en casi todos los parámetros del documento, salvo en uno sólo porque no apareció anteriormente. El resultado que daría NBM sería (0,55) que parece más lógico que cero. Para evitar ese problema hay otras modificaciones como la propuesta por Cestnik [Cestnik1990] (en adelante NBCestnik) que es una de las

más aceptadas. Lo que hace es compensar un poco esa anulación de NB cuando alguno de los términos es 0 dándole a éstos una probabilidad inicial. En la Ecuación 24.1 se muestra el factor original en NB con la equivalencia resultado de aplicar el teorema de Bayes. En la Ecuación 24.2 se muestra el valor de la probabilidad condicionada como el cociente entre casos favorables (número de elementos J_i correspondientes a la clase C) y casos posibles, como el número total de instancias del valor J_i para el atributo i.

$$Q_i(C, J_i) = \frac{P(V_{i,J_i} \mid C)}{P(V_{i,J_i})} = \frac{P(C \mid V_{i,J_i})}{P(C)} \quad \textbf{(1)}$$

$$P(C \mid V_{i,J_i}) = \frac{N(C, V_{i,J_i})}{N(V_{i,J_i})} \quad \textbf{(2)}$$

$$P(C \mid V_{i,J_i}) = \frac{N(C, V_{i,J_i}) + 2P(C)}{N(V_{i,J_i}) + 2} \quad \textbf{(3)}$$

$$P(C) = \frac{N(C) + 1}{N + 2} \quad \textbf{(4)}$$

Ecuación 24. Modificación de Cestnik al factor de Naïve Bayes

En la Ecuación 24.3 se muestra la modificación de Cestnik para la probabilidad condicionada. El objetivo es que si $N(C,V_{iJi})=0$ se compensará con la probabilidad de P(C). Esta probabilidad la calcula usando la fórmula de la Ecuación 24.4. Tom Mitchell [Mitchell1997] muestra una generalización de la Ecuación 24.3, en la que el valor '2' se modifica con un valor genérico *m* al que llama *tamaño equivalente de la muestra* que determina la importancia del peso de P(C) con relación a los datos observados. Dicha modificación se muestra en la Ecuación 25:

$$P(C \mid V_{i,J_i}) = \frac{N(C, V_{i,J_i}) + mP(C)}{N(V_{i,J_i}) + m}$$

Ecuación 25. Generalización de Mitchell de la modificación de Cestnik

Mitchell calcula P(C) asumiendo una distribución uniforme, es decir, si hay *k* posibles clases cada una de ellas tendrá una probabilidad de P(C)=1/k. En el capítulo de experimentación (VI) se comparan los tres tipos de algoritmos para una muestra de 1000 entradas. Para cada entrada se suponen 3 atributos (J_i) con posibles valores elegidos de forma aleatoria. Se utilizan los tres algoritmos NB, NBM y NBCestnik/Mitchell para ordenar los 1000 elementos de acuerdo a un usuario. Para analizar la similitud entre los distintos algoritmos, se ha calculado el factor de correlación entre los tres métodos. La correlación entre cada par de algoritmos indica el parecido entre ellos a la hora de ordenar documentos. Para la prueba se han calculado los $Q(ok,J_i)$ para cada uno de los tres atributos y a partir de ahí se han calculado las $P(ok|V_{1J1,}V_{2J2,}V_{3J3})$. En la Tabla 8 puede verse los resultados que son bastante interesantes. El algoritmo aquí propuesto NBM obtiene resultados muy similares a los de los otros dos algoritmos, con una similitud del 93% en los resultados con NB,

estando cerca del 94% en la comparación con NBCestnik, el cual se supone una mejora de NB.

NB-NBCestnik	NB-NBM	NBM-NBCestnik
0,922	0,930	0,937

Tabla 8. Comparación entre las tres variantes de Naïve Bayes

IV.6.4 EL ALGORITMO WNBM

Los resultados anteriores muestran que los resultados de NBM son del orden de los generados por los clásicos Naïve Bayes, siendo la necesidad de cálculo algo menor, pero, ¿aporta algo más?. Suponiendo que para recomendar un objeto a un usuario se quieren utilizar diferentes parámetros, como por ejemplo para artículos; autor, año y palabras clave, para un artículo como el de la Fig. 7, al utilizar alguno de los dos algoritmos clásicos se tendría la probabilidad de que al usuario le interesase el documento como el producto de las probabilidades de 10 palabras clave por la de 1 año y por la de 2 autores como puede verse de forma simplificada en Ecuación 26. El problema es que al haber muchas más palabras claves que fechas, la aportación de la fecha a la evaluación final del objeto es muy baja. Por lo tanto, sería necesario poder realizar de una forma sencilla algún tipo de ponderación para dar más importancia a los factores que pertenecen a parámetros con pocos atributos como el año.

Título: METIORE: A Personalized Information Retrieval System
Autores: Bueno, D; David, A
Publicado en: 8th International Conference on User Modeling.UM'2001
Páginas: 168-177
Año: 2001
Palabras Clave: information retrieval, user modeling, METIORE, Personalization,…

Fig. 7. Ejemplo de artículo

$$P(C/V_{1,J_1},...,V_{n,J_n}) = P(C)\prod_{i=1}^{n} Qi(C,J_i) =$$
$$P(C)\big(Q(C,bueno)Q(C,david)Q(C,2001)Q(C,usermodel)Q(C,infretrieval)...\big)$$

Ecuación 26. Aplicación de Naïve Bayes a objetos con múltiples parámetros

El algoritmo NBM permite de forma muy sencilla la adaptación en la que cada parámetro del documento pueda ponderarse de acuerdo a su importancia. A esta extensión se le llamará WNBM (*Weighted Naive Bayes Metiore*) La ecuación general se muestra a continuación:

$$P(C/V_{1,J_1},...,V_{n,J_n}) = P(C)\sum_{p=1}^{m}\omega_p \frac{\sum_{i=1}^{n} Q_{pi}(C,J_{pi})}{N_p}$$

Ecuación 27. WNBM:Adaptación de NBM para múltiples parámetros

En la Ecuación 27 el primer sumatorio con índice *p* se suman los valores parciales de las *m* parámetros a analizar (en el ejemplo anterior m=3:año, palabras clave y autor). N_p es

el número total de elementos asociados al parámetro p (en el ejemplo si el parámetro es autor N_p=2 y si es año N_p=1). El factor ω_p indica la importancia que se da a cada parámetro, por ejemplo 0,25 para año, 0,25 para autor y 0,5 para palabras clave. Si se quiere dar un valor equitativo para todos los parámetros ω_p=1/m. Para que el resultado sea el de una media ponderada, la suma de todos los ω_p debe ser igual a 1.

Para ver más clara la aplicación de esta última ecuación, se mostrará su desarrollo para el ejemplo de la Fig. 7. En este caso se dispone de tres parámetros (autor, año y palabras clave). Para esta base de datos, el diseñador ha decidido dar la misma importancia al parámetro autor y a las palabras clave, reduciendo un poco la importancia del año. Para ello asigna los factores de importancia ω_1=2/5 (autor) ω_2=1/5 (año) ω_3=2/5 (palabras clave). En la Ecuación 28 se muestra la aplicación de la ecuación anterior para este ejemplo:

$$P(C/V_{1,J_1},\ldots,V_{n,J_n}) = P(C)\left(\begin{array}{l} \tfrac{2}{5}\dfrac{Q(C,bueno)Q(C,david)}{2} + \\ \tfrac{1}{5}\dfrac{Q(C,2001)}{1} + \\ \tfrac{2}{5}\dfrac{Q(C,usermodel)\ldots Q(C,infretrieval)}{10} \end{array}\right)$$

Ecuación 28. Ejemplo de aplicación de WNBM para múltiples parámetros

De esta forma se evita el problema de los clasificadores NB y NBCestnik de tratar a todos los atributos de todos los parámetros por igual.

IV.7. CONCLUSIONES

En este capítulo se ha descrito el modelo teórico de la propuesta de esta tesis mostrando las principales aportaciones. Éstas son el resultado de analizar, en los capítulos anteriores, las deficiencias de los sistemas clásicos IR y de algunos sistemas de IR con modelo del usuario.

Se ha propuesto desde un punto de vista teórico una herramienta de búsqueda y análisis de datos con muchas opciones que permiten al usuario realizar consultas que van desde una simple lista de palabras a un análisis multicampo con restricciones.

Se ha realizado un análisis aplicable a múltiples sistemas de recuperación de la información que intenten personalizar las respuestas, analizando las preguntas: qué representar, cómo hacerlo y la forma de utilizar esa información o modelo.

Se han propuesto y discutido ventajas e inconvenientes de diversas maneras de evaluar, que difieren de las clásicas me gusta/no me gusta. Todas ellas integradas en el mismo modelo de funcionamiento.

También en relación con las evaluaciones se ha tenido en cuenta algo que en otras propuestas no se considera, quizás por su baja probabilidad, y es que un usuario, a lo largo del tiempo pueda evaluar el mismo documento de forma diferente. Con la propuesta de esta tesis, se deshacen las inferencias realizadas con las anteriores evaluaciones para utilizar sólo la última para cada documento.

Respecto al algoritmo de personalización, se ha mostrado una modificación de Naïve Bayes para recomendación de objetos. El algoritmo NBM permite recomendar utilizando modelos de usuario organizados por objetivos. Además se ha propuesto una mejora WNBM que permite ponderar la importancia y aportación de cada parámetro al resultado final, realizando unos cálculos relativamente simples.

Centrar la necesidad de búsqueda alrededor del concepto de objetivo es otro elemento innovador, no muy utilizado en otros sistemas. Se ha introducido para permitir modelos de usuarios adaptados a sus necesidades de información para cada situación.

La utilización del historial inteligente ofrece un valor añadido para la introspección del usuario en sus decisiones anteriores.

El siguiente capítulo presenta METIORE, una posible implementación de los algoritmos y elementos aquí propuestos.

V. METIORE: APLICACIÓN DEL MODELO TEÓRICO

V.1. INTRODUCCIÓN

En este capítulo se continuará con el análisis de la propuesta desde un punto de vista más práctico. En el capítulo anterior se introducían los conceptos teóricos y ecuaciones necesarias para el diseño de un sistema de recuperación de la información que personalizase la respuesta para cada usuario. A continuación se verá la arquitectura de una posible implementación de esa propuesta: METIORE. También se verá qué condiciones debe cumplir una base de datos para poder utilizarla con METIORE, cual debe ser el formato de los datos y cómo se cargan, es decir, cómo es posible que METIORE sea capaz de ofrecer características de personalización a diferentes tipos de bases de datos multimedia o documentales. En el caso en que un usuario no sea capaz de conseguir los resultados deseados con el sistema, se propone una arquitectura para que pueda cooperar con otros usuarios en Internet para obtener los resultados que desee. También se mostrará cómo se organizan los ficheros para conseguir resultados eficientes. La estructura final del sistema es el resultado de los refinamientos sucesivos del prototipo durante cinco años. A lo largo del capítulo, se ilustrará el funcionamiento con pantallas del sistema en alguna de sus dos versiones: Aplicación o Web.

V.2. DESCRIPCIÓN GENERAL DEL SISTEMA

En este apartado se describirán los detalles del prototipo en el que se ha implementado la propuesta de esta tesis. El sistema METIORE (***M****ultimedia coop****E****rative Informa****TIO****n* ***R****etrieval Syst****E****m*) es un entorno que permite la implementación de distintos sistemas de recuperación de la información personalizados. Entre sus características fundamentales se pueden destacar las siguientes:

- Multi-IRS
- Entrada de datos con XML
- Base de datos Orientada a Objetos
- Análisis de datos

- Gráficos de resultados
- Personalización
- Cooperación
- Interfaz Multilingüe
- Implementación Multiplataforma (Windows, Solaris, Linux, etc.)

V.2.1 MULTI-IRS

Debido a la forma de utilización de los datos de los distintos sistemas de información y a la posibilidad de la gestión de éstos de una forma parametrizada, independiente del contenido, es fácil aplicar METIORE a prácticamente cualquier base de datos que tenga una estructura jerárquica. Algunos ejemplos de posibles bases de datos aplicables en este sistema se muestran a continuación:

- Publicaciones (autor, titulo, palabras clave, etc.)
- Fichas descriptivas: árboles (altura, altitud, nombre científico, etc.)
- Música (grupo, titulo, año, compañía, etc.)
- Ofertas de trabajo (edad, estudios, experiencia)

Los contenidos multimedia como imágenes, videos o sonidos son también utilizables si se utilizan campos de descripción. Hasta el momento se ha utilizado para 4 bases de datos reales, cuya descripción se muestra en la Tabla 9.

Nombre Base de Datos	*Descripción*
STREEMS	Es una base de datos multimedia con información sobre árboles (altura, épocas de floración, tipos de hojas, etc.) con imágenes asociadas a cada uno de los componentes. Este sistema se aplico en el proyecto europeo Leonardo (RITA[6]).
REVUES	Esta implementación contiene datos sobre la revista francesa "relation publiques" que se publica desde 1950 y que no disponía de ningún índice informático. Al introducir este sistema en METIORE se pudieron hacer múltiples tipos de consultas no disponibles hasta el momento. Esta implementación estaba subvencionada por un proyecto regional de la Lorena (Francia).
LORIA	Esta es la base de datos que contiene información de las publicaciones desarrolladas en el laboratorio LORIA de Nancy. Es con ella con la que se han realizado las pruebas sobre las características de personalización.
AH2002	Contiene la mayoría de los artículos publicados sobre temas relacionados con Hipermedia Adaptativos y Modelado del Usuario

Tabla 9. Bases de datos utilizadas en METIORE

V.2.2 USO DE XML COMO FORMATO DE ENTRADA DE DATOS

Para utilizar METIORE, como se comentaba en el apartado anterior, es necesario que la base de datos pueda ser organizada de forma jerárquica. Si la base de datos esta utilizándose en otro sistema que no permita personalización, que puede ser, por ejemplo, una base de datos relacional, puede expresarse utilizando XML y podrá adaptarse para utilizarla con METIORE. El sistema es capaz de interpretar bases de datos en XML y convertirlas en un formato adecuado para poder personalizar la respuesta de los usuarios. En la Fig. 8 se muestran dos ejemplos representativos de

6 RITA: rinaturalizzazione e riforestazione attraverso metodi di formazione a distanza supportati da nuove tecnologie informatiche

bases de datos con características estructurales diferentes que han sido adaptadas para ser utilizadas con METIORE.


```
<revue>
  <titre>RELATIONS PUBLIQUES INFORMATIONS</titre>
  <numero> 663 </numero>
  <date>16 décembre 1970</date>
  <agendadesRP>
    <typemanifestation>
          assemblée générale
    </typemanifestation>
    <lieu>Paris</lieu>
    <organisme>
        Syndicat national des attachés de presse
    </organisme>
  </agendadesRP>
  <agendadesRP>
    <typemanifestation>Remise de prix</typemanifestation>
    <lieu>Paris</lieu>
    <personneconcernee>Marcel Bleustein-Blanchet
    </personneconcernee>
    <organisme>Publicis</organisme>
  </agendadesRP>
  <ondit>
    <typemanifestation>Journées d'information</typemanifestation>
    <lieu>Paris</lieu>
    <secteuractivite>Travaux publics</secteuractivite>
    <organisme>
        Fédération nationale des travaux publics
    </organisme>
  </ondit>
</revue>
```

a)

```
<doc>
  <reference>Rec2380 </reference>
  <title>
    METIORE: A Publications Reference for the Adaptive
Hypermedia Community
  </title>
  <booktitle>
    AH2002 Second International Conference on Adaptive
Hypermedia and Adaptive Web-Based Systems
  </booktitle>
  <author>
    <aut>  Bueno, David </aut>
    <aut>  Conejo, Ricardo </aut>
    <aut>  Carmona, Cristina </aut>
    <aut> David, Amos A </aut>
  </author>
  <editor> Paul De Bra and Peter Brusilovsky  </editor>
  <publisher> Springer Verlag, </publisher>
  <year> 2002 </year>
  <keywords>
    <keyword> METIORE </keyword>
    <keyword> publications </keyword>
    <keyword> objectives </keyword>
    <keyword> retrieval system  </keyword>
    <keyword> search engines  </keyword>
    <keyword> Adaptive Hypermedia </keyword>
    <keyword> personalization </keyword>
    <keyword> information needs </keyword>
    <keyword> evaluations </keyword>
  </keywords>
  <url>
http://sirius.lcc.uma.es/WCTP/doc/paper_236_RH6179RH617
9.pdf
  </url>
  <abstract>
    The Web is one of the most powerful sources of
information on any topic. However looking for scientific
literature is a difficult task. In this paper we propose our system
METIORE as a source of information for the Adaptive
Hypermedia  community. The idea is to put together all the
publications on this research area and provide an adaptive tool
to find papers or people working in the field. METIORE is a
Personalized Information Retrieval system that keeps a user
model based on objectives.
  </abstract>
</doc>
```

b)

Fig. 8. Ejemplos de dos bases de datos utilizadas en METIORE. a) REVUE y b) AH2002

Dependiendo de la estructura jerárquica de los datos se pueden tener sistemas con una jerarquía 'sencilla' como AH2002/LORIA (Ver Fig. 8.b) o 'compleja' como en STREEMS/REVUE (Ver Fig. 8.a). La diferencia está en la cantidad de objetos del mismo tipo que pueden pertenecer a otro objeto. Si esta cantidad es menor o igual que 1, el sistema será sencillo y si es mayor que uno será complejo.

En la Fig. 8.b) se puede ver un ejemplo de la base de AH2002. El objeto principal es `<doc>`. Dentro de `<doc>` la mayoría de las etiquetas son de la forma `<et>texto</et>`, como por ejemplo `<editor> Paul De Bra and Peter Brusilovsky </editor>`. Estas etiquetas son atributos del objeto que la contiene, en este caso el objeto `<doc>`. Se llamará a este tipo de etiquetas **simples**. En la base de AH2002 todas las etiquetas son de esa forma salvo dos: `<author>` y `<keywords>`. La diferencia con las demás es que se puede tener más de un elemento de este tipo dentro del documento, por ejemplo, varios autores o palabras clave que representen a un artículo. Aunque puede haber varios elementos del mismo tipo, solamente tiene un atributo (El nombre del autor o la palabra clave correspondiente). Se llamarán a estas etiquetas **Múltiples Simples.** En el apartado siguiente se verá como afecta esto a la representación interna de los datos. Para resumir, en AH2002 se tiene un único objeto `<doc>` con atributos simples y dos atributos múltiples en los que cada uno de sus componentes es un atributo también

simple. En la Fig. 9 se muestra la gramática asociada (DTD[7]) a los documentos XML para esta base de datos. Esta es la estructura que se utilizará para hacer las búsquedas de los documentos. Lo que no limita que de los autores sólo deba conocerse el nombre. Para los autores se utiliza otro fichero XML con sus datos personales que son irrelevantes para realizar las búsquedas de publicaciones (como e-mail, dirección,...).


```
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT doc (reference, title, booktitle, author, editor, publisher, year, keywords, url, abstract)>
<!ELEMENT author (aut*)>
<!ELEMENT keywords (keyword*)>
<!ELEMENT aut (#PCDATA)>
<!ELEMENT keyword (#PCDATA)>
<!ELEMENT editor (#PCDATA)>
<!ELEMENT booktitle (#PCDATA)>
…(El resto de los elementos que aparecen en doc se definen con #PCDATA)
```

Fig. 9. DTD para la base de datos AH2002

En el otro sistema Fig. 8.a) la jerarquía es algo más complicada porque aparecen varias etiquetas múltiples y compuestas dentro de la etiqueta principal, en este caso `<revue>`. Ésta contiene algunos atributos que pueden aparecer más de una vez: `<agendadesRP>` `<ondit>` `<article>` `<editorial>` `<vientde>`. Se llamaran etiquetas **Múltiples Compuestas**. Cada uno de estos atributos está compuesto de más atributos simples como puede verse en el ejemplo anterior. La DTD para esta base de datos se muestra en la Fig. 10.

```
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT revue (titre, numero, annee, mois, jour, html, agendadesRP*, ondit*, noteque*, editorial, article*, vientde*)>
<!ELEMENT agendadesRP (typemanifestation, lieu, personneconcernee?, organisme?, html?, personneconcernee?, organisme?)>
<!ELEMENT article (titre, auteurs, typearticle?, referencesdocumentsource?, motscles, typearticle?, referencesdocumentsource?,
auteurresumearticle?, html?)>
<!ELEMENT editorial (titre, auteurs, motscles?, html?)>
<!ELEMENT noteque (typeevenement, secteuractivite, personneconcernee?, organisme?, organismeconcernee?, html?,
personneconcernee?, organisme?, organismeconcernee?)>
<!ELEMENT ondit (typemanifestation, lieu?, secteuractivite, organisme?, organismeconcernee?, personneconcernee?, html?,
secteuractivite, organisme?, organismeconcernee?)>
<!ELEMENT vientde (typedocument, titre, auteurs, anneeedition, edition, html)>
<!ELEMENT annee (#PCDATA)>
<!ELEMENT anneeedition (#PCDATA)>
<!ELEMENT auteurresumearticle (#PCDATA)>
…(El resto de los elementos que aparecen en los elementos anteriores se definen con #PCDATA)
```

Fig. 10. DTD para la base de datos REVUE

En el apartado siguiente se muestra como se va a tratar cada uno de esos tipos de etiquetas XML para generar la base de datos interna de METIORE.

V.2.3 BASE DE DATOS ORIENTADA A OBJETOS

Para poder trabajar con los datos, se podría utilizar una base de datos relacional, pero en los sistemas de recuperación de la información esta aproximación no es utilizada, y se trabaja principalmente con listas invertidas y tablas hash como puede verse en el capitulo **II** (**Recuperación de la Información**). Además, para el tipo de consultas que se necesitan para la gestión de clusters es necesario realizar anidamientos de consulta de gran complejidad que no suelen dar el resultado deseado. Otro problema del uso de bases de datos relacionales es que hay que definir tablas, campos clave y relaciones

[7] DTD- Definición de tipo de documento (Document Type Definition). Es una gramática que se utiliza para validar documentos XML

entre tablas que son específicas de cada base de datos, limitando en gran medida las posibilidades de aplicación a distintos tipos de datos.

La decisión de cómo gestionar los datos para este sistema se orientó más a la utilizada en los sistemas de recuperación de la información. De forma que se trabaja con listas invertidas y clusters (Ver V.6) y para el tratamiento en memoria se utilizan objetos. En la Fig. 11 se muestra la DTD para definir los distintos tipos de etiquetas comentadas en ambos sistemas: simples, múltiples simples y múltiples compuestas

```
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT doc (simple, multiplesimple, multiplecompuesta-ceroouna?, multiplecompuesta-masdeunavez*)>
<!ELEMENT multiplecompuesta-ceroouna (simple1, simple2, simple3, simplen)>
<!ELEMENT multiplecompuesta-masdeunavez (simple1, simple2, simple3, simplen)>
<!ELEMENT multiplesimple (simple+)>
<!ELEMENT simple (#PCDATA)>
<!ELEMENT simple1 (#PCDATA)>
<!ELEMENT simple2 (#PCDATA)>
<!ELEMENT simple3 (#PCDATA)>
<!ELEMENT simplen (#PCDATA)>
```

Fig. 11. DTD para definir tipos de etiquetas posibles

Tipo de Etiqueta	*Ejemplo en Fig. 8*	*Implementación*
Principal	revue, doc	Objeto
Simple	numero, lieu, date, title, booktitle	Atributo del objeto en que se encuentre
Múltiple Simple	autor, keywords	Atributo del objeto, pero es una lista dentro de él con cada uno de los diferentes valores.
Múltiple Compuesto (Aparece 0 ó 1 vez)	No hay ejemplo disponible	Los atributos de esa etiqueta serán atributos de la etiqueta que la contiene
Múltiple Compuesto (Aparece más de 1 vez)	agendadesRP, ondit	Un objeto relacionado con el objeto que lo contiene

Tabla 10. Implementación para cada tipo de etiqueta

En la Tabla 10 se muestra cómo se relacionan los distintos tipos de etiquetas con la implementación orientada a objetos. Para cada tipo se muestra un ejemplo en las bases de datos de la Fig. 8. Hay un caso especial que ocurre cuando una etiqueta tiene varias sub-etiquetas, pero ésta sólo puede aparecer una vez en el documento. En ese caso, las sub-etiquetas se tratan como si la etiqueta a la que pertenecen no estuviese, es decir, como si estuvieran al mismo nivel de la etiqueta que las contiene.
En la Fig. 12 se muestra una estructura general de un diagrama de objetos para los posibles tipos de etiquetas. En general todas las etiquetas se convierten en atributos del objeto principal, salvo si son Múltiple-Compuestas que se convierten en objetos relacionados con el principal.

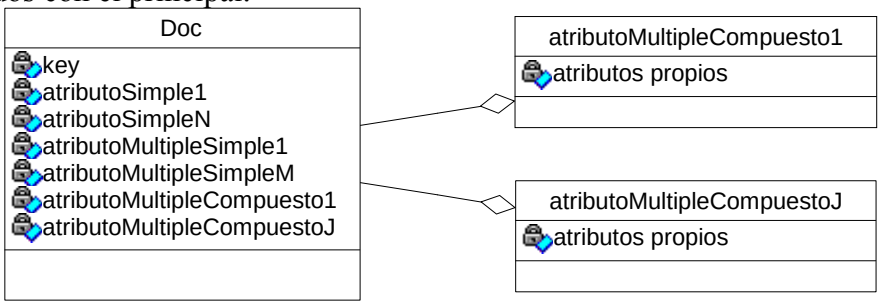


Fig. 12. Diagrama general para creación de objetos asociados a etiquetas XML

Cuando los datos estén en memoria se tendrá un objeto para cada documento y tantos objetos como atributos múltiples compuestos existan.

V.2.4 ANÁLISIS DE DATOS

METIORE dispone de una capacidad de búsqueda que va más allá de la simple introducción de palabras clave. Las opciones de análisis de datos se explicaron en el capítulo IV. La potencia que aporta este tipo de análisis permite que usuarios sin conocimiento previo de la base de datos que se utilice en METIORE puedan descubrir su contenido sin necesidad de tener que hacer consultas explicitas. En el **Apéndice II** se muestra una aplicación real de los análisis que se pueden hacer de una base de datos utilizando METIORE. En la Fig. 13 se muestra la interfaz que permite realizar este tipo de análisis.

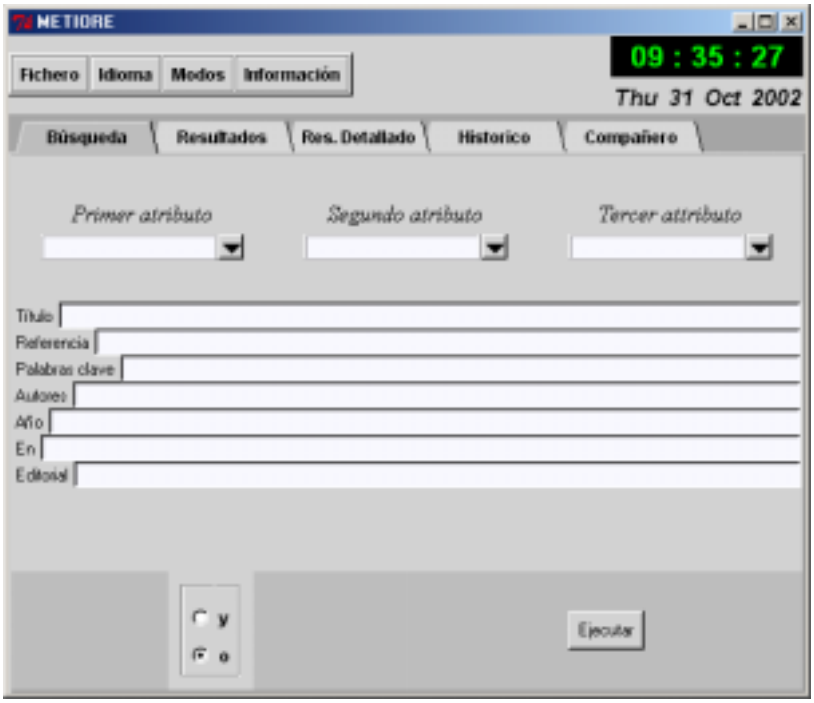


Fig. 13. Interfaz para el análisis de datos (Versión Aplicación)

V.2.5 GRÁFICOS DE RESULTADOS

Otras de las posibilidades de METIORE es la de generar gráficos asociados a las búsquedas. Esto es especialmente útil cuando se quieren hacer análisis visuales sobre el estado de la base de datos. Los gráficos se presentan en función del número de atributos que se utilicen en la consulta. Si se utiliza uno se obtendrá un gráfico en 2D y si se utilizan dos se obtendrá un gráfico en 3D. Cada vez que el usuario realiza una consulta, se genera un fichero[8] con la información gráfica asociada a ella. A continuación se muestra un ejemplo de cada tipo. La Fig. 14 es el gráfico resultado de la búsqueda que utiliza un atributo (autor) en la base de datos AH2002. El resultado de esa consulta es la lista de autores ordenadas por número de publicaciones en la base de datos. En el gráfico se muestra una columna para cada autor y el número de publicaciones de éste. Todos las gráficos que utilicen un solo atributo son de este estilo, se muestran los elementos que responden a la consulta, y la frecuencia de aparición.

[8] El formato del fichero es interpretable por programas de hojas de calculo como Microsoft Excel

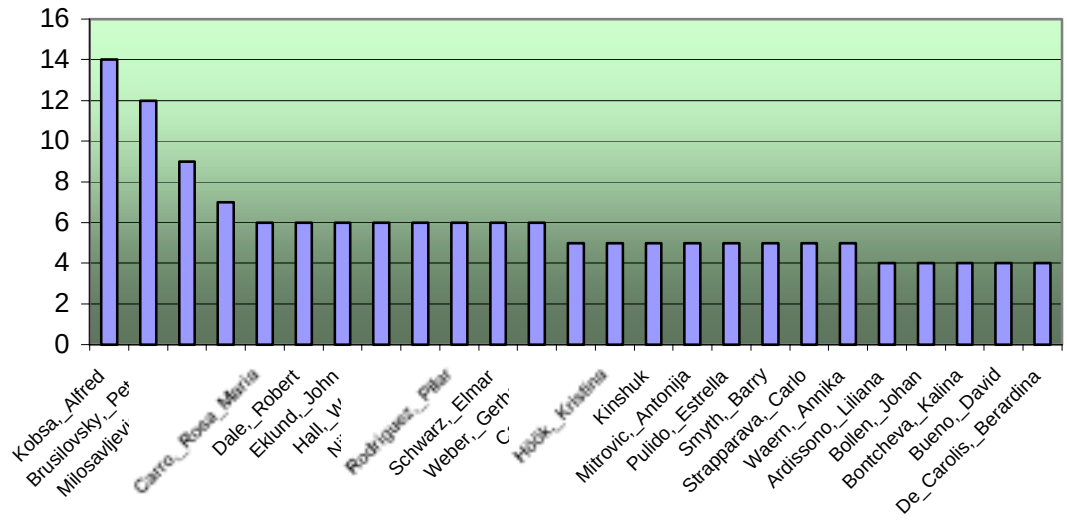
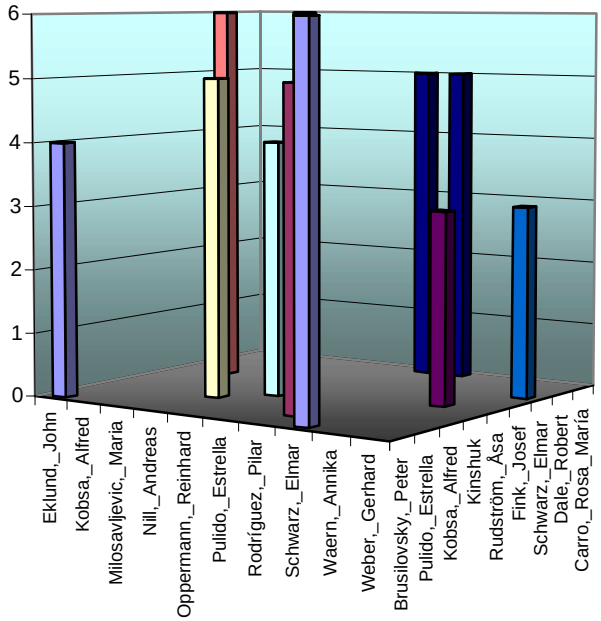


Fig. 14. Gráfica generada que muestra los autores que más han publicado en la BD AH2002

El otro tipo de gráficos es el asociado a dos atributos. Por ejemplo, si se quiere buscar a los autores que trabajan juntos y ver el número de artículos en los que han colaborado, la búsqueda que hay que realizar sería: *atributo1=autor y atributo2=autor*. En la Fig. 15 se muestra el gráfico asociado a esa consulta. En la base se tienen los autores en filas y columnas y la altura indica el número de artículos realizados juntos. Algunos de los datos que aparecen son: (Brusilovsky, Eklund :4 artículos; Carro, Pulido: 5 artículos, …).



	Eklund John	Kobsa, Alfred	Milosavljevic, Mª	Nill Andre	Oppermann ,Reinhard	Pulido, Estrella	Rodríguez, Pilar	Schwar Elmar	Waern, Annika	Weber, Gerhar
Brusilovsky,_Peter	4							6		
Pulido, Estrella							5			
Kobsa, Alfred				5						
Kinshuk					4					
Rudström, Åsa									3	
Fink, Josef		6								
Schwarz, Elmar										3
Dale, Robert			6							
Carro, Rosa_Mª						5	5			

Fig. 15. Gráfica y tabla de datos que muestra los autores que más han publicado juntos en la BD AH2002

V.2.6 PERSONALIZACIÓN

El elemento que hace especial a METIORE es la personalización. La idea principal es que el usuario sienta que el sistema entiende sus necesidades y es capaz de resolver la necesidad de información que motivó su interacción con METIORE. En el apartado **IV.5** (**Modelo de usuario**) se documentó cómo se organizan los datos y los algoritmos que se utilizan para ayudar al usuario. En este apartado se resumirán las características de personalización de METIORE. El elemento fundamental que se intenta corregir es el problema clásico de los buscadores que devuelven muchos resultados como respuesta a una consulta. Los usuarios suelen analizar sólo los primeros resultados, por lo que es muy importante que éstos sean los más relevantes. Para resolver ese problema, cuando se realiza una consulta, METIORE utiliza el modelo de ese usuario para reordenar las soluciones de acuerdo a dos criterios. El primero es la relevancia de la solución para su modelo, y el segundo la relevancia de la solución para la consulta realizada. El orden es éste ya que la consulta realizada es sólo una pieza de una serie de actividades de búsqueda orientadas todas a una meta común que es el objetivo del usuario, alrededor del cual se crea el modelo.

Una segunda forma de personalización consiste en la búsqueda automática del sistema de elementos que puedan ser interesantes sin que se realice ninguna consulta concreta. En este caso toda la acción del usuario se reduce a pulsar el botón *recomiéndame.* Dicha acción hará que METIORE empiece a buscar todos los documentos que puedan ser relevantes, pero la comparación no se hace con ninguna consulta, sino con el propio modelo del usuario. El resultado será una lista de documentos ordenados por similitud con el modelo asociado al objetivo en curso.

El sistema tiene aún otra forma de ayudar al usuario y es mediante el historial que muestra todos los documentos que ha evaluado. Dichos documentos están organizados por objetivos y dentro de cada objetivo, ordenados por la relevancia de la evaluación. Los documentos ya evaluados pueden volver a consultarse o modificar su evaluación. Esta posibilidad es interesante pues permite reflexionar sobre algunas evaluaciones y modificarlas, mejorando la precisión de su modelo.

V.2.7 COOPERACIÓN

METIORE tiene disponible un modo de cooperación. El objetivo de este modo es el de permitir a varios usuarios cooperar a través de la red para conseguir un objetivo común.

Uno puede ser un experto que ayude a los demás a comprender como utilizar el sistema. Cuando los usuarios se conectan, se activa una ventana de charla que les permite expresar textualmente que desean hacer juntos. Además, todas las acciones que realice uno de ellos se podrá visualizar por el otro en su interfaz, facilitando así el aprendizaje. Las características y protocolos de la cooperación se muestran en V.4. En la Fig. 16 puede verse la interfaz de usuario que le permite visualizar los usuarios conectados y seleccionar un compañero con el que iniciar una cooperación. También se visualiza el estado de todos los otros usuarios y si están conectados, con quién.

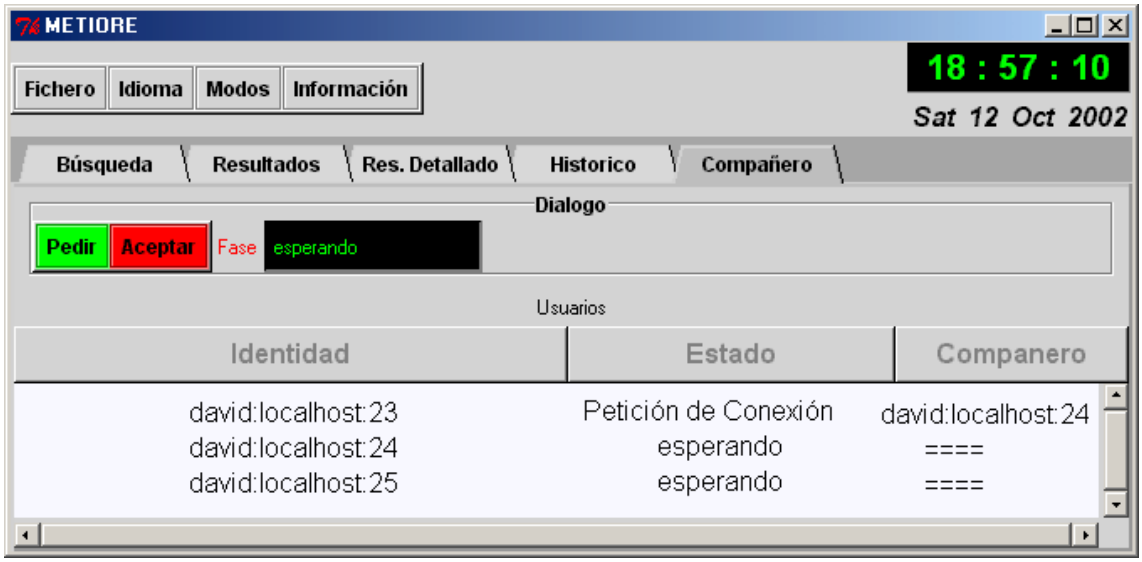


Fig. 16. Pantalla de la zona de cooperación de METIORE

En la Fig. 17 se muestra la ventana de charla que se inicia cuando los usuarios empiezan a cooperar. En esta ventana aparecen los mensajes con una fecha exacta (*timestamp)* que permite saber en que orden se escribieron los mensajes.

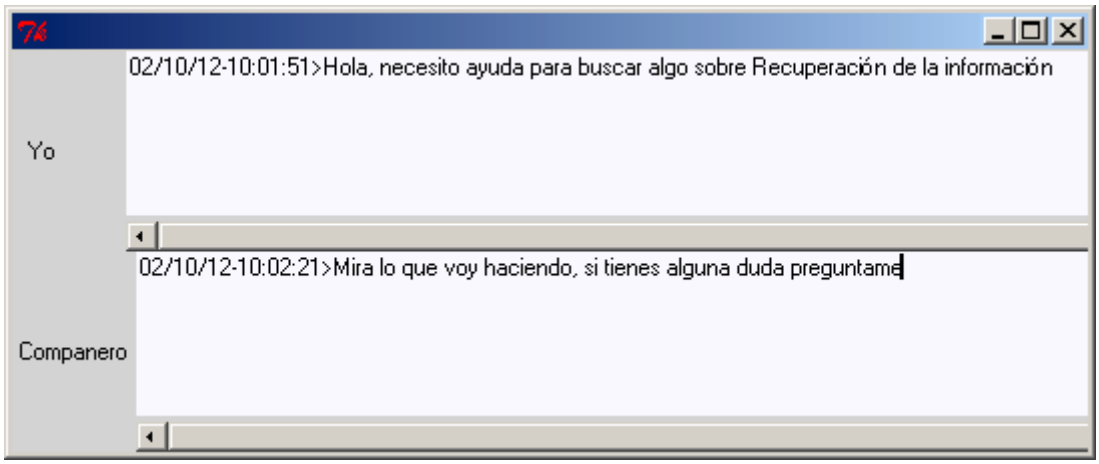


Fig. 17. Ventana de charla que se abre cuando dos usuarios inician una cooperación

V.2.8 OTRAS CARACTERÍSTICAS

Otras características de METIORE que se pueden destacar son:

- Interfaz Multilingüe.- El idioma de la interfaz de METIORE es configurable en tiempo de ejecución, con lo que el usuario podrá interactuar con el sistema usando el

idioma de su preferencia. Actualmente los idiomas son Español, Francés, Ingles e Italiano, aunque no existe limitación en el número de idiomas

- Multiplataforma.- La aplicación de METIORE puede ejecutarse en diferentes máquinas y sistemas operativos (Windows, Solaris, Linux, etc.). Esto es posible gracias al lenguaje de programación utilizado: Incr-Tcl.
- Multi-interfaz.- METIORE puede ejecutarse tanto en modo aplicación, como a través de la Web. En los dos casos el núcleo es el mismo y desde la Web se accede a éste mediante *sockets* utilizando Java (JSP).

V.3. ARQUITECTURA DEL SISTEMA

En este apartado se mostrará la arquitectura general del sistema (ver Fig. 18) y se detallarán los diferentes módulos que forman METIORE. El usuario puede trabajar con METIORE utilizando dos interfaces diferentes, una en modo aplicación (fichero ejecutable) y otra en modo Web. Cualquiera de estas dos interfaces se conecta con el módulo principal de METIORE que realizará todo el proceso de cálculo de la aplicación y gestionará las bases de datos y los modelos de usuario. Otro elemento importante es el módulo encargado de procesar las bases de datos en su formato inicial para adaptarlas a los formatos utilizados en METIORE.

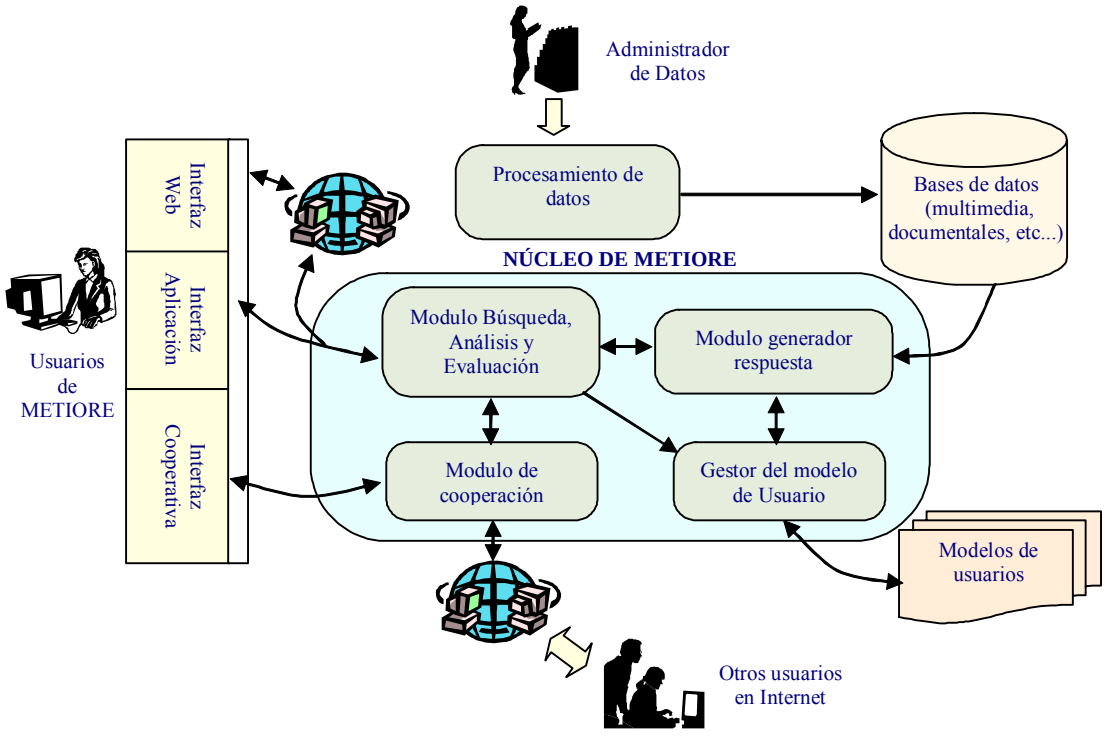


Fig. 18. Arquitectura general de METIORE

I.1.1 INTERFACES DE USUARIO

Un usuario puede acceder a METIORE como un programa instalado en modo aplicación o a través de la Web. En el primer caso, debido a las características multiplataforma, se pueden utilizar diferentes formas de instalación. Por ejemplo en una red Unix el programa puede estar instalado una sola vez para todos los usuarios y cada

usuario accederá utilizando su cuenta a una instalación común. También es posible hacer una instalación autónoma en un PC con Windows, con lo que el usuario tendrá todos los datos en su ordenador. En cualquiera de los casos, se podrá activar una cooperación, que permitirá comunicar a los usuarios a través de Internet para trabajar conjuntamente, sin importar el sistema operativo sobre el que se encuentren.

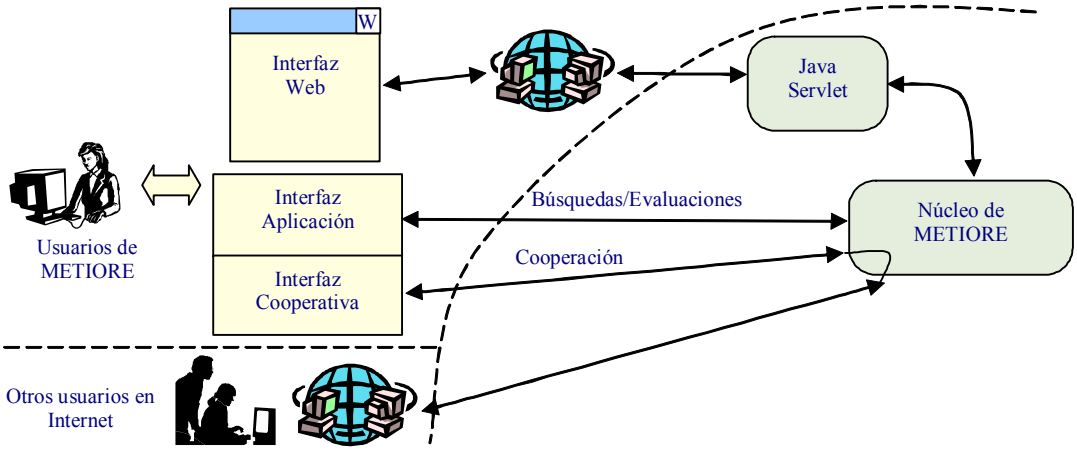


Fig. 19. Interfaces de usuario

Una segunda forma de utilizar METIORE es utilizando un navegador WWW. El navegador se conectará a un servidor Web (aplicación con servlets de Java) que procesará los datos de la interfaz para enviar mediante una conexión con *sockets* los datos al núcleo de METIORE (programado con Incr-Tcl). Dicho núcleo actúa como un servidor que espera peticiones de diferentes usuarios y devuelve resultados personalizados a cada usuario utilizando su modelo. En la Fig. 19 se muestran todas las opciones de interfaz.

V.3.1 CONVERSIÓN DE BASES DE DATOS

En la Fig. 20 se muestran los pasos necesarios para convertir una base de datos a un formato útil para METIORE. Una buena organización permite que puedan aplicarse diferentes bases de datos sin mucha dificultad. En primer lugar, hay que transformar los datos originales a formato XML que será el que interpretará METIORE como formato de entrada.

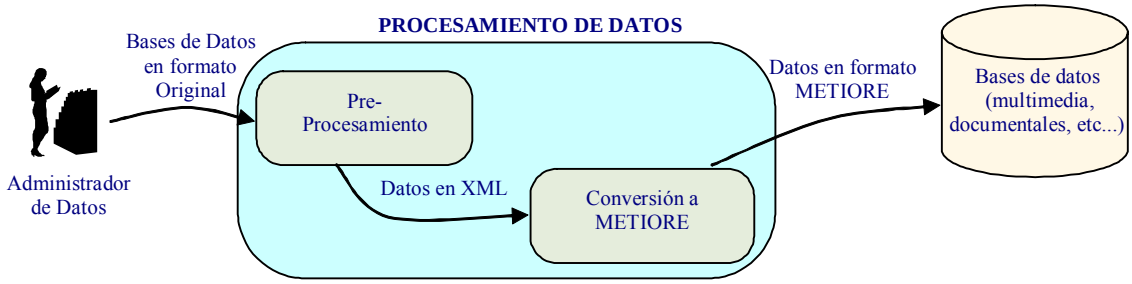


Fig. 20. Procesamiento de datos

Las etiquetas que se utilicen podrán tener los nombres que quieran y adecuarse al tipo de datos que represente. Posteriormente, hay que crear un fichero de balizas/etiquetas en el que se indica la tarea a realizar cuando en un fichero de entrada XML se encuentre cada etiqueta. Básicamente, para cada etiqueta se indica el tipo al que pertenece según

los tipos de etiqueta vistas en la Tabla 10. El algoritmo de procesamiento se muestra a continuación de manera simplificada:

```
PARA CADA etiqueta HACER
  CASO  etiqueta SEA
    simple, múltiple simple o múltiple_compuesto_0_1:
                                    La etiqueta se añade como atributo al último objeto de la lista
    principal  o múltiple compuesta:
                                    Se crea un objeto
                                    Se añade a una lista de objetos creados
                                    SI etiqueta es múltiple_compuesta entonces
                                            Añadir referencia a este objeto en el último objeto de la lista
                                    FINSI
  FINCASO
FINPARA
```


De esta forma se crean todos los objetos asociados a la base de datos en memoria. Este proceso sólo se realiza cuando se crea la base de datos en METIORE, o cuando se incluyen nuevas entradas al sistema. En este segundo caso, si se reciben nuevos documentos, sólo éstos se cargan en memoria, de forma incremental.

En este punto se han creado en memoria una serie de objetos que representan a la base de datos. Durante el tiempo de desarrollo del sistema se vio que cuando la base de datos alcanzaba un tamaño considerable, no era posible tener todos los datos en memoria, además de que había algunas operaciones de búsqueda muy costosas que podían estar preprocesadas. Por lo tanto es necesario generar un conjunto de ficheros que ayudarán a trabajar de una forma rápida y eficiente para cualquier tipo de búsqueda. Estos ficheros son los de *listas invertidas, clusters, objetos e índices de objetos*. Su formato y características se explicarán más adelante en V.6.

V.3.2 NÚCLEO DEL SISTEMA

A continuación en la Fig. 21 se muestran los bloques principales de METIORE comunes a cualquiera de las posibles interfaces. El *módulo de búsqueda/análisis* que recibe las consultas del usuario ya sean para búsquedas simples, complejas con análisis de datos o de historial.

Los datos se normalizan y se envían al *módulo generador de respuesta*. Dicho módulo utilizando las consultas, buscará en la base de datos los posibles resultados, pero teniendo en cuenta que la consulta no se realiza de forma aislada, sino dentro de un contexto de búsqueda determinado por el objetivo actual del usuario. Por lo tanto, los resultados se mostrarán teniendo también en cuenta los datos del modelo ofrecidos por el *gestor del modelo de usuario.* Los datos del modelo ayudarán a mostrar las respuestas en el orden y forma que sean más convenientes. El gestor del modelo del usuario recibe también las evaluaciones sobre los documentos analizados y las utilizará para actualizar los modelos.

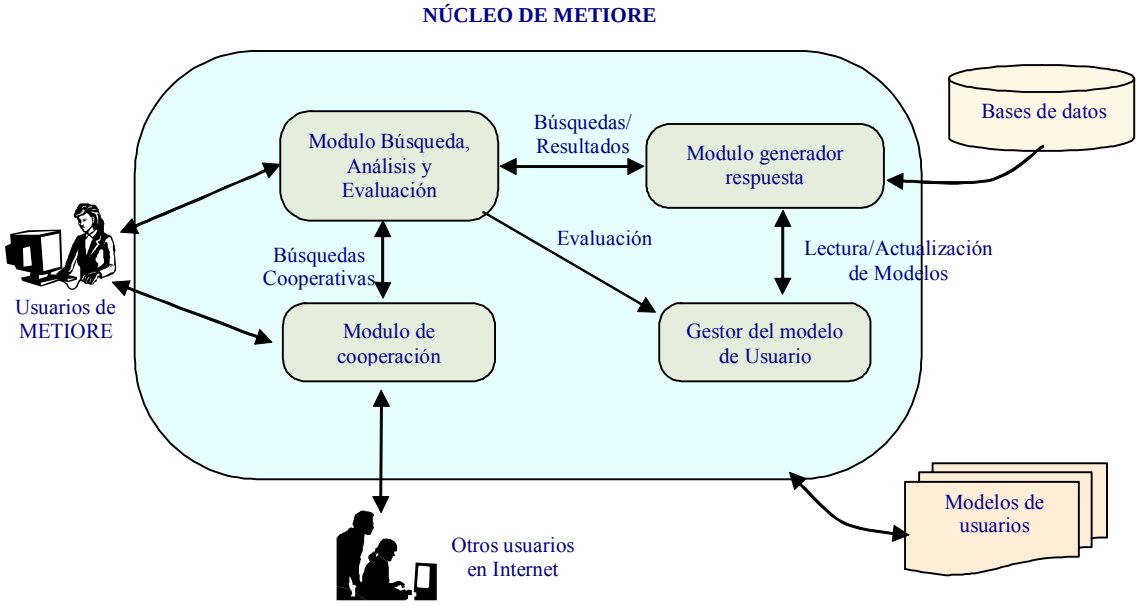


Fig. 21. Núcleo de METIORE

V.4. BÚSQUEDA COOPERATIVA DE INFORMACIÓN

La eficiencia en la recuperación de la información no consiste en obtener toda la información existente [Saracevic1997], sino sólo la que es relevante. Por lo tanto el modelo del usuario en IR debe conocer el concepto de relevancia y el proceso de interacción para reducir la información que puede ser relevante. La herramienta más poderosa en este campo incluye una interacción intermedia, es decir, una interacción que implica a un usuario humano, a un intermediario humano y a un sistema de IR. El intermediario humano es un profesional de la información hábil en la modelización del usuario, las búsquedas para obtener información de varios sistemas IR y grandes bases de datos. Tiene un rol muy importante que consiste en asistir en el diagnostico del problema del usuario y reformular sus cuestiones al sistema, sugerir búsquedas y traducir la consulta del usuario en una o más preguntas que puedan ser aceptables para la base de datos en cuestión. También debe dirigir y modificar la búsqueda y asistir en la evaluación de los resultados. Se puede decir que un intermediario es un agente inteligente que construye, implementa y modifica modelos de usuario.

Según experimentos realizados [Saracevic1997], los usuarios intercambian mucha información con el proceso intermedio. Aunque sólo un 8% está en el contexto de las preguntas y las respuestas. En cambio hay una gran cantidad de intercambios en la conversación que sólo se utilizan para mantener la conversación o para confirmar afirmaciones del otro elemento de la comunicación. Ésta es una táctica que acelera la comunicación e incrementa la comprensión mutua, cosa que parece bastante importante en el modelo del usuario.

A pesar de la potencia que pueda tener un sistema de búsqueda de información, nunca debería abandonarse la posibilidad de obtener ayuda de un experto humano. En METIORE no se ha querido dejar cerrada esta puerta y se ha desarrollado un sistema de recuperación de la información cooperativa (CIRS. *Cooperative Information Retrieval System*) [David1999]. Por lo tanto, se han hecho esfuerzos en tres sentidos: por un lado el sistema que ayuda al usuario, mediante la potencia de búsqueda y la personalización

de respuestas, en segundo lugar se ofrecen herramientas al usuario como el análisis de datos y el uso del historial. Y si el usuario no es capaz de encontrar una solución por si mismo, puede comunicarse con un experto humano y trabajar de forma cooperativa para encontrar la solución.

De forma general, la idea de trabajo en grupo conocida como *groupware* tiene como objetivo facilitar las interacciones entre usuarios con un objetivo común. El campo de estudio de trabajo colaborativo es CSCW (*Computer-Supported Cooperative Work*) y los entornos CSCW [Dewan1993] [Dourish1996] suelen ofrecer herramientas como pizarra compartida o compartición de aplicaciones. Por otro lado, existen protocolos para establecer sesiones entre usuarios y que puedan intercambiar información. Uno de los más utilizados es el protocolo utilizado para charlas en la red (*Internet Relay Chat Protocol*) descrito en el RFC1459 [Oikarinen1993].

Para esta tesis, se ha implementado una arquitectura propia que permite compartir la aplicación y un protocolo para las conexiones, más sencillo que el RFC1459, pero suficiente para las necesidades de este trabajo. A continuación se presentan las características funcionales de esta arquitectura para trabajo cooperativo.

V.4.1 MODOS DE FUNCIONAMIENTO DE UN CIRS

En este CIRS, dos usuarios pueden cooperar para encontrar la mejor forma de consultar al sistema para obtener la información buscada. Los dos usuarios pueden operar de forma remota[9]. Se ofrecen tres modos de funcionamiento:

1. *Autónomo*.- Es el modo de ejecución aislado, en el que el usuario trabaja en su máquina independientemente y no puede ser visto ni contactado por ningún otro usuario que esté utilizando el programa.
2. *Cooperación*.- Si el usuario elige este modo, se registra en un servidor de localización que contiene la lista de todos los usuarios conectados. Cualquiera de ellos puede intentar colaborar con otro para obtener una solución a sus necesidades de información. Esta colaboración se refleja en nuestra propuesta de dos formas. La primera es una ventana de charla, que permite a los dos usuarios intercambiar mensajes textuales en tiempo real para comentar como va a colaborar. En segundo lugar, la herramienta permite que todas las acciones de un usuario realiza en su interfaz se reflejen en la interfaz del otro usuario con el que coopera.
3. *Observación*.- Es similar a la cooperación, pero en este caso sólo uno de los dos puede ver lo que el otro usuario está haciendo, sin la posibilidad de modificar su entorno como en la cooperación.

Una aplicación que se registra en un entorno CIRS puede encontrarse en cinco estados diferentes (ver Fig. 22): espera, requerido, conexión, observación y cooperación. Cuando una aplicación se lanza en un modo distinto del autónomo, automáticamente se registra en el servidor de localización (SL) y entra en un estado de *espera*. La lista de usuarios en el SL es actualizada y enviada a todos los usuarios activos. Cuando un usuario desea conectarse con otro, se le envía una petición de conexión y ambos entran en el estado *requerido.* El segundo usuario puede aceptar o rechazar la conexión, pasando a los estados *conexión* o *espera* respectivamente. Una vez conectado, los usuarios deciden el tipo de comunicación que desean, entrando en modo de *observación*

[9] En las demostraciones realizadas, un usuario estaba en la Universidad de Málaga y el otro en el laboratorio LORIA (Nancy-Francia)

o *cooperación*. Cualquiera de los dos podrá decidir cambiar de modo (cooperación, observación) o terminar conexión.

En cualquiera de los dos modos de conexión debe respetarse el siguiente protocolo:

1. Seleccionar el usuario con el que se quiere interactuar
2. Solicitar petición de conexión, para alguno de los dos modos
3. Esperar la respuesta del otro usuario con la aceptación/rechazo de la conexión
4. Para terminar la cooperación hay que informar al compañero mediante una petición de desconexión y esperar su confirmación.

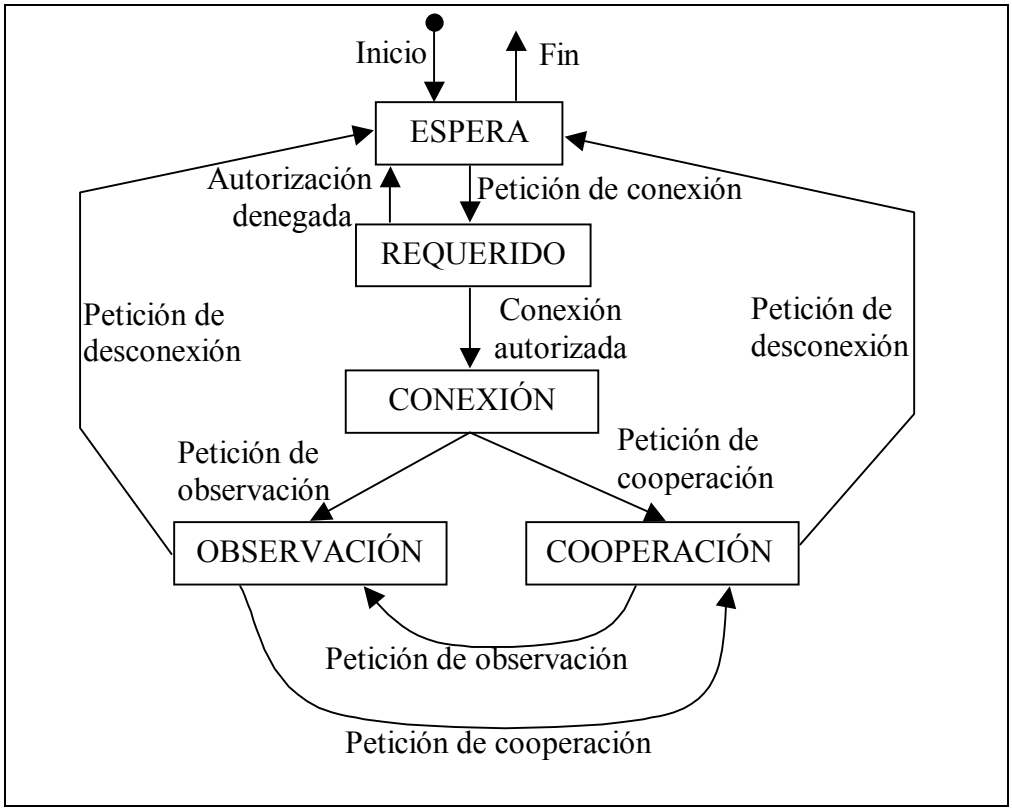


Fig. 22. Diagrama de transición de estados de una conexión CIRS

V.4.2 ARQUITECTURA GENERAL DEL CIRS

La arquitectura de un CIRS se compone de aplicaciones y de un servidor de localización. El servidor de localización centraliza la información que se necesita para la comunicación entre aplicaciones. Como se muestra en la Fig. 23 una aplicación tiene su *interfaz de aplicación* dependiente del dominio. Es la interfaz del usuario específica del programa que utiliza. Por otro lado dispone de un *servidor de aplicación(SA)* que es independiente del dominio, y cuyo objetivo es registrar al usuario en el *servidor de localización*. En la aproximación que se propone, sería independiente que el programa final fuera METIORE o un juego de ajedrez en red. La forma como el usuario selecciona a sus compañeros, y los modos de conexión se realizan a través de la *interfaz de conexión*, también independiente de la aplicación.

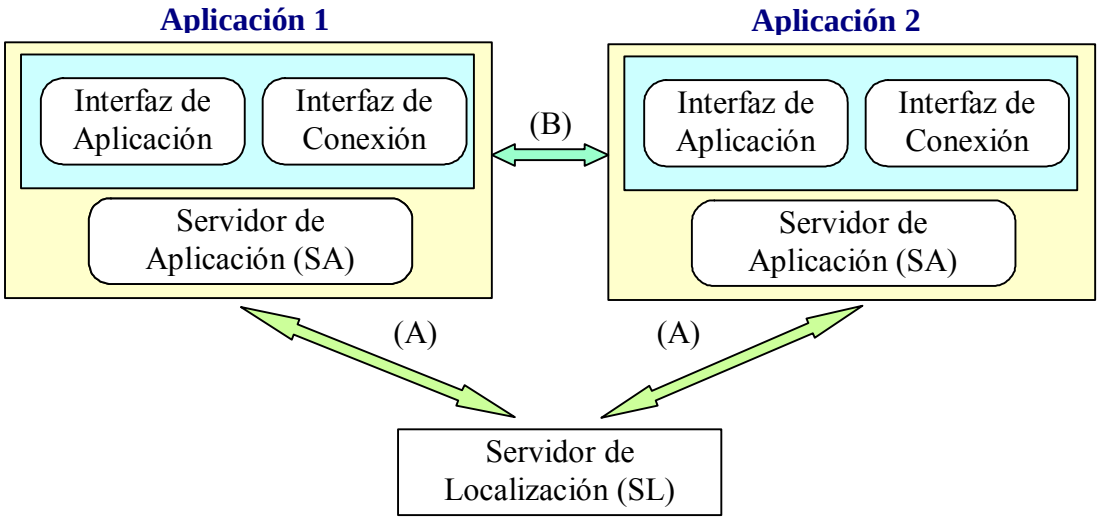


Fig. 23. Arquitectura del CIRS

La interfaz de conexión permite al usuario controlar todas las peticiones de colaboración con otros usuarios, y visualizar el estado de todos ellos, si están conectados con otros, o en estado de espera. Cuando la aplicación se lanza, inicialmente el usuario empieza a trabajar en modo autónomo. Se pueden encontrar distintos tipos de mensajes dependiendo de la fase de la comunicación. En la Fig. 24 se muestran algunos ejemplos de mensajes que representan las comunicaciones más importantes. A continuación se explican con más detalle:

1. **Registro.-** El usuario podrá decidir conectarse al servidor en la interfaz de conexión. Cuando el usuario se conecta, el SA envía un mensaje (*registrar)* al SL para indicarle que ese usuario está disponible. El SL envía al nuevo usuario la lista con el estado de todos los usuarios (*NuevaAplicación(lista))*, permitiendo al usuario que recien llegado saber quien está conectado. A los otros les envía un mensaje para indicar que hay un nuevo usuario (*NuevaAplicación(1))*.
2. **Petición de conexión.-** El usuario dispone en este momento de una lista actualizada de todos los usuarios conectados. Puede en este momento proceder a la petición de conexión con alguno de ellos. Para ello, envía un mensaje al usuario con el que quiere conectarse (*PeticiónConexión)*. Además, informa al SL (*ActualizarEstado(PetConexión)*) de su nuevo estado (*Requerido* en la Fig. 22) es decir, esperando confirmación de conexión. El servidor de localización informa a todos que ha cambiado el estado del usuario 1 (*ActualizarEstado(1,PetConexión))*
3. **Aceptar conexión.-** El usuario al que se pide la conexión, puede aceptar o rechazar la conexión. En el caso del ejemplo, si el usuario 2 acepta la conexión con el usuario 1. Le enviará un mensaje de aceptación (*Aceptar Conexión*) e informará al SL *(ActualizarEstado(2,AceptarConexión))* quien de nuevo difundirá la información a todos los usuarios.

Cuando los dos usuarios se han puesto de acuerdo para la conexión, todos los mensajes para intercambiar información se realizará exclusivamente entre sus aplicaciones sin pasar por el SL. En este momento también se inicia una aplicación de charla, con la que los usuarios pueden entablar conversaciones sobre su cooperación. Al SL sólo se le comunicará algún cambio de estado. Arquitecturas similares a nuestra propuesta que fue publicada en [David1999], se utilizan actualmente para los juegos en red.

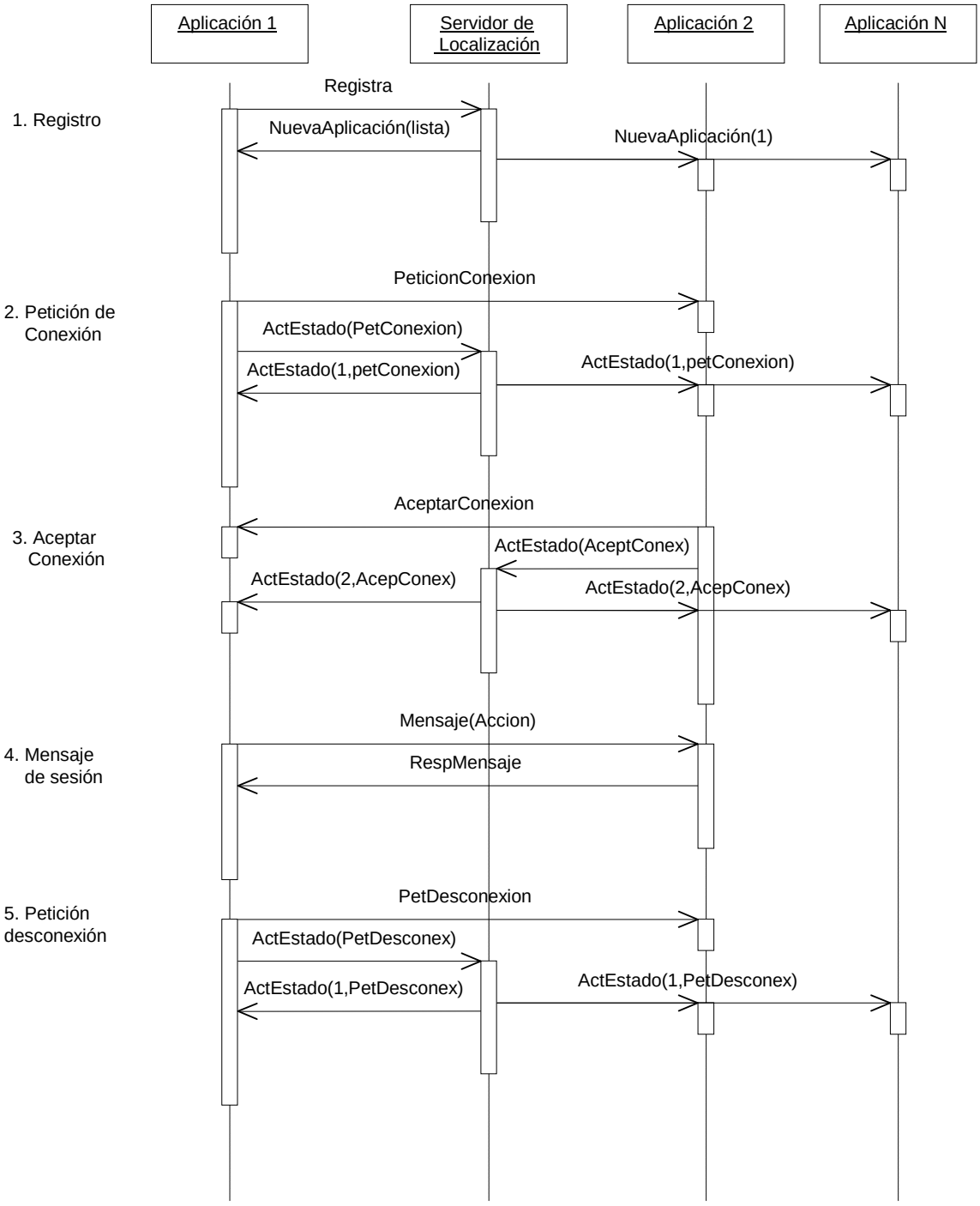


Fig. 24. Diagrama de intercambio de mensajes para la cooperación

4. **Mensajes durante la sesión.-** Una vez conectados, todos los mensajes tanto de la aplicación de charla, como de acciones sobre la aplicación se envían entre los dos usuarios. Ambos procesos actúan como clientes/servidores, permitiendo que cualquiera de los dos inicie un par mensaje-respuesta.
5. **Petición de desconexión.-** Cuando los usuarios han terminado su colaboración, uno de ellos hace la petición de desconexión, al compañero y también se comunica al servidor de localización. La organización de los mensajes es muy similar a los de petición de conexión, que también tienen por parte del otro usuario un conjunto de mensajes de *aceptación de desconexión*, que devuelve a los dos procesos a su estado inicial (*Espera*).

V.5. FUNCIONAMIENTO

En este apartado se pretende aclarar el funcionamiento del sistema mostrando las distintas partes de una sesión con METIORE. Para el ejemplo, se utilizarán pantallas indistintamente de la versión aplicación y de la versión Web, ambas comparten el núcleo de METIORE y sólo se diferencian en características propias de la presentación. Se utilizará la base de datos AH2002 para el ejemplo.

En primer lugar cuando un usuario ejecuta la aplicación debe registrarse. Para ello debe introducir un nombre de usuario y contraseña, que puede crearse en el momento si es un usuario nuevo. Además deberá elegir la base de datos con la que quiere trabajar.

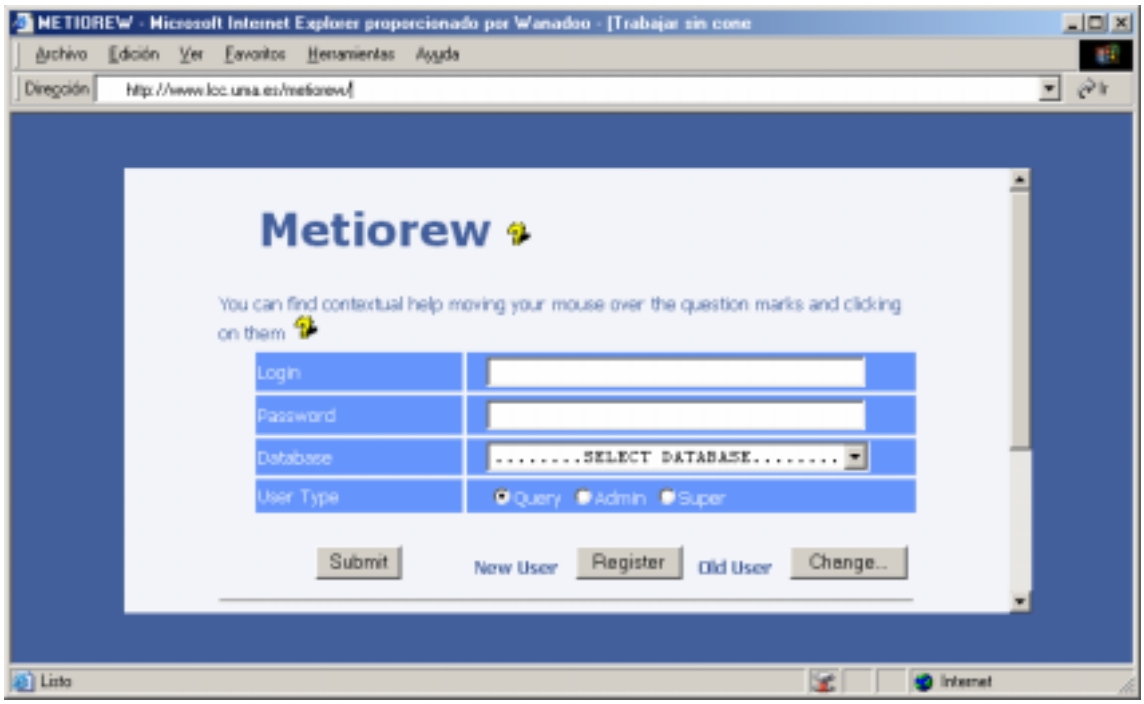


Fig. 25. Registro del usuario en METIORE

Una vez seleccionado el usuario y la base de datos, el siguiente paso es elegir el objetivo. El usuario podrá elegir un objetivo antiguo de una lista con todos sus objetivos anteriores o un objetivo nuevo. En este segundo caso, deberá escribir una cadena que le sirva para posteriormente identificar en futuras sesiones que esta buscando. El sistema creará para este usuario un modelo asociado a su objetivo actual.

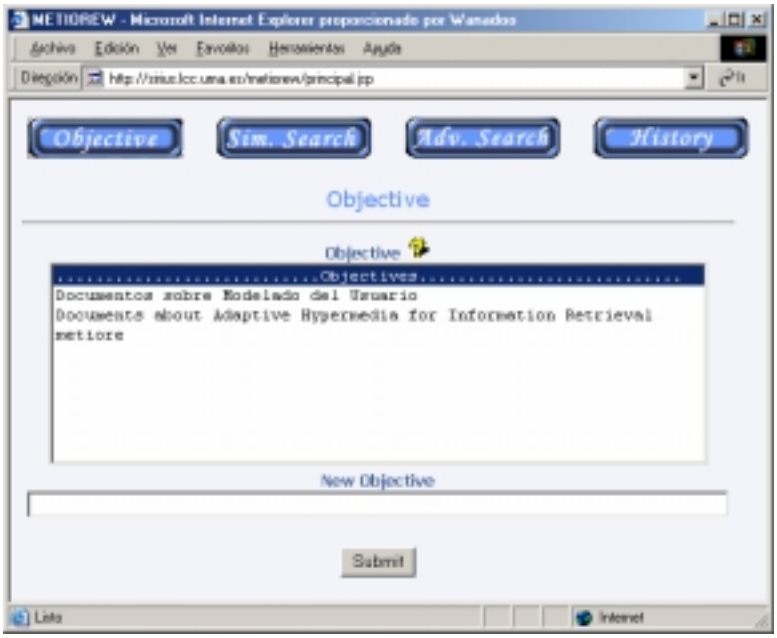


Fig. 26. Selección del objetivo

En este momento el usuario puede comenzar a realizar sus búsquedas o a trabajar con su historial para ver resultados de sus anteriores interacciones con el sistema. Si el usuario desea realizar una búsqueda, podrá utilizar la búsqueda simple donde pondrá la lista de términos que quiera utilizar (Ver Fig. 27). El resultado será una lista de documentos que, ordenando las soluciones en función de dos criterios: Primero se mostrarán los documentos que se correspondan mejor con el modelo del usuario y en el caso de que haya varios con el mismo valor, se elegirán por similitud con la consulta.

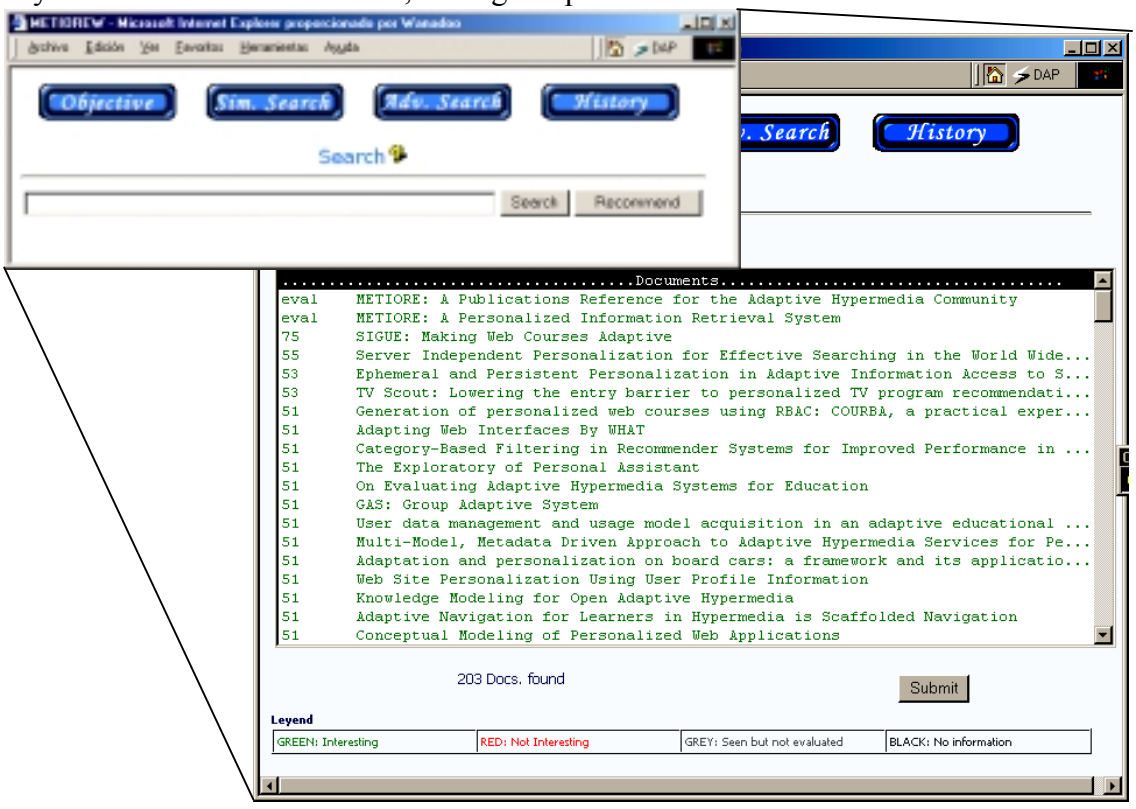


Fig. 27. Búsqueda simple y resultados

El segundo tipo de búsqueda es la búsqueda avanzada, en la que el usuario puede hacer un estudio de la base de datos y para comenzar a trabajar no necesita saber nada sobre su contenido. En la Fig. 28 se muestra esta interfaz. En los atributos de búsqueda se pondrán los elementos que se quiere que aparezcan en los resultados y con las restricciones se limitarán los resultados. En el ejemplo se han seleccionado los atributos *autor, autor* y la restricción del nombre de autor que contenga *brusilovsky*.

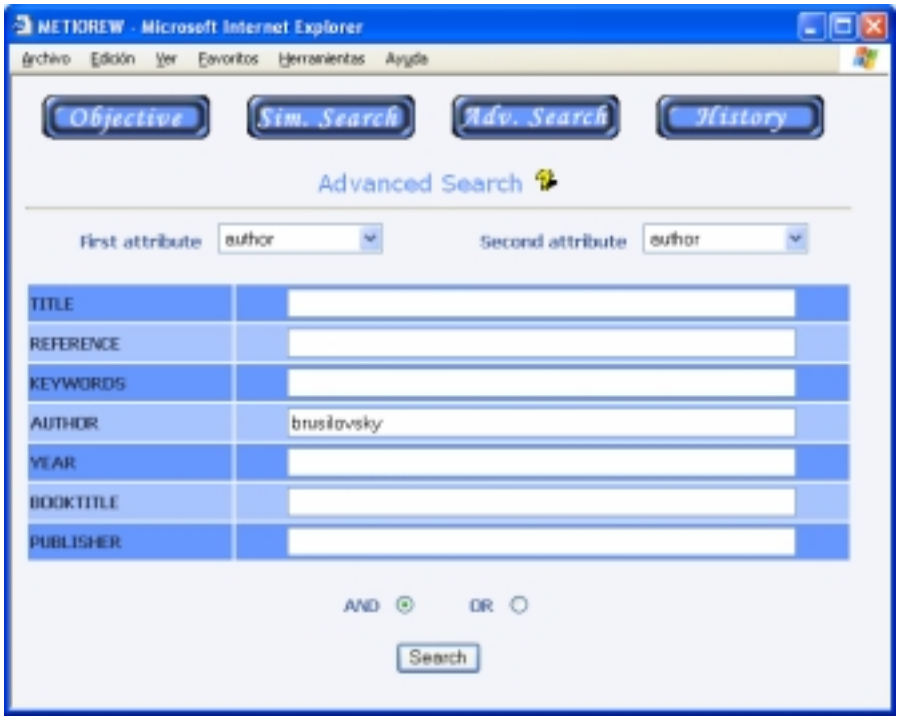


Fig. 28. Búsqueda avanzada (Selección autor-autor con restricciones)

Si no se hubiera introducido ninguna restricción, el resultado sería una lista con todos los pares de autores de la base de datos que han escrito juntos y la frecuencia con la que lo han hecho. Cuando se añaden restricciones, la consulta se limita a aquellos documentos que las cumplen. En este caso saldrán todos los pares de autores que aparezcan en documentos donde *brusilovsky* aparezca como uno de ellos. Cada una de estas líneas representa un cluster de documentos que tienen algo en común. Los clusters resultado de la consulta anterior pueden verse en la parte superior de la Fig. 29.

Cuando el usuario selecciona uno de estos clusters, los documentos del cluster se muestran en la parte inferior de la ventana (Fig. 29) ordenados de acuerdo al perfil del usuario al igual que se hacía con la búsqueda simple.

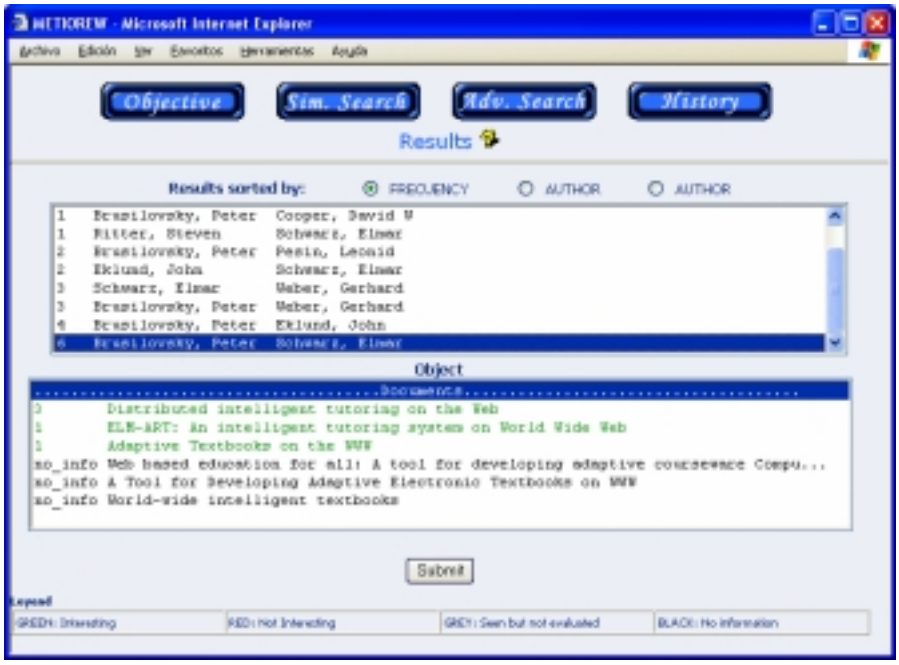


Fig. 29. Resultado de la búsqueda avanzada (Clusters/Lista documentos)

Otra posibilidad de realizar una búsqueda es la que ofrece el botón *recomiéndame*, que se encuentra en la interfaz de búsqueda simple (Fig. 27). Cuando se pulsa, se realiza la consulta a METIORE pero esta vez sólo se utilizan el modelo del usuario para el objetivo actual. Cada documento candidato a contener algún término incluido en el modelo se compara con él y como resultado se mostrará una lista de los documentos ordenados por similitud. A diferencia de los casos anteriores, sólo se mostrarán los documentos que METIORE considera interesantes para el usuario.

En cualquiera de los casos, el resultado final es la selección de uno de los documentos que el sistema le ofrece, para verlo con detalle. En las Fig. 30 y Fig. 31 se muestra la página de resultado detallado, para dos bases de datos diferentes. Aunque el resto de las interfaces sea común, el resultado detallado dependerá de la base de datos y puede contener elementos multimedia, como imágenes, videos o sonido. En el caso de la base de datos REVUE, se puede acceder desde METIORE al documento final en HTML, como se muestra en la Fig. 31.

Fig. 30. Resultado detallado para la Base de Datos AH2002

En estas pantallas de resultado detallado el usuario puede analizar la solución que se le propone y tiene la posibilidad de evaluarla. Esta evaluación es muy importante, para saber si al usuario le interesa o no el documento y supondrá una modificación en el modelo del usuario que implicará una mejora en las recomendaciones futuras. El tipo de evaluaciones también puede variar como se detalla en **IV.5.6**.

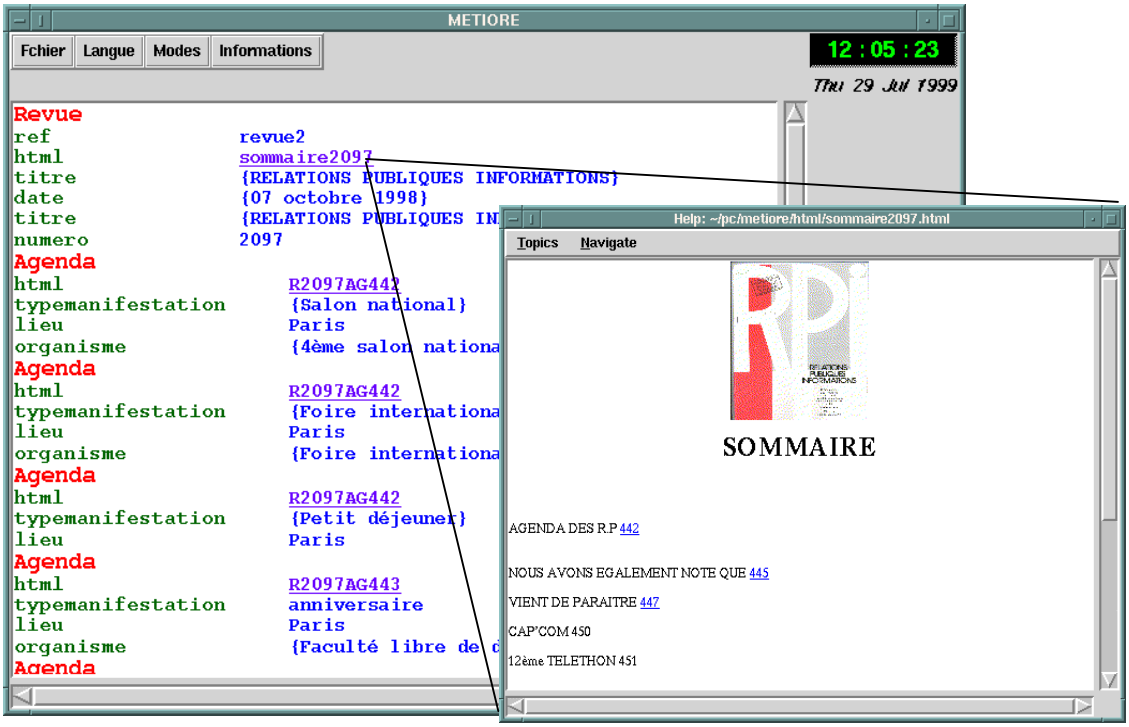


Fig. 31. Resultado detallado + evaluación (Base de Datos Revue)

En la misma interfaz aparece disponible el botón *Ver También*, que permite seleccionar cuáles son las propiedades del documento que parecen interesantes al usuario y que querría encontrar en otro documento. Esto generará una nueva búsqueda utilizando los criterios introducidos. En la Fig. 32 se muestra un ejemplo en el que el usuario está interesado en buscar documentos similares al actual en algunas palabras claves y autores.

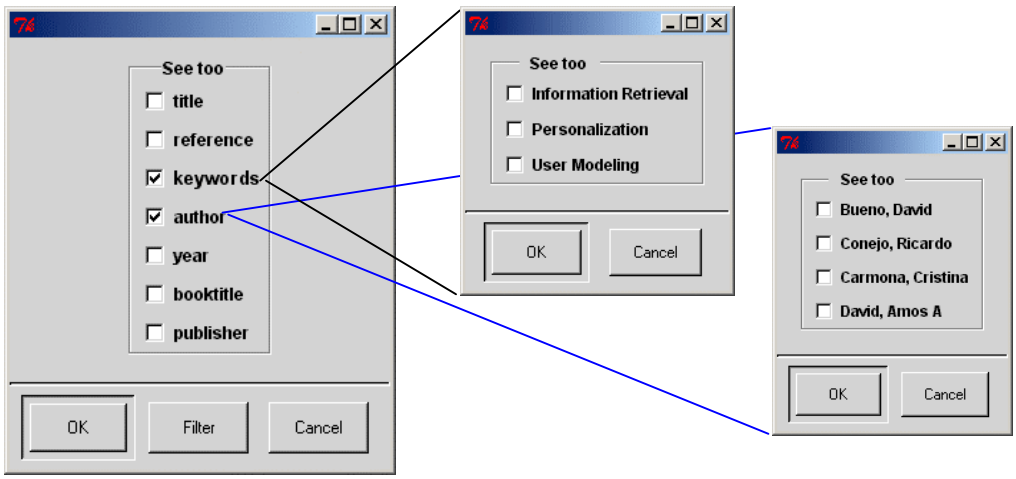


Fig. 32. Ejemplo de 'Ver También'

Otro elemento al que el usuario puede acceder es a su historial, en el que aparecen todos los objetivos que ha tenido con el sistema, y para cada uno, los documentos que evaluó. En la Fig. 33 se muestra los objetivos del usuario en la parte superior. Cuando selecciona alguno de ellos aparece una lista en la parte inferior con los documentos que ha visitado ordenados por sus respectivas evaluaciones. Si se selecciona alguno de ellos se mostrará el resultado detallado, permitiendo modificar su evaluación.

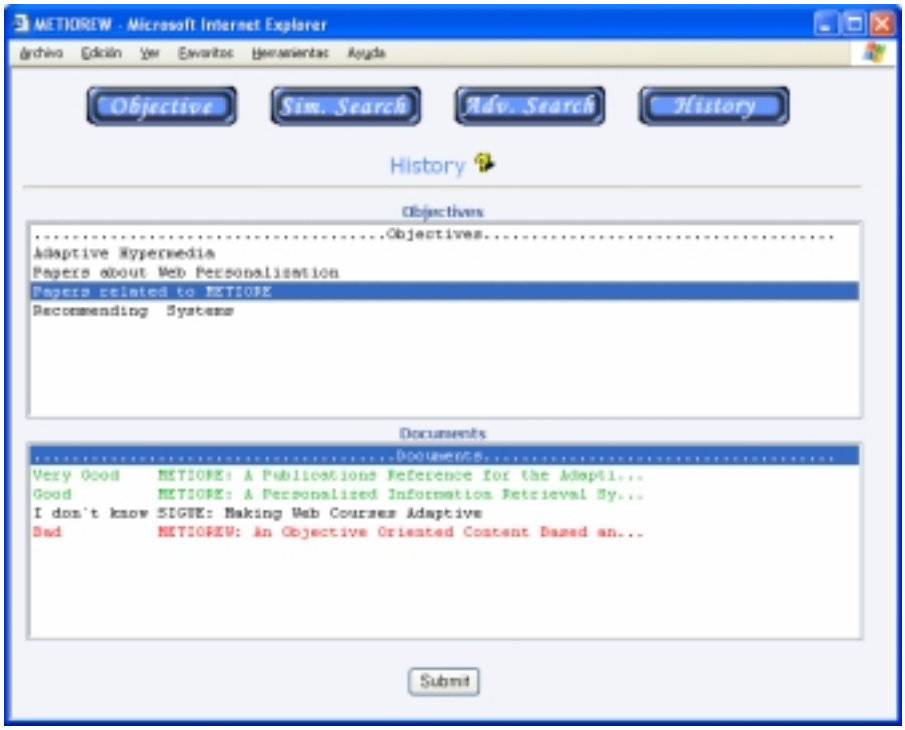


Fig. 33. Historial

V.6. FICHEROS

En este apartado se documenta la estructura de los ficheros que se utilizan por un lado para almacenar las diferentes bases de datos. Para tener una estructura que permita ofrecer al usuario una respuesta rápida a sus búsquedas, los datos se suelen organizar en diferentes ficheros que almacenan tablas hash o índices. Uno de los ejemplos más relevantes es el utilizado en el buscador Web más conocido en la actualidad: *Google*. En [Brin1998] se documentan los tipos de ficheros utilizados para almacenar y buscar datos en *Google*. En su caso, necesitan guardar todas las páginas HTML comprimidas, un índice de esas páginas, un diccionario con 14 millones de palabras, asociada cada una de ellas con una tabla hash de punteros a los documentos, una lista de aciertos (*hit list*), que muestra las ocurrencias de una palabra en un documento y otras estructuras similares. Como puede verse para este caso, sería complicado utilizar de forma eficiente una base de datos relacional clásica. A continuación se muestran los tipos de ficheros utilizados en METIORE para gestionar los datos.

V.6.1 LISTAS INVERTIDAS

La forma clásica de guardar los datos consiste en tener un documento y para ese documento analizar sus propiedades. La unidad de medida en ese caso es el documento. Si se quiere dar prioridad a las propiedades, sobre los documentos, los datos se guardarán en una lista invertida.

Una lista invertida guarda la información asociada a una propiedad. Para cada posible valor de esa propiedad, se guarda una lista de todos los documentos que la contienen. En las Fig. 34 y Fig. 35 se muestran ejemplos de dos ficheros de listas invertidas. La primera almacena las palabras clave y la segunda los autores, en ambos casos junto con los identificadores de los documentos donde aparecen. En METIORE se guarda cada

lista invertida en un fichero y hay una lista invertida para cada propiedad de la base de datos.

hypermedia_design {book12 book125} movies {book1 book48} multi-modality book60 information_goals book125 lesson book144 mapping_rules book12 individualization {book64 book66 book67 book103 book103} apprenticeship book91 domain {book10 book19 book23 book56 book87 book90 book91 book110 book114 book127 book151 book152} search_engines {book62 book87 book140 book294} **personalization {book4 book62 book86 book87 book96 book97 book100 book118 book129 book147 book149 book150 book172 book192 book204 book236 book256 book260 book293 book295 book352 book354 book396}**

Fig. 34. Lista invertida para palabras clave en AH2002

Gmytrasiewicz,_Piotr_J {book171 book366} de_Buen,_Pablo_R book110 **Bueno,_David {book62 book63 book64 book77}** Mauney,_Jennifer_Mitchell book69 Siegel,_Polly book88 Brailsford,_Tim book330 Kashihara,_Akihiro {book216 book217 book224 book318}

Fig. 35. Lista invertida para autores en AH2002

Las listas invertidas se utilizan para varias tareas. La más inmediata es para encontrar aquellos documentos que pueden satisfacer una consulta. Por ejemplo, tomando como referencia los datos de las figuras anteriores, si en la búsqueda del usuario aparecen las restricciones:

```
palabra clave=”personalization” AND autor=”bueno”
```

La forma de aprovechar la organización con listas invertidas, dará como resultado para *personalization* una serie de documentos que lo contienen (en negrita en la Fig. 34), y para *bueno* hay otra. Dependiendo del operador booleano AND/OR, la forma como se mezclarán esas dos listas será diferente. Si es AND sólo se cogerán los elementos comunes a las dos listas (en este caso el documento *book62*) y si es OR la unión de las dos.

Existe además una lista invertida que engloba todos los valores de todos los parámetros unidos. Esta lista invertida se utiliza para permitir una búsqueda clásica del tipo de un buscador tradicional en el que se introducen una serie de términos que pueden pertenecer a cualquier parámetro de la base de datos (palabras clave, año, autores,...).

Una segunda aplicación de las listas invertidas es como paso intermedio para generar los clusters, como se verá en el apartado siguiente.

V.6.2 CLUSTERS

Para poder realizar las operaciones de análisis de datos en las que se hacen consultas de cierta complejidad, como “¿Qué autores han trabajado en algo relacionado con Recuperación de la Información con David Bueno y cuantas veces lo han hecho?”, es aconsejable tener algunos datos preprocesados. Si se hace esa consulta al sistema sin tener ninguna información adicional, primero habría que buscar todos los artículos donde apareciera el autor *David Bueno* y la palabra clave *Recuperación de la Información*. Para todos los documentos resultantes habría que mirar todos los coautores y para cada uno de ellos contar cuantas veces aparecen. Cuando el volumen de datos es grande, esta operación puede durar demasiado.

Este tipo de consultas tienen en común la búsqueda de parejas de elementos que aparezcan juntos en un documento. Posiblemente, estas parejas de elementos están sometidas a algunas restricciones. En METIORE se guardan todas las posibles parejas en ficheros (clusters). Cada uno de estos ficheros contiene los pares de valores de dos

propiedades, que si son diferentes dan lugar a consultas inter-campo, pero si son la misma propiedad, dan lugar a consultas intra-campo. En la Fig. 36 se muestra una porción del fichero de clusters autor-año de la base de datos AH2002. Para cada pareja autor-año, se almacena la lista de documentos donde se encuentran ambos. El cálculo de las consultas multiatributo tiene un coste en tiempo considerable, como para realizar la consulta en línea. Disponer de los ficheros de cluster permite por un lado tener preprocesadas las consultas generales que utilizan dos atributos. En algunos casos se impondrán restricciones a la consulta, es decir, que se seleccionen dos atributos (autor y año), pero con la restricción por ejemplo de que en el documento aparezca una palabra clave concreta. En ese caso, habrá que buscar entre los clusters ya creados, aquellos que tengan los mismos documentos que los que aparecen en la lista invertida de la palabra clave que forma parte de la restricción.

```
<Stolze,_Markus><2001> book364 <Fernández_de_Arriba,_Marta><2002> book143 <Gu,_Xiaodong><2002> book182
<Fink,_Josef><1998> book147 <Shoval,_Peretz><2001> book176 <Bueno,_David><2002> {book62 book77}
<Hurley,_S><1998> book294 <Abrahão,_Silvia><2002> book4 <Rodríguez,_Ismael><2002> book254
```

Fig. 36. Ejemplo de cluster inter-campo (Autor-Año) en AH2002

```
<Kohlhase,_Michael><Sorge,_Volker> book345 <Krems,_Josef_F><Naumann,_Anja> book50
<Abbattista,_Fabio><Ferilli,_Stefano> book342 <Davis,_Joseph_G><Ghose,_Aditya_K> book140
<Bueno,_David><Conejo,_Ricardo> {book62 book63 book77} <Bull,_Susan><Shurville,_Simon> book66
```

Fig. 37. Ejemplo de cluster intra-campo (Autor-Autor) en AH2002

En la Fig. 37 se muestran una porción del fichero de clusters autor-autor, es decir, un fichero en el que se buscan apariciones en un documento de valores diferentes para una misma propiedad. Esto simplificará las consultas intra-campo.

En la versión aplicación se pueden utilizar hasta 3 atributos para combinar los clusters. En principio no hay limitación en el número de atributos a utilizar, aunque el espacio en disco ocupado por los clusters preprocesados crecerá de forma considerable.

V.6.3 OBJETOS

En los apartados anteriores se han mostrados ficheros que facilitaban la velocidad de acceso para las búsquedas. También es necesario guardar información de todos los objetos/documentos que forman parte de la base de datos. En principio podría pensarse en utilizar XML, aunque cada vez que hubiese que consultar un fichero habría que interpretar el XML. En la medida de lo posible, se ha limitado su a la conversión inicial de los datos por el coste que conlleva su interpretación. Hay una forma más rápida de leer los datos de un documento y es aprovechando las posibilidades de lectura/escritura de ficheros del lenguaje de programación utilizado. En nuestro caso[10], el formato más rápido es el que se muestra en la Fig. 38, donde hay una línea para cada parámetro con el formato: *-nombreparametro lista_valores*. De forma muy rápida se leen las líneas y se reconstruyen los objetos.

```
…
book77
-ref book77
-reference Rec2270
-title {SIGUE: Making Web Courses Adaptive}
-year 2002
-booktitle {AH2002 Second International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems}
-editor {Paul De Bra and Peter Brusilovsky}
```

[10] El lenguaje para el núcleo es [incr-Tcl]

```
-publisher {Springer Verlag,}
-url  http://sirius.lcc.uma.es/WCTP /doc/paper_208_AK3971AK3971.pdf
-abstract {Most of the information of the WWW is not adaptive, rather it is dispersed and disorganized. Another difficulty ...}
-author {Carmona, Cristina} {Bueno, David} {Conejo, Ricardo}
-motscles student adaptivity SIGUE navigation prerequisites recommendation MEDEA {knowledge units} {domain model}
book78
-ref book78
…
```

Fig. 38. Ejemplo de almacenamiento de los objetos/documentos de la Base de Datos (AH2002)

V.6.4 ÍNDICES DE OBJETOS

En memoria se cargan sólo los objetos necesarios. Para evitar accesos lentos al fichero de objetos (Apdo. V.6.3) y acceder únicamente a la zona del fichero donde se encuentra el documento, se utiliza un fichero de índices. Para la aplicación AH2002 se muestra a continuación. En cada línea se tiene el identificador del objeto y la posición física del fichero. Esta información se guarda en memoria en un array que permite cargar un documento de forma inmediata.

```
book1 0
book2 1561
book3 2867
book4 4582
book5 5745
book6 6780
book7 8208
book8 9630
book9 10954
```

Fig. 39. Fichero de indice de objetos para AH2002

V.6.5 HISTORIAL DE ACTIVIDADES

En el fichero de historial de actividades se guardan todas las interacciones que el usuario realiza con METIORE de forma estructurada, usando los conceptos vistos de objetivo, sesión, actividades, evaluación, etc. Estas interacciones se almacenan en formato XML y pueden ser analizados posteriormente por la herramienta METIORE-MU que se explica en V.7.

```
<user>
    <nomlogin> bueno </nomlogin>
    <session>
        <objectif> Qu'est-ce Amos a fait en 1998 </objectif>
        <activite>
            <heure> 10:24:55 </heure>
            <type> observation </type>
            <classification>
                <attribut1> author </attribut1>
                <lcontraintes>
                    <contrainte> year {* 1998} </contrainte>
                    <contrainte> author {* david} </contrainte>
                </lcontraintes>
                <feedback>
                    <solution> book719 </solution>
                    <evaluation> ok </evaluation>
                    <date> 99:06:18 </date>
                    <heurefb> 10:25:11 </heurefb>
                </feedback>
                <feedback>
                    <solution> book828 </solution>
                    <evaluation> known </evaluation>
                    <date> 99:06:18 </date>
                    <heurefb> 10:29:05 </heurefb>
                </feedback>
            </classification>
        </activite>
    </session>
</user>
```

Fig. 40. Ejemplo de historial de actividades (metiore.dat)

V.6.6 EVALUACIONES DE USUARIO

Para personalizar las repuestas, es necesario almacenar información sintetizada sobre las actividades del usuario. En las figuras Fig. 41 y Fig. 42 se muestra de dos formas diferentes la información que se necesita almacenar.

Historial_de_evaluaciones::=	elemento+
elemento ::=	objetivo objetos_evaluados+
objetos_evaluados ::=	atributos_evaluados+
atributos_evaluados ::=	atributo valor_evaluado+
valor_evaluado::=	valor ev1 ev2 ev3 ev4

Fig. 41. Historial de evaluaciones en formato EBNF

El historial de evaluaciones se compone de elementos. Cada *elemento* contiene un objetivo y una lista de objetos evaluados. Una de las suposiciones que deben hacerse para poder indexar y tratar un documento es que: "*La evaluación de un documento se considera como la evaluación de sus atributos*". De forma similar, evaluar un atributo se considera como evaluar los posibles valores de ese atributo para el documento actual. Es importante destacar que un usuario evalúa un documento como un conjunto y no cada uno de los valores de sus atributos. Por ejemplo, si en una base de datos de referencias bibliográficas se evalúa un documento como *ok*, eso quiere decir que todos los valores de todos los atributos de ese documento se evalúan también como *ok* (es decir, los valores del título, autor, año, palabras clave, etc...).

{objetivo$_1$ {atributo$_1$ {valor ev1 ev2 ev3 ev4} {valor ...}}...
{atributo$_n$ {...}}} ...
{objetivo$_m$ {atributo$_1$ {valor ev1 ev2 ev3 ev4} {valor ...}}...
{atributo$_n$ {...}}}

Fig. 42. Historial de evaluaciones en modo lista

Cada vez que se evalúa un objeto en el contexto de un objetivo, el tipo de evaluación se asocia con los valores de sus atributos. Lo que se representa para cada valor es en realidad la frecuencia para cada tipo concreto de evaluación. Por lo tanto *ev1, ev2, ev3 y ev4* son las frecuencias de las evaluaciones para *ok, known, bof* y *wrong*. Para ilustrarlo con un ejemplo, un objetivo podría tener un atributo *palabra_clave* con un posible valor *user_modeling* y las evaluaciones 5 3 0 5, que quieren decir el número de veces que algún documento que contenía esta palabra clave ha sido evaluado como *ok, known, bof* y *wrong* respectivamente. Los diferentes ev_i dependerán del modelo de evaluación que será alguno de los vistos en **IV.5.6**.

V.7. METIORE-USER

En este apartado se presenta la herramienta METIORE-USER (MU) [Bueno2000a]que se utiliza para analizar las interacciones del usuario con METIORE. En realidad no es una herramienta adicional, sino el propio sistema, que recibe como base de datos el historial de actividades de un usuario (Ver Fig. 40). Esta herramienta permite:

- Obtener información sobre el conocimiento del usuario
- Obtener una visión global del comportamiento de usuario en el sistema
- Obtener la correlación entre las actividades del usuario y sus objetivos

El análisis del historial puede representarse en forma de gráficos. Estos gráficos pueden ser interpretados por el mismo usuario o por un colaborador en el contexto de recuperación de la información cooperativa.

Sin esta herramienta, el análisis del historial de un usuario sería muy complicado. El historial de usuario puede compararse a los registros de una base de datos. Aquí los registros son las actividades del usuario. El problema es por lo tanto similar al de obtener una visión global de una base de datos. Por eso, se utilizan los mismos métodos de análisis de datos para los usuarios.

V.7.1 ANÁLISIS SIMPLE DEL HISTORIAL DE USUARIO

En la interfaz de búsqueda avanzada (Ver Fig. 28), para MU estarán disponibles los atributos (objetivo, sesión, evaluación, solución, atributo, restricciones, hora/fecha) que podrán aplicarse con o sin restricciones. Esto ofrecerá las posibilidades de análisis simple, utilizando un solo atributo con restricciones, que se muestran en la Tabla 11.

Attributes	*Result*
Objetivo	Lista de los objetivos utilizados por ese usuario
Sesión	Lista de todas las sesiones
Evaluación	Lista de todas las evaluaciones agrupadas por tipo de evaluación
Solución	Lista todas las soluciones analizadas por el usuario ordenadas por frecuencia de aparición
Atributo	Lista los atributos utilizados para las búsquedas
Restricciones	Lista de todas las restricciones utilizadas, agrupadas y ordenadas por frecuencia de uso

Tabla 11. Análisis simple del historial de usuario

Si se combinan con los atributos algunas restricciones, el sentido de las consultas cambia sensiblemente, como puede verse en los siguientes ejemplos:

- ***Atributo:*** *Objetivo****; Restricciones:*** *evaluación='ok'*.- Esto dará como resultado una lista de todos los objetivos del usuario donde al menos uno de los objetos evaluados se evaluó como *ok*.
- ***Atributo:*** *Sesión****; Restricciones:*** *solución='docX'*.- Devuelve una lista de todas las sesiones donde las solución docX haya sido evaluada. Esto puede utilizarse por ejemplo para ver cuatas veces el usuario a revisado un documento concreto.
- ***Atributo:*** *Solución****; Restricciones:*** *evaluación='error' Y sesion='sesionX'*.- Muestra todas las soluciones que han sido evaluadas como error durante la sesión X

V.7.2 ANÁLISIS CRUZADO

También es posible combinar más de un atributo con restricciones para realizar un análisis. Pueden combinarse hasta tres atributos para realizar el análisis cruzado combinado con restricciones. A continuación se muestran algunos ejemplos de análisis cruzado con restricciones para analizar el historial de un usuario:

- ***Attributo1****: Evaluación;* ***Attributo2****: Fecha;* ***Restricciones****: Solución = 'Solx'*
  Con este análisis se puede ver la evolución de las evaluaciones de un usuario para una solución dada. Es posible que un usuario evalúe la misma solución de forma diferente por alguna de las siguientes razones: a) Su nivel de conocimiento en el campo ha mejorado b) Al principio no tenía conocimiento suficiente sobre el sistema y por lo tanto analizó de forma incorrecta alguna solución, c) El usuario está evaluando de forma indiferente sin adecuar estas evaluaciones a su objetivo real.

- ***Atributo1****: Objetivo/Sesión ;* ***Attributo2****: Evaluación*
  Esta consulta devolverá la frecuencia de cada tipo de evaluación que ha sido realizada para cada objetivo o sesión. Si la mayoría de las evaluaciones son de error, esto podría indicar que las soluciones que el sistema propone no son relevantes para las necesidades de información de los usuarios. Pero esta interpretación debería ser investigada ya que puede haber varias causas para que el usuario identifique una solución como errónea. Por ejemplo, que el usuario no sea capaz de expresar su necesidad de información con la interfaz de búsqueda del sistema o que un atributo discriminante no este disponible para permitirle esa expresividad. Esto puede observarse por ejemplo en el caso de un usuario que esta buscando en una base de datos bibliográfica por autores españoles. Es posible que en la referencia bibliográfica de la base de datos no haya un campo que represente el origen de los autores, lo que dificultará al usuario formular esta consulta.

V.8. OBTENCIÓN DE DATOS DE DOCUMENTOS

En este apartado se documentan los pasos que se utilizan en METIORE cuando se quieren indexar documentos que no provienen de una base de datos ya establecida. El ejemplo de aplicación es la base de datos sobre Hipermedia Adaptativos (AH2002) en la que los artículos que la forman se obtenían de forma integra de la Web, ya sea en formato *pdf*, *.ps*, texto o *Word*. Para poder incluirlos en la base de datos es necesario obtener los siguientes campos (titulo, lugar de publicación, editores, autores, año, resumen, palabras clave, URL,...). La mayoría de ellos se pueden extraer directamente del documento. Uno de los campos que normalmente es muy útil para las búsquedas documentales y en algunos sistemas es el único que se utiliza es el campo *palabras clave*. Casi ningún documento contiene las palabras clave que lo describen, por lo tanto ha sido necesario desarrollar un programa que recibiera un documento completo y extrajera las palabras clave.

El algoritmo utilizado para la obtención de las palabras claves es el siguiente: Primero se filtra lo que se conoce como palabras vacías (*stop words)* que son palabras no relevantes para identificar el documento (artículos, preposiciones,...) después se aplica el algoritmo de *Porter* [Porter1980]. Dicho algoritmo recibe una palabra y devuelve su raíz. Esto permite homogeneizar palabras de una misma familia para que se identifiquen como una sola y su peso en el documento sea mayor. Pero para METIORE, debido a sus posibilidades de análisis de datos, no interesaba quedarse solamente con las raíces para representar palabras clave, sino que se necesitaba tener las palabras claves completas. Para ello se decidió utilizar los resultados de *Porter* para agrupar todas las palabras con la misma frecuencia pero sin perder su valor inicial. Después de ordenan por frecuencia (*Term Frequency-TF*) y se muestran al usuario de la forma siguiente:

US[121]: users[27] user [64] use [19] User [10] uses [1]

Esto significa que hay 121 palabras con la raíz US de las cuales 27 se corresponde a *users* , 64 a *user,* etc. De esta forma, se pueden seleccionar de forma sencilla y rápida las palabras que representan al documento entre las propuestas por el programa. Una muestra del programa que realiza estas acciones puede verse en la Fig. 43.

Fig. 43. Aplicación para extraer palabras clave usando el algoritmo de Porter

V.9. CONCLUSIONES

En este capítulo se ha mostrado una implementación de las ideas propuestas en el capítulo anterior mediante METIORE. De este sistema se ha mostrado su arquitectura, forma de uso, y los mecanismos que utiliza para convertir bases de datos externas a un formato propio que permite utilizarlas. Se han introducido los detalles de las estructuras de datos utilizadas para almacenar los documentos de una forma eficiente que permite hacer consultas complejas en un tiempo razonable. También se ha tratado un caso de base de datos especial que permite utilizar el histórico de los usuarios como datos para el sistema. Por último, se incluye una herramienta que permite extraer palabras claves de los documentos utilizando el algoritmo de porter. Esto es muy útil cuando se quieren indexar documentos Web, o artículos que no disponen de palabras clave.

La implementación de METIORE ha servido para probar que las propuestas teóricas del capítulo anterior son viables y se han podido aplicar en un sistema real. Además servirá para poder hacer pruebas con usuarios reales y verificar que la personalización que ofrece satisface a los usuarios. Esto se verá en el capítulo siguiente.

VI. EXPERIMENTACIÓN

VI.1. INTRODUCCIÓN

En los dos capítulos anteriores se ha expuesto una propuesta de personalización para los sistemas de recuperación de la información, basada en un algoritmo de personalización e implementadas en el sistema METIORE. En este capítulo se describen las pruebas realizadas para valorar la propuesta de esta tesis. En primer lugar se muestran unas pruebas que comparan la forma de ordenar una lista de documentos utilizando diferentes algoritmos. Esto muestra que en igualdad de condiciones, es decir, utilizando un solo parámetro, los algoritmos Naïve Bayes y Naïve Bayes Cestnik se comportan de forma similar al algoritmo aquí propuesto NBM. En segundo lugar se introducen las pruebas realizadas con la aplicación METIORE con un grupo controlado de usuarios, personal investigador del laboratorio LORIA de Nancy, en las que se muestran las ventajas reales de la aplicación. Esto permitió comprobar que los resultados que generaba el algoritmo y la aplicación en general eran satisfactorios para los usuarios. Por último, se analizan los resultados de METIORE en la Web con un grupo no controlado de usuarios. Esto permitió comprobar que usuarios no instruidos para el uso del sistema fueron capaces de obtener un buen rendimiento del sistema, además de reflejar la viabilidad del algoritmo en bases de datos diferentes.

VI.2. COMPARACIÓN CON NAÏVE BAYES

En este apartado se muestran los resultados de comparar tres variantes del algoritmo Naïve Bayes. El objetivo de esta prueba es comprobar que los resultados del algoritmo aquí propuesto NBM (NB-Metiore) son similares a la hora de ordenar una serie de documentos para un usuario que los que daría Naïve Bayes en su versión clásica (NB) o en su modificación por Cestnik (NB-Cestnik). Si eso se cumple, se muestra que en igualdad de condiciones el funcionamiento es similar, con la ventaja del algoritmo aquí propuesto de ser aplicable a múltiples parámetros. La comparación se va a realizar calculando la correlación entre los algoritmos comparados dos a dos. Lo esperado es que la correlación que se mueve en un rango entre 0 y 1 sea lo más próxima a 1 en las

comparaciones de los otros algoritmos como el aquí propuesto siendo deseable que su parecido sea mayor al NB-Cestnik por ser una mejora de NB.

Los datos de los experimentos se muestran en el Apéndice III. Para el experimento se suponen 1000 objetos analizados, los cuales pueden contener tres atributos diferentes. Para cada atributo se va a calcular $Q_i(C,J_i)$, (factor fundamental de la ecuación Naïve Bayes. Ver IV.6.1) según corresponda en cada algoritmo, como se muestra en la Ecuación 29. Para la comparación se van a suponer dos clases posibles de evaluación del usuario: *ok, error* de las cuales es suficiente utilizar una de ellas (*ok*) para ver como se ordenan los documentos.

En el diseño de las pruebas lo importante es poder comparar como los tres algoritmos generan sus probabilidades utilizando los Q_i. Con la peculiaridad de que para el algoritmo de Cestnik, para el cálculo del Q_i hay que suavizar los valores $P(V_i)$ y $P(V_i|C)$ por si alguno de ellos es cero. Estos valores se han generado de forma aleatoria como se muestra en la Ecuación 29a. Para cada supuesto documento se han calculado los valores de las ecuaciones simplificadas mostradas en la Ecuación 29b.

Para cada uno de los tres algoritmos, los documentos se han ordenado de menor a mayor y se ha calculado el coeficiente de correlación entre las tres series posibles. En la Tabla 12 se muestra el resultado de esa comparación. Estos resultados indican que los tres algoritmos en general realizan una ordenación parecida, pues en las tres comparaciones el factor de correlación es cercano a 1. Siendo mayor la correlación entre NBM y NB-Cestnik. Es decir, nuestra propuesta y la mejora de Cestnik que suaviza los casos en que algún atributo valga 0. En las gráficas (Apéndice III) puede verse que los documentos mejor evaluados son prácticamente los mismos, con pequeños cambios de orden en algunos casos.

Estas pruebas muestran que la variación de Naïve Bayes propuesta da resultados del orden de los obtenidos con otras variaciones aceptadas y que sus resultados son más próximos a las mejoras del algoritmo básico que a el NB original.

	NB-NBCestnik	NB-NBMetiore	NBMetiore-NBCestnik
Factor Correlación	0,922	0,930	0,937

Tabla 12. Resumen de resultados

$$Aleatorio() \rightarrow [0,1]$$
$$P(V_{i,J_i}) = CP_i = 20 * Aleatorio()$$
$$P(V_{i,J_i} \mid ok) = CF_i = CPi * Aleatorio()$$
$$Q(ok, J_i) = Valor_i = \frac{CF_i}{CP_i}$$
$$Q'(ok, J_i) = Valorp_i = \frac{CF_i + 1}{CP_i + 2}$$

a)

$$NB = \prod_{i=1}^{3} Q(ok, J_i)$$
$$NB\_Cest = \prod_{i=1}^{3} Q'(ok, J_i)$$
$$NBMetiore = \frac{\sum_{i=1}^{3} Q(ok, J_i)}{3}$$

b)

Ecuación 29. Ecuaciones utilizadas para el experimento

VI.3. EXPERIMENTACIÓN CON EL PROTOTIPO APLICACIÓN

La importancia de la evaluación es mostrar si las hipótesis iniciales se han verificado durante el uso del sistema. En este apartado se mostraran los métodos utilizados para evaluar el prototipo y los resultados obtenidos.

VI.3.1 LOS MÉTODOS

Una de las formas de evaluar un sistema para dar respuestas personalizadas es utilizar los datos recopilados en cada interacción con los usuarios. Esto quiere decir que si se propone una solución, se puede utilizar como medida de buen funcionamiento el porcentaje de éxito de estas propuestas. Este tipo de evaluaciones fue utilizado en el asistente personal de Tom Mitchell [Mitchell1994]. En nuestro caso se utiliza este sistema para calcular la precisión de los resultados desde el punto de vista del sistema. Se han realizado algunas otras observaciones durante la evaluación del sistema supervisando las interacciones de los usuarios con el sistema. Además se ha utilizado un cuestionario que rellenaba al finalizar la sesión. Uniendo todos estos elementos se han obtenido algunas conclusiones interesantes.

En la Tabla 13 se presentan las principales hipótesis y la forma de verificarlas. Para ello se utiliza un cuestionario y también la información del sistema para validar las respuestas de los usuarios.

Hipótesis	Sistema	Cuestionario
La integración de las actividades del usuario y su asociación con el objetivo para el cálculo de soluciones debería proponer más soluciones que el usuario evaluará como relevantes, que si esta integración no se hiciese.	% de soluciones predichas correctamente	5,6,7,8,9,10,11, 12
La exploración del historial activo debería facilitar la operación de recuperación de la información y ayudar a ofrecer soluciones relevantes	Analizar la frecuencia de uso del historial	19,20
La posibilidad del usuario de dar evaluaciones debería ayudar a comprender mejor sus necesidades de información.	% de soluciones predichas correctamente	16,17,18
Clasificar las soluciones por tipo de evaluación y la asociación de códigos de colores debería facilitar la recuperación se soluciones.		5, 6, 7
La posibilidad de recuperación cooperativa debería acelerar el proceso de encontrar soluciones relevantes.	No evaluado	

Tabla 13 Hipótesis y formas de comprobarlas

A continuación se muestra el cuestionario que se pasó a los participantes.

Cuestionario para METIORE

Interfaz

1. ¿Es fácil de utilizar la interfaz? (Entre 0[No] - 5[Sí])
2. ¿Piensa que puede expresar todas las operaciones de búsqueda que necesita? (0-5)
3. ¿Cuáles son las mayores dificultades que encuentra en la interfaz?
4. ¿Qué cambiaría de la interfaz?.....

Ordenación de las soluciones (y códigos de colores)

5. ¿Entiende la utilidad de los códigos de colores? (0-5)
6. ¿Son útiles? (0-5)
7. ¿Podría indicar la utilidad que le encuentra a los códigos de colores? .....

Recomendación de soluciones

8. ¿Piensa que las soluciones son personalizadas? (0-5)
9. ¿Piensa que el sistema sabe realmente lo que busca? (0-5)
10. ¿Esta contento con las soluciones del sistema? (0-5)

11. ¿Encuentra útil el botón *ver también?* (0-5)
12. ¿Son interesantes las soluciones finales? (0-5)

Objetivo
13. ¿Entiende la necesidad de indicar su objetivo? (0-5)
14. ¿Tiene el mismo objetivo en diferentes sesiones? (0 [nunca] – 5 [siempre])
15. ¿Le resulta util conocer que documentos han interesado a otros usuarios con objetivos similares? (0-5)

Evaluación
16. ¿Le gusta evaluar las soluciones propuestas por el sistema? (0-5)
17. ¿Piensa que la clasificación (correcto, conocido, ?, error) es suficiente? (0-5)
18. Si la respuesta 17 es negativa, ¿ Qué otros valores sugiere?.....

Historial activo
19. ¿Utiliza el historial? (0-5)
20. ¿Lo encuentra útil? (0-5)

Comparación de METIORE con otros sistemas
21. ¿Cuáles son las ventajas/desventajas que encuentra en METIORE con relación a otros sistemas similares que conozca (escriba sus nombres)?

Velocidad	(0 [METIORE peor] – 5 [METIORE mejor])
Rendimiento	(0-5)
Interfaz	(0-5)
Cooperación	(0-5)
Personalización	(0-5)
Historial	(0-5)
Otros	(0-5)

22. ¿Piensa continuar utilizando el sistema? (Si/No)
23. Lo mejor de METIORE.....
24. Lo peor de METIORE.....

VI.3.2 RESULTADOS

Los resultados de los experimentos pueden frustrar las esperanzas de muchos meses o años de investigación si no son satisfactorios. Si ese fuera el caso, debería hacer replantearse a los investigadores que quizás la línea que están siguiendo no es la correcta. En el ámbito subjetivo en el que se mueven los sistemas de recuperación de la información, el bajo porcentaje de éxito de las recomendaciones no indica un fallo significativo. Hay dos frases célebres de autores de sistemas recomendadores que sustentan esta hipótesis. Por un lado, Lieberman [Lieberman1995] dice: "*Its guesses only need be better than no guess at all, and so even wake heuristics can be employed*". Es decir, que sugería que para estar satisfechos es suficiente con que lo que se recomienda sea mejor que nada ya es bueno. Una segunda frase de Armstrong con el mismo sentido decía de su sistema WebWatcher [Armstrong1995]: "*Although learned knowledge may provide only imperfect advice, even a modest reduction in the number of hyperlinks considered at each page leads to an exponential improvement in the overall search*". Con eso el sistema estaría aportando su granito de arena. Es decir, con que el sistema haga algunas recomendaciones interesantes es mejor que no tener ninguna recomendación. Aun así, en nuestros experimentos con METIORE encontramos resultados mucho mejores de los esperados, como se verá a continuación.

Los experimentos se han realizado en el laboratorio de investigación LORIA en Francia. La base de datos utilizada es la base de datos de ese laboratorio. Las personas envueltas en el experimento fueron 20 miembros de investigación y doctorandos del laboratorio. Se llevaron a cabo durante tres meses, en sesiones de una hora aproximadamente. La experimentación se dividió en tres fases:

1. *Fase de explicación.*- Se mostraba a los usuarios las características funcionales del prototipo. (Tipo de sistema, características generales, interfaz, etc...).

2. *Fase de utilización*.- Se dejaba al usuario interactuar con el sistema, teniendo un objetivo en mente para realizar búsquedas e intentar encontrar el máximo número de documentos relacionados con su objetivo. Durante esta fase el usuario trabajaba solo con el sistema aunque se supervisaban sus acciones en silencio.
3. *Fase de evaluación*.- Los usuarios rellenaban el cuestionario y podían hacer comentarios sobre el sistema.

Para poder generar estadísticas de las predicciones del sistema después de cada evaluación del usuario es necesario almacenar alguna información adicional. La información que se guarda son las frecuencias de los pares (evaluación, predicción), es decir, el número de veces que el sistema predice una evaluación y la evaluación real del usuario. Un ejemplo de los datos almacenados en el fichero de un usuario real se muestra en la Fig. 44. Las evaluaciones del usuario pueden ser (ok ,known ,? y wrong) y las predicciones que hace el sistema, se preceden con una p (pok, pknown, pnormal, p?, pwrong). En la Fig. 45 se muestra la representación gráfica para esos datos.

```
?,pnormal 1 ok,pok 11 ?,pok 5 wrong,pwrong 1 wrong ,pnormal 2 wrong,pok 2 ok,pnormal 7 ok,pwrong 1
```

Fig. 44. Predicciones

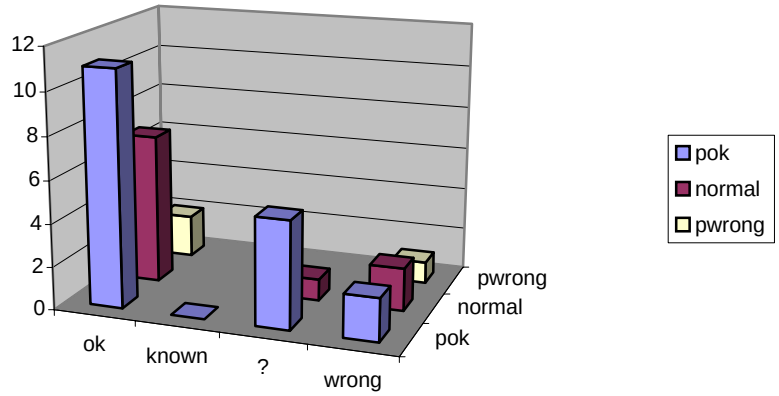
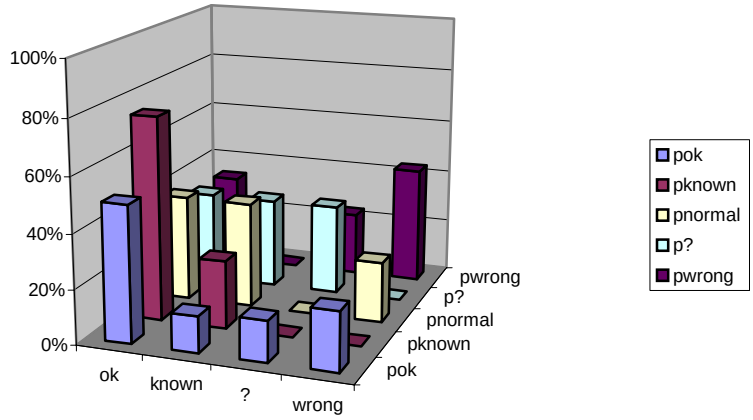


Fig. 45. Representación gráfica de las predicciones

De esos datos puede concluirse para ese usuario que el 61% de las veces que el sistema predecía que la solución le interesaría, realmente sería así. Es importante destacar que en las primeras interacciones, el sistema no tiene suficiente información para recomendar a un usuario y no es capaz de saber si un documento puede o no interesar al usuario. Es en estos casos, cuando el sistema predice como **normal.** Incluso en estos casos los documentos son relevantes en un bien número de casos.

Los resultados más interesantes aparecen cuando se unen los cálculos realizados para todos los usuario, lo que da una idea más general. La Fig. 46 representa la gráfica de todas las predicciones hechas por el sistema junto con lo que el usuario evaluó realmente. De esos datos puede concluirse que:



	ok	known	?	wrong
pok	50,00%	13,33%	15,00%	21,67%
pknown	75,00%	25,00%	0,00%	0,00%
pnormal	38,89%	38,89%	0,00%	22,22%
p?	33,33%	33,33%	33,33%	0,00%
pwrong	33,33%	0,00%	23,33%	43,33%

Fig. 46. Resultados de evaluaciones para usuarios de METIORE-LORIA

- El porcentaje de (ok/pok) es muy interesante. El 50% de las veces que el sistema propone una solución, la propuesta es correcta para ese usuario. El 13,33% de las veces que el sistema predice que le interesará acierta aunque el usuario responde que ya lo conoce. Esto quiere decir que el 63,33% de las predicciones del sistema son correctas, lo que consideramos un buen resultado.
- La predicción de *known* tiene un alto porcentaje de evaluaciones como *ok(*75%). Esto puede considerarse razonable ya que cuando un usuario evalúa un documento como conocido quiere decir que el documento le interesa, y que documentos similares a los evaluados también le interesan aunque quizás no los conozca.
- El número de éxitos mejora con el tiempo. Como cabría esperar, cuanto más se usa el sistema, más conoce el sistema las preferencias del usuario para ese objetivo y por lo tanto las propuestas del sistema son cada vez más acertadas.
- La mayoría de los usuarios aceptan las recomendaciones y raras veces evalúan las que se le sugieren como erróneas. Estas acciones por parte del usuario pueden ser algo peligrosas al comienzo cuando el sistema no tiene información suficiente sobre el usuario y las recomendaciones no son totalmente precisas. Una forma de evitar recomendaciones sin tener demasiados dados es imponer un número mínimo de evaluaciones correctas antes de que el sistema empiece a hacer recomendaciones.

Los cuestionarios se diseñaron siguiendo un proceso de refinamiento tras varias pruebas iniciales antes de llegar a su forma final, definiendo que se quería que aportase cada una de las preguntas. Los resultados de los cuestionarios y la observación durante la realización de los experimentos han llevado a las siguientes conclusiones:

- El 90% de los usuarios quedaron satisfechos con las soluciones que le daba, y sentían que el sistema había comprendido cual era su objetivo. En particular, algunos usuarios al final de la evaluación pedían al sistema una recomendación automática, es decir, sin realizar ninguna consulta, utilizando sólo los datos del modelo.

- Se pensó al principio que hacer que los usuarios evaluaran los documentos propuestos iba a ser una tarea que les disgustaría. Entre los comentarios de los usuarios destacaban comentarios como que evaluar los documentos era la única forma de obtener una asistencia más personalizada y correcta, así que no les preocupaba hacerlo.
- Los usuarios que utilizaron el sistema en varias sesiones usaron el historial activo y lo encontraron como una herramienta interesante. Por otro lado, los usuarios que sólo interactuaron una vez no pudieron apreciar su utilidad.

VI.4. EXPERIMENTACIÓN CON EL PROTOTIPO WEB

Las características de un experimento en la Web difieren en cierta medida de las realizadas con usuarios bajo un entorno controlado. Las motivaciones de un usuario que accede al sistema a través de la WWW pueden ser varias. En un primer lugar, curiosidad por ver cómo es ese sistema. Si el contenido le resulta interesante, podría utilizarlo como cualquier otro buscador. Sólo en el caso en que quiera aprovechar las características de personalización, el usuario evaluará los documentos para obtener resultados personalizados.

Para hacer las pruebas en la Web, no era suficiente con tener el sistema y cualquier base de datos, sino que era necesario disponer de una base de datos que resultase atractiva para algún grupo considerable de usuarios. Si además este grupo de usuarios estaba interesado en la personalización se conseguiría tener usuarios más motivados. Por eso se desarrollo una base de datos, que recopila publicaciones científicas sobre hipermedia adaptativos (en adelante AH2002) seleccionadas de la página de hipermedia adaptativos[11], Citeseer[12], buscadores genéricos como Google[13] y de páginas de conferencias relacionadas. Cada publicación debía ser indexada extrayendo los datos típicos de una publicación (titulo, autor, conferencia/revista,...). Además era necesario extraer las palabras clave para lo que se utilizó el programa visto en V.8. Las características de esta versión de METIORE aparecen en [Bueno2002].

Una vez detallada la base de datos, el estudio que se ha realizado ha sido observar las interacciones de los usuarios con el sistema y analizar los resultados. Los usuarios del sistema son en su mayoría investigadores, profesores universitarios o estudiantes. Un total de 146 usuarios se han registrado para utilizar este sistema. En la Tabla 14 se muestra la procedencia de las visitas a METIORE y el porcentaje según el país.

Spain	38,7%	***Ireland***	2,5%	***Malaysia***	0,5%
Adresse IP	15,5%	***Netherlands***	2,5%	***Colombia***	0,4%
Network	9,1%	***US Commercial***	2,0%	***Finland***	0,4%
Germany	6,2%	***Greece***	2,0%	***Poland***	0,4%
United Kingdom	5,5%	***Canada***	1,4%	***Belgium***	0,2%
Brazil	3,4%	***Switzerland***	1,2%	***Japan***	0,2%
France	3,2%	***US Educational***	1,1%	***Australia***	0,2%
Italy	2,8%	***Denmark***	0,7%	***Austria***	0,2%

Tabla 14. Distribución de visitas de usuarios a METIORE

[11] Adaptive Hypermedia. http://wwwis.win.tue.nl/ah

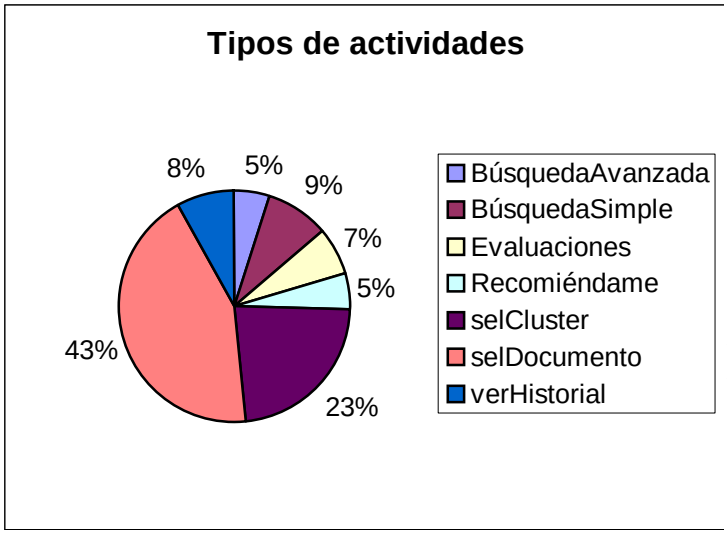
[12] http://citeseer.com

[13] http://www.google.com

Para comenzar el estudio, en la Fig. 47 se muestran los datos obtenidos de las interacciones de los usuarios. Las actividades de búsqueda se han agrupado en dos, las que utilizan una lista de palabras clave al estilo de cualquier buscador (*BúsquedaSimple*) y las que utilizan la interfaz de análisis de datos (*BúsquedaAvanzada)*. También se muestra el número de veces que los usuarios han pedido recomendación al sistema (*Recomiéndame)*. *SelCluster* indica el número de clusters que se han consultado como resultado de la Búsqueda Avanzada. Cada vez que el usuario visualiza en detalle un documento se corresponde con una actividad de tipo *selDocumento*. Una actividad de *Evaluaciones* ocurre cada vez que el usuario evalúa el documento. Por último, si consulta su historial, la actividad es *verHistorial*.

De los datos de la Fig. 47 se obtiene los siguientes resultados, algunos quizás inesperados:

- Por ejemplo, se esperaba que el número de búsquedas realizadas de la forma tradicional (BúsquedaSimple) iba a ser muy superior a la búsqueda avanzada. La experiencia en cursos y en ver como los usuarios utilizan normalmente los buscadores (pocos utilizan las opciones de búsqueda avanzada) hacía pensarlo así. De los resultados analizados el 65% de las búsquedas eran simples y el 35% avanzadas. Además la operación de analizar los clusters (selCluster), a la que sólo puede accederse a través de la búsqueda avanzada se ha utilizado bastante, una media de 4,66 veces por cada búsqueda avanzada.



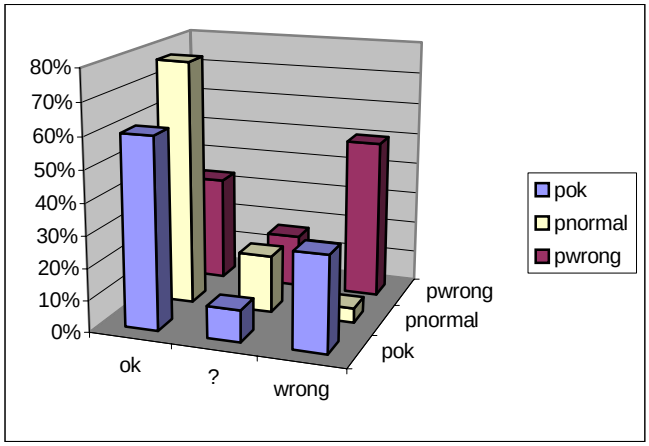
Actividades	Frecuencia
BúsquedaAvanzada	85
BúsquedaSimple	156
Evaluaciones	118
Recomiéndame	88
selCluster	396
selDocumento	762
verHistorial	140

Fig. 47. Actividades realizadas por los usuarios de METIORE en la Web

- Una de las hipótesis que también se han confirmado con este estudio es la utilidad del historial. Un 8% de las actividades realizadas era para consultar los documentos ya evaluados.
- Otro dato interesante es el número de evaluaciones por parte de los usuarios. Al contrario que en el estudio anterior, donde los usuarios estaban motivados a evaluar los documentos, en este caso, como los usuarios que accedían al sistema no tenían ningún tipo de instrucción previa sobre el funcionamiento, hay muchos que sólo se han limitado a hacer las búsquedas sin evaluar documentos, con lo que han obtenido resultados no personalizados. El número de evaluaciones se corresponde con el 15,5% de documentos vistos en detalle. Este porcentaje es bastante bajo y además se concentra por usuarios, es decir, hay usuarios que evalúan todos o casi todos los documentos que visualizan y otros que no evalúan ninguno.

- La actividad de recomendación supone el 5% de las actividades. En los datos puede verse que usuarios que no han evaluado ningún documento solicitan esta actividad alguna vez, aunque la mayoría de las solicitudes de recomendación están relacionadas con usuarios que han evaluado documentos.

Dentro del estudio de los resultados de las pruebas, hay otro apartado importante relacionado con las evaluaciones de los usuarios. Al igual que se hizo en las otras pruebas se ha realizado un estudio sobre los porcentajes de acierto de METIORE en sus predicciones sobre las preferencias de los usuarios con respecto a los documentos evaluados. En el caso de la base de datos AH2002, y para probar que el funcionamiento de METIORE es independiente del tipo de evaluación, se ha utilizado un modelo de evaluación diferente al de la base de datos LORIA. El modelo utilizado en este caso se corresponde con las evaluaciones del usuario: *muy interesante(ok_2), interesante(ok_1), no lo sé(?), poco interesante(wrong_1), nada interesante(wrong_2)*. En dos primeros casos, los parámetros del documento se evalúan como *ok*, recibiendo un peso mayor si se evalúa como *muy interesante*. En el caso contrario los documentos se evalúan como *wrong* recibiendo un peso mayor si la evaluación es *nada interesante*. Si un documento se evalúa como *no lo sé*, se almacena en el historial, pero no se hará ninguna suposición para hacer predicciones. Resumiendo, METIORE hará predicciones para *ok, wrong o normal* (si no se tiene otra información), aunque para la evaluación haya más opciones (*ok_2,ok_1,?,wrong_1,wrong_2*).



	ok	?	wrong
pok	60,00%	10,00%	30,00%
pnormal	77,27%	18,18%	4,55%
pwrong	33,33%	16,67%	50,00%

Fig. 48. Resultados de evaluaciones para los usuarios de METIORE-AH2002

En la Fig. 48 se muestran los resultados de las predicciones de METIORE para las evaluaciones realizadas en las pruebas de AH2002. Aunque hubiese sido muy interesante disponer de un porcentaje mayor de evaluaciones sobre el total de documentos visualizado por los usuarios, los resultados para las evaluaciones realizadas parecen interesantes. A continuación se comentan alguno de los datos más relevantes que puede extraerse de la Fig. 48:

- El 60% de las veces que el sistema predice que un documento interesará a un usuario (*pok*) el usuario evalúa ese documento como interesante

- El hecho de que un alto porcentaje (77,27%) de los documentos que no tenían ninguna predicción (estado normal), hayan sido evaluados como correctos indica que en los casos iniciales cuando no se tienen datos suficientes para saber si un documento es interesante o no, el criterio de ordenación se realiza en función de la relevancia de la consulta, por lo que es indicio de que el mecanismo de búsqueda funciona bien.
- Es interesante, al igual que pasó en las otras pruebas, constatar que los usuarios han evaluado muy pocos documentos que el sistema había predicho como incorrectos, aun así, cuando los usuarios lo han evaluado, en un 50% de las veces el sistema había predicho correctamente los resultados.

VI.5. CONCLUSIONES

Las experimentaciones son un elemento importante para poder probar, que las previsiones que se tenían para un determinado algoritmo o sistema se satisfacen. En este capítulo se ha mostrado por un lado que el algoritmo propuesto como variación de Naïve Bayes para adaptar la clasificación a la recomendación organiza los resultados de forma similar a las variaciones aceptadas hasta el momento, con el valor añadido de poder utilizar este nuevo algoritmo para múltiples parámetros, pudiendo ponderarse su aportación para el resultado final.

Por otro lado se han mostrado las pruebas realizadas para el sistema METIORE en dos situaciones diferentes. La primera utilizaba una interfaz de aplicación, la base de datos del laboratorio LORIA, unos usuarios a los que se había introducido el funcionamiento de METIORE y un sistema de evaluación en el que el usuario decía si le interesaba un documento, si le interasaba pero lo conocía, si no era capaz de dar una evaluación para ese documento o si el documento no se correspondía con lo que buscaba. También se han realizado pruebas con el mismo sistema en otro entorno, la Web, con una base de datos de publicaciones de hipermedia adaptativos, con usuarios no instruidos en el funcionamiento de METIORE, procedentes de diferentes partes del mundo y un sistema de evaluación simplificado en el que el usuario decía si el documento le parecía interesante, muy interesante, indiferente, poco interesante o nada interesante.

Con eso se ha pretendido mostrar la versatilidad del sistema para distintas bases de datos, interfaces, tipos de usuarios y formas de evaluación. En ambos casos el sistema ha sido capaz de dar resultados satisfactorios.

VII. CONCLUSIONES

Hasta comienzo de la década de los 90 los sistemas informáticos en general tenían un enfoque tradicional basado en "uno para todos", es decir, un tipo de sistema que funcionaba de la misma forma para cualquier usuario. En los comienzos de la WWW y prácticamente hasta la actualidad, esa filosofía se mantiene en los buscadores como Altavista, Yahoo o Google. La tendencia actual es la de hacer sistemas que tiendan a adaptarse cada vez más a los usuarios. Ejemplos cotidianos se tienen en las últimas versiones de los sistemas operativos *Windows* donde cada escritorio es personalizado, y cuando se va a ejecutar una aplicación, aparecen las más recientes destacadas de las demás. En los sistemas de recuperación de la información también se tiende a limitar los cientos de posibles respuestas de las cuales el usuario sólo examina unas pocas. Por lo tanto la personalización de las respuestas se muestra como un tema de investigación necesario y de actualidad.

En esta tesis se ha profundizado en el tema de personalización en los sistemas de recuperación de la información proponiendo unos métodos para poder llevarla a cabo con resultados satisfactorios como se ha probado en los experimentos realizados.

VII.1. PRINCIPALES APORTACIONES

Las principales contribuciones de este trabajo son las siguientes:

- Se ha realizado un estudio comparativo de las diferentes técnicas de recuperación de la información seleccionando las características más relevantes para proponer una arquitectura de recuperación de la información completa y eficiente.
- Se han estudiado las técnicas de modelado del usuario utilizadas en la actualidad para proveer respuesta personalizada, haciendo un estudio exhaustivo de los distintos métodos y sistemas utilizados en los sistemas recomendadores. A partir de este estudio se han visto las carencias comunes en los sistemas existentes con la idea de subsanarlas en este trabajo.
- Se ha propuesto una arquitectura de recuperación de la información que combina los modelos boléanos, vectorial y probabilístico que permiten un amplio rango de tipos de consultas que permiten tanto a usuarios principiantes como expertos sacar el

máximo provecho de los datos contenidos en el sistema. Además se permite al usuario trabajar con su historial y analizar documentos consultados anteriormente. La ventaja de esta arquitectura es su posible aplicación diferentes tipos de sistemas multimedia y documentales.

- Se han desarrollado algoritmos originales, inspirados en otros existentes como *Naïve Bayes*, para mostrar resultados personalizados al usuario. La propuesta mejora los algoritmos existentes en el sentido de que se pueden agrupar los datos que describen los objetos con múltiples parámetros ponderando la importancia de cada parámetro como sea necesario.
- En el análisis de los sistemas recomendadores existentes la mayor carencia observada ha sido el uso de un modelo de usuario único que servia para recomendar documentos en cualquier situación. Una visión novedosa de esta tesis es la de centrar el modelado del usuario alrededor del concepto de objetivo. Dependiendo del objetivo actual del usuario las recomendaciones que se le proponen serán diferentes.
- Se ha propuesto una arquitectura alternativa a los usuarios basada en la cooperación a través de la red con otros usuarios que puedan ayudarle a solucionar los problemas que encuentren en su tarea de búsqueda.
- Por último se ha implementado la arquitectura propuesta en el sistema METIORE y se ha evaluado su funcionamiento con usuarios en dos entornos y bases de datos diferentes. En ambos, casos los resultados de las recomendaciones del sistema han sido interesantes.

VII.2. TRABAJOS FUTUROS

En cuanto a los trabajos futuros quedan varias líneas abiertas muy interesantes que se muestran a continuación:

- La aplicación de este trabajo para la recomendación de documentos Web, es un paso casi inmediato, de hecho ya se han hecho algunas propuestas en esta línea [Bueno2001], donde se propone una arquitectura que tiene como base esta tesis y que permite a diferentes usuarios compartir los documentos encontrados con otros que tengan objetivos similares. La idea consiste en hacer un buscador Web personalizado, orientado a objetivos y cooperativo. Es importante estudiar cómo puede ayudar el historial de otros usuarios con objetivos similares al usuario actual, una posibilidad es el razonamiento basado en casos propuesto en [David2001], o la correlación de modelos de usuarios.
- El uso de un modelo de usuario inspeccionable puede ser interesante para que el usuario puede modificar las predicciones del sistema para adaptarlas a su verdadero objetivo. Dependiendo de los conocimientos de los usuarios, permitirles modificar su modelo puede ser beneficioso o peligroso, pues malas decisiones en esas modificaciones pueden llevar al sistema a recomendarle de forma incorrecta.
- La aplicación de las técnicas aquí propuestas a periódicos personalizados o tiendas virtuales es también atractiva y de fácil aplicación. Los periódicos ofrecen muchas noticias que a un usuario no interesan. Utilizando sus evaluaciones o las noticias que lee, se puede generar un modelo donde se almacenen sus gustos. Cuando se conecte al periódico, encontrará las noticias más interesantes para él en la primera página, o se les enviarán por correo cada mañana. Por otro lado, las tiendas virtuales pueden sacar provecho de las compras de los usuarios, para obtener un perfil de sus

preferencias y recomendarle productos relacionados con los que ha comprado o por los que ha mostrado interés. Ambas propuestas pueden llevarse a cabo utilizando los algoritmos aquí propuestos: NBM o WNBM.

- La cooperación en línea de varios usuarios se ha introducido, aunque es un elemento en el que se puede profundizar por ejemplo para hacer búsquedas simultáneas entre varios usuarios con un objetivo común.

APÉNDICE I. ABREVIATURAS UTILIZADAS

En este apéndice se muestran las distintas abreviaturas utilizadas en este documento además de aquellos términos que podría resultar confusos. El formato general vendrá dado por su significado en español y opcionalmente entre paréntesis y cursiva su significado en ingles, si el acrónimo está en este idioma.

BIR Es un sistema de recuperación probabilístico que representa un documento como un vector booleano de términos y asume que los términos de un documento son independientes entre sí (*Binary Independence Retrieval*)

CIRS Sistema Cooperativo de Recuperación de la información (*Cooperative Information Retrieval System*)

Documento Se llama documento a cualquier objeto que puede utilizarse como elemento principal en un sistema de recuperación de la información. Por ejemplo, un documento textual (articulo, libro, revista, correo electrónico, noticia...) o un documento multimedia (imagen, video, sonido,...). Se corresponde con el término *item* del inglés. No se ha utilizado su traducción al español 'ítem' pues puede confundirse con los elementos que forman parte de un dato (según definición del diccionario de la real academia española).

DTD Definición de tipo de documento (*Document Type Definition*). Es una gramática que se utiliza para validar documentos XML.

IDF Hace referencia a que la importancia de un término debe ser inversamente proporcional a la frecuencia de aparición de ese término en la base de datos (*Inverse Document Frequency*)

IF Filtrado de la Información (*Informatión Filtering*)

IR Acrónimo de Recuperación de la Información (Information Retrieval)

IRS Se corresponden con las siglas en ingles de Sistemas de Recuperación de la Información (*Information Retrieval Systems*)

METIORE Sistema Multimedia Cooperativo de Recuperación de la información (Mult*imedia coopErative InformaTIOn Retrieval SystEm*). Es el sistema implementado para probar las propuestas de esta tesis.

NB Bayesiano Ingenuo. Método de clasificación simple basado en el teorema Bayes (*Naïve Bayes*)

NB-CESTNIK Adaptación de NB propuesta por Cestnik

NB-METIORE Adaptación de NB propuesta en esta tesis

Parámetro Un parámetro representa alguna característica de un documento. Por ejemplo, en un artículo, posibles parámetros son el autor, titulo, palabras clave, etc.

Peso Se llama peso de un término en un documento a la importancia que ese término tiene en ese documento

RF Evaluación Pertinente. (*Relevant Feedback*). Este concepto hace referencia en los sistemas de recuperación de la información al uso de las evaluaciones de los usuarios para mejorar su consulta.

SA Servidor de Aplicación. Utilizado en la arquitectura propuesta en V.4. Es la parte de la aplicación encargada de las comunicaciones.

SGDB Sistema Gestor de Base de Datos.

SL Servidor de Localización. Utilizado en la arquitectura propuesta en V.4. Es el servidor al que se conectan las aplicaciones.

Término Es el valor concreto de un parámetro de un documento.

TF Es el número de apariciones de un término dentro de un documento (*Term Frequency Inverse Document Frequency*)

TFIDF Combinación de TF e IDF para dar pesos a los terminos de un documento (*Term Frequency*

TREC Conferencia sobre Recuperación de Textos (*Text REtrieval Confenrence*)

UM Modelado del Usuario. (*User Modeling)*

XML Lenguaje de etiquetado extendido (eXtended Mark-up Language)

APÉNDICE II. ANÁLISIS BIBLIOMÉTRICOS CON METIORE

AP.II.1. INTRODUCCIÓN

El prototipo creado para poder demostrar las hipótesis de esta tesis, tiene características que le permiten ofrecer de forma sencilla resultados que no serían posibles o muy complejos en otro tipo de sistemas. Como ejemplo, en este apartado se muestran los resultados de aplicar METIORE para realizar un estudio sobre la calidad de la BDs de publicaciones del laboratorio LORIA/INRIA Lorraine. El documento completo [Kislin2000] contiene 137 páginas. Todas las tablas y gráficas que se muestran a continuación son generadas a partir de ficheros con formato Microsoft Excel que METIORE genera para cada consulta.

Existen técnicas para medir la información sobre actividades científicas y técnicas en un momento dado bajo la forma de indicadores de tendencias, que permiten, mediante el estudio de las características de las publicaciones ver la evolución sobre temas de investigación, grupos de trabajo, etc. Los resultados no son pruebas irrefutables, aunque sí índices de actividad científica.

AP.II.2. ORIGEN DE LOS DATOS

Para realizar este estudio se utiliza la base de publicaciones de LORIA. Esto permite disponer de un *corpus* de estudio significativo para poder extraer indicadores de tendencias en informática, y las principales áreas temáticas del laboratorio de investigación.

Respecto al lenguaje de las publicaciones de esta BDs., la mayoría está en inglés. Esto tiene que tenerse en cuenta a la hora de realizar las consultas. Aunque los títulos y resúmenes pueden estar en francés, todos las palabras claves (motscles/keywords) estan en inglés.

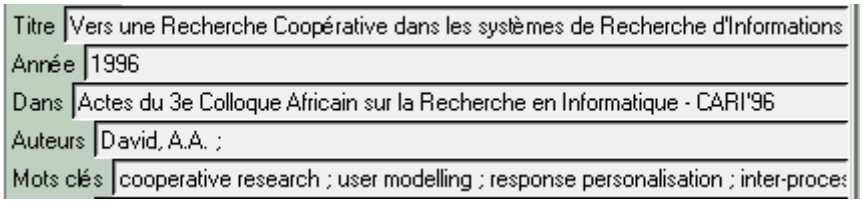


Fig. 49. Ejemplo de publicación con título en francés y palabras clave en inglés

Otros problemas encontrados en la BDs tras las múltiples consultas con METIORE son:

- **la aparición de redundancias.-** (consecuencia de un mal método de inserción de artículos). Pueden encontrarse términos diferentes que deberían ser el mismo (p.ej: distributed system/distributed systems, user modeling/user modelling, etc.)
- **Publicaciones inaccesibles por autor o palabras clave**.- En la BDs METIORE encontró 201 publicaciones sin autor (201 desconocidos/1546 autores → 13'01%). También aparecen 513 artículos que no tienen definidas palabras claves→ 13'7%
- **Identificadores diferentes para el mismo autor**.- Por ejemplo pueden encontrarse 4 referencias diferentes a la misma persona (David, M./David, Amos./David, A./David, A A.)
- **Referencias duplicadas**.- El mismo articulo aparece varias veces.

AP.II.3. ESTUDIO DE LA PRODUCCIÓN CIENTÍFICA

En 1976, De Solla Price [Price1976] construye un modelo probabilístico denominado corrientemente como la ley de *las ventajas acumuladas* que dice: "Cuanto más elementos produce una fuente de información, mayor será su oportunidad de producir". Las propiedades matemáticas de esas distribuciones estadísticas son estudiadas con el nombre 'Zipfiennes'. Para el caso de estudio, la fuente serán los autores y los elementos las publicaciones. Probar esta ley puede ser interesante para un comité de dirección, pues le ayudará a saber cuales son los investigadores más eficientes dentro de un centro.

METIORE pueda a ayudar a esta tarea realizando la siguiente consulta:

```
ATRIBUTO1= Author ; Restricciones = ""
```

Que dará como resultado una lista ordenada de forma decreciente de los autores y el número de publicaciones.

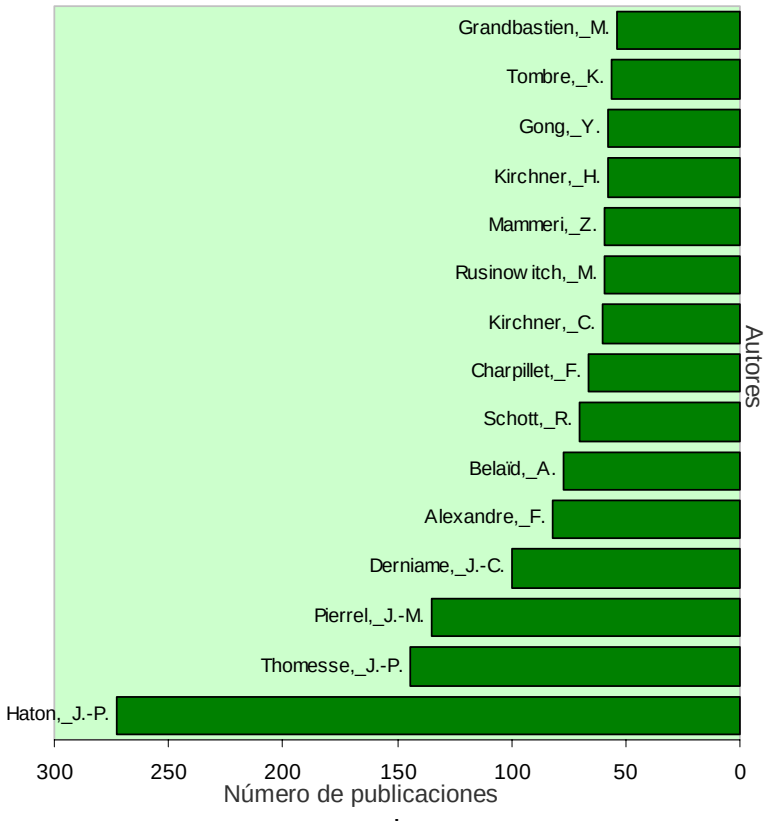


Fig. 50. Autores con mayor número de publicaciones en el laboratorio LORIA

La bibliometría utiliza técnicas basadas en el estudio de trabajos científicos con las mismas particularidades. Por ejemplo se pueden querer conocer la cantidad de trabajos realizados por un individuo, publicaciones de una fecha concreta, publicaciones realizadas por un autor, organismo o difundidas por una revista científica. Todas estas propiedades que caracterizan los documentos se encuentran en sus referencias bibliográficas. Ordenando los elementos comunes a uno de esos aspectos, se encuentra que todos ellos siguen prácticamente la misma forma de distribuirse, y por lo tanto tienen el mismo aspecto gráfico. Estas distribuciones se podrían dividir en dos partes. El núcleo (*core*) y la dispersión (scatter). Estos nombres son sugeridos por su representación gráfica. Fig. 51

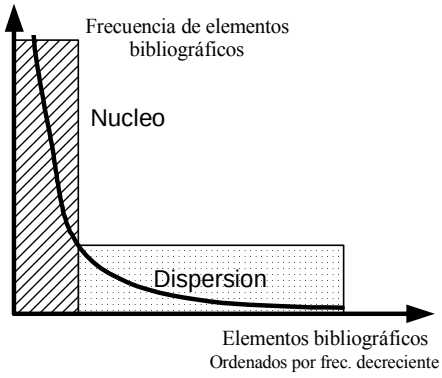


Fig. 51. Distribución Núcleo-Dispersión

El núcleo representa el grupo de elementos que aparecen con más frecuencia en el conjunto de referencias estudiadas. La dispersión representa los otros numerosos elementos de baja frecuencia de ese conjunto de referencias.

Los *colegios invisibles* [Price1966] muestran el fenómeno de núcleo-dispersión como un ejemplo de perversión de la ciencia. Se suelen encontrar dos formas de comportamiento entre los investigadores:

1. La creación de un núcleo debido a que sólo se investiga, lee y habla con el mismo grupo de personas. Esto genera una especie de monopolio de una revista o de una cierta terminología, haciendo muy difícil la inclusión de nuevos investigadores en ese grupo.
2. La creación del efecto de dispersión debido a las novedades y a las diferentes personas e ideas, que provocan una extensión continua de la red de trabajo.

En la Fig. 52 se muestra la distribución 'zipfienne' utilizando los datos del laboratorio. Se puede ver entre los dos autores con más publicaciones J.P.Haton y J.P. Tómese hay un hueco que se corresponde a publicaciones anónimas (201 publicaciones que no tienen definido un autor), eso muestra un fallo en la base de datos de las publicaciones. La gráfica de la Fig. 52 permite conocer la identidad de los autores más productivos que podría utilizarse para una actividad de vigilancia sistemática sobre productividad.

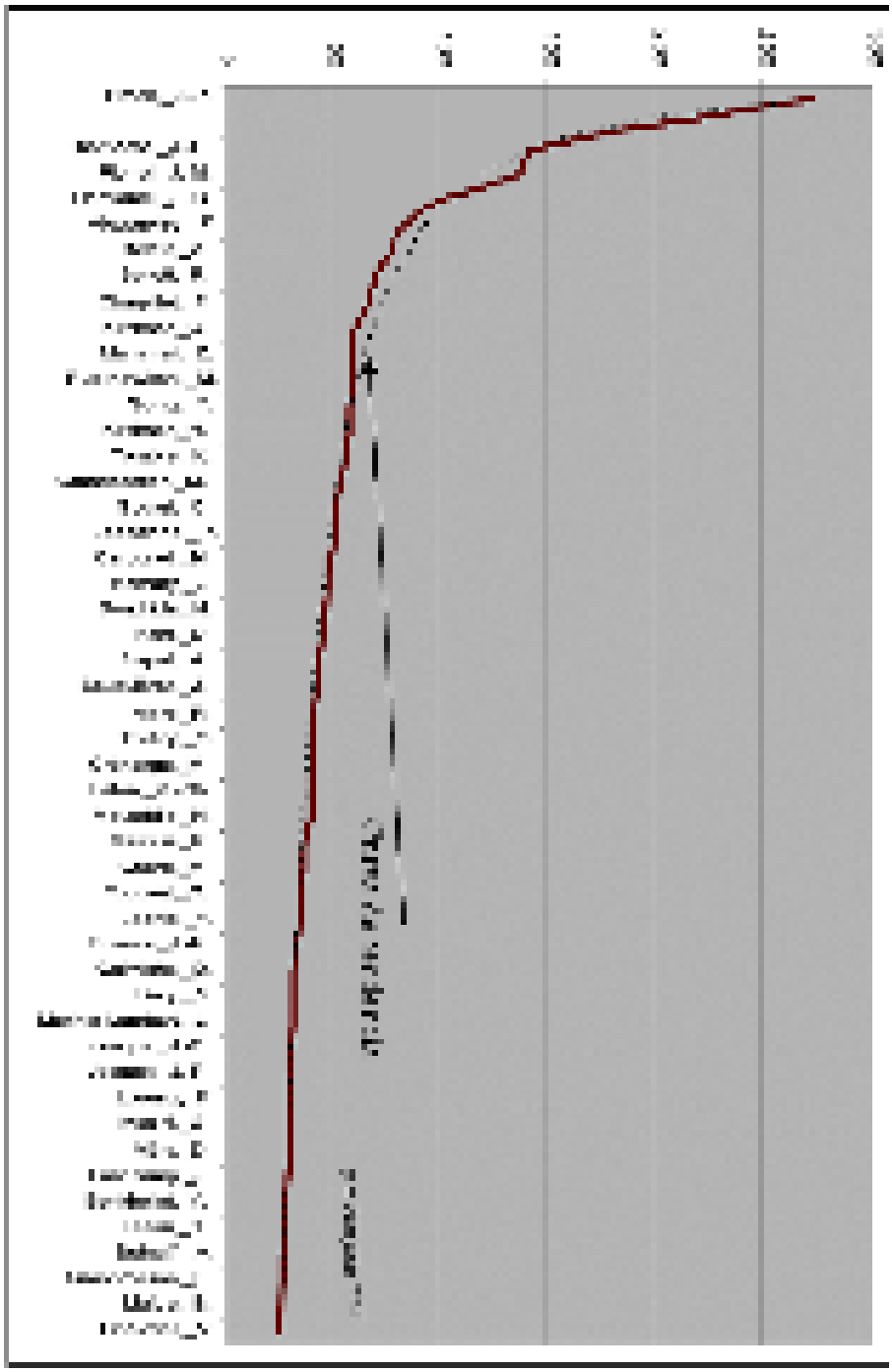


Fig. 52. Los 50 autores que ha publicado más entre 1986-1999

Si se quiere hacer un análisis sobre las publicaciones de las revistas en el laboratorio, la consulta a METIORE sería:

```
ATRIBUTO1= Journal; Restricciones = ""
```

Esa consulta devolverá las publicaciones periódicas en las que algún miembro del laboratorio ha publicado. En la Tabla 15 se muestran las 10 revistas con más publicaciones, que representan el 27'65% de todos los artículos de revistas, es decir, 86 de los 311 artículos en revistas.

Las 10 revistas más con mayor número de publicaciones 86/311	Frecuencias
Theoretical_Computer_Science	26
Speech_Communication	8
Technique_et_Science_Informatique	7
Information_and_Computation	7
Traitement_du_Signal	6
Science_of_Computer_Programming	6
J._Symbolic_Computation	6
International_Journal_of_Pattern_Recognition_and_Artificial_Intelligence	6
Journal_of_Symbolic_Computation	5
Information_Processing_Letters	5
Technique_et_Science_Informatiques	4
Total :	86
	27,65%

Tabla 15. Las 10 revistas con mayor número de publicaciones del laboratorio

Puede verse en la Tabla 15 que la revista "Technique et Science Informatique" aparece de esa forma y como "Technique et Science Informatiques", debido esto de nuevo a fallos del personal encargado de actualizar la base de datos.

Continuando con el análisis parece también interesante saber cuáles son los autores que han publicado en la revista "Theoretical Computer Science", para eso la consulta a METIORE sería:

```
ATRIBUTO1= Journal; ATRIBUTO2=Autor;
Restricciones =  {Journal="Theoretical Computer Science"}
```

Esto dará la lista de autores que publicaron en esa revista, junto con el número de veces que lo hicieron. En la Tabla 16 se muestran los 9 primeros autores entre los 20 que han firmado los 26 artículos de esta revista. También se podría hacer una consulta para ver cuáles son los descriptores más utilizados en los artículos de una revista. Para ello la consulta podría ser:

```
ATRIBUTO1= Keywords
Restricciones =  {Journal="Theoretical Computer Science"}
```

Autores	Frecuencia
Galmiche,_D.	3
Kirchner,_C.	2
Kirchner,_H.	2
Louchard,_G.	2
Méry,_D.	2
Randrianarimanana,_B.	3
Rusinowitch,_M.	1
Schott,_R.	4
Zimmermann,_Paul	1

Tabla 16. Autores de la revista Theoretical Computer Science

Muchas otros análisis de mayor o menor complejidad podrían realizarse cruzando los datos obtenidos. Los análisis dependerán de la tema de investigación que sea objetivo de estudio.

AP.II.4. VERIFICACIÓN DE LA LEY DE LOTKA

Además del análisis anterior en que se mostraba que las distribución de las publicaciones seguían una distribución 'zipfienne', se va a intentar verificar otra ley bibliométrica: La ley de Lotka [Lotka1926].

Lotka deseaba determinar la contribución que cada investigador hacía al progreso de la ciencia. Para aplicar sus ideas utilizó el dominio de la química. Contabilizo todos los autores que habían publicado un artículo en el índice de 'Chemical Abstract' entre 1907-1916, también los que habían publicado 2, 3,... Presentó sus resultados en forma de histograma y se dio cuenta que existía una relación inversa entre el número de publicaciones en un dominio específico y el número de sus miembros. La relación se muestra en la Ecuación 30. Según esto, tomando como relación el número de autores que publican 1 artículo, habrá 4 veces menos autores con dos ($1/2^2$), 9 veces menos autores con 3 ($1/3^2$), y en general n^2 menos autores que publiquen n. Es decir, que la productividad científica disminuye en función de una ley de tipo cuadrática inversa.

$$y = \frac{A}{X^2}$$

donde

$A = Número\ de\ autores\ que\ han\ publicado$

$X = Número\ \ publicaciones$

$y = Número\ de\ autores\ con\ X\ \ publicaciones$

Ecuación 30. Ley de Lotka

La posibilidad de demostrar esa ley utilizando METIORE es prácticamente inmediata. Esta vez será suficiente con la consulta:

```
ATRIBUTO1= Author ; Restricciones = ""
```

Tras contar el número de autores y aplicar la fórmula, se obtiene la siguiente tabla:

Nº Autores	Nº Publicaciones	Lotka
773	1	773
214	2	193
138	3	86
84	4	48
53	5	31
43	6	21
44	7	16
32	8	12
20	9	10
17	10	8

Tabla 17. Aplicación de la ley de Lotka al laboratorio

Los valores obtenidos siguen más o menos la distribución 'ideal' de Lotka. Aunque el coeficiente calculado es inferior a 2 (1,6 en lugar del cuadrado previsto por la fórmula). En la Fig. 53 se muestran gráficamente los resultados de las dos curvas calculadas.

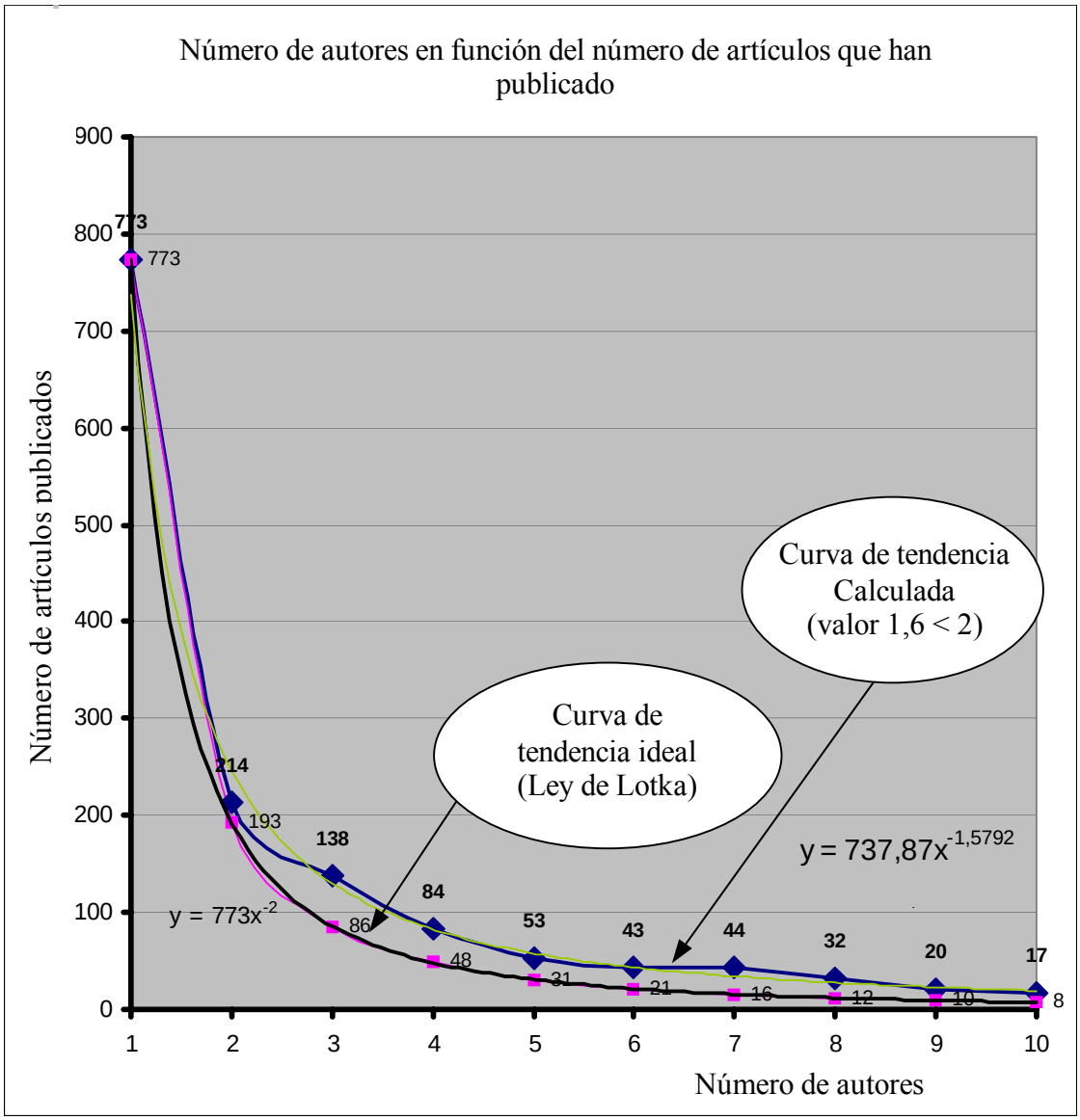


Fig. 53. Curva de la ley de Lotka y su equivalente para las publicaciones de LORIA

AP.II.5. DETERMINAR TENDENCIAS DE LA INVESTIGACIÓN EN UN DOMINIO PARTICULAR

Para este tercer ejemplo de actividad de análisis científico, se hará el estudio asociado a un autor. Se intentarán encontrar sus diferentes colaboradores, analizando los autores de sus artículos, sus temas de interés e incluso su orientación actual, es decir, hacia que áreas de investigación se dirige. Para ello se utilizarán las características de METIORE de análisis intra-campos y Inter-campos, que permiten ver la correlación de distintos atributos de la base de datos. Para comenzar se van a analizar los contenidos principales de investigación utilizando la consulta:

```
ATRIBUTO1= Author ; Restricciones = ""
```

Como resultado se obtienen las palabras claves utilizadas por orden de frecuencia decreciente. Esto genera una curva 'Zipfienne' en la que podrían identificarse 3 zonas, de las cuales son interesantes la zona de altas frecuencias y la de bajas:

- **Términos triviales.-** Son aquellos utilizados con más frecuencia. Representan las grandes orientaciones del laboratorio. Según informaciones del laboratorio, las dos areas son:
  1. *Teorías y Técnicas de producción de software* (Desarrollo de métodos, de lenguajes y de herramientas que permitan desarrollar y mantener software seguro y eficaz) y
  2. *Comunicación Hombre-Máquina e Inteligencia Artificial* (Definición de nuevos modelos que permitan adquirir las capacidades de razonamiento y de comunicación de ordenadores con el hombre y probarlo en entornos realistas).

  Utilizando METIORE se han podido verificar como ciertas esas dos áreas ya que entre los términos más frecuentes (Fig. 54.a) aparecen relacionados con la primera directriz del laboratorio palabras clave como: *(specification, parallelism, classification, connectionism, rewriting, software engineer, ...)*. Por otro lado, asociados a la 2ª directriz tambien aparecen con mucha frecuencia los términos (speech-recognition, artificial intelligence, expert systems, man-machine dialogue,...).

speech_recognition	166
artificial_intelligence	99
specification	83
expert_system	80
man-machine_dialogue	68
document_analysis	67
rewriting	66
computer_vision	61
software_engineering	55
image_processing	52
natural_language	50
real-time	48
knowledge_representation	43
fieldbus	40
parallelism	40
classification	38
neural_network	38
theorem_proving	37
connectionism	36
completion	35
cooperation	34
scheduling	34
FIP	33
unification	33
knowledge-based_systems	32

distributed_systems	8
equational_theories	8
explanation	8
folding	8
genetic_algorithms	8
hierarchical_systems	8
Hough_transform	8
image_analysis	8
interaction	8
interoperability_verification	8
multi-expert_systems	8
networks	8
normalisation	8
operator	8
program_development	8
rewriting_logic	8
technical_document	8
time_constraint	8
unfolding	8
viewpoint	8
vision	8

Fig. 54 Lista de palabras claves utilizadas a) Más frecuentes b) Menos frecuentes

- Términos 'emergentes' o 'marginales'.- En la zona baja de frecuencias, los términos que aparecen pueden pertenecer a dos grupos. Por un lado, aquellos de reciente uso, que están comenzando a aparecer en la terminología científica y que pueden indicar nuevas líneas de investigación. Por otro lado, también aparecen términos que podrían indicar temas secundarios o áreas de investigación no muy utilizadas que han sido abandonadas. Para identificar a que grupo de los dos pertenece un términos, se podría utilizar la consulta con atributos= keyword, año. Esto permitiría ver si esas palabras de baja frecuencia son recientes o no.

Con METIORE se puede analizar información sobre una persona en particular. En la base de datos de LORIA, parece interesante estudiar al autor más 'productivo', con 273 publicaciones entre 1984-1999 (Ver Fig. 50). Para ello se empezará con la consulta:

```
ATRIBUTO1= Keyword ; Restricciones = {Autor=Haton}
```

Como resultado, las primeras palabras claves que se obtienen son: 73 speech recognition; 54 artificial intelligence; 24 expert systems. Si esto se compara con la lista de la Fig. 54.a), puede verse que las 3 palabras claves más utilizadas por Haton aparecen entre los 4 descriptores más utilizados en el laboratorio, por lo tanto este autor parece ser una referencia para los grandes temas del laboratorio. Por eso sería interesante saber quienes trabajan con él. Según los estudios de Peters y Van Raan [Peter1991], el análisis de los autores que firman los mismos artículos (coautores) permite:

- Identificar las relaciones intelectuales y/o cohesiones sociales entre individuos
- Conocer las evoluciones de esos grupos, mediante comparaciones en el tiempo
- Identificar las especialidades emblemáticas o pivotes
- E identificar los lideres de cada especialidad

Para realizar este tipo de análisis se consultará a METIORE:

```
ATRIBUTO1= Author;ATRIBUTO2=Author ;
Restricciones = {Autor=Haton}
```

Autores	Frecuencia
Charpillet, F.	53
Haton, M,Ch. (su mujer)	49
Gong, D.	47
Pierrel, J.M.	32
Fohr, D.	22
Alexandre, F.	20
Carbonell, N.	16

Tabla 18. Coautores de J.P.Haton con mayor número de publicaciones

En la Tabla 18 se muestran los autores que más han publicado con J.P.Haton, aunque hay muchos más autores que han realizado algún tipo de colaboración en un número menor de artículos. Como puede verse, en la tabla anterior, se ha encontrado un pequeño grupo. Si nos interesa saber exactamente en que ha colaborado con el autor, se puede utilizar la consulta:

```
ATRIBUTO1= Author;ATRIBUTO2=Author ;
ATRIBUTO3= Keyword;Restricciones = {Autor=Haton}
```

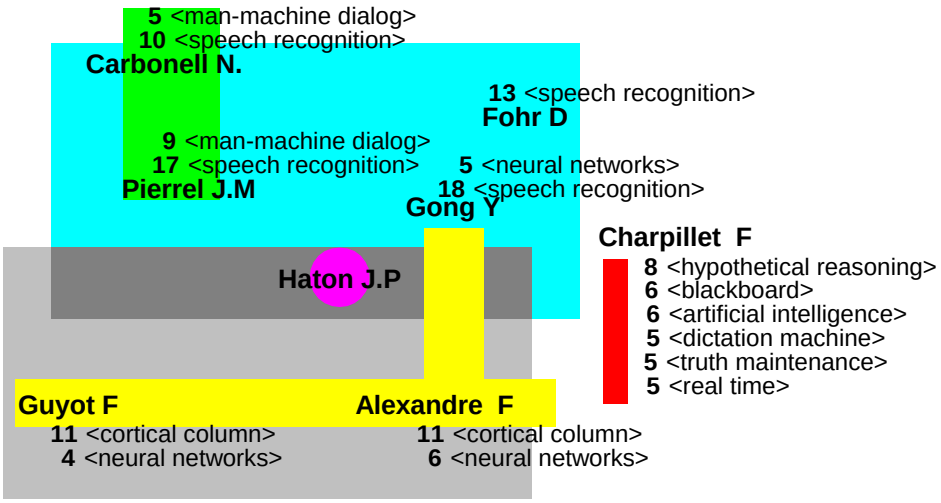


Fig. 55. Coautores de Haton y términos más utilizados

La consulta anterior es bastante compleja. Se piden todos los autores que han escrito con Haton y además cuales son los temas principales de sus artículos. En la Fig. 55 se muestran los resultados obtenidos, eligiendo las palabras más utilizadas y realizando una distribución gráfica por temas. Puede verse que el grupo de coautores de Haton a su vez está compuesto por subgrupos organizados por autores con las mismas inquietudes de investigación. Por ejemplo, N. Carbonell y J.M. Pierrel se interesan por el reconocimiento de voz y la comunicación hombre-máquina. Por otro lado F.Charpillet, aunque ha publicado bastante con Haton, no parece estar 'relacionado' con los otros coautores, por los temas que utiliza.

Para verificar algunas de las hipótesis expuestas, se han analizado los informes de actividad del laboratorio de 1996/1998, donde se ha podido verificar que todos los componentes pertenecen al mismo equipo RFIA (Modelos fundamentales y aplicaciones de procesos perceptivos y cognitivos). Los otros autores minoritarios eran mayormente estudiantes de doctorado que hacían la tesis en este equipo.

Otra consulta que permite analizar las evoluciones en el tiempo respecto a los temas de investigación de este autor sería la siguiente:

```
ATRIBUTO1= Author;ATRIBUTO2=Keyword ;
ATRIBUTO3= Year;Restricciones = {Autor=Haton}
```

Esa consulta a METIORE ofrece una distribución de términos utilizados en cada año por ese autor, que permite saber cuales son los *temas recurrentes* (los temas que siempre ha utilizado y utiliza en sus publicaciones, que representan líneas de investigación estables). También permite saber cuales son los *temas obsolescentes* (aquellos que se utilizaron durante un tiempo, pero que han dejado de tener interes) y por último los temas emergentes (que aparecen en los últimos años). Estas distribuciones pueden verse claramente en la Fig. 56.

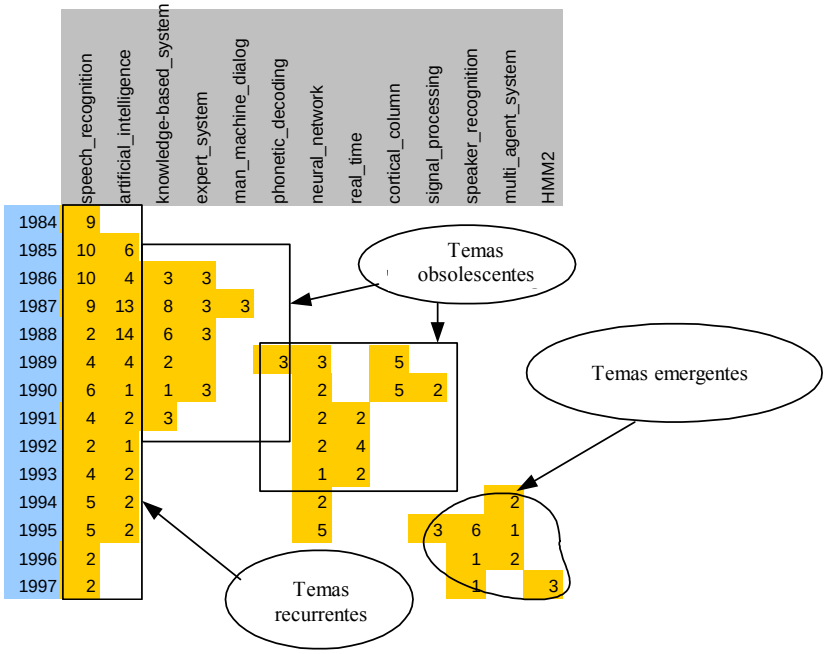


Fig. 56. Distribución de publicaciones de J.P. Haton entre 1984-1997

APÉNDICE III. DATOS DE LOS EXPERIMENTOS

AP.III.1. INTRODUCCIÓN

En este apéndice se presenta una muestra de los datos de los experimentos. En primer lugar se detallan los datos utilizados para poder comparar las diferentes versiones de Naïve Bayes. En segundo lugar, es muestra una porción de los ficheros de registro de METIORE donde se muestran las sesiones de dos usuarios y las actividades que estos realizan.

AP.III.2. COMPARATIVA NAÏVE BAYES

A continuación se muestran los datos utilizados para comparar los tres algoritmos Naïve Bayes que se comentaron en el capítulo 0. Para comparar los tres algoritmos se ha utilizado una hoja de cálculo que se muestra a continuación y de la que se van a comentar sus diferentes campos.
Las entradas de la tabla *Valor1,Valor2* y *Valor3* representan los valores $Q(ok,V_i)$ correspondiente a las ecuaciones de Naïve Bayes(NB) y Naïve Bayes METIORE (NBM en la tabla), que se han calculado como se muestra en la Ecuación 29 y para lo que ha sido necesario calcular los valores de $P(V_{iJi})$, representados en la tabla como casos posibles (*CP1,CP2 y CP3*), calculados de forma aleatoria como un número entre 0 y 20. Dichos casos posibles son los mismos para todos los documentos. También se ha tenido que calcular para cada atributo y documento el valor $P(V_{iJi}|ok)$, representado como casos favorables (*CF1,CF2 y CF3*). Que se calculan como un número aleatorio entre 0 y el número de casos posibles de cada variable. Para la modificación de Cestnik los valores correspondientes a los $Q'(ok,J_i)$ (*Valorp1, Valorp2 y Valorp3)*, se suavizan con un factor para los casos en que CF_i sea cercano a cero como se muestra en la Ecuación 29a. El valor de la recomendación que haría cada algoritmo utilizando los valores que correspondan se muestra en la Ecuación 29.b) y en la tabla como (*NB, NB_Cest y NBM)*. Por último, en las otras 3 columnas se muestra la posición que cada uno de los 3 algoritmos ha dado a dicho documento. Por ejemplo, en la columna NB se tiene el valor que Naïve Bayes daría a cada documento. Utilizando esa columna, se ordenan todos los documentos y se rellena la columna correspondiente Ord. NB. Lo mismo se hace para los otros dos algoritmos.

$$Aleatorio() \rightarrow [0,1]$$
$$P(V_{i,J_i}) = CP_i = 20 * Aleatorio()$$
$$P(V_{i,J_i} \mid ok) = CF_i = CPi * Aleatorio()$$
$$Q(ok, J_i) = Valor_i = \frac{CF_i}{CP_i}$$
$$Q'(ok, J_i) = Valorp_i = \frac{CF_i + 1}{CP_i + 2}$$

a)

$$NB = \prod_{i=1}^{3} Q(ok, J_i)$$
$$NB\_Cest = \prod_{i=1}^{3} Q'(ok, J_i)$$
$$NBMetiore = \frac{\sum_{i=1}^{3} Q(ok, J_i)}{3}$$

b)

Ecuación 31. Ecuaciones utilizadas para el experimento

De los 1000 objetos analizados, se han seleccionado los ordenados según NBM en las posiciones 1-60, 500-530, 700-730, 939-996. Esta selección se ha realizado con el objetivo de mostrar como se distribuían los valores para cada uno de los algoritmos, dando una visión de los objetos peor valorados, de la zona media y de los mejor valorados.

CP1	CP2	CP3
1,57	2,34	10,31

CF1	CF2	CF3	Valor1	Valor2	Valor3	Valorp1	Valorp2	Valorp3	NB	NB_Cest	NBM	Ord. NB	Ord.Cest	Ord.NBM
0,02	0,17	0,52	0,01	0,07	0,05	0,29	0,27	0,12	0,00004	0,00948	0,04423	2	3	1
0,24	0,02	0,30	0,15	0,01	0,03	0,35	0,24	0,11	0,00004	0,00863	0,06390	3	1	2
0,26	0,04	0,28	0,17	0,02	0,03	0,35	0,24	0,10	0,00008	0,00882	0,07118	5	2	3
0,05	0,11	1,76	0,03	0,05	0,17	0,30	0,26	0,22	0,00029	0,01697	0,08449	12	21	4
0,27	0,17	0,37	0,17	0,07	0,04	0,36	0,27	0,11	0,00045	0,01067	0,09423	16	5	5
0,07	0,43	0,53	0,05	0,18	0,05	0,30	0,33	0,12	0,00044	0,01236	0,09431	15	7	6
0,19	0,04	1,86	0,12	0,02	0,18	0,33	0,24	0,23	0,00037	0,01853	0,10590	13	31	7
0,30	0,01	1,33	0,19	0,00	0,13	0,37	0,23	0,19	0,00011	0,01610	0,10933	7	15	8
0,12	0,57	0,15	0,08	0,24	0,01	0,31	0,36	0,09	0,00028	0,01065	0,11225	10	4	9
0,17	0,40	0,68	0,11	0,17	0,07	0,33	0,32	0,14	0,00122	0,01445	0,11496	25	10	10
0,35	0,08	1,01	0,22	0,04	0,10	0,38	0,25	0,16	0,00077	0,01537	0,11877	20	12	11
0,28	0,19	1,10	0,18	0,08	0,11	0,36	0,27	0,17	0,00154	0,01673	0,12151	32	20	12
0,20	0,31	1,22	0,13	0,13	0,12	0,34	0,30	0,18	0,00201	0,01833	0,12628	41	28	13
0,23	0,42	0,62	0,15	0,18	0,06	0,35	0,33	0,13	0,00160	0,01487	0,12935	34	11	14
0,07	0,44	1,65	0,05	0,19	0,16	0,30	0,33	0,22	0,00134	0,02135	0,13032	26	41	15
0,21	0,45	0,79	0,13	0,19	0,08	0,34	0,33	0,15	0,00199	0,01651	0,13478	40	19	16
0,10	0,41	1,81	0,07	0,17	0,18	0,31	0,32	0,23	0,00201	0,02290	0,13854	42	46	17
0,59	0,02	0,34	0,38	0,01	0,03	0,45	0,23	0,11	0,00010	0,01145	0,14021	6	6	18
0,36	0,41	0,24	0,23	0,18	0,02	0,38	0,33	0,10	0,00094	0,01247	0,14240	23	8	19
0,30	0,14	1,83	0,19	0,06	0,18	0,37	0,26	0,23	0,00210	0,02207	0,14380	44	44	20
0,29	0,45	0,57	0,19	0,19	0,06	0,36	0,33	0,13	0,00197	0,01539	0,14423	38	13	21
0,31	0,42	0,85	0,20	0,18	0,08	0,37	0,33	0,15	0,00286	0,01791	0,15191	67	26	22
0,12	0,31	2,76	0,08	0,13	0,27	0,31	0,30	0,31	0,00277	0,02895	0,15925	62	83	23
0,04	0,26	3,94	0,02	0,11	0,38	0,29	0,29	0,40	0,00096	0,03372	0,17156	24	118	24
0,53	0,12	1,35	0,34	0,05	0,13	0,43	0,26	0,19	0,00232	0,02112	0,17354	48	40	25
0,16	0,82	0,80	0,10	0,35	0,08	0,33	0,42	0,15	0,00278	0,01992	0,17617	63	37	26
0,06	0,78	1,72	0,04	0,33	0,17	0,30	0,41	0,22	0,00219	0,02693	0,17963	47	72	27
0,68	0,11	0,68	0,43	0,05	0,07	0,47	0,26	0,14	0,00139	0,01642	0,18220	29	17	28
0,27	0,55	1,62	0,17	0,23	0,16	0,36	0,36	0,21	0,00622	0,02690	0,18663	119	71	29

CF1	CF2	CF3	Valor1	Valor2	Valor3	Valorp1	Valorp2	Valorp3	NB	NB_Cest	NBM	Ord. NB	Ord.Cest	Ord.NBM
0,04	0,32	4,13	0,03	0,14	0,40	0,29	0,30	0,42	0,00155	0,03701	0,18812	33	136	30
0,19	0,43	2,72	0,12	0,19	0,26	0,33	0,33	0,30	0,00590	0,03328	0,18995	116	112	31
0,41	0,64	0,41	0,26	0,27	0,04	0,39	0,38	0,11	0,00279	0,01701	0,19050	65	23	32
0,17	0,13	4,26	0,11	0,05	0,41	0,33	0,26	0,43	0,00245	0,03638	0,19215	52	134	33
0,08	0,88	1,57	0,05	0,37	0,15	0,30	0,43	0,21	0,00296	0,02734	0,19262	72	74	34
0,48	0,59	0,38	0,31	0,25	0,04	0,42	0,37	0,11	0,00286	0,01706	0,19905	68	24	35
0,12	0,80	1,92	0,08	0,34	0,19	0,31	0,41	0,24	0,00494	0,03090	0,20163	100	95	36
0,15	0,02	5,27	0,10	0,01	0,51	0,32	0,23	0,51	0,00042	0,03854	0,20504	14	148	37
0,45	0,03	3,28	0,29	0,01	0,32	0,41	0,24	0,35	0,00135	0,03358	0,20600	27	115	38
0,16	0,75	2,06	0,10	0,32	0,20	0,33	0,40	0,25	0,00665	0,03270	0,20832	128	106	39
0,02	1,07	1,66	0,01	0,46	0,16	0,29	0,48	0,22	0,00091	0,02941	0,20972	21	85	40
0,32	0,90	0,46	0,20	0,38	0,04	0,37	0,44	0,12	0,00346	0,01913	0,21002	81	34	41
0,14	0,93	1,51	0,09	0,40	0,15	0,32	0,44	0,20	0,00508	0,02891	0,21023	103	82	42
0,15	0,78	2,11	0,09	0,33	0,21	0,32	0,41	0,25	0,00644	0,03331	0,21033	124	113	43
0,40	0,06	3,65	0,25	0,03	0,35	0,39	0,25	0,38	0,00248	0,03628	0,21180	54	133	44
0,15	1,01	1,15	0,10	0,43	0,11	0,32	0,46	0,17	0,00460	0,02606	0,21321	93	67	45
0,67	0,43	0,32	0,43	0,18	0,03	0,47	0,33	0,11	0,00240	0,01646	0,21362	49	18	46
0,46	0,78	0,15	0,30	0,33	0,01	0,41	0,41	0,09	0,00138	0,01561	0,21390	28	14	47
0,58	0,52	0,53	0,37	0,22	0,05	0,44	0,35	0,12	0,00420	0,01923	0,21408	91	35	48
0,16	1,00	1,20	0,10	0,43	0,12	0,33	0,46	0,18	0,00510	0,02679	0,21512	104	70	49
0,61	0,21	1,71	0,39	0,09	0,17	0,45	0,28	0,22	0,00578	0,02769	0,21567	114	76	50
0,10	0,19	5,22	0,06	0,08	0,51	0,31	0,27	0,51	0,00259	0,04267	0,21688	58	163	51
0,31	0,36	3,13	0,20	0,15	0,30	0,37	0,31	0,34	0,00917	0,03851	0,21795	153	147	52
0,07	0,16	5,62	0,04	0,07	0,54	0,30	0,27	0,54	0,00165	0,04309	0,21924	35	167	53
0,34	0,82	0,93	0,22	0,35	0,09	0,38	0,42	0,16	0,00689	0,02468	0,21967	130	59	54
0,09	1,17	1,09	0,06	0,50	0,11	0,31	0,50	0,17	0,00313	0,02595	0,22095	77	66	55
0,32	0,77	1,32	0,21	0,33	0,13	0,37	0,41	0,19	0,00868	0,02849	0,22105	147	80	56
0,74	0,14	1,41	0,47	0,06	0,14	0,49	0,26	0,20	0,00391	0,02511	0,22339	85	63	57
0,62	0,66	0,01	0,39	0,28	0,00	0,45	0,38	0,08	0,00013	0,01424	0,22549	8	9	58
0,50	0,48	1,61	0,32	0,21	0,16	0,42	0,34	0,21	0,01018	0,03033	0,22631	167	91	59
0,54	0,58	0,95	0,34	0,25	0,09	0,43	0,36	0,16	0,00789	0,02491	0,22828	139	61	60

CF1	CF2	CF3	Valor1	Valor2	Valor3	Valorp1	Valorp2	Valorp3	NB	NB_Cest	NBM	Ord. NB	Ord.Cest	Ord.NBM

1,44	0,55	3,72	0,92	0,24	0,36	0,68	0,36	0,38	0,07810	0,09377	0,50555	530	427	500
0,71	1,55	4,12	0,45	0,66	0,40	0,48	0,59	0,42	0,12035	0,11740	0,50564	631	536	501
1,35	0,92	2,71	0,86	0,39	0,26	0,66	0,44	0,30	0,08895	0,08782	0,50574	564	399	502
0,53	0,85	8,42	0,34	0,36	0,82	0,43	0,43	0,76	0,10054	0,13993	0,50617	589	633	503
1,25	0,23	6,46	0,80	0,10	0,63	0,63	0,28	0,61	0,04866	0,10791	0,50645	421	501	504
1,30	0,75	3,83	0,83	0,32	0,37	0,64	0,40	0,39	0,09841	0,10182	0,50659	585	474	505
0,35	2,11	4,13	0,22	0,90	0,40	0,38	0,72	0,42	0,07966	0,11266	0,50753	535	519	506
0,11	1,48	8,46	0,07	0,63	0,82	0,31	0,57	0,77	0,03797	0,13724	0,50872	364	620	507
0,88	0,67	6,96	0,56	0,29	0,68	0,53	0,39	0,65	0,10949	0,13161	0,50879	612	600	508
0,92	0,08	9,39	0,59	0,03	0,91	0,54	0,25	0,84	0,01709	0,11229	0,50943	235	515	509
0,12	1,37	8,94	0,08	0,58	0,87	0,31	0,55	0,81	0,03952	0,13846	0,50947	374	628	510
0,03	2,27	5,58	0,02	0,97	0,54	0,29	0,75	0,53	0,01067	0,11645	0,51027	175	532	511
1,50	0,50	3,72	0,96	0,21	0,36	0,70	0,35	0,38	0,07348	0,09266	0,51037	513	422	512
0,62	0,53	9,41	0,39	0,23	0,91	0,45	0,35	0,85	0,08105	0,13485	0,51046	541	610	513
1,50	1,32	0,08	0,96	0,56	0,01	0,70	0,53	0,09	0,00414	0,03290	0,51082	90	108	514
1,38	1,10	1,90	0,88	0,47	0,18	0,67	0,48	0,24	0,07629	0,07605	0,51119	525	339	515
1,21	1,38	1,81	0,77	0,59	0,18	0,62	0,55	0,23	0,07961	0,07737	0,51149	534	345	516
0,98	1,17	4,20	0,63	0,50	0,41	0,56	0,50	0,42	0,12781	0,11742	0,51171	649	537	517
0,78	1,13	5,73	0,50	0,48	0,56	0,50	0,49	0,55	0,13347	0,13379	0,51196	659	606	518
0,42	0,95	8,88	0,27	0,41	0,86	0,40	0,45	0,80	0,09386	0,14369	0,51202	578	645	519
1,18	0,63	5,30	0,76	0,27	0,51	0,61	0,38	0,51	0,10431	0,11754	0,51275	595	539	520
1,26	1,36	1,62	0,80	0,58	0,16	0,63	0,54	0,21	0,07317	0,07318	0,51312	511	326	521
1,50	1,30	0,30	0,95	0,56	0,03	0,70	0,53	0,11	0,01523	0,03905	0,51318	222	150	522
0,19	1,19	9,37	0,12	0,51	0,91	0,33	0,50	0,84	0,05734	0,14226	0,51359	454	641	523
0,68	0,44	9,53	0,43	0,19	0,92	0,47	0,33	0,85	0,07512	0,13334	0,51459	520	605	524
0,93	0,21	8,87	0,59	0,09	0,86	0,54	0,28	0,80	0,04582	0,12093	0,51479	408	552	525
0,80	0,29	9,35	0,51	0,13	0,91	0,51	0,30	0,84	0,05832	0,12660	0,51488	459	578	526
1,04	0,29	7,84	0,66	0,13	0,76	0,57	0,30	0,72	0,06319	0,12216	0,51574	482	559	527
1,40	1,27	1,17	0,89	0,54	0,11	0,67	0,52	0,18	0,05496	0,06199	0,51628	449	264	528
1,26	1,01	3,25	0,81	0,43	0,32	0,63	0,46	0,35	0,10938	0,10132	0,51734	611	471	529
0,98	0,54	7,18	0,63	0,23	0,70	0,56	0,36	0,66	0,10126	0,13116	0,51808	591	597	530

CF1	CF2	CF3	Valor1	Valor2	Valor3	Valorp1	Valorp2	Valorp3	NB	NB_Cest	NBM	Ord. NB	Ord.Cest	Ord.NBM

1,10	1,58	4,27	0,70	0,67	0,41	0,59	0,59	0,43	0,19605	0,14976	0,59675	770	665	700
1,00	1,08	7,11	0,64	0,46	0,69	0,56	0,48	0,66	0,20361	0,17725	0,59688	779	768	701
1,31	1,33	4,01	0,83	0,57	0,39	0,65	0,54	0,41	0,18446	0,14131	0,59731	754	638	702
0,50	2,16	5,69	0,32	0,92	0,55	0,42	0,73	0,54	0,16143	0,16606	0,59736	710	729	703
1,22	0,60	7,81	0,78	0,25	0,76	0,62	0,37	0,72	0,15056	0,16391	0,59751	695	721	704
1,46	0,12	8,35	0,93	0,05	0,81	0,69	0,26	0,76	0,03816	0,13502	0,59798	365	611	705
1,46	1,53	2,16	0,93	0,65	0,21	0,69	0,58	0,26	0,12783	0,10331	0,59862	650	482	706
0,45	1,50	8,98	0,28	0,64	0,87	0,41	0,58	0,81	0,15887	0,18930	0,59884	708	798	707
0,61	2,26	4,57	0,39	0,97	0,44	0,45	0,75	0,45	0,16649	0,15333	0,59918	720	679	708
1,05	0,54	9,25	0,67	0,23	0,90	0,58	0,35	0,83	0,13793	0,16943	0,59921	665	740	709
0,75	1,27	8,02	0,48	0,54	0,78	0,49	0,52	0,73	0,20182	0,18778	0,59941	777	793	710
0,09	2,09	8,79	0,05	0,89	0,85	0,30	0,71	0,80	0,04145	0,17213	0,59950	388	749	711
0,35	1,71	8,75	0,22	0,73	0,85	0,38	0,62	0,79	0,13877	0,18703	0,60044	669	789	712
1,56	0,20	7,50	0,99	0,09	0,73	0,72	0,28	0,69	0,06271	0,13709	0,60244	481	618	713
1,16	1,70	3,55	0,74	0,72	0,34	0,61	0,62	0,37	0,18455	0,13898	0,60280	755	630	714
1,56	0,68	5,41	1,00	0,29	0,52	0,72	0,39	0,52	0,15110	0,14434	0,60340	697	647	715
0,49	2,01	6,60	0,31	0,86	0,64	0,42	0,69	0,62	0,17172	0,17878	0,60375	732	771	716
0,31	2,15	7,25	0,20	0,92	0,70	0,37	0,72	0,67	0,12580	0,17765	0,60456	644	770	717
1,27	1,12	5,40	0,81	0,48	0,52	0,64	0,49	0,52	0,20353	0,16178	0,60470	778	711	718
1,27	0,64	7,54	0,81	0,27	0,73	0,64	0,38	0,69	0,16243	0,16698	0,60567	714	733	719
1,41	0,16	8,77	0,90	0,07	0,85	0,68	0,27	0,79	0,05106	0,14280	0,60593	430	642	720
0,94	1,44	6,24	0,60	0,61	0,61	0,54	0,56	0,59	0,22268	0,17948	0,60615	816	774	721
0,85	1,38	7,07	0,55	0,59	0,69	0,52	0,55	0,66	0,22000	0,18662	0,60643	807	788	722
1,41	0,21	8,58	0,90	0,09	0,83	0,68	0,28	0,78	0,06595	0,14611	0,60714	493	652	723
1,03	1,03	7,50	0,65	0,44	0,73	0,57	0,47	0,69	0,20952	0,18338	0,60736	786	780	724
1,15	1,08	6,45	0,74	0,46	0,63	0,60	0,48	0,60	0,21213	0,17487	0,60748	795	758	725
0,80	1,55	6,71	0,51	0,66	0,65	0,51	0,59	0,63	0,22069	0,18587	0,60830	811	787	726
1,48	0,75	5,81	0,94	0,32	0,56	0,70	0,40	0,55	0,16908	0,15442	0,60841	727	682	727
0,74	1,88	5,67	0,47	0,80	0,55	0,49	0,66	0,54	0,20888	0,17542	0,60845	784	761	728
0,49	1,48	9,08	0,32	0,63	0,88	0,42	0,57	0,82	0,17532	0,19570	0,60891	738	811	729
1,06	1,83	3,85	0,68	0,78	0,37	0,58	0,65	0,39	0,19650	0,14786	0,60945	772	659	730

CF1	CF2	CF3	Valor1	Valor2	Valor3	Valorp1	Valorp2	Valorp3	NB	NB_Cest	NBM	Ord. NB	Ord.Cest	Ord.NBM
0,99	2,24	7,30	0,63	0,95	0,71	0,56	0,75	0,67	0,42765	0,28054	0,76513	943	940	939
1,31	1,21	9,80	0,84	0,51	0,95	0,65	0,51	0,88	0,40814	0,28822	0,76658	936	945	940
1,22	1,51	9,06	0,78	0,64	0,88	0,62	0,58	0,82	0,44093	0,29388	0,76742	947	947	941
1,50	1,46	7,48	0,96	0,62	0,73	0,70	0,57	0,69	0,43349	0,27374	0,76924	945	932	942
0,93	1,85	9,58	0,60	0,79	0,93	0,54	0,66	0,86	0,43753	0,30589	0,77174	946	955	943
0,94	1,99	9,04	0,60	0,85	0,88	0,54	0,69	0,82	0,44776	0,30558	0,77563	950	954	944
1,40	2,06	5,85	0,89	0,88	0,57	0,67	0,70	0,56	0,44404	0,26323	0,77885	949	923	945
1,35	1,25	9,67	0,86	0,54	0,94	0,66	0,52	0,87	0,43345	0,29673	0,77888	944	948	946
1,13	1,89	8,36	0,72	0,80	0,81	0,60	0,66	0,76	0,47054	0,30160	0,77888	955	951	947
0,87	2,19	8,75	0,56	0,94	0,85	0,52	0,74	0,79	0,44116	0,30530	0,77999	948	953	948
0,74	2,18	9,72	0,47	0,93	0,94	0,49	0,73	0,87	0,41524	0,31135	0,78224	938	959	949
1,36	1,68	7,83	0,87	0,72	0,76	0,66	0,62	0,72	0,47475	0,29362	0,78266	957	946	950
1,14	1,70	9,26	0,73	0,73	0,90	0,60	0,62	0,83	0,47415	0,31089	0,78375	956	958	951
1,46	2,27	4,73	0,93	0,97	0,46	0,69	0,75	0,47	0,41356	0,24145	0,78603	937	900	952
1,22	1,78	8,52	0,78	0,76	0,83	0,62	0,64	0,77	0,48722	0,30751	0,78738	959	956	953
1,18	1,53	9,86	0,76	0,65	0,96	0,61	0,58	0,88	0,46998	0,31388	0,78738	954	961	954
1,51	2,26	4,49	0,96	0,96	0,44	0,70	0,75	0,45	0,40414	0,23530	0,78750	934	894	955
1,15	1,72	9,33	0,73	0,73	0,90	0,60	0,63	0,84	0,48748	0,31679	0,79095	960	963	956
1,36	2,20	5,88	0,87	0,94	0,57	0,66	0,74	0,56	0,46428	0,27225	0,79207	953	930	957
0,83	2,24	9,20	0,53	0,96	0,89	0,51	0,75	0,83	0,45289	0,31740	0,79304	951	964	958
1,26	1,48	9,80	0,81	0,63	0,95	0,63	0,57	0,88	0,48363	0,31764	0,79589	958	965	959
1,54	1,20	9,36	0,98	0,51	0,91	0,71	0,51	0,84	0,45769	0,30382	0,80099	952	952	960
1,28	1,44	10,01	0,82	0,62	0,97	0,64	0,56	0,89	0,48957	0,32201	0,80180	961	966	961
1,45	1,58	8,37	0,93	0,67	0,81	0,69	0,59	0,76	0,50456	0,30976	0,80288	964	957	962
1,33	1,82	8,10	0,85	0,78	0,79	0,65	0,65	0,74	0,51766	0,31343	0,80354	967	960	963
1,22	1,70	9,40	0,78	0,72	0,91	0,62	0,62	0,84	0,51305	0,32612	0,80429	966	970	964
1,07	2,29	7,82	0,68	0,98	0,76	0,58	0,76	0,72	0,50777	0,31554	0,80707	965	962	965
1,24	1,95	8,32	0,79	0,83	0,81	0,63	0,68	0,76	0,53112	0,32280	0,81002	968	967	966
0,88	2,15	10,09	0,56	0,92	0,98	0,53	0,72	0,90	0,50380	0,34397	0,81872	963	978	967
1,44	1,70	8,36	0,92	0,73	0,81	0,68	0,62	0,76	0,54144	0,32360	0,81891	972	968	968
1,08	2,07	9,14	0,69	0,88	0,89	0,58	0,71	0,82	0,53878	0,33904	0,81914	970	974	969

CF1	CF2	CF3	Valor1	Valor2	Valor3	Valorp1	Valorp2	Valorp3	NB	NB_Cest	NBM	Ord. NB	Ord.Cest	Ord.NBM
1,51	1,29	9,77	0,96	0,55	0,95	0,70	0,53	0,87	0,50359	0,32489	0,82067	962	969	970
1,30	1,63	9,79	0,83	0,70	0,95	0,65	0,61	0,88	0,54953	0,34279	0,82573	973	976	971
1,44	1,66	8,84	0,92	0,71	0,86	0,68	0,61	0,80	0,55727	0,33450	0,82772	974	971	972
1,28	1,74	9,63	0,81	0,74	0,93	0,64	0,63	0,86	0,56551	0,34782	0,83061	977	982	973
1,41	2,34	6,12	0,90	1,00	0,59	0,68	0,77	0,58	0,53322	0,30043	0,83067	969	949	974
1,09	2,21	8,86	0,70	0,94	0,86	0,59	0,74	0,80	0,56302	0,34650	0,83223	976	980	975
1,54	2,13	6,23	0,99	0,91	0,60	0,71	0,72	0,59	0,54041	0,30150	0,83253	971	950	976
1,25	2,21	7,87	0,80	0,94	0,76	0,63	0,74	0,72	0,57186	0,33505	0,83351	979	972	977
1,43	1,55	9,57	0,91	0,66	0,93	0,68	0,59	0,86	0,56196	0,34397	0,83486	975	977	978
1,46	1,79	8,44	0,93	0,77	0,82	0,69	0,64	0,77	0,58337	0,33993	0,83838	980	975	979
1,19	1,95	9,62	0,76	0,83	0,93	0,62	0,68	0,86	0,59151	0,36033	0,84232	981	984	980
1,50	1,92	7,80	0,96	0,82	0,76	0,70	0,67	0,71	0,59257	0,33646	0,84405	982	973	981
1,28	1,82	9,75	0,82	0,78	0,95	0,64	0,65	0,87	0,60275	0,36337	0,84762	983	985	982
1,28	2,28	7,84	0,82	0,97	0,76	0,64	0,75	0,72	0,60452	0,34663	0,85014	984	981	983
0,91	2,33	10,16	0,58	0,99	0,99	0,54	0,77	0,91	0,57027	0,37246	0,85370	978	987	984
1,31	2,27	7,95	0,84	0,97	0,77	0,65	0,75	0,73	0,62509	0,35459	0,85886	985	983	985
1,44	2,30	7,16	0,92	0,98	0,69	0,69	0,76	0,66	0,62756	0,34480	0,86563	986	979	986
1,36	1,91	9,54	0,87	0,81	0,93	0,66	0,67	0,86	0,65292	0,37892	0,86874	989	989	987
1,43	1,74	10,01	0,91	0,74	0,97	0,68	0,63	0,89	0,65545	0,38346	0,87431	990	990	988
1,11	2,30	9,60	0,71	0,98	0,93	0,59	0,76	0,86	0,64912	0,38766	0,87442	987	991	989
1,54	2,04	8,08	0,98	0,87	0,78	0,71	0,70	0,74	0,66823	0,36674	0,87804	992	986	990
1,16	2,12	10,23	0,74	0,90	0,99	0,61	0,72	0,91	0,66516	0,39691	0,87919	991	992	991
1,05	2,28	10,29	0,67	0,97	1,00	0,58	0,76	0,92	0,65263	0,39865	0,88104	988	993	992
1,18	2,17	10,01	0,76	0,93	0,97	0,61	0,73	0,89	0,68102	0,40029	0,88488	993	994	993
1,51	2,33	7,58	0,96	0,99	0,73	0,70	0,77	0,70	0,70166	0,37499	0,89650	994	988	994
1,37	2,30	9,30	0,87	0,98	0,90	0,66	0,76	0,84	0,77323	0,42207	0,91898	995	995	995
1,45	2,25	9,52	0,93	0,96	0,92	0,69	0,75	0,85	0,82076	0,43930	0,93642	996	996	996

	NB-Cest	NB-NBM	Cest-NBM
Coef. Correl.	0,922	0,930	0,937

AP.III.3. DATOS PARA LA EVALUACIÓN DE METIORE

02/11/12-22:28:14:150.214.108.233:usuarioActual -user user28 -passwd *****
02/11/12-22:28:15:150.214.108.233:listaObjetivos -user user28
02/11/12-22:28:38:150.214.108.233:crearObjetivo -user user28 -objetivo {Find papers on mobile adaptive systems}
02/11/12-22:28:38:150.214.108.233:listaAtributos
02/11/12-22:28:38:150.214.108.233:nuevaSesion -user user28 -objetivo {Find papers on mobile adaptive systems}
02/11/12-22:28:53:150.214.108.233:busquedaSimple -user user28 -objetivo {Find papers on mobile adaptive systems} -busqueda {city guide}
02/11/12-22:29:37:150.214.108.233:selDocumento -objeto book179
02/11/12-22:30:04:150.214.108.233:evalDocumento -user user28 -objetivo {Find papers on mobile adaptive systems} -doc book179 -valor noeval -prediccion ok_1
02/11/12-22:30:12:150.214.108.233:listaObjetivos -user user28
02/11/12-22:30:31:150.214.108.233:crearObjetivo -user user28 -objetivo e-commerce
02/11/12-22:30:31:150.214.108.233:listaAtributos
02/11/12-22:30:31:150.214.108.233:nuevaSesion -user user28 -objetivo e-commerce
02/11/12-22:30:42:150.214.108.233:busquedaSimple -user user28 -objetivo e-commerce -busqueda store
02/11/12-22:30:55:150.214.108.233:selDocumento -objeto book6
02/11/12-22:31:20:150.214.108.233:listaAtributos
02/11/12-22:31:20:150.214.108.233:nuevaSesion -user user28 -objetivo e-commerce
02/11/12-22:31:55:150.214.108.233:busqueda -user user28 -objetivo e-commerce -op AND -busqueda {title year {} {{title {* commerce}} {year {* 2000}} }}
02/11/12-22:32:16:150.214.108.233:busqueda -user user28 -objetivo e-commerce -op AND -busqueda {title {} {} {{title {* commerce}} }}
02/11/12-22:32:24:150.214.108.233:selCluster -user user28 -objetivo e-commerce -cluster {Electronic_Commerce_via_Personalised_Virtual_Electronic_Catalogues}
02/11/12-22:32:40:150.214.108.233:selDocumento -objeto book282
02/11/12-22:32:52:150.214.108.233:evalDocumento -user user28 -objetivo e-commerce -doc book282 -valor ok_2 -prediccion no_info
02/11/12-22:33:05:150.214.108.233:verHistorico -user user28
02/11/12-22:33:23:150.214.108.233:selDocumento -objeto book282
02/11/12-22:33:25:150.214.108.233:listaAtributos
02/11/12-22:33:25:150.214.108.233:nuevaSesion -user user28 -objetivo e-commerce
02/11/12-22:33:33:150.214.108.233:busquedaSimple -user user28 -objetivo e-commerce -busqueda commerce
02/11/12-22:33:56:150.214.108.233:selDocumento -objeto book1
02/11/12-22:34:06:150.214.108.233:evalDocumento -user user28 -objetivo e-commerce -doc book1 -valor ok_1 -prediccion ok_1
02/11/12-22:34:19:150.214.108.233:selDocumento -objeto book168
02/11/12-22:34:38:150.214.108.233:evalDocumento -user user28 -objetivo e-commerce -doc book168 -valor ok_2 -prediccion ok_1
02/11/12-22:34:50:150.214.108.233:selDocumento -objeto book234
02/11/12-22:35:11:150.214.108.233:evalDocumento -user user28 -objetivo e-commerce -doc book234 -valor noeval -prediccion no_info
02/11/12-22:35:15:150.214.108.233:listaObjetivos -user user28
02/11/12-22:35:20:150.214.108.233:listaAtributos
02/11/12-22:35:20:150.214.108.233:nuevaSesion -user user28 -objetivo e-commerce
02/11/12-22:35:28:150.214.108.233:busquedaSimple -user user28 -objetivo e-commerce -busqueda shop
02/11/12-22:35:34:150.214.108.233:listaObjetivos -user user28
02/11/12-22:35:42:150.214.108.233:listaAtributos
02/11/12-22:35:42:150.214.108.233:nuevaSesion -user user28 -objetivo e-commerce
02/11/12-22:35:57:150.214.108.233:busquedaSimple -user user28 -objetivo e-commerce -busqueda commerce
02/11/12-22:36:11:150.214.108.233:selDocumento -objeto book159
02/11/12-22:36:28:150.214.108.233:evalDocumento -user user28 -objetivo e-commerce -doc book159 -valor ok_1 -prediccion ok_1
02/11/12-22:36:35:150.214.108.233:listaObjetivos -user user28
02/11/12-22:36:39:150.214.108.233:listaAtributos
02/11/12-22:36:39:150.214.108.233:nuevaSesion -user user28 -objetivo e-commerce
02/11/12-22:36:41:150.214.108.233:recomiendame -user user28 -objetivo e-commerce
02/11/12-22:37:44:150.214.108.233:listaObjetivos -user user28
02/11/12-22:37:53:150.214.108.233:crearObjetivo -user user28 -objetivo {user model servers}
02/11/12-22:37:53:150.214.108.233:listaAtributos
02/11/12-22:37:53:150.214.108.233:nuevaSesion -user user28 -objetivo {user model servers}
02/11/12-22:37:56:150.214.108.233:recomiendame -user user28 -objetivo {user model servers}
02/11/12-22:38:10:150.214.108.233:busquedaSimple -user user28 -objetivo {user model servers} -busqueda server
02/11/12-22:38:45:150.214.108.233:busquedaSimple -user user28 -objetivo {user model servers} -busqueda {user model server}
02/11/12-22:39:13:150.214.108.233:busquedaSimple -user user28 -objetivo {user model servers} -busqueda personis
02/11/12-22:39:21:150.214.108.233:selDocumento -objeto book220
02/11/12-22:39:31:150.214.108.233:evalDocumento -user user28 -objetivo {user model servers} -doc book220 -valor ok_2 -prediccion ok_1
02/11/12-22:39:34:150.214.108.233:listaObjetivos -user user28
02/11/12-22:39:39:150.214.108.233:listaAtributos
02/11/12-22:39:39:150.214.108.233:nuevaSesion -user user28 -objetivo {user model servers}
02/11/12-22:39:41:150.214.108.233:recomiendame -user user28 -objetivo {user model servers}
02/11/12-22:40:16:150.214.108.233:busquedaSimple -user user28 -objetivo {user model servers} -busqueda shell
02/11/12-22:40:33:150.214.108.233:selDocumento -objeto book230

02/11/12-22:40:41:150.214.108.233:evalDocumento -user user28 -objetivo {user model servers} -doc book230 -valor ok_1 -prediccion no_info
02/11/12-22:40:45:150.214.108.233:listaAtributos
02/11/12-22:40:45:150.214.108.233:nuevaSesion -user user28 -objetivo {user model servers}
02/11/12-22:40:56:150.214.108.233:busquedaSimple -user user28 -objetivo {user model servers} -busqueda client
02/11/12-22:41:37:150.214.108.233:selDocumento -objeto book87
02/11/12-22:42:00:150.214.108.233:evalDocumento -user user28 -objetivo {user model servers} -doc book87 -valor wrong_2 -prediccion ok_1
02/11/12-22:42:04:150.214.108.233:listaObjetivos -user user28
02/11/12-22:42:08:150.214.108.233:listaAtributos
02/11/12-22:42:08:150.214.108.233:nuevaSesion -user user28 -objetivo {user model servers}
02/11/12-22:42:12:150.214.108.233:recomiendame -user user28 -objetivo {user model servers}
02/11/12-22:43:32:150.214.108.233:verHistorico -user user28
02/11/12-22:43:53:150.214.108.233:listaObjetivos -user user28
02/11/12-22:43:57:150.214.108.233:listaAtributos
02/11/12-22:43:57:150.214.108.233:nuevaSesion -user user28 -objetivo {Find papers on mobile adaptive systems}
02/11/12-22:44:06:150.214.108.233:busquedaSimple -user user28 -objetivo {Find papers on mobile adaptive systems} -busqueda guide
02/11/12-22:44:27:150.214.108.233:selDocumento -objeto book350
02/11/12-22:44:50:150.214.108.233:evalDocumento -user user28 -objetivo {Find papers on mobile adaptive systems} -doc book350 -valor ok_2 -prediccion no_info
02/11/12-22:44:55:150.214.108.233:listaAtributos
02/11/12-22:44:55:150.214.108.233:nuevaSesion -user user28 -objetivo {Find papers on mobile adaptive systems}
02/11/12-22:44:57:150.214.108.233:recomiendame -user user28 -objetivo {Find papers on mobile adaptive systems}
02/11/12-22:45:18:150.214.108.233:selDocumento -objeto book7
02/11/12-22:45:28:150.214.108.233:evalDocumento -user user28 -objetivo {Find papers on mobile adaptive systems} -doc book7 -valor ok_1 -prediccion ok_1
02/11/12-22:45:34:150.214.108.233:evalDocumento -user user28 -objetivo {Find papers on mobile adaptive systems} -doc book7 -valor ok_1 -prediccion no_info
02/11/12-22:45:38:150.214.108.233:listaAtributos
02/11/12-22:45:38:150.214.108.233:nuevaSesion -user user28 -objetivo {Find papers on mobile adaptive systems}
02/11/12-22:45:41:150.214.108.233:recomiendame -user user28 -objetivo {Find papers on mobile adaptive systems}

02/11/27-18:43:34:150.214.108.233:nuevoUsuario -user user124 -passwd *****
02/11/27-18:43:55:150.214.108.233:usuarioActual -user user124 -passwd *****
02/11/27-18:43:56:150.214.108.233:listaObjetivos -user user124
02/11/27-18:44:08:150.214.108.233:listaObjetivos -user user124
02/11/27-18:44:57:150.214.108.233:crearObjetivo -user user124 -objetivo {find articles on information technology}
02/11/27-18:44:57:150.214.108.233:listaAtributos
02/11/27-18:44:57:150.214.108.233:nuevaSesion -user user124 -objetivo {find articles on information technology}
02/11/27-18:58:49:150.214.108.233:busquedaSimple -user user124 -objetivo {find articles on information technology} -busqueda {information technology}
02/11/27-18:59:49:150.214.108.233:recomiendame -user user124 -objetivo {find articles on information technology}
02/11/27-19:00:05:150.214.108.233:recomiendame -user user124 -objetivo {find articles on information technology}
02/11/27-19:00:14:150.214.108.233:verHistorico -user user124
02/11/27-19:00:44:150.214.108.233:busquedaSimple -user user124 -objetivo {find articles on information technology} -busqueda technology
02/11/27-19:00:58:150.214.108.233:selDocumento -objeto book137
02/11/27-19:01:26:150.214.108.233:evalDocumento -user user124 -objetivo {find articles on information technology} -doc book137 -valor ok_1 -prediccion no_info
02/11/27-19:01:43:150.214.108.233:selDocumento -objeto book133
02/11/27-19:01:53:150.214.108.233:evalDocumento -user user124 -objetivo {find articles on information technology} -doc book133 -valor ok_2 -prediccion no_info
02/11/27-19:02:33:150.214.108.233:busquedaSimple -user user124 -objetivo {find articles on information technology} -busqueda information
02/11/27-19:03:11:150.214.108.233:selDocumento -objeto book191
02/11/27-19:03:32:150.214.108.233:evalDocumento -user user124 -objetivo {find articles on information technology} -doc book191 -valor ok_2 -prediccion ok_1
02/11/27-19:03:48:150.214.108.233:verHistorico -user user124
02/11/27-19:04:13:150.214.108.233:listaAtributos
02/11/27-19:04:13:150.214.108.233:nuevaSesion -user user124 -objetivo {find articles on information technology}
02/11/27-19:05:01:150.214.108.233:busqueda -user user124 -objetivo {find articles on information technology} -op AND -busqueda {title keywords {} {{title {* technology}} }}
02/11/27-19:05:39:150.214.108.233:selCluster -user user124 -objetivo {find articles on information technology} -cluster {<Integrating_Neural_Network_Technology_with_Hypermedia><semantic_links>}
02/11/27-19:05:46:150.214.108.233:selCluster -user user124 -objetivo {find articles on information technology} -cluster {<Integrating_Neural_Network_Technology_with_Hypermedia><user_model>}
02/11/27-19:05:50:150.214.108.233:selCluster -user user124 -objetivo {find articles on information technology} -cluster {<Integrating_Neural_Network_Technology_with_Hypermedia><hypermedia>}
02/11/27-19:05:54:150.214.108.233:selCluster -user user124 -objetivo {find articles on information technology} -cluster {<Integrating_Neural_Network_Technology_with_Hypermedia><semantic_links>}
02/11/27-19:05:55:150.214.108.233:selCluster -user user124 -objetivo {find articles on information technology} -cluster {<Integrating_Neural_Network_Technology_with_Hypermedia><user_model>}

02/11/27-19:06:13:150.214.108.233:selCluster -user user124 -objetivo {find articles on information technology} -cluster {<Integrating_Neural_Network_Technology_with_Hypermedia><semantic_links>}
02/11/27-19:06:17:150.214.108.233:selCluster -user user124 -objetivo {find articles on information technology} -cluster {<Integrating_Neural_Network_Technology_with_Hypermedia><hypermedia>}
02/11/27-19:06:18:150.214.108.233:selCluster -user user124 -objetivo {find articles on information technology} -cluster {<Integrating_Neural_Network_Technology_with_Hypermedia><semantic_links>}
02/11/27-19:06:18:150.214.108.233:selCluster -user user124 -objetivo {find articles on information technology} -cluster {<Integrating_Neural_Network_Technology_with_Hypermedia><user_model>}
02/11/27-19:06:33:150.214.108.233:selCluster -user user124 -objetivo {find articles on information technology} -cluster {<Integrating_Neural_Network_Technology_with_Hypermedia><neural_network>}
02/11/27-19:06:45:150.214.108.233:selCluster -user user124 -objetivo {find articles on information technology} -cluster {<Integrating_Neural_Network_Technology_with_Hypermedia><intelligent_tutoring>}
02/11/27-19:06:46:150.214.108.233:selCluster -user user124 -objetivo {find articles on information technology} -cluster {<Integrating_Neural_Network_Technology_with_Hypermedia><user_model>}
02/11/27-19:06:56:150.214.108.233:selDocumento -objeto book301
02/11/27-19:07:05:150.214.108.233:evalDocumento -user user124 -objetivo {find articles on information technology} -doc book301 -valor wrong_2 -prediccion ok_1

REFERENCIAS

Armstrong, R., Freitag, D., Joachims, T., & Mitchell, T. (1995). "Webwatcher: A learning apprentice for the world wide web". AAAI Spring Symposium on Information Gathering from Heterogeneous Distributed Environments

Asnicar, F., & Tasso, C. (1997). "ifWeb: A prototype of user model - Based intelligent agent for document filtering and navigation in the world wide web". UM97 Workshop on "Adaptive Systems and User Modeling on the World Wide Web". Sixth International Conference on User Modeling . URL= http://www.dimi.uniud.it/~ift/um97/positionp.html.

Balabanovic, M. (1997). "An Adaptive Web Page Recommendation Service". Proceedings of the First International Conference on Autonomous Agents Marina del Rey, CA

Balabanovic, M., & Shoham, Y. (1995). "Learning Information Retrieval Agents: Experiments with Automated Web Browsing". AAAI Spring Symposium on Information Gathering, Stanford, CA

Balabanovic, M., & Shoham, Y. (1997). "Combining Content-Based and Collaborative Recommendation". Communications of the ACM, 40(3)

Barra, M. (2000). "Distributed Systems for Group Adaptivity on the Web". Adaptive Hypermedia and Adaptive Web-Based Systems Springer-Verlag

Bauer, M. (1995). "Dempster Shafer Approach to Modeling Agent Preferences for Plan Recognition". User Modeling and User-Adapted Interaction, 5(3-4), 317-348

Bauer, M., Gmytrasiewicz, P. J., & Vassileva, J. (2001). "User Modeling". 8th International Conference UM'2001 Sonthofen, Germany

Belkin, N. J., & Croft, W. B. (1992). "Information Filtering and Information Retrieval: Two sides of the Same Coin?". Communications of the ACM, 35(12), 29-38

Billsus, D., & Pazzani, M. (1997). "Learning Probabilistic User Models". *Workshop Notes of "Machine Learning for User Modeling", Sixth International Conference on User Modeling, Chia Laguna, Sardinia.* URL= http://www.ics.uci.edu/~pazzani/Publications/ProbUserModels.pdf.

Billsus, D., & Pazzani, M. (1999). "A Hybrid User Model for News Story Classification". Proceedings of the Seventh International Conference on User Modeling (UM '99) Banff, Canada

Bornscheuer, S., McIntyre, Y., Steffen Holldobler, S., & Storr, H. (2001). "User Adaption in a Web Shop System". (pp. 208-213). Proceedings of the IASTED International Conference Internet and Multimedia Systems and Applications. URL= citeseer.nj.nec.com/523289.html.

Bradley, K., Rafter, R., & Smyth, B. (2000). "Case-Based User Profiling for Content Personalisation". Adaptive Hypermedia and Adaptive Web-Based Systems Springer-Verlag

Brajnik, G., & Tasso, C. (1994). "Ashell for developing non-monotonic user modeling systems". International Journal of Human-Computer Studies, 40, 31-62

Brewer, R. S., & Johnson, P. M. (1994). "Toward Collaborative Knowledge Management within Large, Dynamically Structured Information Systems". University of Hawaii, Dpt. of Information and Computer Science. Honolulu:

Brin, S., & Page, L. (1998). The Anatomy of a Large-Scale Hypertextual Web Search Engine (pp. 107-117). WWW7/Computer Networks. URL= http://dbpubs.stanford.edu/pub/1998-8.

Bueno, D., Conejo, R., Carmona, C., & David, A. A. (2002). "METIORE: A Publication Reference for Adaptive Hypermedia Community". Adaptive Hypermedia and Adaptive Web-Based Systems. AH'2002 Málaga

Bueno, D., Conejo, R., & David, A. A. (2001). "METIOREW: An Objective Oriented Content Based and Collaborative Recommending System". Third Workshop on Adaptive Hypertext and Hypermedia (pp. 123-133). Sonthofen (Alemania)

Bueno, D., & David, A. A. (2000a). "Experimenting user model in Cooperative IRS". The 6th International

Conference on Information Systems, Analysis and Synthesis ISAS 2000 (IEEE) Florida

Bueno, D., & David, A. A. (2000b). "Processing the user model in IRS". International Journal of Knowledge Organization, 27(1/2), 17-26

Bueno, D., & David, A. A. (2001). "METIORE: A Personalized Information Retrieval System". 8th International Conference on User Modeling.UM'2001 Alemania

Callan, J. P., Croft, W. B., & Hardling, S. M. (1992). "The INQUERY Retrieval System, (pp. 78-83). Valencia, Spain: Springer-Verlag.". Third International Conference on Database and Expert Systems Applications Valencia, SpainSpringer-Verlag

Cestnik, B. (1990). "Estimating probabilities: A cricual task in machine learning". Proceedings of the Ninth european Conference on Artifical Intelligence (pp. 147-149). London

Chin, D. N. (1989). "KNOME: Modeling What the User Knows in UC". User Models in Dialog Systems,

Cotter, P., & Smyth, B. (2000). "WAPing the Web: Content Personalisation for WAP-Enabled Devices". Adaptive Hypermedia and Adaptive Web-Based Systems Springer-Verlag

Croft, W. B. (1995). "What do people want from information retrieval". D-Lib Magazine,

David, A. A., & Bueno, D. (1999). "User Modeling and Cooperative Information Retrieval in Information Retrieval Systems". International Journal of Knowledge Organization, 26(1), 30-45

David, A. A., & Bueno, D. (2001). "Case-Based Reasoning, User model and IRS". The 5th World Multiconference on Systemics, Cybernetics and Informatics (SCI 2001).IEEE Orlando

Dewan, P. (1993). "Tool for implementing multiuser interfaces". User interface software, vol 1, Trends in software. URL=ftp://ftp.cs.unc.edu/pub/users/dewan/papers/trends.ps

Dourish, P. (1996). "Open implementation and flexibility in CSCW toolkits". Ph.D. Thesis, Department of Computer Science, University College London, UK, 1996. URL=shell4.ba.best.com/pub/jpd/thesis/thesis.pdf

Fuhr, N. (2000). "Models in Information Retrieval". Lectures on Information Retrieval (pp. 21-50). Berlin Springer-Verlag

Good, N., Schafer J. , Konstan, J., Borchers, A., Sarwar, B., Herlocker, J., & Riedl, J. (1999). "Combining collaborative filtering with personal agents for better recommendations". In Proceedings of the Sixteenth National Conferenceon Artificial Intelligence

Joachims, T. (1997). "A Probabilistic Analysis of the Rocchio Algorithm with TFIDF for Text Categorization". Proc. of the 14th International Conference on Machine Learning ICML97 (pp. 143-151).

Kay, J. (1994). "Lies, damned lies and stereotypes: pragmatic approximations of users". 4th International Conference on User Modelling. UM'94

Kay, J. (1995a). "The um Toolkit for reusable, long term user models". User Modeling and User-Adapted Interaction, 4(3), 149-196

Kay, J. (1995b). "Vive la difference! Individualised interaction with users". IJCAI'95

Keogh, E., & Pazzani, M. (1999). "Learning augmented Bayesian classifiers: A comparison of distribution-based and classification-based approaches.". Uncertainty 99, 7th. Int'l Workshop on AI and Statistics, (pp. 225-230). Ft. Lauderdale, Florida

Kislin, P., & David, A. A. (2000). "Application du prototype METIORE dans un cadre d'intelligence économique : De la compétitivité à la coopération…Utilisation de la base bibliographique du LORIA/INRIA Lorraine pour la veille scientifique.". Technical Report. LORIA France: Pages 1-137

Kobsa, A. (1993). "User Modeling: Recent Work, Prospects and Hazards". Adaptive User Interfaces: Principle and Practise. Amsterdam

Kobsa, A. (2001). "Generic User Models". User Modeling and User-Adapted Interaction, 11, 49-63

Kobsa, A., Nill, A., & Dietmar Müller. (1996). "KN-AHN: An Adaptative Hypertext Client of the User Modeling System BGP-MS". Review of Information Science 1(1). URL= http://www.inf-wiss.uni-konstanz/RIS.

Kobsa, A., & Pohl, W. (1995). "The BGP-MS user modeling system". User Modeling and User-Adapted Interaction, 4(2), 59-106

Kononenko, I. (1990). "Comparison of Inductive and Naive Bayesian Learning Approaches to Automatic Knowledge Acquisition". Current Trends in Knowledge Adquisition, 190-197

Kowalski, G. J., & Maybury, M. T. (2001). "Information Storage and Retrieval Systems. Theory and Implementation". Kuwer Academic.

Lesh, N., Rich, C., & Sidner, C. (1997). "Using Plan Recognition in Human-Computer Collaboration". User Modeling Conference. UM'97

Lieberman, H. (1995). " Letizia: An Agent That Assists Web Browsing". International Joint Conference on Artificial Intelligence Montreal, CA.

Lotka, A. J. (1926). "The frequency distribution of scientific productivity". Journal of the Washington Academy of Sciences, 16, 317-323

Magnini, B., & Strapparava, C. (2001). "Imporving User Modeling with Content-Based Techniques". 8th International Conference on User Modeling.UM'2001 (pp. 74-83). Springer-Verlag

Mitchell, T. (1997). "Machine Learning". The McGraw-Hill Companies, Inc.

Mitchell, T., Caruana, R., McDermott, J., & Zabowski D. (1994). " Experience With a Learning Personal Assistant". Communications of the ACM, 37(7)

Mizzarro, S., & Tasso, C. (2002). "Ephemeral and Persistent Personalization in Adaptive Information Access to Scholarly Publications on the Web". Adaptive Hypermedia and Adaptive Web-Based Systems. AH'2002 (pp. 306-316).

Pazzani, M., Muramatsu, J., & Billsus, D. (1996). "Syskill & Webert: Identifying interesting web sites". AAI Spring Symposium on Machine Learning in Information Access . URL=

http://www.parc.xerox.com/istl/porjects/mlia/papers/pazzani.ps.
Peter, H. P. J., & Van Raan. (1991). "Structuring scientific activities by co-author analysis". Scientometrics, 20(1), 235-255
Porter, M. F. (1980). "An algorithm for suffix stripping". (pp. 130-137).
Price, D. S. (1976). "A general theory of bibliometric and other cumulative advantage processes". Journal of the American Society for Information Science, 27(5), 292-306
Price, D. S., & Beaver, D. (1966). "Collaboration in an Invisible College". American Psychologist, 21, 1011-1018
Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., & Riedl, J. (1994). "GroupLens: An Open Architecture for Collaborative Filtering of Netnews". Proceedings of ACM 1994 Conference on Computer Supported Cooperative Work (pp. 175-186).
Rich E. (1979). "User Modeling via Stereotypes". International Journal of Cognitive Science, 3, 329-354
Rich, E. (1983). "Users are individuals: individualizing user models". Int. J. Man-Machine Studies, 18, 199-214
Rijsbergen, C. J. v. (1979). "Information Retrieval". (2nd ed.). London: Butterworths.
Rijsbergen, C. J. v. (2000). "Getting into Information Retrieval". Lectures on Information Retrieval (pp. 1-20). Berlin Springer-Verlag
Robertson, S. (1997). "Special Issue on Okapi". Journal of Documentation, 53
Robertson, S. (2000). "Evaluation in Information Retrieval". Lectures on Information Retrieval (pp. 81-92). Berlin Springer-Verlag
Robertson, S., & Sparck Jones, K. (1976). "Relevance weighting of search terms". Journal of the American Society for Information Science (pp. 127-146).
Robertson, S., & Walker, S. (1993). "Okapi at TREC". The First Text REtrieval Conference (TREC) (pp. 21-30).
Robertson, S., & Walker, S. (1994). "Okapi at TREC-2". Second Text Retrieval Conference (TREC-2)
Rölleke, T. (1999). "POOL: Probabilistics Object-Oriented Logical Representation and Retrieval of COmplex Objects; A model for hypermedia retrieval. PhD Thesis". University of Dormund: Springer Verlag.
Rocchio, J. (1971), “Relevance Feedback in Information Retrieval”, in Salton: The SMART Retrieval System: Experiments in Automatic Document Processing, (pp.313-323). Prentince-Hall.
Oikarinen, J., & Reed, D. (1993). “Internet Relay Chat Protocol”. RFC1459. 1993.URL=ftp://ftp.rfc-editor.org/in-notes/rfc1459.txt
Salton, G. (1971). "The SMART Retrieval System- Experiments in Automatic Document Processing". New Jersey: Prentice Hall, Englewoods, Cliffs.
Saracevic, T., Spink, A., & Wu, M.-M. (1997). "Users and Internmediaries in Information Retrieval: What are they talking about?". UM'97 New YorkSpringer
Schwab I. , & Pohl W. (1999). "Learning Information Interest from Positive Examples". User Modeling (UM'99)
Singh, M., & Provan, G. M. (1996). "Efficient learning of selective Bayesian network classifiers". . Proceedings of the 13th International Conference on Machine Learning
Tasso, C., & Mizzarro, S. (2002). "Ephemeral and Persistent Personalization in Adaptive Information Access to Scholarly Publications on the Web". Adaptive Hypermedia and Adaptive Web-Based Systems. AH'2002 Springer Verlag
TREC. (2001) TREC Conferences. http://trec.nist.gov.
Vassileva, J. (1994). "A Practical Architecture for User Modeling in a Hypermedia-Based Information System". (pp. 115-120). Proceedings of the 4-th International Conference on User Modeling
Vel, O., & Nesbitt, S. (1997). "A Collaborative Filtering Agent System for Dynamic Virtual Communities on the Web". URL= http://citeseer.nj.nec.com/de-collaborative.html.
Versteegen, L. (2000). "The Simple Bayesian Classifier as a Classification Algorithm". URL= http://www.cs.kun.nl/nsccs/artikelen/leonv.ps.Z.
Yan, T. W., & Garcia-Molina, H. (1995). "SIFT - A Tool for Wide-Area Information Dissemination". (pp. pages 177-186). In Proceedings of the 1995 USENIX Technical Conference. URL= ftp://db.stanford.edu/pub/yan/1994/sift.ps.
Zukerman, I., Albrecht, D. W., & Nicholson, A. E. (1999). "Predicting Users' Request on the WWW".