Modelos de Recuperación de la Información basados en Información Lingüística Difusa y Algoritmos Evolutivos. Mejorando la Representación de las Necesidades de Información

TESIS DOCTORAL

MARÍA LUQUE RODRÍGUEZ

DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN E INTELIGENCIA ARTIFICIAL

UNIVERSIDAD DE GRANADA

Granada, 2005

Editor: Editorial de la Universidad de Granada
Autor: María Luque Rodríguez
D.L.: Gr. 338 - 2005
ISBN: 84-338-3288-3

UNIVERSIDAD DE GRANADA

E.T.S. DE INGENIERÍA INFORMÁTICA

DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN E INTELIGENCIA ARTIFICIAL

Modelos de Recuperación de la Información basados en Información Lingüística Difusa y Algoritmos Evolutivos. Mejorando la Representación de las Necesidades de Información

TESIS DOCTORAL

María Luque Rodríguez

Granada, Enero de 2005

Modelos de Recuperación de la Información basados en Información Lingüística Difusa y Algoritmos Evolutivos. Mejorando la Representación de las Necesidades de Información

MEMORIA QUE PRESENTA

MARÍA LUQUE RODRÍGUEZ

PARA OPTAR AL GRADO DE DOCTOR EN INFORMÁTICA

ENERO DE 2005

DIRECTORES

DR. D. OSCAR CORDÓN GARCÍA Y DR. D. ENRIQUE HERRERA VIEDMA

Fdo. Oscar Cordón García

Fdo. Enrique Herrera Viedma

Fdo. María Luque Rodríguez

AGRADECIMIENTOS

Cuando por fin se ve la meta y tan sólo faltan unos pocos metros para cruzarla quiero daros las gracias a todos, a los que habéis recorrido conmigo este camino, a los que me habéis alentado a seguir, a los que me habéis ayudado a levantarme cuando caía y también a los que me habéis empujado para que cayera. Gracias, sin vosotros no estaría aquí.

En primer lugar, quiero darles las gracias a mis directores de tesis, a Oscar Cordón y Enrique Herrera, por haber confiado en mí desde el primer momento. Gracias por impulsarme a seguir, por el tiempo que me habéis dedicado y por vuestros desvelos leyendo los mismos folios una y otra vez para que consiguiera llegar al final. Sin vuestra dedicación esta memoria seguiría siendo un proyecto en vez de una realidad.

También quiero dar las gracias al doctor Francisco Herrera, por ayudarme a dar mis primeros pasos en este "mundo", y a todos mis compañeros del grupo de investigación SCI2S por su continúo apoyo e interés.

A mis padres mi más sincero agradecimiento, por haber estado ahí en todo momento, por su compresión y su amor y sobre todo por haberme enseñado desde pequeña la importancia de la constancia y el trabajo (sólo ellos saben las veces que me han enviado a mi cuarto....). Sin ellos y todos sus consejos aun seguiría corriendo. Os quiero mucho.

A mi hermano, por portarse conmigo como seguramente yo nunca lo hubiera hecho con él. Paco, gracias de todo corazón y no cambies nunca.

Mi más sentida gratitud va dirigida, al sufridor incondicional de mi mal humor y mis desaires, a la persona que me ha aguantado día y noche durante todos estos meses, y siempre con una sonrisa en los labios. No tengo palabras para agradecerle todo lo que ha hecho por mí, por hacerme ver que no merece la pena agobiarse por tonterías y que todo tiene un fin que tarde o temprano conseguiré alcanzar. Gracias Antonio, por haber estado ahí, por abrazarme y

consolarme cada vez que me estrellaba, y haberme ayudado a proseguir. Espero poder estar a la altura de las circunstancias cuando llegue tu momento.

Gracias también a Marifeli, Mariano y Ester, por todo el cariño que me han demostrado y acogerme en tantas ocasiones.

A mis queridos Belén, Carlos, Carmen y Jesús (beeee!!!!) por su amistad, su apoyo y su sonrisa. Gracias por aguantarme y mantener el tipo. Sois auténticos héroes.

No puedo olvidarme de mis compañeros de faena, los que compartieron mis inicios en Granada, meceneros, exmeceneros y miembros de DECSAI; y los que me aguantan ahora en Córdoba: Eva (tú sigues aguantándome), Pedro, Sebas, Cristobal y el resto de miembros del departamento de Informática y Análisis Numérico. Gracias.

Por último, y no por ello menos importantes, agradecer a Chechu, Alo, Jose, Javi, Gador, Mª Carmen, Mariano, Ana, Sonia, Nacho, Maribel, Inma, en fin, a todos por haberme permitido desaparecer del mapa durante tantos meses.

Y por fin, a la pregunta que tanto tiempo lleva haciéndome mi padre, puedo contestar, "ya sólo me queda poner "***GRACIAS A TODOS""*.**

ÍNDICE DE CAPÍTULOS

ÍNDICE DE CONTENIDOS

ÍNDICE DE FIGURAS

PLANTEAMIENTO

En un mundo globalizado que cambia rápidamente, el estar permanentemente informado se ha convertido en una necesidad apremiante, en fuente de conocimiento y también de dinero. Una oleada reciente de subscripciones a servicios on-line de noticias pone de manifiesto la importancia que la sociedad da a estar permanentemente informada sobre temas que son de su interés. Esta "puesta al día" permite tanto a la persona individual como a las organizaciones ser competitivas y tomar mejores decisiones.

Uno de los problemas principales de Internet es el crecimiento constante y descontrolado de la información a la que los usuarios pueden acceder [120][121]. Este crecimiento de Internet, tanto en sitios Web como en documentos y servicios Web, está contribuyendo a que los usuarios tengan difícil el acceso a la información que precisan de manera simple y eficiente. Son necesarios, por tanto, sistemas que les ayuden a hacer frente a esta gran maraña de información en que se ha convertido Internet [121][111].

Como consecuencia de esto, las investigaciones en áreas relacionadas con la búsqueda o acceso a la información, ya sea en la Web o en cualquier otro sistema, han aumentado considerablemente en los últimos años [4][33][43][48][60]. Estas investigaciones están basadas en diferentes filosofías de trabajo, que se pueden englobar dentro del concepto de *Acceso a la Información* (en inglés, "*Information Seeking*" [130]), término que describe a cualquier proceso mediante el cual los usuarios son capaces de obtener información de un sistema. Algunos de estos procesos son: la recuperación de información (RI), el filtrado de información (FI), el acceso a bases de datos, la extracción de información y el "browsing". En los últimos años estamos asistiendo a la aplicación creciente de distintas ciencias en estos procesos de acceso a la información con objeto de mejorarlos. En concreto, métodos, conceptos y técnicas de Inteligencia Artificial (IA) están siendo aplicados en los procesos de obtención de información con notable éxito [5][112][164].

Por tanto, el estudio de algunos de los procesos de acceso a la información, así como la introducción de mejoras en ellos con el fin de obtener una mayor eficiencia en la búsqueda, se muestra como una línea de investigación muy activa. En esta memoria nos centraremos en el

estudio de la RI y el FI, a la vez que propondremos su combinación con técnicas de IA para conseguir mejorar la búsqueda de información.

Las Dos Caras de una Misma Moneda: Recuperación de Información y Filtrado de Información

La RI y el FI son procesos de búsqueda de información que comparten su objetivo principal, presentar a los usuarios solamente la información relevante para ellos, empleando de manera óptima el tiempo con que cuentan [84]. Sin embargo, mientras que la RI se encarga de dar respuesta a las necesidades de información puntuales de un usuario (representadas como consultas), el FI se asocia a necesidades que permanecen constantes en el tiempo (expresadas como perfiles de usuario). Por otro lado, la RI no tiene ningún conocimiento sobre los usuarios, al contrario del FI, que almacena las preferencias de los usuarios con el fin de facilitar la selección de información.

Ahora bien, Belkin y Croft [7] determinaron que el FI y la RI constituyen las dos caras de una misma moneda que, trabajando en estrecha relación, consiguen ayudar a los usuarios en la obtención de la información que necesitan para lograr sus objetivos. De hecho, usando sistemas de filtrado de información (SFI), podemos depurar la información seleccionada por los sistemas de recuperación de información (SRI), de manera que la información mostrada finalmente a los usuarios se adapte lo mejor posible a sus necesidades.

Aún así, ambos sistemas presentan deficiencias a la hora de capturar y representar las necesidades de información de los usuarios [61]. En lo que respecta al FI, la estructura más común de perfil es la denominada “bag of words”, la cual consiste en un conjunto de palabras clave que representan los intereses del usuario. El problema radica en que la mayoría de los métodos actuales requieren que sean los propios usuarios los que indiquen cómo representar sus necesidades de información, lo que los coloca en una situación comprometida. Se encuentran con la dificultad de tener que seleccionar las palabras adecuadas para representar sus necesidades y comunicarse así con el sistema, lo que se conoce clásicamente como el “*problema del vocabulario*” en la interacción hombre-ordenador [76].

Se necesitan, por tanto, formas más inteligentes de capturar y representar las necesidades de información. En esta línea, Belkin y Croft sugirieron que las técnicas de RI podían

aplicarse con éxito en el FI [7]. De esta forma, el perfil de usuario puede representarse mediante una consulta formulada usando cualquier modelo de RI, las llamadas "*consultas persistentes*" [61], y pueden aplicarse técnicas de aprendizaje automático de consultas para su construcción.

Por otro lado, un factor importante a la hora de obtener buenos resultados en un proceso de RI es la habilidad del usuario para expresar sus necesidades de información mediante una consulta. Se ha demostrado que, con frecuencia, el usuario no tiene una imagen clara de lo que está buscando y sólo puede representar sus necesidades de información de forma imprecisa y vaga. Por ello, es necesario disponer de lenguajes de consulta flexibles que permitan expresar las necesidades de información subjetivas de forma simple y aproximada [141], mejorando de esta forma la interacción entre el usuario y el sistema.

Todo esto pone de manifiesto la necesidad de contar con técnicas que mejoren la representación de las necesidades de información de los usuarios en los SRI documentales. Esta necesidad se puede afrontar desde una doble vertiente:

- Desarrollando técnicas de actuación off-line que permitan caracterizar las preferencias de los usuarios con vistas a filtrar la información que les puede devolver el SRI [7][84].
- Desarrollando técnicas de actuación on-line que permitan representar mejor las necesidades de información en la interacción directa SRI-usuario.

El objetivo principal de esta memoria es mejorar la manera de representar las necesidades de información de los usuarios, tanto las permanentes como las puntuales, para así conseguir un mayor rendimiento de los SRI haciendo uso de técnicas de IA. A continuación, describimos brevemente la forma en que nos proponemos llevarlo a cabo.

Caracterización de los Perfiles de Usuario como Consultas Clásicas de Recuperación de Información. Consultas Persistentes

Aparte del habitual enfoque "bag of words", las necesidades persistentes pueden representarse siguiendo un "enfoque explícito de categorías" [61], donde los intereses se

identifican seleccionándolos de un conjunto de categorías predefinidas.

Sin embargo, este segundo enfoque presenta también una serie de inconvenientes: i) las categorías pueden no ser suficientemente precisas; ii) los usuarios pueden necesitar mucho tiempo para encontrar, si es que la encuentran, la categoría o subcategoría que represente sus necesidades; y iii) puede haber discrepancia entre el usuario y el sistema sobre la categoría en la que clasificar una información.

Necesitamos, por tanto, una nueva forma de representar este tipo de necesidades. Como hemos comentado, Belkin y Croft sugirieron que las técnicas de RI podrían ser aplicadas con éxito en el FI [7]. De esta forma, el perfil de usuario puede representarse mediante una consulta formulada usando cualquier modelo de RI, las llamadas "*consultas persistentes*" [61].

Representar el perfil como una consulta persistente, en lugar de como la clásica estructura de "bag of words", nos proporciona:

- Más flexibilidad, al poder usar cualquier modelo de RI para formularla: Booleano, Booleano extendido, Lingüístico, ...
- Más expresividad, al usar términos y operadores Booleanos (Y, O, NO) para representar las necesidades de información, lo que es más interpretable para el ser humano.

NECESIDAD DEL APRENDIZAJE AUTOMÁTICO DE CONSULTAS

Hemos visto que una de las ventajas de representar los perfiles de usuario como consultas persistentes es la flexibilidad que nos proporciona el poder utilizar cualquier modelo de RI para formularlas (Booleano, Booleano extendido, Lingüístico, ...). Sin embargo, el usuario necesita conocer en profundidad el lenguaje de consulta para poder expresar sus necesidades de información en forma de una consulta interpretable por el sistema y, así, conseguir recuperar información relevante.

Entre los modelos de RI existentes (Booleano, Vectorial, Probabilístico, Booleano Extendido, ...), los más complejos son el Booleano y el Booleano extendido. Esto se debe a la dificultad con la que se encuentra el usuario a la hora de tener que formular una consulta, puesto que su lenguaje de consulta, basado en la construcción de sentencias con términos

unidos mediante los operadores Booleanos Y y O (que pueden negarse mediante el operador de negación NO), es bastante complejo.

En definitiva, a un usuario le suele resultar muy difícil formular una consulta sea cual sea su estructura [61][60][138].

Se han desarrollado diferentes aproximaciones para ayudar al usuario en el proceso de la formulación de consultas en diferentes tipos de SRI [49]. Una de las más conocidas se basa en la generación automática de consultas que describan adecuadamente las necesidades del usuario —representadas en forma de un conjunto inicial de documentos relevantes (y opcionalmente no relevantes)— mediante un proceso *off-line* en el que su interacción no es necesaria. Esta operación se incluye en el paradigma de Aprendizaje Automático [135] y Chen y otros la han denominado *Aprendizaje Inductivo de Consultas a partir de Ejemplos* (IQBE) [49].

Por tanto, las técnicas IQBE trabajan de la misma forma que los métodos de aprendizaje de perfiles (el perfil se aprende automáticamente a partir de un conjunto de documentos de entrenamiento proporcionado por el usuario [60][138]), lo que permite aplicarlas directamente a la construcción de consultas persistentes para el FI. Además, el uso de estas técnicas permitirá aprender no sólo los términos de las consultas (las características), sino también la composición de la consulta en sí.

TÉCNICAS DE SOFT COMPUTING EN RECUPERACIÓN DE INFORMACIÓN

Como hemos mencionado, en los últimos años se ha experimentado un interés creciente en la aplicación de técnicas basadas en IA al campo del acceso a la información con el propósito de resolver distintos problemas. Las técnicas de Soft Computing (SC) son una de las técnicas de IA que más se están usando.

El concepto de Soft Computing (SC) fue introducido por Zadeh [188] como una sinergia de metodologías (lógica difusa, computación evolutiva, redes neuronales, razonamiento probabilístico, ...) que proporcionan los fundamentos para la concepción, diseño, construcción y utilización de sistemas inteligentes de información. De acuerdo a [189], el principio básico del SC es la tolerancia a la imprecisión, incertidumbre y aproximación. El hecho de que la subjetividad y la incertidumbre sean propiedades típicas de cualquier proceso de búsqueda de

información ha dado lugar a que las técnicas de SC se hayan revelado como una excelente herramienta para el manejo de la subjetividad y la imprecisión en la definición de los sistemas de acceso a la información [42]. En concreto, Crestani y Pasi enuncian que el uso de SC puede aportar una mayor flexibilidad a estos sistemas.

Existe una gran cantidad de contribuciones que afrontan el uso de las técnicas de SC en el campo del acceso a la información [42][43][47][136]. En particular, la Lógica Difusa [186][190] y los Algoritmos Evolutivos (AEs) [1][2] están consiguiendo resultados prometedores. La primera está siendo utilizada para modelar la subjetividad y la incertidumbre existentes en la actividad de la RI (p.e, en la estimación de la relevancia de un documento respecto a una consulta o en la formulación de una consulta que representa las necesidades de información del usuario), mientras que los AEs se emplean para adaptar componentes del SRI desde el punto de vista del Aprendizaje Automático [135].

El Uso de Algoritmos Evolutivos en la Recuperación de Información

Los AEs no son específicamente algoritmos de aprendizaje automático, pero ofrecen una metodología de búsqueda potente e independiente del dominio que puede ser aplicada a gran cantidad de tareas de aprendizaje. De hecho, los AEs han sido muy utilizados en problemas de aprendizaje automático de reglas de producción a partir de conjuntos de ejemplos [32][78][82] o de extracción de conocimiento [74]. Este amplio uso se debe a que, cuando los individuos de la población genética representan conocimiento, pueden ser tratados al mismo tiempo como datos que son manipulados por el AE y como código ejecutable que lleva a cabo cierta tarea [133].

Debido a estas razones, la aplicación de los AEs a distintos campos de la ciencia se ha incrementado en los últimos años. Dentro del marco de la RI, los AEs se han aplicado en la resolución de un gran número de problemas, siendo el aprendizaje de consultas uno de los que más aportaciones presenta [38]. Su facilidad para generar consultas con distintas estructuras (consultas Booleanas, Booleanas extendidas, lingüísticas, etc.) los hace muy adecuados para generar perfiles de usuario representados como consultas.

A este respecto, debemos destacar que la mayoría de las aproximaciones basadas en el paradigma IQBE de aprendizaje automático de consultas evalúan el rendimiento de las consultas derivadas empleando los dos criterios habitualmente considerados en el enfoque

algorítmico de RI, precisión y exhaustividad [5][153][155][176]. Claramente, la optimización de los componentes de un SRI y, más concretamente, el aprendizaje automático de consultas, es un claro ejemplo de un problema "multiobjetivo".

Este tipo de problemas se caracterizan por el hecho de tener que optimizar simultáneamente diferentes objetivos [30]. Por esa razón, no existe una única mejor solución para resolver el problema. En un típico problema multiobjetivo de optimización, tendremos un conjunto de soluciones que serán superiores al resto cuando se consideran todos los objetivos. A este conjunto de soluciones se le conoce como "Pareto" y a las soluciones contenidas en él, que son mejores que el resto, se las conoce como soluciones no dominadas o Pareto-optimales. En cambio, al resto de soluciones no incluidas en el conjunto Pareto se las denomina soluciones dominadas. Mientras que ninguna de las soluciones del Pareto sea absolutamente mejor que las otras soluciones no dominadas, todas ellas son igualmente aceptables para satisfacer los objetivos planteados.

Los AEs resultan ser una herramienta muy apropiada para la resolución de problemas multiobjetivo. Sin embargo, la aplicación de los AEs en el área de la RI se ha basado normalmente en la combinación de los criterios exhaustividad y precisión en una sencilla función de adaptación mediante un esquema ponderado. Ahora bien, con la posibilidad de aplicar los AEs multiobjetivo, seríamos capaces de obtener varias soluciones con diferentes balances de precisión y exhaustividad para una determinada necesidad de información, en una sola ejecución del algoritmo [41].

El Uso de Información Lingüística Difusa en Recuperación de Información

Una de las técnicas capitales de la Lógica Difusa es el Modelado Lingüístico [187] para la representación y el manejo de la información cualitativa en los sistemas. En particular, se ha aplicado con notable éxito en el diseño de controladores difusos y en sistemas de información, donde se precisa interacción con los usuarios, entre los cuales tenemos los SRI.

Los SRI basados en información lingüística difusa [10][16][17][114] se diseñan usando el concepto de variable lingüística [187] para representar mejor la información cualitativa en el subsistema de consulta. Estos SRI cuentan con lenguajes de consulta ponderados lingüísticos que mejoran la interacción SRI-usuario. Por un lado, dichos lenguajes de consulta incrementan las posibilidades de expresión de los usuarios puesto que con ellos es posible

asignar pesos a los términos de las consultas indicando importancia relativa o umbrales de satisfacción. Por otro lado, facilitan a los usuarios la expresión de sus necesidades de información porque pueden expresar los pesos mediante valores lingüísticos más propios del lenguaje humano. Se han propuesto diferentes modelos de SRI lingüísticos usando una aproximación lingüística difusa ordinal que facilita la expresión y el procesamiento de los pesos de las consultas [93][94][95][96] .

La principal limitación de los anteriores SRI lingüísticos es que cuentan con pocos medios para que el usuario pueda expresar sus necesidades de información o su idea del concepto de relevancia. Por ejemplo:

i) No permiten que los elementos de una consulta se ponderen simultáneamente de acuerdo a varias semánticas.

ii) Las entradas y la salida de los SRI se valoran sobre el mismo conjunto de etiquetas S, reduciendo las posibilidades de comunicación entre el usuario y el sistema, a la vez que disminuyendo la expresividad, al utilizar un mismo conjunto de etiquetas para expresar conceptos diferentes.

iii) Solamente consideran la ponderación lingüística de los términos, no contemplando la posibilidad de ponderar los distintos elementos de una consulta (términos, sub-expresiones, conectivos Booleanos y la consulta completa) simultáneamente, lo cual podría contribuir a incrementar las posibilidades de expresión de los usuarios.

En concreto, nuestro aporte en esta memoria consistirá en: i) la caracterización de los perfiles de usuario como consultas clásicas de RI (consultas persistentes), formuladas bajo las disposiciones de los modelos Booleano o Lingüístico; ii) el desarrollo de algoritmos de aprendizaje que permitan generarlas de manera automática, sin que el usuario tenga que interaccionar con el sistema, usando para ello AEs multiobjetivo [41][116][165]; y iii) el uso de Información Lingüística Difusa [187] en la definición del subsistema de consultas [10][17][114] y de su correspondiente subsistema de evaluación para dotar de mayor expresividad a las consultas.

OBJETIVOS

En la comunidad científica de RI, cada vez cobra mayor importancia la idea de que el rendimiento de los SRI puede mejorarse incrementando la participación de los usuarios en los procesos de recuperación [107]. El modelado lingüístico difuso, técnica de SC, es una herramienta que puede contribuir a ello. El uso de información lingüística difusa en el diseño del subsistema de consultas ponderadas facilita a los usuarios la expresión de sus necesidades de información de una forma más natural.

Por otro lado, los AEs constituyen una buena alternativa para resolver algunos de los problemas existentes en el ámbito de la RI, tal y como muestra el gran número de publicaciones aparecidas recientemente en la literatura especializada [38].

Sin embargo, los SRI Lingüísticos existentes no permiten que los usuarios puedan usar las distintas partes de una consulta para expresar sus necesidades de información y ello limita su rendimiento y la expresividad de los usuarios. Las técnicas de aprendizaje automático existentes presentan problemas como la escasez de soluciones (suelen devolver una única solución por ejecución), así como la simplicidad del AE, lo que disminuye su rendimiento. Por tanto, pensamos que el desarrollo de nuevos SRI Lingüísticos que introduzcan más medios para que los usuarios expresen sus necesidades de información y que incorporen técnicas basadas en AEs avanzados y en el paradigma IQBE para asistir al usuario en el proceso de formulación de consultas es una propuesta viable y oportuna.

Más concretamente, los objetivos científicos que se persiguen son los siguientes:

1. ***Estudiar los distintos modelos de RI.*** Nos proponemos llevar a cabo un repaso general a todos los modelos de RI (Booleano, espacio vectorial, probabilístico y Booleano Extendido). Se procederá a una evaluación crítica de estos modelos teniendo en cuenta sus limitaciones y posibles mejoras.

2. ***Estudiar los AEs y su aplicación en RI***. Los AEs constituyen una buena alternativa para resolver algunos de los problemas existentes en la RI. Pretendemos realizar un estudio de esta familia de algoritmos de optimización y búsqueda. Para ello, plantearemos y

justificaremos su aplicación en el campo de la RI, haciendo especial énfasis en los modelos para el aprendizaje de consultas y en los enfoques multiobjetivo.

3. ***Caracterizar los perfiles de usuario como consultas***. Las necesidades permanentes de información de un usuario se representan en forma de perfiles de usuario. La estructura más común para los perfiles es la denominada "bag of words". Buscamos dotar de mayor expresividad a los perfiles representándolos como consultas clásicas de RI que indiquen una necesidad de información del usuario que perdura en el tiempo (consultas persistentes).

4. ***Proponer un enfoque evolutivo multiobjetivo basado en el paradigma IQBE para el aprendizaje automático de consultas Booleanas persistentes***. Los AEs son muy adecuados para generar consultas persistentes por su facilidad para generar consultas con distintas estructuras (consultas Booleanas, Booleanas extendidas, lingüísticas, etc.). Pretendemos usar el enfoque multiobjetivo de este tipo de algoritmos para obtener más de una consulta Booleana (perfil) con distinto comportamiento precisión-exhaustividad en una sola ejecución.

5. ***Diseñar un subsistema de consultas ponderadas con información lingüística difusa y su correspondiente subsistema de evaluación.*** Buscamos dotar de mayor flexibilidad y facilidad de representación a las consultas instantáneas en base a la posibilidad de usar información lingüística difusa multigranular para valorar los pesos.

6. ***Adaptar las técnicas de aprendizaje automático de consultas persistentes para considerar la información lingüística multigranular.*** Con esto, pretendemos combinar los resultados de los objetivos anteriores: las técnicas multiobjetivo para el aprendizaje de consultas y el subsistema de consultas lingüísticas. Buscamos dotar a los perfiles de mayor interpretabilidad representándolos como consultas lingüísticas multigranulares, a la vez que generar varias consultas persistentes de este tipo (perfiles) con distinto comportamiento precisión-exhaustividad en una sola ejecución, haciendo uso de los AEs multiobjetivo.

7. ***Analizar experimentalmente el funcionamiento de los procesos evolutivos de IQBE propuestos empleando las bases documentales existentes del área.*** De este modo, se trabajará con colecciones clásicas (Cranfield y CACM) y se desarrollarán experimentos basados en las consultas más significativas de las mismas.

ESTRUCTURA DE LA MEMORIA

Esta memoria está compuesta por esta Introducción: cinco Capítulos en los que se desarrolla la investigación realizada; una sección de Comentarios Finales en la que se incluyen las conclusiones generales de la misma; y los trabajos futuros, y dos Apéndices.

En el primer capítulo vamos a repasar los conceptos básicos de la RI. Para ello, presentaremos los Sistemas de Información que se emplean en este campo, los denominados SRI, analizando sus componentes principales. A continuación, estudiaremos los distintos modelos de recuperación que se han propuesto en la literatura y, posteriormente, dedicaremos una sección a la evaluación de los SRI, un aspecto muy importante del área. Para finalizar, en las siguientes secciones mostraremos dos técnicas para la mejora del rendimiento de estos sistemas: la retroalimentación por relevancia y el IQBE. Por último, analizaremos la relación entre la RI y el FI.

El segundo capítulo de esta memoria está dedicado al estudio de las técnicas de SC y su aplicación en el desarrollo de los SRI. Para ello, empezaremos planteando y justificando la aplicación de los AEs en diferentes áreas de la RI, como son la indización de documentos, el agrupamiento de documentos y términos, el aprendizaje de consultas o el aprendizaje de funciones de similitud para un SRI específico. Profundizaremos especialmente en las propuestas de aprendizaje de consultas, directamente relacionadas con el tema de esta memoria. Continuaremos con una introducción a la Lógica Difusa y a la Información Lingüística, como herramientas para manejar información imprecisa y tratar los aspectos cualitativos de los problemas, respectivamente. Para terminar, daremos una visión de los diferentes modelos de RI Lingüísticos que han sido propuestos a lo largo de los años.

En el tercer capítulo propondremos un AE multiobjetivo basado en Programación Genética (PG) que permita aprender de forma automática los perfiles que representan las necesidades de información de un usuario. Para dotar de mayor expresividad a los perfiles, en vez de representarlos como la clásica estructura “bag o words”, propondremos hacerlo como consultas clásicas de RI (consultas persistentes [61]), formuladas mediante un modelo de RI Booleano. Para ello, repasaremos los AEs multiobjetivo, las diferentes familias de este tipo de algoritmos y una propuesta multiobjetivo existente para el aprendizaje de consultas Booleanas extendidas, así como los principales componentes de la propuesta de Smith y Smith [165], la cual será nuestro punto de referencia al constituir el algoritmo básico de PG para el aprendizaje de consultas Booleanas.

En el capítulo cuarto nos plantearemos el uso del Modelado Lingüístico Difuso Multigranular para dotar de más flexibilidad y facilidad de representación a las consultas instantáneas en el proceso de RI. Desarrollaremos un SRI basado en información lingüística multigranular y expondremos una serie de ejemplos de uso.

En el último capítulo propondremos combinar los resultados de los Capítulos 3 y 4. Para ello, nos plantearemos usar consultas lingüísticas como consultas persistentes (perfiles de usuario), lo que hará más interpretables los perfiles; y presentaremos un AG multiobjetivo basado en el enfoque IQBE [49] que permita aprenderlas automáticamente a partir de un conjunto de documentos proporcionados por el usuario, solucionando lo que se conoce como “*problema del vocabulario*” en la interacción hombre-ordenador [76]. Finalmente, compararemos nuestra propuesta con uno de los algoritmos clásicos para generar perfiles de usuario que mejor comportamiento presenta basado en el modelo de espacio vectorial [60].

La sección de “Comentarios Finales” resumirá los resultados obtenidos en esta memoria, presentará algunos comentarios sobre los mismos y planteará algunos de los trabajos futuros que se pueden abordar en el área.

Por último, el primer apéndice está dedicado a describir el entorno experimental de aprendizaje automático de consultas persistentes que utilizaremos para evaluar el rendimiento

de las propuestas que introduciremos en los Capítulos 3 y 5. El segundo introducirá la optimización evolutiva multiobjetivo, que consideraremos en nuestras propuestas para la derivación de varias consultas persistentes (perfiles) con diferentes balances entre exhaustividad y precisión en una sola ejecución de los algoritmos.

1.- INTRODUCCIÓN A LOS SISTEMAS DE RECUPERACIÓN DE INFORMACIÓN

En este primer capítulo vamos a repasar los conceptos básicos de la Recuperación de Información. Para ello, presentaremos los Sistemas de Información que se emplean en este campo, los denominados Sistemas de Recuperación de Información, analizando sus componentes principales. Estudiaremos los distintos modelos de recuperación que se han propuesto en la literatura, y posteriormente, dedicaremos una sección a la evaluación de los Sistemas de Recuperación de Información, un aspecto muy importante del área. Para finalizar, en las siguientes secciones mostraremos dos técnicas para la mejora del rendimiento de estos sistemas: la retroalimentación por relevancia y el aprendizaje automático de consultas o "*Inductive Query by Example*". Finalmente, analizaremos la relación entre la Recuperación de Información y el Filtrado de Información.

1.1.- Introducción

Los avances tecnológicos de los últimos cincuenta años han provocado un aumento exponencial de la información. El proceso de digitalización y la transformación de documentos que se está llevando a cabo son dos claros ejemplos de la revolución de la información, la cual ha permitido su acceso a un número ilimitado de usuarios.

Además, hay que tener en cuenta que el uso masivo de las tecnologías y de los ordenadores no se reduce a la producción editorial, sino que está presente en todos los ámbitos de la vida, sobre todo en el trabajo, y hasta en el hogar donde cada vez es mayor el número de personas que no sólo tienen ordenador sino que poseen equipos multimedia.

A ello habría que sumar la distribución de información mediante las llamadas "autopistas de la información", la proliferación de las conexiones ADSL y el coste cada vez menor de los medios de almacenamiento. Todo ello nos sitúa dentro de un entorno en desarrollo de información electrónica a la que se puede acceder por medios automáticos. Otro aspecto que tenemos que considerar es la diversificación de los medios, que trae consigo una mayor

cantidad de información no normalizada, imagen, sonido, texto, etc.

La Recuperación de Información (RI) se puede definir como el problema de la selección de información desde un mecanismo de almacenamiento en respuesta a consultas realizadas por un usuario [5] [153] [155] [176].

Los Sistemas de Recuperación de Información (SRI) son una clase de sistemas de información que tratan con bases de datos compuestas por documentos y procesan las consultas de los usuarios permitiéndoles acceder a la información relevante en un intervalo de tiempo apropiado (véase la Figura 1.1). Estos sistemas fueron originalmente desarrollados en la década de los años 40 con la idea de auxiliar a los gestores de la documentación científica.

Un SRI permite la recuperación de la información, previamente almacenada, por medio de la realización de una serie de consultas *(queries)* a los documentos contenidos en la base de datos. Estas preguntas son *sentencias formales de expresión de necesidades de información* y suelen venir expresadas por medio de un lenguaje de consulta.

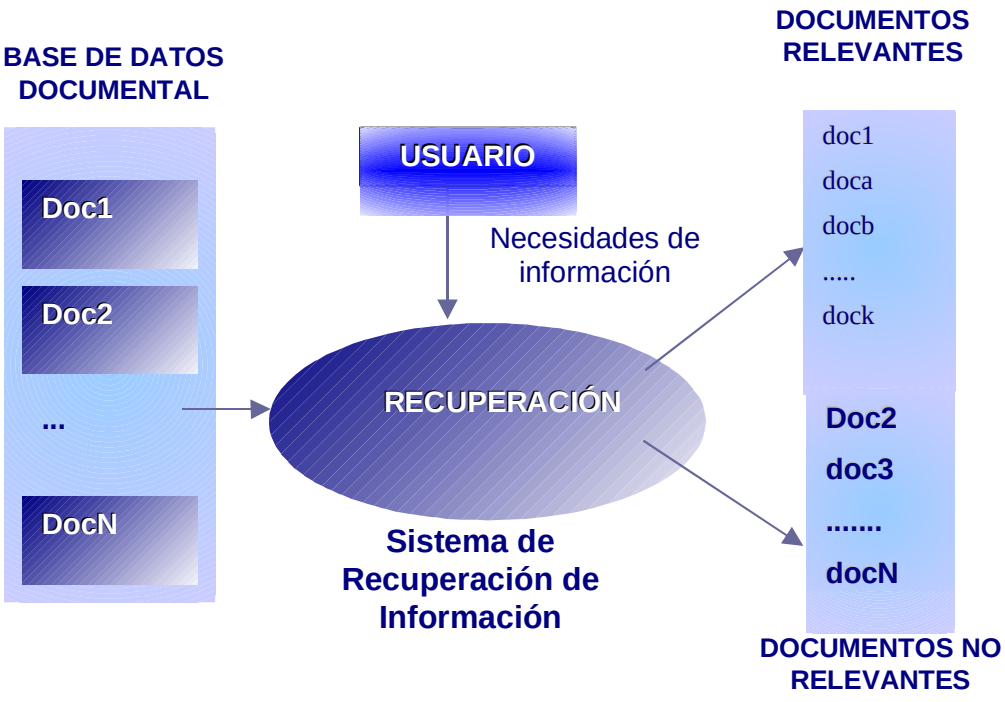


Figura 1.1: Proceso de recuperación de información

Un SRI debe soportar una serie de operaciones básicas sobre los documentos almacenados en el mismo, como son: introducción de nuevos documentos, modificación de los que ya estén almacenados y eliminación de los mismos. Debemos también contar con algún método de localización de los documentos (o con varios, generalmente) para presentárselos posteriormente al usuario. Este proceso se resume gráficamente en la Figura 1.2.

Los SRI implementan estas operaciones de varias formas distintas, lo que provoca una amplia diversidad en lo relacionado con la naturaleza de los mismos. Por tanto, para estudiarlos es necesario establecer en primer lugar una clasificación de estos sistemas. Para ello, veremos a continuación cuáles son los componentes principales de un SRI.

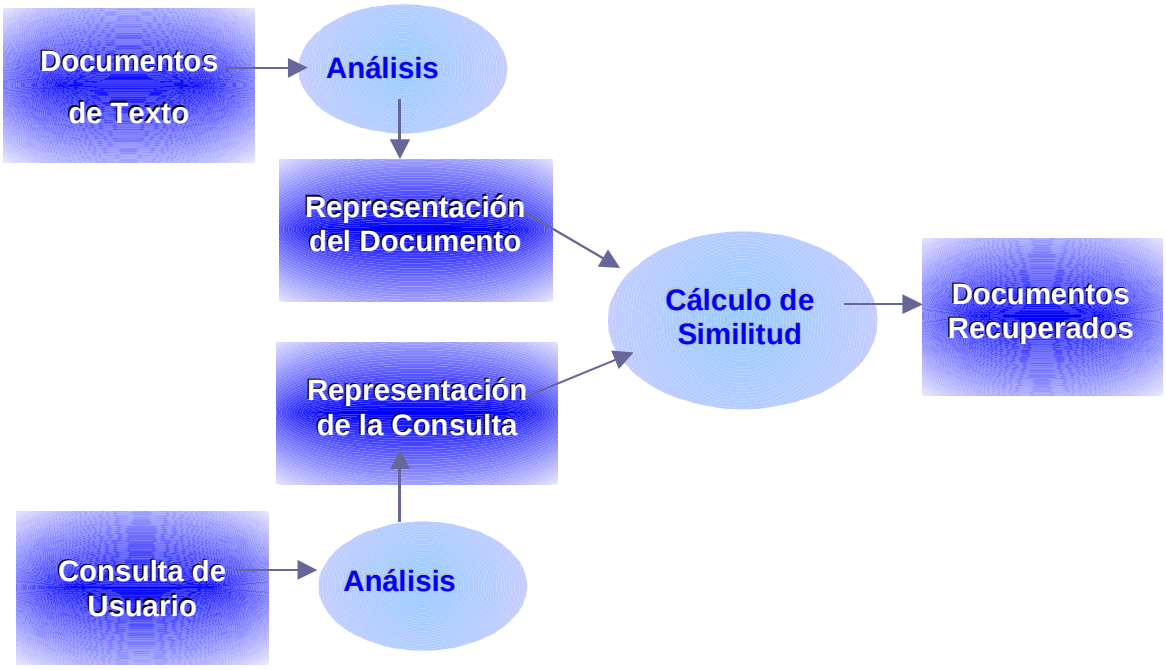


Figura 1.2: Operaciones para la recuperación de documentos

1.2.- Componentes

Un SRI está compuesto por tres componentes principales: la base de datos documental, el subsistema de consultas y el mecanismo de emparejamiento o evaluación (Figura 1.3). Las tres secciones siguientes están dedicadas a estudiar la composición de cada uno de ellos.

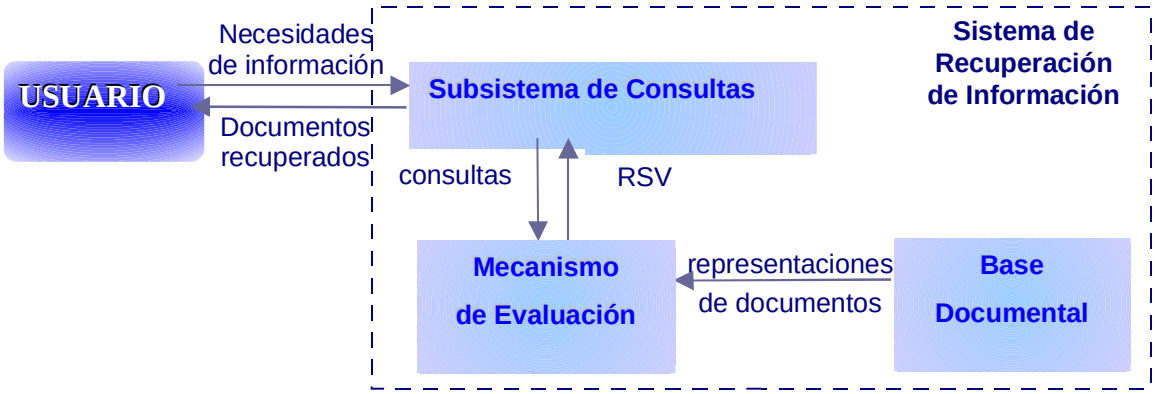


Figura 1.3: Composición genérica de un Sistema de Recuperación de Información

1.2.1.- La Base de Datos Documental

Un documento es un objeto de datos, de naturaleza tradicionalmente textual, aunque la evolución tecnológica ha propiciado la profusión de documentos multimedia, incorporándose al texto fotografías, ilustraciones gráficas, vídeos animados, audio, etc. Estos objetos no se introducen directamente en el SRI, sino que estarán representados por unos elementos llamados *descriptores*. La razón de ser de estos descriptores se basa en darle una mayor eficiencia a la base de datos, la cual será más pequeña, provocando que el tiempo de búsqueda en ella sea mucho menor.

Por tanto, un documento se compondrá de una serie de descriptores. Desde un punto de vista matemático, la base de datos es una tabla o matriz en la que cada columna indica las asignaciones de un determinado descriptor y cada línea o fila representa un documento. En principio, en cada fila aparecen "*unos*" en las columnas relativas a los descriptores asignados al documento y "*ceros*" en las restantes. Así, cada documento estará representado por un vector de ceros y unos [176].

Podemos pensar que esta representación se podría mejorar introduciendo información numérica sobre la asignación de un descriptor al documento en lugar de simplemente 0 y 1. Como veremos a continuación, esta operación se tendría que hacer teniendo en cuenta toda la base documental y el universo de conceptos.

La información numérica de la asignación de un concepto a un documento puede tener diferentes significados dependiendo del modelo de recuperación que se trate. Por ejemplo, en

el modelo de *Espacio Vectorial* [155], que estudiaremos en la sección 1.3.2, puede considerarse como el grado en el que ese descriptor describe el documento; mientras que en el modelo *Probabilístico* [14] (Sección 1.3.3), se considera como la probabilidad de que el documento sea relevante para ese descriptor.

Podemos considerar una base documental D, compuesta por documentos d_i, indizada por un conjunto de términos formado por n términos t_j, en la que cada documento d_i contiene un número no especificado de términos de indización t_j. De esta forma, sería posible representar cada documento como un vector perteneciente a un espacio n-dimensional, siendo n el número de términos de indización que forman el conjunto T:

$$d_i = (t_{i1}, t_{i2}, t_{i3}, ..., t_{in})$$

donde cada uno de los elementos de este vector t_{ij} puede representar la presencia o ausencia del término t_j en el documento d_i en la indización binaria, la relevancia del término t_j en el documento d_i en el modelo de espacio vectorial, o la probabilidad de que el documento d_i sea relevante al término t_j en el modelo probabilístico.

La indización (proceso de construcción de los vectores documentales) puede realizarse de forma manual o automática. En este último caso, la base de datos documental comprende un módulo llamado *módulo indizador* que se encarga de generar automáticamente la representación de los documentos extrayendo los contenidos de información de los mismos. La labor del módulo indizador consistirá en asociar automáticamente una representación a cada documento en función de los contenidos de información de éste, es decir, determinar los pesos de cada término en el vector documental. Su función de indización o ponderación será:

$$F: D \times T \rightarrow [0,1]$$

La representación de cada vector tendrá n componentes, de los cuales los que estén referenciados en el documento tendrán un valor diferente de 0, mientras que los que no estén referenciados tendrán un valor nulo o 0. Es importante señalar que la indización juega un papel fundamental en la calidad de la recuperación puesto que en virtud de ella se definen los coeficientes correspondientes.

De este modo, para obtener estas representaciones se aplica un proceso de "construcción de la base documental". Para ello, solemos partir de una información mucho menos específica, es decir, del estado puro del documento (información textual). Partiendo de esta

información, el sistema realizará un conjunto de operaciones que permitirán obtener la base de datos documental [5] [155]. Dichas operaciones están representadas gráficamente en la Figura 1.4.

Los documentos de tipo textual se pueden representar bien por una componente estructurada en campos (título, autor, resumen, palabras clave...) o bien por una componente no estructurada, es decir, el texto literal. La representación textual de cada documento se basará normalmente en los términos de indización (que pueden ser tanto palabras como frases), los cuales son identificadores de los propios documentos.

El primer paso para la construcción de la base consiste en extraer los términos del texto del documento. Cada una de las palabras se comparará con una lista de palabras vacías o "stoplist", que eliminará las palabras que no tienen interés o carecen de significado propio. Después, las palabras podrán sufrir un proceso de stemming o recorte de sus raíces (por ejemplo, palabras como *informática*, *información*... pueden reducirse a la raíz "*infor*"). Posteriormente, se aplica una función de ponderación para obtener los pesos asociados a cada término en los vectores de documentos y se introducen éstos en la base de datos documental. El proceso completo está representado en la Figura 1.4

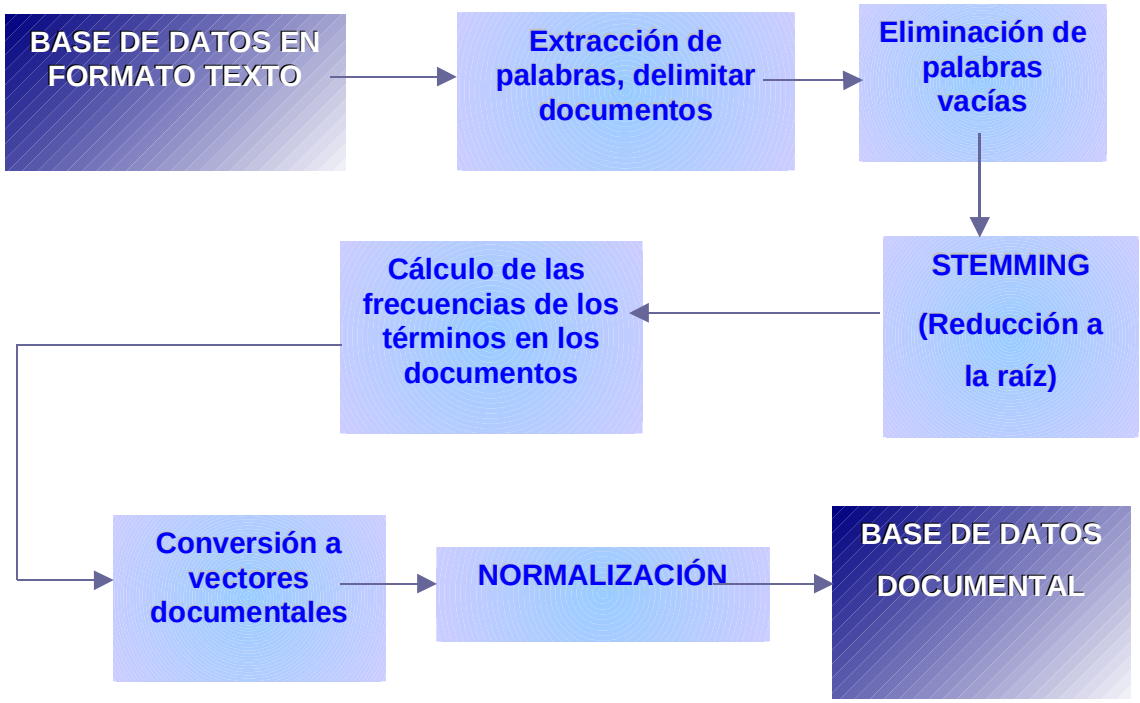


Figura 1.4: El proceso documental

A continuación, analizaremos más detenidamente el proceso que siguen los documentos para pasar a formar parte de la base de datos documental.

1.2.1.1.- Preprocesamiento

El primer paso, incluso anterior a los que hemos nombrado antes, es el denominado "preprocesamiento", el cual consiste en eliminar aquellos fragmentos de texto que no tienen nada que ver con el documento a tratar.

Se trata, por tanto, de un análisis de patrones léxicos en el flujo del texto. Como resultado de este preprocesamiento obtendremos los documentos delimitados y sin cabeceras informativas que no nos sean útiles.

1.2.1.2.- Eliminación de palabras vacías

A continuación, se trata de eliminar las palabras funcionales del lenguaje (artículos, preposiciones, etc.), cuyo valor como términos para la indización y para la discriminación es muy bajo. La eliminación de estas palabras antes del proceso de indización hace que éste sea más rápido, que se ahorre gran cantidad de espacio en los índices y que no influyan en la efectividad de la recuperación.

Para ello, se crea una lista de palabras vacías (en inglés, "stoplist") que nos ayudará a ir eliminando cada una de las palabras carentes de interés y hará que el funcionamiento del SRI sea mejor en la indización automática.

Normalmente, estas listas incluyen las palabras que aparecen con más frecuencia en el idioma en cuestión. Sin embargo, eso no es del todo correcto ni aconsejable ya que, por ejemplo, en la lengua inglesa, los términos que más aparecen en la literatura son: "time", "war", "home", "life", "water" y "world". Por lo tanto, si se siguiera esa regla tradicional, sería muy difícil encontrar un libro cuyo título fuera "La Segunda Guerra Mundial" y lo estuviéramos buscando por el campo "Título". En el caso contrario, también existen varias bases de datos especializadas que contienen términos realmente técnicos que resultarían completamente inútiles al público en general.

De esta forma, la política usada habitualmente para construir una "stoplist" variará dependiendo de la base documental, las características de los usuarios y del proceso de indización.

En cuanto a las formas de eliminación de las palabras vacías, existen principalmente dos:

- ☞ Examinar la salida del analizador léxico y eliminar aquellas que sean realmente vacías.
- ☞ Eliminar las palabras vacías como parte del análisis léxico.

1.2.1.3.- Stemming (Reducción a la Raíz)

Llegados a este momento, tenemos todas las palabras que nos interesan para la indización correcta del documento, pero aún así necesitamos ser un poco más parcos con nuestra información para mejorar el funcionamiento del SRI.

El siguiente paso consiste en ofrecer al usuario la posibilidad de encontrar las variantes morfológicas de los términos que busque. Procederemos por tanto a la reducción a la raíz de las palabras restantes. Este proceso se denomina normalmente "stemming" y se utiliza también para reducir el tamaño de ficheros índice. Almacenando sólo las raíces de los términos en cuestión, se puede llegar a reducir su dimensión hasta un 50%.

La reducción de los términos puede realizarse bien durante la indización o bien en la propia búsqueda. La primera variante presenta la ventaja de ser más eficiente y ahorrar espacio, pero tiene la desventaja de perder información sobre los términos completos.

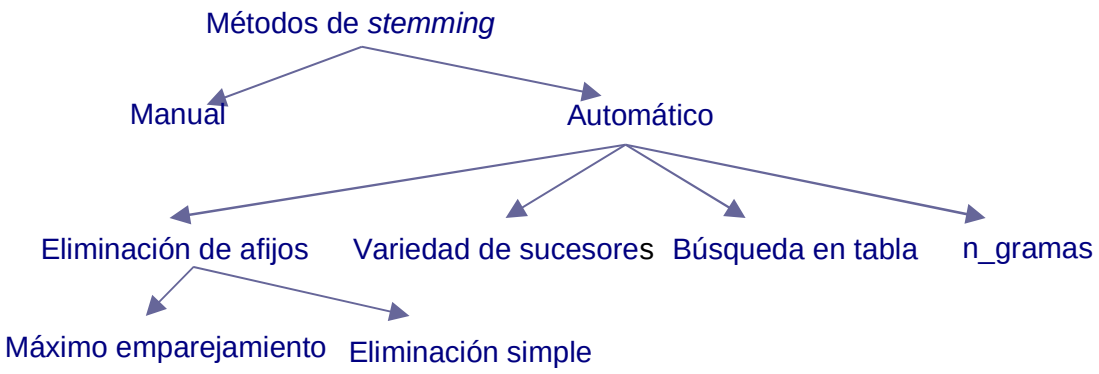


Figura 1.5: Modalidades de stemming

Existen cuatro variedades automáticas de stemming [72] (véase la Figura 1.5) que analizaremos a continuación:

1. Eliminación de afijos: trata de eliminar los prefijos y/o los sufijos de los términos, quedando la raíz. Este método es el más utilizado. Uno de los algoritmos de este tipo más conocidos y empleados es el de Porter [143].
2. Variedad de sucesores: basándose en la frecuencia de las secuencias de letras en un texto.
3. N-gramas: combinación de términos basados en el número de diagramas o n-gramas que comparten.
4. Búsqueda en tabla: en la que están contenidos los términos y sus correspondientes raíces.

Sólo nos resta decir sobre este proceso que el stemming dejará de ser correcto tanto si las palabras se recortan en exceso como si no se recortan lo suficiente, ya que provocaría ruido (recuperación de documentos no relevantes) o silencio (la no recuperación de documentos relevantes).

1.2.1.4.- Vectorización

En este momento, contamos finalmente con todos los términos existentes en la base de datos documental. El siguiente paso consistirá en seleccionar aquellos términos con mayor poder discriminatorio y proceder a su ponderación. Para ello, nos basaremos en la Ley de Zipf, también conocida como la *ley del mínimo esfuerzo*.

Esta ley trata la frecuencia del uso de las palabras en cualquier lengua. Establece que dentro de un proceso de comunicación (sobre todo en el escrito), suelen utilizarse con mayor frecuencia algunos términos, pues los autores suelen evitar la búsqueda de más vocabulario para expresar sus ideas [191].

Zipf formuló la ley de la frecuencia de palabras en un texto, que expresa que si contamos el número de ocurrencias de cada palabra diferente en un texto y que si las palabras encontradas en dicho texto se ordenan en una tabla de modo que la primera palabra sea la más frecuente, la segunda palabra sea la segunda más frecuente, y así sucesivamente, obtenemos una ecuación del tipo [155]:

$$R \times F = C$$

donde:

- R = es el orden de la palabra en la lista
- F = es la frecuencia o el número de ocurrencias de esa palabra
- C = es la constante para el texto

Esto significa que podemos identificar algunos términos de indización para cualificar el contenido del texto. Es decir, podemos indizar el libro, artículo, tesis, disertación, etc. y, lo que es más importante, esto se puede hacer usando procesamiento automático, siempre y cuando se apliquen y respeten ciertas reglas.

Una vez que hemos obtenido todos los términos con mayor poder discriminatorio, es decir, los más representativos y cargados de información, procederemos a la vectorización. Este proceso consiste en la construcción de vectores con el tamaño de los términos significativos que han quedado.

Es decir, un documento d_i se identificará mediante una colección de términos t_{i1}, t_{i2}, t_{i3},…, t_{it} , donde t_{ij} representa el peso, o importancia, del término j en el documento i, como hemos visto al principio de la Sección 1.2.1. Por "término" entendemos una especie de identificador de contenido, como una palabra extraída de un documento, de una frase, o una entrada de un tesauro. Por tanto, una base documental podría representarse como una ordenación, o matriz, de términos donde cada fila de la matriz representa un documento y cada columna representa la asignación de un término específico a los documentos en cuestión (véase la Figura 1.6).

	t_1	t_2	....	t_t
doc_1	t_{11}	t_{12}	....	t_{1t}
doc_2	t_{21}	t_{22}	....	t_{2t}
....	....	....	....	....
doc_n	t_{n1}	t_{n2}	....	t_{nt}

Figura 1.6: Representación matemática de una base documental

A continuación, se construyen los vectores con el tamaño de los términos significativos escogidos finalmente y se les asigna un peso usando la función de ponderación.

1.2.1.5.- Métodos de indización

Como ya hemos comentado, existen dos tipos de funciones de ponderación [155] [5]:

- La función basada en la indización binaria, aquella que indizará con 0 y 1, asignando un 1 a aquellos términos que aparecen al menos una vez en el documento y un 0 a aquellos que no aparezcan ninguna.
- Funciones basadas en pesos reales, que son menos estrictas que la binaria dando pesos encontrados normalmente en el intervalo [0,1].

Dentro de estas últimas se pueden identificar diferentes tendencias.

1) El uso de ponderación de términos basada en la distribución de un término en una colección siempre mejora el rendimiento. La más utilizada es la ponderación IDF (Inverted Document Frecuency), ideada por Salton [153] [155], que utiliza tanto la frecuencia de una palabra en el documento como el número de documentos de la base en los que aparece para calcular su *poder de resolución*. De esta manera pretende indicar la capacidad de discriminación de un término. Es lógico que la capacidad de discriminación sea una función decreciente con respecto al número de documentos de la base en los que aparece la palabra en cuestión, de ahí su nombre Inverted Document Frecuency.

Otras ponderaciones posibles dentro de este grupo son:

Normalizaciones de IDF [166][167][45]

$$IDF_i = \log_2 \frac{N}{n_i} + 1 \qquad IDF_i = \log_2 \frac{maxn}{n_i} + 1 \qquad IDF_i = \log_2 \frac{N - n_i}{n_i}$$

donde N es el número de documentos en la colección, n_i el número total de apariciones del término *i* en la colección y *maxn* la frecuencia máxima de cualquier término en la colección.

Medida de ruido [86]

$$Ruidonormalizado_i = maxnoise - noise_i$$

$$noise_i = \sum_{k=1}^{N} \frac{Frec_{ik}}{TFrec_i} \log_2 \frac{TFrec_i}{Frec_{ik}}$$

donde N es el número de documentos en la colección, *maxnoise* el ruido (noise) más alto de cualquier término en la colección, $Frec_{ik}$ la frecuencia del término *i* en el documento *k* y $Tfrec_i$ la frecuencia total del término *i* en la colección.

Medida de entropía [125]

$$entropia_i = 1 - \frac{\sum_{k=1}^{N} \frac{Frec_{ik}}{TFrec_i} \log_2 \frac{TFrec_i}{Frec_{ik}}}{\log_2 N}$$

donde N es el número de documentos en la colección, *maxnoise* el ruido (noise) más alto de cualquier término en la colección, $Frec_{ik}$ la frecuencia del término *i* en el documento *k* y $Tfrec_i$ la frecuencia total del término *i* en la colección.

2) La combinación de la frecuencia en el documento con la ponderación IDF a menudo ofrece una mejora aún más importante. No obstante, es muy importante normalizar de algún modo la frecuencia en el documento para moderar el efecto de los términos de alta frecuencia y compensar la longitud del documento. Cualquiera de las fórmulas normalizadas de frecuencia en documentos que se muestran a continuación garantizan un buen funcionamiento.

$$cfrec_{ij} = K + (1 - K) \frac{frec_{ij}}{maxfrec_j} \quad [44] \qquad nfrec_{ij} = \frac{\log_2 (frec_{ij} + 1)}{\log_2 longitud_j} \quad [86]$$

donde $frec_{ij}$ es la frecuencia del término *i* en el documento *j*, $maxfrec_j$ la frecuencia máxima de cualquier término en el documento *j* y $longitud_j$ el número de términos distintos en el documento *j*.

1.2.2.- El subsistema de Consultas

Este subsistema está compuesto por la interfaz que permite al usuario formular sus consultas al SRI y por un analizador sintáctico que toma la consulta escrita por el usuario y la desglosa en sus partes integrantes. Para llevar a cabo esta tarea, incluye un *lenguaje de consulta* que recoge todas las reglas para generar consultas apropiadas y la metodología para seleccionar los documentos relevantes.

La interfaz ofrecerá facilidades al usuario a la hora de formular su consulta, ya que éste no tiene por qué saber exactamente el funcionamiento tanto externo como interno del sistema. También se ocupará de mostrar al usuario el resultado de su búsqueda, una vez procesada su consulta.

En muchas ocasiones los usuarios de SRI realizan sus peticiones basándose en la estructura de consultas Booleanas (con operadores Booleanos, es decir, Y, O, NO). Cada uno de los elementos básicos de la consulta puede ser un término (descriptor o concepto), de la forma:

$$\text{Estructura}(q \mid c_1, \ldots\ldots, c_m) = \bigcap_{\text{Subpartes de } q} \left[\bigcup_{c1 \in \text{subparte}} c_i \right]$$

Como hemos comentado, la consulta que facilite el usuario no puede procesarse directamente en su forma original. Ha de recibir un tratamiento previo que consiste en desglosar la consulta en sus componentes básicos, además de comprobar que corresponde con el formato que se espera de ella (es decir, que su composición es correcta y cuadra con las reglas del lenguaje de consulta). Esta comprobación se podrá llevar a cabo tanto a priori como a posteriori. Si se realiza a priori, el sistema directamente no permite al usuario ejecutar su consulta hasta que no esté en el formato correspondiente. Si la comprobación se realiza a posteriori, el sistema devolverá al usuario un mensaje de error o un resultado incongruente.

El análisis de la consulta se llevará a cabo mediante un analizador sintáctico, que

determinará si la consulta es correcta o no y la desglosará en sus componentes. Después de esta partición, podrá llevar a cabo el proceso de stemming para obtener las raíces de los términos de consulta.

Finalmente la consulta se indizará o vectorizará y será enviada al mecanismo de evaluación para estudiar qué documentos se consideran relevantes para las necesidades de información que representa.

1.2.3.- El Mecanismo de Evaluación

Llegados a este punto, tenemos una representación del contenido de los documentos en nuestra base documental y también una representación de las consultas que queremos realizar proveniente del subsistema de consulta. Lo que nos queda por resolver es la selección de los documentos que se consideran relevantes, de entre los documentos que forman la base documental, de acuerdo con los criterios de nuestra consulta.

De esto precisamente se encargará el mecanismo de evaluación que evalúa el grado en el que las representaciones de los documentos satisfacen los requisitos expresados en la consulta y recupera aquellos documentos que son relevantes a la misma. Este grado es lo que se denomina RSV (*Retrieval Status Value*).

Principalmente, existen dos modalidades de evaluación: sistemas que emparejan los documentos individualmente con la consulta, uno por uno; y otros que los emparejan en su conjunto [72]. Dedicaremos la sección siguiente a analizar los modelos de RI más conocidos.

1.3.- Clasificación de los Sistemas de Recuperación de Información

Existen varios modelos de RI y, como en todo, cada uno tiene sus ventajas e inconvenientes. En esta sección haremos una introducción a varios de los modelos existentes y analizaremos las componentes que los forman.

1.3.1.- Modelo Booleano

Este modelo se basa en la teoría del álgebra de Boole. Se denomina Algebra de Boole o Algebra Booleana a las reglas algebraicas, basadas en la teoría de conjuntos, para manejar ecuaciones de lógica matemática. La lógica matemática trata con proposiciones, elementos de circuitos de dos estados, etc., asociados por medio de operadores como Y, O, NO, EXCEPTO, SI...ENTONCES. Por tanto, permite cálculos y demostraciones como cualquier parte de las matemáticas, además de posibilitar la codificación de la información en el ámbito computacional. Se denomina así en honor de George Boole, famoso matemático, que la introdujo en 1847.

A continuación introduciremos las componentes principales de este modelo [176].

1.3.1.1.- Indización de documentos en el modelo Booleano

Dentro de un sistema Booleano, los documentos se encuentran representados por conjuntos de palabras clave. La indización se realiza asociando un peso binario a cada término del índice: 0 si el término no aparece en el documento y 1 si aparece aunque sea una sola vez. Las búsquedas consisten en expresiones de palabras claves conectadas con algún/os operador/es lógico/s (Y, O y NO).

El grado de similitud entre un documento y una consulta será también binario y un documento será relevante cuando su grado de similitud sea igual a 1, de lo contrario el documento no tendrá ninguna relevancia en cuanto a la consulta. Por tanto, en el caso de los SRI Booleanos, la función de indización quedaría así:

$$F: D \times T \rightarrow \{0,1\}$$

1.3.1.2.- El Subsistema de consulta en el modelo Booleano

Como hemos comentado, las consultas en este modelo se compondrán de expresiones Booleanas que comprenden el conjunto de términos T y los operadores Booleanos Y, O y NO. Un ejemplo de este tipo de consultas sería:

$$(t_7 \text{ O } t_2) \text{ Y } (t_1 \text{ Y NO } t_5)$$

Cuando se ejecute la consulta, el subsistema de consulta extraerá el RSV de cada documento y decidirá qué conjunto de documentos es el que se considera relevante para dicha consulta. En este modelo, esta operación es muy sencilla ya que no existe gradación de relevancia (el documento es totalmente relevante a la consulta o no lo es en absoluto). Por tanto, los valores del RSV serán 0 o 1 y formarán el conjunto de documentos recuperados aquellos que tengan el RSV igual a 1.

1.3.1.3.- El Mecanismo de evaluación en el modelo Booleano

El trabajo del mecanismo de evaluación de este modelo consiste en emparejar la consulta *C* con los vectores de los documentos de la base documental para obtener, de este modo, el RSV de cada uno de ellos. Al darse la característica de la coincidencia exacta, propia de este modelo, se trabajará a nivel de conjunto de documentos y no de documentos individuales y, por tanto, el RSV sólo podrá tomar los valores 0 y 1.

Para facilitar la evaluación de consultas Booleanas sobre toda la base documental, se suele seguir una filosofía de trabajo basada en conjuntos de documentos relevantes consistente en representar las consultas en forma de árbol, donde los términos índice serán las hojas (la Figura 1.7 representa claramente el ejemplo de la consulta anterior).

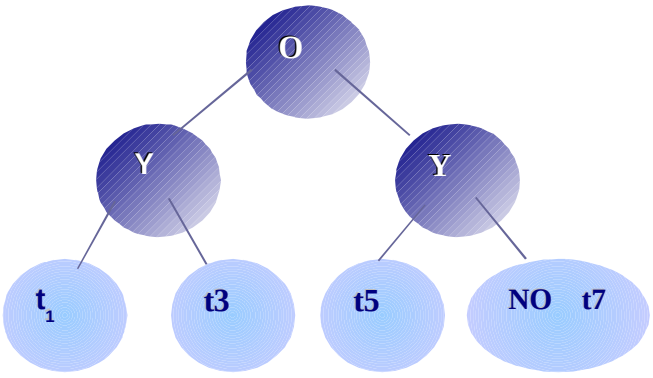


Figura 1.7: Representación de una consulta en forma de árbol

Para obtener el conjunto de documentos relevantes, se recorrerá el árbol de abajo a arriba, es decir, de las hojas a la raíz. Para ello, nos situamos en una hoja y determinamos el conjunto de documentos relevantes para el término situado en ella, es decir, aquellos que tienen dicho término (o que no lo tengan en caso de negación).

Posteriormente, vamos subiendo en el árbol aplicando la operación correspondiente en cada nodo para obtener el conjunto de documentos asociado (intersección de conjuntos para el caso del Y, y unión de conjuntos con el O). Finalmente, el conjunto de documentos devuelto por el sistema es el contenido en el nodo raíz.

La ventaja del modelo Booleano es que es un modelo muy simple, basado en el Álgebra de Boole, lo que le da un marco teórico sólido. Su principal desventaja es el criterio de recuperación binario tan tajante y estricto, por lo que es más un sistema de recuperación de datos que de información.

1.3.2.- Modelo de Espacio Vectorial

Salton fue el primero en proponer los SRI basados en Espacio Vectorial a finales de los 60 dentro del marco del proyecto SMART [155]. Partiendo de que se pueden representar los documentos como vectores de términos, los documentos podrán situarse en un espacio vectorial de *n* dimensiones, es decir, con tantas dimensiones como elementos tenga el vector.

Hace tiempo que se trabaja matemáticamente con espacios n-dimensionales. Situado en ese espacio vectorial, cada documento cae entonces en un lugar determinado por sus coordenadas, al igual que en un espacio de tres dimensiones cada objeto queda bien ubicado si se especifican sus tres coordenadas espaciales. Se crean así grupos de documentos que quedan próximos entre sí a causa de las características de sus vectores. Estos grupos o *clusters* están formados, en teoría, por documentos similares, es decir, por grupos de documentos que serían relevantes para la misma clase de necesidades de información.

En una base de datos documental organizada de esta manera, resulta muy rápido calcular la relevancia de un documento a una pregunta (su RSV), y siendo muy rápida también la ordenación por relevancia, ya que, de forma natural, los documentos ya están agrupados por su grado de semejanza. En la fase de la consulta, cuando se formula una pregunta, también se la deja caer en este espacio vectorial y, así, aquellos documentos que queden más próximos a

ella serán, en teoría, los más relevantes para la misma.

La representación de los documentos y las consultas se realiza mediante la asociación de un vector de pesos no binarios (un peso por cada término de índice). Por ejemplo,

$$d_i = (t_{i1}, t_{i2}, t_{i3}, ..., t_{in})$$

El hecho de que tanto los documentos como las consultas tengan la misma representación dota al sistema de una gran potencialidad.

1.3.2.1.- Indización de documentos en el modelo vectorial

Sea D = {d_1,…., d_m} el conjunto de documentos y T = {t_1,…, t_m} el conjunto de términos índice. El mecanismo de indización de este modelo se presentará de la siguiente forma:

$$F: D \times T \rightarrow I$$

Lo más habitual será trabajar con una función de evaluación normalizada donde los vectores tengan los pesos reales en [0,1].

Como hemos dicho anteriormente, una de las múltiples formas de definir la función F es la *frecuencia inversa del documento* (IDF) [153][155], que presenta la forma siguiente:

$$F(d_i, t_j) = f_{tj,d_i} \cdot \log\left(\frac{N}{n_{t_j}}\right)$$

donde $f_{tj,di}$ es el número de veces que el término t_j aparece en el documento d_i , N es el número de documentos de la base y n_{tj} es el número de documentos en los que aparece t_j.

La bondad de la indización IDF está en que pondera la importancia de los términos en función de su aparición en el resto de los documentos de la base además de su frecuencia de aparición en el documento actual.

1.3.2.2.- El subsistema de consulta en el modelo vectorial

Como hemos indicado, en este modelo tanto las consultas como los documentos tienen la misma representación, es decir, vectores n-dimensionales, donde n es el número de términos índice considerados. Cada una de las posiciones del vector contiene un peso, el cual indica la importancia relativa del término concreto de la consulta o del documento. Este peso es un número real positivo que puede estar o no normalizado.

Cuando un usuario formula una pregunta, la mayoría de los pesos de la misma serán 0, con lo que bastará con proporcionar los términos con peso distinto de 0 para poder definirla. El sistema se encargará de representar la consulta completa en forma de vector n-dimensional de modo automático.

Un ejemplo de consulta para un SRI de Espacio Vectorial (SRI-EV) sería:

	t_1	t_2	t_3		t_n
C =	0	3	6	...	1

Una de las diferencias que existen entre este modelo y el Booleano es que los términos individuales considerados en la consulta no están conectados por ningún operador (ni conjunción, ni disyunción, ni negación). En el modelo vectorial, la consulta se considera como un todo.

La ventaja del modelo vectorial es que permite hacer correspondencias parciales, es decir, ordena los resultados por grado de relevancia.

Su principal inconveniente es que no incorpora la noción de correlación entre términos (problema de todos los modelos clásicos). Aunque este modelo se creó hace cuatro décadas y se ha investigado mucho sobre él, no se ha extendido su uso en los SRI comerciales, donde sigue demandándose el modelo Booleano a pesar de todos sus inconvenientes.

1.3.2.3.- El Mecanismo de evaluación en el modelo vectorial

El mecanismo de evaluación de los SRI-EV empareja la consulta C contra la representación (el vector) asociado a cada documento de la base, $d_i \in D$, para obtener el grado de relevancia RSV del documento con respecto a la consulta, RSV_i. El RSV toma un valor real que será tanto mayor cuanto más similares sean documento y consulta.

Existen diferentes funciones para medir la similitud entre documentos y consultas. Todas ellas están basadas en considerar ambos como puntos en un espacio n-dimensional. Como ejemplo, citaremos las siguientes:

- *Producto escalar*: Corresponde al clásico producto escalar de vectores:

$$RSV(C,D) = \Sigma_{j=1}^{n} d_j \cdot c_j$$

  donde d_j y c_j son, respectivamente, los pesos asociados al término t_j en la representación del documento y la consulta.

- *Medida del coseno*: Está basada en el cálculo del coseno del ángulo que forman ambos vectores, la consulta y el documento:

$$RSV(C,D) = \frac{\Sigma_{j=1}^{n} d_j \cdot c_j}{\sqrt{\Sigma_{j=1}^{n} d_j^2 \cdot c_j^2}}$$

- *Índice de Dice*:

$$RSV(C,D) = \frac{2 \cdot \sum_{j=1}^{n} d_j \cdot c_j}{\sum_{j=1}^{n} (d_j^2 + c_j^2)}$$

- *Índice de Jaccard*:

$$RSV(C,D) = \frac{\sum_{j=1}^{n} d_j \cdot c_j}{\sum_{j=1}^{n} (d_j^2 + c_j^2 - d_j \cdot c_j)}$$

- *Distancia euclídea*: Calcula la distancia existente entre ambos vectores en el espacio:

$$RSV(C,D) = -\sqrt{\sum_{j=1}^{n} d_j^2 - c_j^2}$$

Nótese que, al usar la distancia euclídea habitual, documentos y consultas serían tanto más similares cuanto menor fuera el valor del RSV. Para evitar tener que darle un trato distinto que a las anteriores, se cambia el signo de la medida.

1.3.3.- Modelo Probabilístico

Este modelo se basa en la mejora del rendimiento de los SRI por medio del uso de la información procedente de la distribución estadística de los términos, en tanto que la frecuencia de aparición de un término en un documento o conjunto de documentos podría considerarse un dato relevante a la hora de establecer una consulta a la base de datos documental.

Una forma de ver los SRI es como sistemas que aplican una función de búsqueda que es capaz de diferenciar un documento relevante de otro que no lo es. El modelo probabilístico intenta utilizar la teoría de la probabilidad para construir la función de búsqueda y para establecer su modo de uso [14] [75]. La información utilizada para componer la función de búsqueda se obtiene del conocimiento de la distribución de los términos de indización a lo largo de la colección de documentos o de un subconjunto de ella. Esta información se usa para establecer los valores de ciertos parámetros asociados con la función de búsqueda.

Realmente, la función de la consulta se compone de una serie de pesos asociados a los términos de indización. La diferencia entre este modelo y el vectorial está en el modo de calcular el peso de los términos en la consulta. En el modelo probabilístico, los pesos de los términos que aparecen en los documentos relevantes en una consulta anterior, deberían ser incrementados frente a los pesos de los términos que no aparecían en la consulta previa.

La ventaja más clara de este modelo es que introduce el concepto de probabilidades. En cuanto a sus desventajas observamos que:

- no incorpora la noción de correlación entre términos (problema de todos los modelos clásicos);
- es difícil determinar la asignación de las probabilidades iniciales; y
- no toma en cuenta las frecuencias de aparición de los términos (pesos binarios).

1.3.4.- Modelo Booleano Extendido

Cualquier SRI debe ser capaz de tratar con dos características inherentes al proceso de RI: la imprecisión y la subjetividad [17]. Estos dos factores juegan un papel fundamental en los diferentes estados de procesamiento de la información, tales como:

- en la formulación de las necesidades de información,
- en la estimación del grado en que cada ítem de información es relevante para las necesidades del usuario y
- en la decisión de qué items de información deben recuperarse en función a una petición determinada.

Los SRI Booleanos no incorporan herramientas adecuadas para manejar las dos características anteriores. Debido a ello, los SRI basados en este modelo de recuperación presentan los siguientes problemas:

- Una de sus mayores inconvenientes es la *indización de los documentos*. Un término puede aparecer en un documento y ser más significativo en éste que en cualquier otro. Sin embargo, no existen mecanismos para representar esta distinción en el modelo Booleano. Este inconveniente afecta directamente al *módulo indizador* de la base documental.
- Otra fuente de imprecisión que caracteriza a la RI es el conocimiento vago que el usuario tiene sobre el tema sobre el que está preguntando. Si el usuario es un entendido, le gustaría tener la habilidad de expresar en su consulta la importancia o relevancia que tienen unos términos sobre otros, es decir, expresar la importancia relativa a través del lenguaje de consulta. La incapacidad de realizar esta tarea viene a ser una carencia muy representativa del *subsistema de consulta* de los SRI Booleanos.
- Por último, la recuperación será tajante: 1 si el documento es relevante y 0 si no lo es. El RSV será 0 o 1, sin permitir que exista una gradación en la recuperación que maneje mejor la incertidumbre. Este problema se centra en el *mecanismo de evaluación*.

Sin embargo, a pesar de las carencias anteriores, el modelo Booleano sigue estando muy extendido en el ámbito comercial. Por esta razón, se han llevado a cabo varias extensiones sobre el mismo que permiten salvar algunas de las limitaciones que presenta sin proceder a su completa redefinición. La teoría de conjuntos difusos [186] se ha empleado como herramienta para tal propósito, especialmente por su habilidad para tratar con la imprecisión y la incertidumbre en el proceso de RI. Este hecho se debe fundamentalmente a dos razones principales [16]:

- es un marco formal diseñado para tratar con imprecisión y vaguedad, y
- facilita la definición de una superestructura del modelo Booleano, de forma que los SRIs basados en este modelo pueden modificarse sin tener que ser completamente rediseñados.

El modelo Booleano extendido, resultante de la aplicación de las técnicas difusas al modelo Booleano, extiende a este último en tres aspectos principales:

- La representación de los documentos se convierte en conjuntos difusos definidos sobre el universo de términos, y los términos se transforman en conjuntos difusos definidos sobre el universo de los documentos tratados, introduciendo así un grado de relevancia (relación) entre un documento y un término.
- Se consideran pesos numéricos en las consultas con diferentes semánticas (podemos encontrar una revisión de éstas en [16]), permitiendo así al usuario cuantificar la "importancia subjetiva" de los requisitos de selección.
- Puesto que la evaluación de la relevancia de un documento a una consulta es una proceso impreciso, se introduce el grado de relevancia del documento, RSV. Para ello, el enfoque de coincidencia completa y el conjunto de operadores Booleanos se modelan mediante operadores difusos que ejecutan de forma apropiada el emparejamiento de las consultas y los documentos de tal forma que preservan el significado de los anteriores.

1.4.- Evaluación de los Sistemas de Recuperación de Información

Un SRI puede evaluarse empleando diversos criterios. Frakes [73] selecciona los tres siguientes como los más importantes: *ejecución eficaz* (eficacia), *almacenamiento correcto* y *recuperación efectiva* (efectividad). La importancia relativa de estos factores debe decidirla el diseñador del sistema, y la selección de la estructura de datos y los algoritmos apropiados para su implementación dependerá de esa decisión.

La *eficacia en la ejecución* se medirá por el tiempo que toma el sistema o una parte del mismo para llevar a cabo una operación. Este parámetro ha sido siempre una preocupación principal en un SRI, especialmente desde que muchos de ellos son interactivos y un tiempo de recuperación excesivo interfiere con la utilidad del sistema, llegando a alejar a los usuarios del mismo. Los requerimientos no funcionales de un SRI normalmente especifican el tiempo máximo aceptable para una búsqueda y para las operaciones de mantenimiento de una base documental, tales como añadir y borrar documentos.

La *eficiencia del almacenamiento* se medirá por el número de bytes que se precisan para almacenar los datos. El espacio general, una forma común de medir la eficacia del almacenamiento, es la razón del tamaño de los ficheros índice más el tamaño de los archivos del documento sobre el tamaño de los archivos del documento.

Tradicionalmente, se le ha dado mucha importancia a la *efectividad de la recuperación*, normalmente basada en la relevancia de los documentos recuperados a las necesidades reales de información del usuario, lo cual ha representado un problema ya que medir la relevancia es un proceso subjetivo y sin confianza. Esto es, diferentes juicios personales asignarían diferentes valores de relevancia a un documento recuperado en respuesta a una búsqueda.

Por otro lado, Salton y McGill [155] señalan que, además de los criterios anteriores que se centran principalmente en el punto de vista del diseñador del sistema, se debe considerar también el punto de vista del usuario ya que los criterios de evaluación de diseñador y usuario no tienen por qué coincidir. Los seis criterios siguientes han sido identificados como los más importantes en lo que respecta a las características que un SRI debe ofrecer al usuario [28], [118]:

1. La *exhaustividad*, o habilidad del sistema para presentar todos los items relevantes.
2. La *precisión*, o habilidad del sistema para presentar solamente items relevantes.
3. El *esfuerzo*, intelectual o físico, requerido por el usuario en la formulación de las consultas, en el manejo de la búsqueda y en el proceso de examinar los resultados.
4. El *intervalo de tiempo* transcurrido entre que el sistema recibe la consulta del usuario y presenta las respuestas.
5. La *forma de presentación de los resultados de la búsqueda*, la cual influye en la habilidad del usuario para utilizar la información recuperada.
6. El *alcance o cobertura de la colección documental*, o la proporción en la que están incluidos en la recuperación todos los items relevantes del sistema ya conocidos por el usuario.

Una vez repasados los distintos factores que se pueden considerar en el proceso de evaluación de un SRI, en importante destacar el hecho de que el propio concepto de evaluación puede verse desde dos perspectivas distintas en el área [5], [72], [153], [155], [176]. Existen dos grandes corrientes de investigación en evaluación de la RI, denominadas respectivamente la *corriente algorítmica*, basada en el *modelo tradicional de evaluación*, y la *corriente cognitiva*. Mientras que el primer modelo, el más antiguo, se centra en los algoritmos y en las estructuras de datos necesarias para optimizar la eficacia y la eficiencia de las búsquedas en bases documentales, el segundo, más reciente, considera el papel del usuario y de las fuentes de conocimiento implicadas en la RI [108].

En esta memoria nos centraremos en el modelo tradicional o algorítmico; aún así, dedicaremos los párrafos siguientes a comentar brevemente los conceptos básicos del paradigma cognitivo.

La base del modelo cognitivo la constituyen los trabajos de Dervin y Nilan [54], y Ellis [57], que buscaban proponer una alternativa al modelo clásico de evaluación. Este modelo ha provocado un interés creciente por incorporar a los usuarios en el proceso de evaluación, considerando ésta desde el punto de vista del propio usuario final del SRI.

Desde esta perspectiva, la evaluación se centra en la representación de los problemas de información, el comportamiento en las búsquedas y los componentes humanos de los SRI en situaciones reales, y se fundamenta en la psicología cognitiva y en las ciencias sociales. La búsqueda de información y la formulación de la necesidad de información se contemplan como procesos cognitivos del usuario individual, siendo el SRI y los intermediarios funcionales (como la interfaz del sistema) componentes fundamentales de este proceso de contextualización [139].

La naturaleza compleja de las necesidades de información han puesto de manifiesto que la investigación orientada solamente a técnicas algorítmicas de RI no puede ofrecer una panorámica global del proceso de recuperación. Para lograrla, es necesario incorporar las características del sistema, las características situacionales del usuario y los intermediarios imprescindibles, el más importante de los cuales es la interfaz de usuario, al ser el mecanismo principal de enlace entre este último y el sistema [108].

En el marco del modelo cognitivo, se han propuesto distintas medidas de evaluación del SRI relacionadas con el concepto de relevancia basada en el usuario. Entre ellas se encuentran la *proporción de cobertura* o *alcance* ("*coverage ratio*"), definida como la fracción de documentos relevantes conocidos por el usuario que han sido recuperados, y la *proporción de novedad* ("*novelty ratio*"), que se define como la fracción de documentos relevantes recuperados que son desconocidos por el usuario [112] [5]. También se ha considerado la *satisfacción del usuario* como medida de la eficacia del SRI en este marco de trabajo [112], aunque no se ha propuesto una forma adecuada para medirla al ser un criterio muy subjetivo.

Precisamente, esta última es la mayor crítica al modelo cognitivo, que también utiliza otras medidas como *beneficios y frustraciones*, *utilidad*, etc. que no son objetivas y que no evalúan directamente el sistema sino el efecto que provoca en el usuario.

Para finalizar esta sección, retomaremos el modelo algorítmico de evaluación de la RI para describir algunas de las medidas de eficacia habitualmente consideradas en él, puesto que nos referiremos a ellas en el resto de la memoria.

Se han propuesto múltiples medidas de efectividad de la RI, siendo las más empleadas y conocidas la *exhaustividad* y la *precisión* [176].

La **exhaustividad** es la proporción de documentos relevantes recuperados en una búsqueda determinada sobre el número de documentos relevantes para esa búsqueda en la base de datos, siendo su fórmula:

$$E = \frac{n^{o}\ de\ documentos\ relevantes\ recuperados}{n^{o}\ de\ documentos\ relevantes}$$

La **precisión** es la proporción de documentos relevantes recuperados sobre el número total de documentos recuperados, siendo su fórmula:

$$P = \frac{n^{o}\ de\ documentos\ relevantes\ recuperados}{n^{o}\ de\ documentos\ recuperados}$$

En tanto que se quiere comparar la efectividad del SRI en los términos de exhaustividad y precisión, se han desarrollado métodos para evaluarlos de forma simultánea. De este modo, es habitual trabajar con un sistema de coordenadas en el que un eje es para exhaustividad y otro para precisión (véase la Figura 1.8).

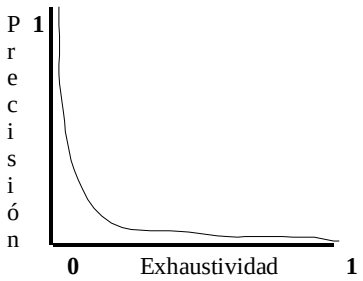


Figura 1.8: Precisión y exhaustividad

En teoría, los puntos de exhaustividad-precisión están inversamente relacionados [27]. Esto es, cuando la precisión sube, la exhaustividad normalmente baja y viceversa. Esto se debe a que, mientras que la precisión da importancia a la ausencia o "no recuperación" de documentos no relevantes, la exhaustividad se basa fundamentalmente en la recuperación de todos los documentos relevantes, aunque esto implique recuperación de documentos no

relevantes. Por tanto, el fin de la precisión es la ausencia de lo que en el ámbito documental llamamos “ruido”, mientras que en la exhaustividad lo que se intenta evitar es el “silencio”.

Una medida de evaluación combinada de exhaustividad y precisión, **M**, ha sido desarrollada por Van Rijsbergen [176] y definida como:

$$M = 1 - [(1 + b^2)\ P\ E\ /\ (b^2\ P + E)]$$

donde {P = precisión, E = exhaustividad}, y **b** es una medida de la importancia relativa, para un usuario, de exhaustividad y precisión. Los investigadores suelen manejar valores de M que reflejen la exhaustividad y precisión que interese al usuario típico.

Como todos los SRI, los basados en el modelo vectorial también se evalúan considerando las medidas de precisión y exhaustividad. Sin embargo, debido a que los SRI-EV devuelven todos los documentos de la base como respuesta a una consulta concreta, es necesario aplicar un tratamiento especial.

Una posibilidad consiste en trabajar con una filosofía de umbral y considerar como conjunto de documentos recuperados final aquel que se obtiene una vez aplicado dicho umbral. En ese caso, se trabajaría de la misma manera que en el caso de los SRI Booleanos.

Sin embargo, este modo de trabajo presenta varios inconvenientes:

- ☞ No es bueno que exista una dependencia del umbral en la evaluación del sistema. Una mala elección de éste puede llevar a pensar que el SRI responde mal a la consulta cuando en realidad lo está haciendo correctamente.
- ☞ Puesto que los SRI-EV presentan la potencialidad de devolver los documentos ordenados según su relevancia, sería interesante que el mecanismo de evaluación fuese capaz de medir también esta capacidad.

Por estas razones, lo más habitual consiste en ignorar el umbral y estudiar el comportamiento del sistema ante una consulta considerando todos los documentos recuperados, es decir, todos los de la base junto con su RSV asociado. Para ello, se procede del siguiente modo:

1. Se fija un conjunto creciente de *p* valores de exhaustividad equidistante en [0,1]. Se consideran estos valores como marcas en el conjunto de documentos recuperados. Nos situamos en el primer documento de la lista.

2. Comprobamos la relevancia del documento actual. Si no es relevante, descendemos en la lista hasta localizar el siguiente documento relevante.

3. Consideramos el conjunto de documentos existentes entre el principio de la lista y el documento actual y medimos la precisión y exhaustividad obtenidas.

4. Descendemos al siguiente documento de la lista y repetimos los pasos 2 y 3 hasta procesar el último documento relevante.

5. Interpolamos los valores de precisión y exhaustividad obtenidos para calcular los asociados a los *p* valores fijados inicialmente

6. Finalmente, calculamos la media de los *p* valores y consideramos esa medida como índice de la calidad del SRI-EV.

Observamos que un valor alto de esta medida indicará un buen comportamiento del sistema. Cuanto mayor sea este valor, más documentos relevantes habremos encontrado entre los primeros de la lista. El valor óptimo es 1 y se obtiene cuando todos los documentos relevantes se encuentran en las primeras posiciones de la lista y no hay ningún documento no relevante entre ellos. La aparición de documentos no relevantes en esas posiciones provoca que las marcas se desplacen hacia abajo y que, consecuentemente, los valores de precisión desciendan.

También es destacable que se puedan calcular tanto la precisión media para una serie de valores de exhaustividad (caso del algoritmo anterior) como la exhaustividad media para una serie de valores de precisión.

Finalmente, diremos que en cualquiera de los dos casos, son prácticas habituales escoger los once (p=11) valores siguientes {0, 0.1, 0.2, 0.3, … , 0.9, 1}, o los tres (p=3) siguientes {0.25, 0.5, 0.75}.

Veamos un ejemplo de cálculo de precisión media [127]. Sean los siguientes 14 documentos de una determinada base documental con un número asignado dentro de la misma. Después de lanzar una consulta, se obtienen en total cinco documentos relevantes para la misma, con sus valores de exhaustividad y precisión correspondientes (véase la Tabla 1.1).

Orden	Número	Relevante	E	P
1	588	x	0.2	1.0
2	589	x	0.4	1.0
3	576		0.4	0.67
4	590	x	0,6	0.75
5	986		0,6	0,6
6	592	x	0,8	0.67
7	984		0.8	0.57
8	988		0.8	0.5
9	578		0.8	0.44
10	985		0.8	0.4
11	103		0.8	0.36
12	591		0.8	0.33
13	772	x	1.0	0.38
14	990		1.0	0.36

Tabla 1: Ejemplo para el cálculo de precisión media

Debido a que hay cinco documentos relevantes, tendremos cinco puntos de exhaustividad-precisión en la gráfica generada (véase Figura 1.9), concretamente los siguientes:

```
(e,p): {(0.2,1), (0.4,1), (0.6,0.75), (0.8,0.67), (1,0.38)}.
```

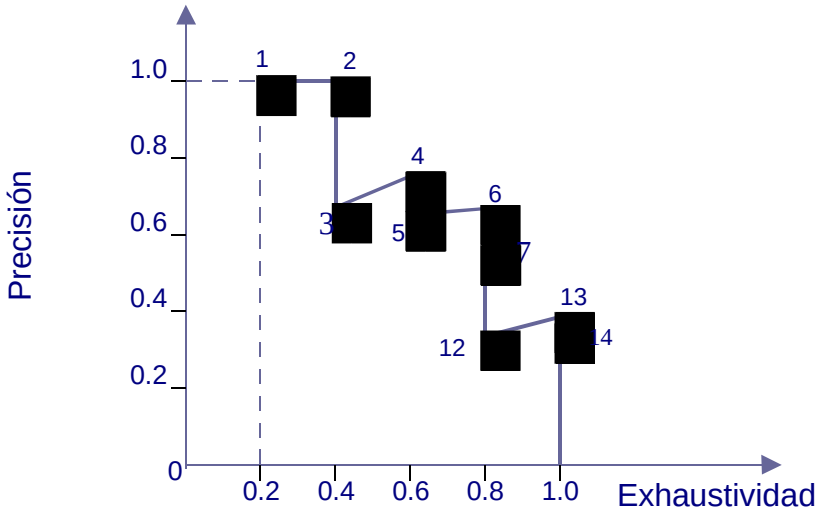


Figura 1.9: Valores de exhaustividad y precisión

Para obtener las 11 marcas necesarias para el promedio, tenemos que interpolar, dando a cada marca de exhaustividad ausente el valor de precisión de la siguiente marca conocida: (e,p)= { *(0,1)*, *(0.1,1)*, (0.2,1), *(0.3,1)*, (0.4,1), *(0.5,0.75)*, (0.6,0.75), *(0.7,0.67)*, (0.8,0.67), *(0.9, 0.38)*, (1,0.38) } (véase Figura 1.10).

Por último, calculamos la precisión media:

$$P = \sum_{m=1}^{11} P_m = 0{,}782$$

De este modo, vemos cómo la interpolación sólo consiste en quitarle los dientes de sierra a la gráfica para evitar que haya más de un valor de exhaustividad para un valor de precisión y viceversa.

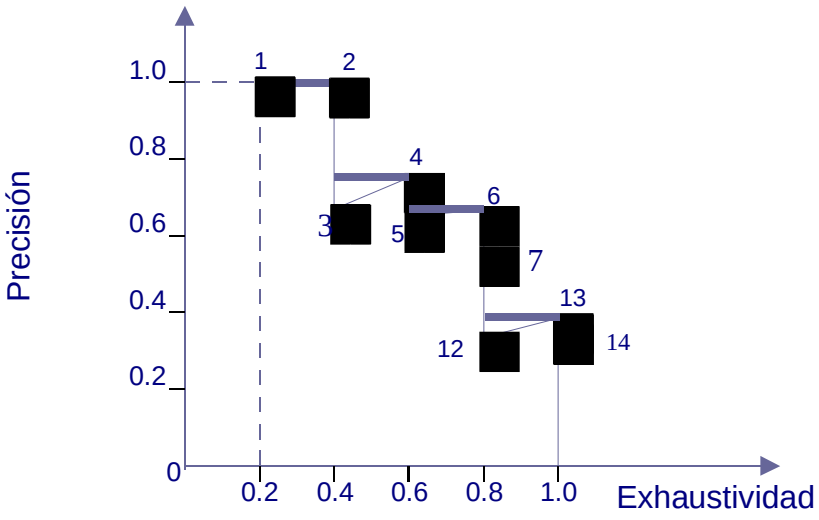


Figura 1.10: Interpolación de valores de exhaustividad y precisión

1.5.- Mejoras en la Recuperación: Retroalimentación por Relevancia

Por muy completo que sea un SRI, por regla general tiene una carencia destacable, la exhaustividad [86]. Los usuarios pueden recuperar algunos documentos relevantes como respuesta a sus consultas, pero casi nunca recuperan todos los documentos relevantes relacionados con las mismas. Existen casos en que esto no tiene mucha importancia para los usuarios, pero en aquellas ocasiones en las que la exhaustividad es un parámetro crítico, hay algunas formas de recuperar más documentos relevantes que los que se recuperaron en un principio.

Una alternativa para resolver este problema es que el SRI ofrezca al usuario la posibilidad de modificar la pregunta original mediante *retroalimentación por relevancia* (*relevance feedback*).

El concepto de retroalimentación (feedback) puede verse desde varias perspectivas diferentes. Pioneros en abordarlo fueron Wiener [181] en Cibernética, Maruyama [132] en Ciencias Sociales y Rocchio [151] en RI. Éste último afirmaba en su obra "*Retroalimentación por relevancia en la recuperación de información*" que a la formulación de la pregunta óptima se llegará mediante una serie de interacciones entre el sistema y el propio usuario, ya que la pregunta inicial que dicho usuario puede realizar es una mera recuperación de datos.

En las operaciones de RI, la mayoría de los usuarios, que no conocen los detalles de la estructura de la colección y del entorno de recuperación, encuentran difícil formular una pregunta bien diseñada para el propósito de recuperación que tienen. Esto sugiere que la primera operación de recuperación debe verse como un tanteo, como una ejecución de prueba solamente, cuyo objetivo es recuperar algunos elementos útiles de la colección dada. Estos elementos inicialmente recuperados pueden ser examinados entonces para ver su relevancia y puede entonces construirse una definición de la consulta, nueva y mejorada, con la aspiración de recuperar elementos adicionales útiles en las siguientes búsquedas.

La retroalimentación por relevancia es la más popular de las estrategias de modificación de consultas. Introducida en los años 60, es un proceso controlado y automático de definición de consultas, sencillo de utilizar y extraordinariamente efectivo. La idea principal consiste en la elección de términos importantes ligados a ciertos documentos que previamente se han identificado como relevantes por el usuario, para incrementar la importancia de estos

términos en la nueva formulación de la pregunta. Análogamente, se puede disminuir la importancia de los términos incluidos en documentos no relevantes previamente recuperados en la futura formulación de la pregunta.

El efecto de este proceso de alteración de la pregunta es el de "mover" la consulta en la dirección de los documentos relevantes y alejarla de los no relevantes, con la esperanza de recuperar así más documentos deseados y menos documentos no deseados en una búsqueda posterior.

Como hemos comentado, el proceso de retroalimentación por relevancia consiste en formular una consulta inicial, ejecutarla sobre el SRI, especificar la relevancia de los documentos recuperados, aplicar un mecanismo de retroalimentación por relevancia para modificar la consulta inicial acercándola a los documentos considerados como relevantes y alejándola de los no relevantes, y ejecutar la nueva consulta obtenida en el sistema. La Figura 1.11 representa gráficamente esta forma de trabajo. Puede observarse que el proceso se efectúa en línea, al requerir interacción con el usuario (que debe proporcionar los juicios de relevancia ante cada ejecución de una nueva consulta), y que es cíclico, al poder repetirse tantas veces como se desee, modificando sucesivamente la ultima consulta ejecutada con el objetivo de obtener nuevos documentos relevantes.

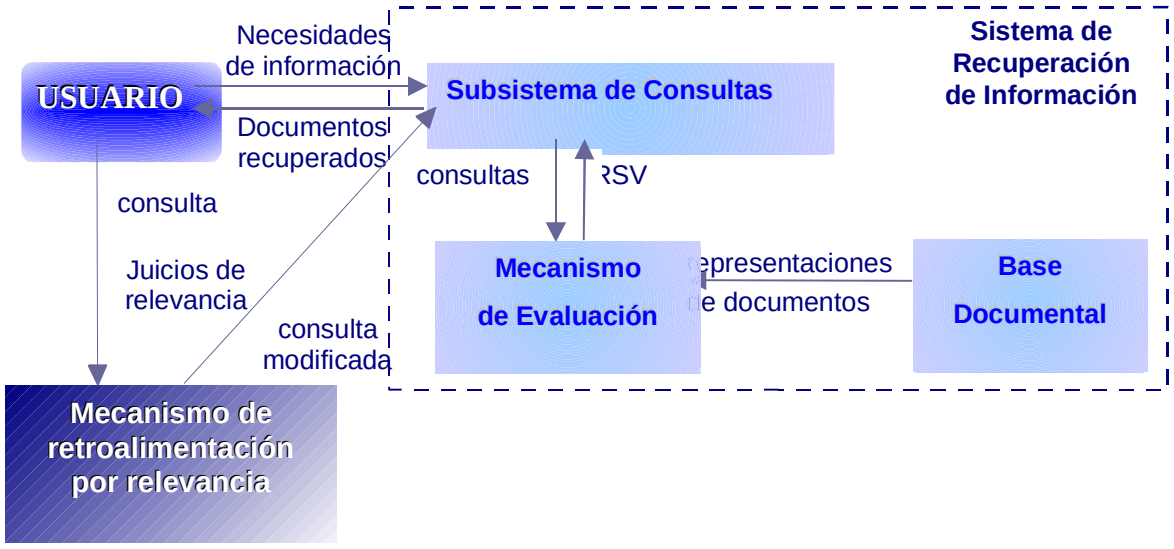


Figura 1.11: Proceso de Retroalimentación por Relevancia

Las principales ventajas de la retroalimentación por relevancia son [127]:

- Es transparente al usuario, de forma que éste puede construir búsquedas útiles sin tener que conocer los detalles del proceso de reformulación de la pregunta, de la estructura de la colección ni del entorno.
- Descompone la operación de búsqueda en una secuencia de pequeños pasos, diseñados para acercarse al área temática gradualmente.
- Proporciona un proceso controlado de alteración de la pregunta que enfatiza algunos términos y perjudica a otros como se requiere en entornos de búsqueda determinados.

La aplicación de la retroalimentación por relevancia está más extendida en los SRI-EV que en otros modelos debido a la sencillez de modificación de las consultas. En este caso, esta modificación puede realizarse mediante dos operaciones distintas:

- Cambiando los pesos de los términos incluidos en la consulta inicial sin alterar los términos considerados en la misma. Para ello, se aumentan los pesos de aquellos términos incluidos en los documentos relevantes recuperados y se disminuyen los de los términos existentes en los documentos irrelevantes recuperados.
- Cambiando los términos existentes en la consulta, añadiendo nuevos términos incluidos en documentos relevantes o eliminado términos existentes en los documentos irrelevantes recuperados. Este último caso suele denominarse *expansión de consultas.*

La técnica de retroalimentación más conocida y empleada en el modelo vectorial se denomina Ide dec-hi [106] y realiza las dos tareas anteriores para generar nuevas consultas.

En cambio, en otros modelos de RI, la retroalimentación es un proceso más complicado debido a la estructura más compleja de las consultas. Así, en los modelos Booleano y Booleano extendido es necesario que el método automático que modifica la consulta sepa cómo conectar los términos mediante los operadores Booleanos [165]. Los primeros enfoques

desarrollados se basaban en conectar estos términos en base a sus frecuencias de aparición en documentos y colecciones [55], [56]. Posteriormente, se han aplicado técnicas evolutivas [2] basadas en Programación Genética [113] para derivar automáticamente las consultas Booleanas y Booleanas extendidas como veremos en el capítulo siguiente. Por último, en el marco de la RI difusa, encontramos también enfoques basados en la modificación de los pesos numéricos asociados a los términos manteniendo inalterada la estructura de la consulta (términos existentes y operadores Booleanos que los relacionan). Es el caso de la técnica propuesta en [157], que analizaremos en la Sección 2.1.3.2 de esta memoria.

1.6.- Aprendizaje Automático de Consultas: Inductive Query by Example

Uno de los problemas principales que los usuarios no expertos encuentran cuando se enfrentan a un SRI es la necesidad de conocer en profundidad el lenguaje de consulta del mismo para poder expresar sus necesidades de información en forma de una consulta interpretable por el sistema que les permita recuperar información relevante. Este problema es habitual en los SRI Booleanos puesto que su lenguaje de consulta, basado, como vimos en la Sección 1.3.1, en la construcción de consultas compuestas de sentencias con términos unidas mediante operadores Booleanos Y y O (que pueden negarse mediante el operador de negación NO), es bastante complejo.

Para resolver el problema de la formulación de consultas en diferentes clases de SRI, se han desarrollado diferentes aproximaciones para ayudar al usuario en dicho proceso [49]. Una de las más conocidas se basa en la generación automática de consultas que describan adecuadamente las necesidades del usuario —representadas en forma de un conjunto inicial de documentos relevantes (y opcionalmente no relevantes) — mediante un proceso *off-line* en el que su interacción no es necesaria. Esta operación se incluye en el paradigma de Aprendizaje Automático [135] y Chen y otros la han denominado *Aprendizaje Inductivo de Consultas a partir de Ejemplos* (IQBE) [49].

La consulta obtenida de este proceso podrá ejecutarse en otros SRI para obtener nuevos documentos relevantes. De esta forma, tal y como se muestra en la Figura 1.12, no es necesario que el usuario interactúe con el sistema como en otras técnicas de refinamiento de consultas como la retroalimentación por relevancia, analizada en la sección anterior.

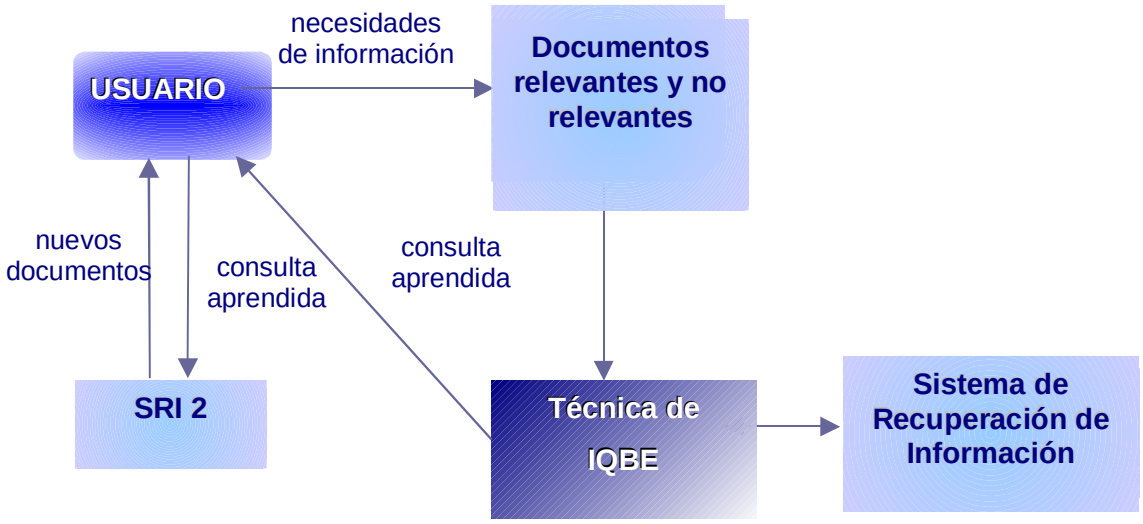


Figura 1.12: Proceso de Inductive Query by Example

Es importante señalar que, en la mayoría de los casos, la diferencia entre el IQBE y la retroalimentación por relevancia es muy sutil. Habitualmente, dicha diferencia se centra únicamente en la existencia o no de la interacción del usuario, la cual es consecuencia del objetivo final del proceso. En el caso de la retroalimentación, esta interacción está presente al encontrarnos en un proceso cíclico, en línea, que persigue refinar (iterativamente y en varios pasos) una consulta ya existente para obtener nuevos documentos relevantes. En cambio, el objetivo final del IQBE es asistir al usuario en el proceso de formulación de la consulta derivando automáticamente una consulta que represente un conjunto de documentos relevantes e irrelevantes para sus necesidades de información, por lo que es un proceso fuera de línea, en un único paso y en el que no se requiere ninguna acción por su parte (aparte, lógicamente, de la tarea previa de determinar el conjunto de documentos inicial).

De este modo, las mismas técnicas de aprendizaje/modificación de consultas pueden ser empleadas en muchas ocasiones para aplicar IQBE y retroalimentación por relevancia. En ambos casos, cada vez que se ejecuta la técnica, recibe como entrada una serie de documentos a los que se tiene que aproximar la nueva consulta y otro grupo de los que se tiene que alejar. La única diferencia es que en el caso de la retroalimentación se parte de una consulta inicial, lo que no ocurre en el IQBE, en el que la consulta se genera de la nada.

Se han propuesto varios enfoques de IQBE para los distintos modelos de RI existentes. Por ejemplo, Smith y Smith [165] proponen un algoritmo Programación Genética para derivar consultas Booleanas de modo automático; y Fernández-Villacañas [65] proponen un

algoritmo de PG y un AG para derivar arboles de consultas Booleanas. Analizaremos más detenidamente ambas propuestas en el Capítulo 2 de la presente memoria.

En lo que respecta al modelo de espacio vectorial, encontramos los métodos de aprendizaje automático considerados en [49]: Árboles de Regresión, Algoritmos Genéticos y Enfriamiento Simulado (*simulated annealing*). Como comentan los autores, la herramienta que mejores resultados proporcionó en su estudio fueron los Algoritmos Genéticos.

Finalmente, en lo que respecta a los SRI difusos, existe un enfoque muy conocido para la derivación de consultas Booleanas con pesos: el algoritmo de Programación Genética de Kraft y otros [116]. Otros trabajos en este área son [39][40][41]. Repasaremos su aplicación en el Capítulo 2.

1.7.- Filtrado de Información *versus* Recuperación de Información.

Hoy en día la recopilación de información en Internet es una actividad compleja para la que los usuarios necesitan sistemas que los ayuden en la selección de la información que les interesa. Los diferentes sistemas propuestos hacen uso de métodos, conceptos y técnicas procedentes de diversas áreas de investigación, tales como: recuperación de información, filtrado de información, inteligencia artificial o ciencia del comportamiento.

A pesar de estar basados en diferentes filosofías, todos estos sistemas comparten su objetivo principal, exponer a los usuarios solamente a información relevante para ellos, de forma que empleen de manera optima el tiempo con que cuentan [84].

De hecho, Belkin y Croft [7] determinaron que el filtrado de información (FI) y la RI son las dos caras de una misma moneda, que trabajan conjuntamente para ayudar a los usuarios en la obtención de información que necesitan para lograr sus objetivos. Usando sistemas de filtrado de información (SFI) podemos depurar la información seleccionada por los SRI de forma que la que sea mostrada a los usuario se adapte lo más posible a sus necesidades.

Así, el FI es un proceso de búsqueda de información donde las necesidades de información del usuario perduran en el tiempo [84][138] y cuya idea es muy simple. Un usuario proporciona sus necesidades de información a un SFI y éste le devuelve una serie de documentos de forma que él indicará que documentos recuperados son relevantes para sus necesidades y cuales no. Una vez obtenida esta información, se procede a almacenar los

documentos relevantes e irrelevantes para esa necesidad de información específica y para ese usuario (*perfil de usuario*). De esta forma, cuando un usuario vuelve a realizar una consulta con unas necesidades de información similares, el sistema tendrá en cuenta qué documentos marcó el usuario como irrelevantes y los tratará como tales. En la Figura 1.13 se representa el funcionamiento de este tipo de sistemas.

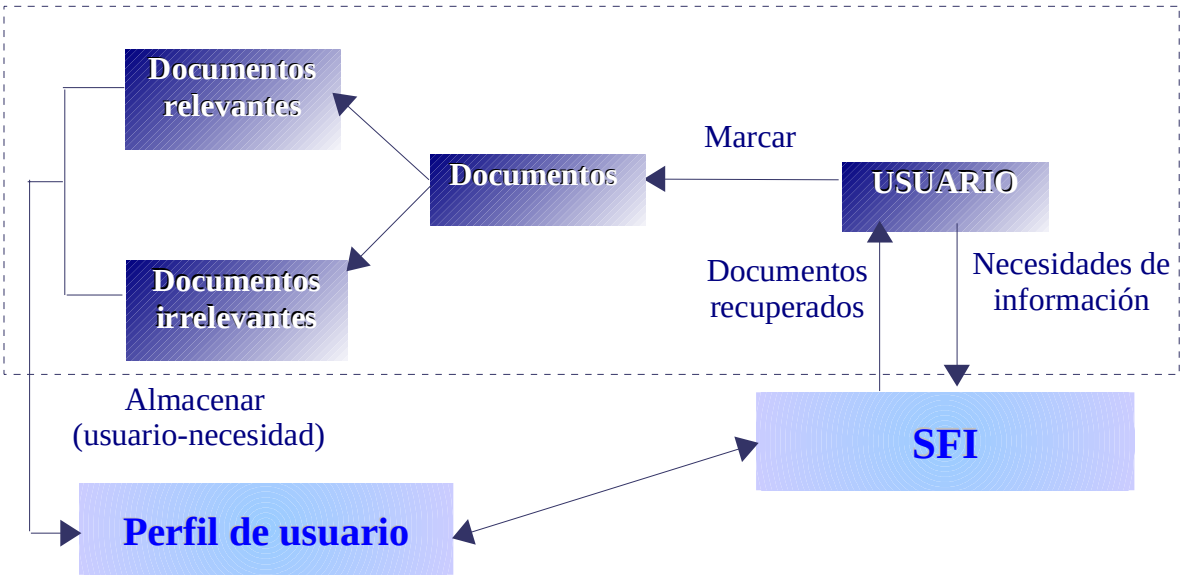


Figura 1.13: Perfil de usuario

A pesar de que ambos sistemas tienen en común su objetivo, la selección de información relevante, difieren en otra serie de aspectos. La Tabla 1.2 recoge algunas de las diferencias existentes.

Parámetro	***Recuperación de información***	***Filtrado de información***
Frecuencia de uso	Uso puntual	Uso repetitivo. Usuarios con necesidades constantes de información.
Representación de las necesidades de información	Consultas	Perfiles
Base de datos	Relativamente estáticas	Datos dinámicos
Tipo de usuarios	El sistema no tiene conocimientos sobre ellos	El sistema tiene almacenados los perfiles de los usuarios

Tabla 2: Comparación entre RI y FI

Las necesidades de información de un usuario en un SFI se representan por medio de un "*perfil*". La estructura más común de perfil es la denominada "bag of words", la cual consiste en un conjunto de palabras clave que representan los intereses del usuario.

Muchos de los sistemas que trabajan con perfiles asumen que será el propio usuario el que defina el perfil, identificando las palabras que lo formaran. Sin embargo, esta forma de trabajo lleva ligada la dificultad con la que se encuentra el usuario a la hora de seleccionar las palabras adecuadas para comunicarse con el sistema. Es lo que clásicamente se conoce como el "problema del vocabulario" en la interacción humano-ordenador [76].

Debido a esta razón, se han aplicado técnicas de aprendizaje automático a la construcción implícita de perfiles [60][138]. En estos casos, el perfil se aprende automáticamente a partir de un conjunto de documentos de entrenamiento proporcionado por el usuario.

Muchos de los algoritmos utilizados para la creación de perfiles aprenden un conjunto de características que podrían ayudar a distinguir documentos relevantes de aquellos que no lo son. En base a las ocurrencias de estas características en un nuevo documento, dicho documento será considerado potencialmente útil y mostrado al usuario, o considerado irrelevante y descartado. La mayoría de los sistemas actuales también asignan pesos a las características con el fin de indicar la importancia que tienen en la estimación de la relevancia [60].

PERFILES COMO CONSULTAS CLÁSICAS DE RI

Belkin y Croft sugirieron que las técnicas de RI podían ser aplicadas con éxito en el FI [7]. De esta forma, un perfil puede ser representado mediante una consulta clásica de RI, las llamadas "*consultas persistentes*" [61].

Representar el perfil como una consulta persistente, en lugar de como la clásica estructura de "bag of words", nos proporciona:

- ☞ Más flexibilidad, al poder usar cualquier modelo de RI para formularla: Booleano, Booleano extendido, Lingüístico, ...
- ☞ Más expresividad, al usar términos y operadores Booleanos (Y, O, NO) para representar las necesidades de información, lo que es más interpretable para el ser humano.

Con esta nueva forma de representar los perfiles de usuario, los nuevos documentos que se presenten al sistema, se consideraran potencialmente válidos si son relevantes a esa consulta.

A hora bien, si los perfiles se pueden representar como consultas clásica de RI, también las técnicas de formulación de consultas, tales como retroalimentación por relevancia o el IQBE, se podrán aplicar al proceso de FI.

Como se ha dicho en el apartado anterior, IQBE es un proceso para asistir a los usuarios en el proceso de formulación de consultas desarrollado por medio de métodos de aprendizaje automático. Trabaja tomando un conjunto de documentos relevantes (y opcional mente no relevantes) proporcionados por el usuario y aplicando un proceso de aprendizaje off-line para generar automáticamente una consulta que describa las necesidades del usuario (véase la Figura 1.12).

Por tanto, las técnicas IQBE trabajan de la misma forma que los métodos de aprendizaje de perfiles explícitos, lo que permite aplicarlas directamente a la construcción de consultas persistentes para el FI. Además, el uso de estas técnicas permitirá aprender no sólo los términos de las consultas (las características), sino también la composición de la consulta en sí.

2.- TÉCNICAS DE SOFT COMPUTING PARA EL DESARROLLO DE SRI

El concepto de Soft Computing (SC), introducido por Zadeh [188] como una sinergia de metodologías (Lógica Difusa, Computación Evolutiva, Redes Neuronales, Razonamiento Probabilístico, ...), tiene como característica básica la tolerancia a la imprecisión, la incertidumbre y la aproximación. Por otro lado, la subjetividad, y la incertidumbre son propiedades típicas de cualquier proceso de búsqueda de información. Como resultado, las técnicas de SC se han revelado como una excelente herramienta para el manejo de la subjetividad y la imprecisión en la definición de los sistemas de acceso a la información [42], como demuestra la gran cantidad de contribuciones que afrontan la aplicación de estas técnicas en el campo del acceso a la información [42][43][47][136]. En concreto, dos de estas técnicas, la Lógica Difusa y los Algoritmos Evolutivos (AEs), están consiguiendo resultados prometedores.

Empezaremos describiendo el concepto de SC y su aplicación a la RI Seguidamente, plantearemos y justificaremos la aplicación de los AEs en diferentes áreas de la RI, como son la indización de documentos, el agrupamiento de documentos y términos, el aprendizaje de consultas o el aprendizaje de funciones de similitud para un SRI específico, profundizando en la propuestas del aprendizaje de consultas, directamente relacionadas con el tema de esta memoria. Continuaremos con una introducción a la Lógica Difusa y a la Información Lingüística, como herramientas para manejar información imprecisa y tratar los aspectos cualitativos de los problemas, respectivamente. Para terminar daremos una visión de los diferentes modelos de RI Lingüísticos que han sido propuesto a lo largo de los años.

2.1.- Soft Computing y su Aplicación a la Recuperación de Información

El término Soft Computing (SC) hace referencia a una familia de técnicas de computación que originalmente, cuando Zadeh, el padre de la lógica difusa, introdujo el concepto [188], estaba compuesta de cuatro técnicas: lógica difusa, computación evolutiva, redes neuronales y razonamiento probabilístico. El término SC distingue estas técnicas de la "hard computing", considerada menos flexible y con más demandas computacionales. La clave de la transición de "hard" a SC estriba en el hecho de que los esfuerzos computacionales requeridos por las técnicas de computación convencionales no es sólo un problema intratable la mayoría de las veces, sino que también son innecesarios, puesto que en muchas aplicaciones la precisión puede ser sacrificada en aras de alcanzar soluciones factibles menos complejas y más económicas. De acuerdo a [189], el principio básico del SC es "explotar la tolerancia a la imprecisión, incertidumbre y aproximación para conseguir robustez, un menor coste de las soluciones y una mayor semejanza con el mundo real.

Todas la metodologías que constituyen el área de SC (las cuatro anteriormente mencionadas y algunas otras que se han incorporado en los últimos años como, por ejemplo, "rough sets" o computacional caótica) son consideradas complementarias de manera que las características que no aparecen en un enfoque están presentes en otro [11]. Por lo tanto, parece conveniente trabajar con sistemas híbridos que combinen dos o más técnicas del área de SC junto con características complementarias.

Una de estas características complementarias es la RI, y la pregunta es: ¿Qué puede hacer el SC por la RI? Crestani y Pasi dieron su visión al responder a esta cuestión en el prefacio de su libro [42]: "nosotros pensamos que un dirección prometedora para mejorar la eficacia de los SRI es modelar la subjetividad y parcialidad intrínseca al proceso de RI, y que dichos sistemas permitan adaptar ciertos parámetros, por ejemplo, pueden aprender el concepto de relevancia que tiene el usuario". En pocas palabras, ellos creen que el uso de SC puede aportar una mayor flexibilidad a los SRI y, en vista de las características de este área de investigación, este parecer ser el caso.

Algunas de las aplicaciones de SC en RI son las siguientes:

- Lógica Difusa: fusión de información, extracción de texto, lenguajes de consulta, y clustering de documentos.
- Redes Neuronales: clasificación y clustering de documentos y términos, y recuperación multimedia.
- Algoritmos Evolutivos: clasificación de documentos, recuperación de imágenes, retroalimentación por relevancia, y aprendizaje de consultas.
- "Rough sets" y lógicas multivaluadas: clustering de documentos.
- Redes Bayesianas: modelos de recuperación, construcción de tesauros, y retroalimentación por relevancia.

Un problema crucial a resolver en RI es el desánimo de los usuarios ante la gran cantidad de ruido que acompaña a los resultados de sus búsquedas, por ejemplo como sucede en Internet, donde la cantidad de información existente es desorbitada. Con el objetivo de minimizar ese ruido y mejorar el rendimiento de los SRI se están utilizando diferentes técnicas de SC en el desarrollo de SRI [42]. En concreto, la Lógica Difusa [186][190] y los AEs [1][2] se han destacado como dos potentes herramientas. La primera está siendo utilizada para modelar la subjetividad y la incertidumbre existente en la actividad de la RI (p.e, en la estimación de la relevancia de un documento respecto a una consulta o en la formulación de una consulta que representa las necesidades de información del usuario); mientras que los Algoritmos Evolutivos se utilizan para adaptar componentes desde el punto de vista del aprendizaje automático.

En las siguientes secciones llevaremos a cabo un análisis más detallado de la aplicación de estas dos técnicas al campo de la RI.

2.2.- Aplicación de los Algoritmos Evolutivos a la Recuperación de Información

En los últimos años, se ha experimentado un interés creciente en la aplicación de técnicas basadas en la IA al campo de la RI con el propósito resolver varios de los problemas considerados en el área. En concreto, el paradigma del A*prendizaje Automático* [135], basado en el diseño de sistemas que presenten la capacidad de adquirir conocimiento por si mismos, parece interesante para el área de la RI [49].

Una de las áreas de la IA con un mayor crecimiento en los últimos años es la Computación Evolutiva [2], la cual se basa en el empleo de modelos de procesos evolutivos para el diseño e implementación de sistemas de resolución de problemas mediante el ordenador.

Los distintos modelos computacionales que se han propuesto dentro de esta filosofía suelen recibir el nombre genérico de *Algoritmos Evolutivos* (AEs) [1]. Existen cuatro tipos de AEs bien definidos que han servido como base a la mayoría del trabajo desarrollado en el área: los *Algoritmos Genéticos* (AG) [101] [79], las *Estrategias de Evolución* (EE) [3] [162], la *Programación Evolutiva* [67][66] y la *Programación Genética* (PG) [113] [6].

Un AE se basa en mantener una población de posibles soluciones del problema a resolver, llevar a cabo una serie de alteraciones sobre las mismas y efectuar una selección para determinar cuáles permanecen en generaciones futuras y cuáles son eliminadas. Aunque todos los modelos existentes siguen esta estructura general, existen algunas diferencias en cuanto al modo de ponerla en práctica. Los AGs se basan en operadores que tratan de modelar los operadores genéticos existentes en la naturaleza, como el cruce y la mutación en un punto, los cuales son aplicados a los individuos que codifican las posibles soluciones. En cambio, las EEs y la Programación Evolutiva aplican transformaciones basadas en mutaciones efectuadas sobre los padres para obtener los hijos, lo que permite mantener la línea general de comportamiento del individuo en su descendencia. Finalmente, la PG codifica las soluciones al problema en forma de programas, habitualmente codificados en una estructura de árbol, y adapta dichas estructuras empleando operadores muy específicos.

Los AE no son específicamente algoritmos de aprendizaje automático, pero ofrecen una metodología de búsqueda potente e independiente del dominio que puede ser aplicada a gran cantidad de tareas de aprendizaje.

Debido a estas razones, la aplicación de los AE a distintos campos de la ciencia se ha incrementado en los últimos años. Un claro ejemplo lo constituye el área de la RI, tal y como demuestra el gran número de publicaciones aparecidas recientemente en la literatura especializada. Entre otros, los AE se han aplicado en la resolución de los siguientes problemas dentro del marco de la RI:

- *Indización de documentos*
- *Agrupamiento* (clustering) *de documentos y términos*
- *Mejoras en la definición de consultas*
- *Aprendizaje de funciones de similitud*
- *Recuperación de imágenes*
- *Diseño de perfiles de usuario para la Recuperación de Información en Internet*
- *Clasificación de páginas web*
- *Agentes de búsqueda en Internet.*

A continuación, comentaremos brevemente la aplicación de los AE en cada una de estas áreas profundizando en la tercera, *Mejoras en la definición de consulta*, concretamente en aquellas que aprenden la consulta completa. Nos detendremos en este tipo de aplicaciones por su relación con nuestras propuestas de los Capítulos 3 y 5, ya que su objetivo es generar automáticamente consultas persistentes (perfiles) que representen las necesidades del usuario.

2.2.1.- Aplicación de los Algoritmos Evolutivos a la Indización de Documentos

Las aplicaciones existentes dentro de este primer grupo están orientadas al aprendizaje mediante adaptación de las descripciones de los documentos existentes en la base documental con objeto de facilitar la recuperación de los mismos ante consultas relevantes. La adaptación se puede llevar a acabo mediante el aprendizaje de los términos que describen a los documentos [80], o de sus pesos [177][174] y mediante el diseño de una función de ponderación para los términos [63][62].

2.2.2.- Aplicación de los Algoritmos Evolutivos al Agrupamiento de Documentos y Términos

En este segundo grupo, destacan dos enfoques distintos, el presentado por *Robertson* y *Willet* para el agrupamiento de términos equifrecuentes en una colección de documentos [145], y la propuesta de *Gordon* para el agrupamiento de documentos cuyas descripciones están siendo adaptadas a lo largo del tiempo [81]. En ambos casos, se utiliza un AG para obtener la configuración de clusters.

2.2.3.- Aplicación de los Algoritmos Evolutivos al Aprendizaje de Consultas

Esta familia de aportaciones de los AEs al campo de la RI es la más numerosa de las estudiadas. Todas las aplicaciones incluidas en este grupo tienen en común el empleo de los AEs bien como técnica de retroalimentación por relevancia (véase la Sección 1.5), bien como algoritmo de Inductive Query by Example (IQBE) (véase la Sección 1.6), en distintos tipos de SRI.

Pueden distinguirse tres subgrupos dependiendo de los componentes de la consulta adaptados en el proceso genético: aprendizaje de los términos, los pesos o la consulta completa (términos, pesos y operadores Booleanos).

2.2.3.1.- Aprendizaje de términos

En [49], *Chen* y otros emplean un AG como técnica de IQBE para aprender los términos de consulta que mejor representan un conjunto de documentos relevantes proporcionados por el usuario.

2.2.3.2.- Aprendizaje de pesos

En [146], *Robertson* y *Willet* proponen un AG con el propósito de determinar un umbral de rendimiento para las técnicas de retroalimentación por relevancia en SRI basados en el modelo espacio vectorial.

Yang y Korfaghe presentan un AG similar al de Robertson y Willet para la misma tarea en [185]. Los dos elementos que cambian con respecto al algoritmo de Robertson y Willet son

el esquema de selección y la función de adaptación.

En [157], *Sánchez* y otros proponen un AG para aprender los pesos de los términos en consultas Booleanas extendidas para SRI difusos. Durante el proceso adaptativo, la estructura de la consulta no varía, únicamente lo hacen los pesos asociados a los términos de la misma.

Otra investigación interesante es la realizada por Horng y Yeh [103] que también consiste en la utilización de un AG para reajustar los pesos asociados a los términos de una consulta con el fin de obtener el vector consulta más cercano al óptimo.

Por último, las propuestas de *López-Pujalte y otros* estudian el uso de AG para la mejora de la retroalimentación por relevancia mediante la adaptación de los pesos de los términos en modelo vectorial. Para ello, en primer lugar estudia las diferentes aplicaciones existentes, determinando las mejores características que debe presentar el AG [128], para en trabajos posteriores centrar su atención en las funciones de evaluación, proponiendo varias nuevas que tienen en cuenta el orden de aparición de los documentos [129].

2.2.3.3.- Aprendizaje de la consulta al completo

Por ultimo, para el aprendizaje automático de consultas completas se han propuesto técnicas tanto para SRI Booleanos como para SRI difusos.

Propuesta de Smith y Smith

Smith y Smith proponen un algoritmo de IQBE para SRI Booleanos en [165]. Por tanto, las consultas consideradas están compuestas por términos unidos por operadores Booleanos, sin considerar pesos. Aunque introducen la propuesta como un algoritmo de retroalimentación por relevancia, los experimentos desarrollados en el trabajo están más cerca de un entorno IQBE. Las componentes del algoritmo están descritas a continuación:

- Las consultas Booleanas se codifican en árboles de expresión, donde los nodos terminales son los términos de la consulta y los nodos internos los operadores Booleanos Y, O y NO.
- Cada generación se basa en seleccionar dos padres, aquellos con mejor valor de adaptación tienen más posibilidades de ser elegidos, y generar dos descendientes.

Ambos descendientes son incluidos en la población actual, aumentando el tamaño de la misma.

- Se considera el cruce usual para PG [113], mientras que no se aplica operador de mutación.
- La población inicial se genera mediante la selección aleatoria de los términos incluidos en el conjunto de documentos relevantes proporcionados por el usuario, teniendo más probabilidad de ser seleccionados aquellos que aparecen en más documentos.
- La función de adaptación considerada proporciona una evaluación compuesta de la recuperación que abarca los dos parámetros principales de la recuperación (precisión y exhaustividad).

Para su experimentación, los autores trabajan con la base documental *Cranfield.* En la práctica, sólo consiguen generar consultas perfectas para aquellos casos en los que la colección inicial de documentos relevantes es pequeña.

Propuesta de Fernández-Villacañas y Shackleton

En [65], Fernández-Villacañas y Shackleton introducen dos técnicas evolutivas de IQBE para el aprendizaje de consultas Booleanas y comparan su funcionamiento. En primer lugar, repasan un algoritmo de PG, denominado BTGP (British Telecom Genetic Programming), que ya publicaron en [64]. BTGP es bastante similar a la propuesta de Smith y Smith descrita en la sección anterior, al ser un algoritmo de PG que evoluciona árboles de consulta Booleanos compuesto por una selección por ruleta, un esquema elitista, el cruce habitual de PG por intercambio de subárboles entre los padres y un operador de mutación basado en intercambiar un subárbol por otro generado aleatoriamente. En lo que respecta a la función de adaptación, trabajan con dos funciones; una clásica que combina linealmente la precisión y la exhaustividad y una nueva variante (procedente del campo del diseño de sistemas de clasificación y aprendizaje automático) basada en la minimización del número de documentos relevantes que no han sido recuperados y del número de documentos irrelevantes recuperados.

En lo que respecta al segundo algoritmo, al que denominan MGA (*Mapping Genetic Algorithm*), se trata de un AG binario clásico que adapta árboles de consulta. Para ello, los autores proponen un esquema de codificación que permite representar árboles de expresiones en forma de cadenas binarias. De este modo, cada nodo del árbol se codifica en una cadena binaria y los cromosomas se obtienen concatenando representaciones de los nodos. Como comentábamos, el AG considerado es un algoritmo binario clásico, con selección por ruleta, elitismo, mutación aleatoria y cruce en un punto. Sin embargo, los autores no emplean el cruce en sus experimentos al indicar que produce convergencia prematura a mínimos locales.

En lo que respecta a la experimentación, consideran dos bases documentales muy simples. Los autores dividen la colección en dos conjuntos de entrenamiento y prueba. Con la primera base de datos, no obtienen buenos resultados en ningún caso, ya que cuando emplean la primera función de adaptación no son capaces de generar consultas que resuman adecuadamente las necesidades de información representadas por los documentos de dicho conjunto, y cuando emplean la segunda, se encuentran con un alto sobreaprendizaje al aplicar las consultas aprendidas sobre el conjunto de entrenamiento a la recuperación de los documentos del conjunto de prueba. Con la segunda base, si consiguen buenos resultados con los dos algoritmos tanto en entrenamiento como en prueba, superando el MGA al BTGP en ambos casos.

Propuesta de Kraft, Petry y otros.

En [116], Kraft y otros propone una técnica IQBE aprender automáticamente consultas Booleanas extendidas (términos, pesos y operadores Booleanos) para SRIs difusos. Se basaba en un algoritmo de PG cuyas componentes se describen a continuación:

- **Esquema de codificación**: las consultas difusas se codifican en árboles de expresión, cuyos nodos terminales son términos de la consulta, positivos o negados, con sus respectivos pesos y sus nodos interiores son operadores Booleanos Y o O.

- **Esquema de selección**: el algoritmo evoluciona según el esquema generacional clásico.

- **Operadores genéticos**: se considera el operador de cruce convencional de la PG, y se selecciona aleatoriamente una de las tres posibilidades siguientes para el operador de

mutación: cambiar un operador, negar un término o cambiar un término negado por su equivalente sin negar.

- **Función de pertenencia**: Kraft y otros proponen dos posibilidades diferentes basadas en las medidas clásicas de la precisión y la exhaustividad. Mientras que una sólo considera la exhaustividad obtenida por la consulta, la otra además tiene en cuenta la precisión.

Para los experimentos, los autores usan una base documental compuesta de 483 resúmenes tomados de ejemplares consecutivos de ACM. Los resultados preliminares indican que la selección aleatoria de términos del conjunto de todos los términos de la población de consultas no se comporta de forma eficiente; para resolver este inconveniente, los términos se seleccionan de aquellos documentos indicados como relevantes.

Propuesta de Cordón, Moya y Zarco

Aunque el algoritmo propuesto por Kraft obtiene buenos resultados, sufre de una de las principales limitaciones del paradigma de la PG: los pesos considerados en la codificación sólo pueden ser alterados mediante mutación. Por lo tanto, al algoritmo le resulta muy difícil obtener los pesos de los términos, constituyendo esto una importante limitación. Con el fin de resolver este inconveniente, Cordón y otros proponen dos aproximaciones dirigidas a mejorar el rendimiento del algoritmo de Kraft.

En la primera aproximación [39], hacen uso del paradigma GA-P [104] basado en la combinación de los AG tradicional con la técnica de PG. Mientras la parte GP del GA-P se encarga de generar las expresiones, la parte GA se encarga de derivar los coeficientes utilizados en las mismas. La parte de la expresión (parte GP) codifica la composición de la consulta —términos y operadores lógicos— y la cadena de valores (parte GA) representa los pesos de los términos

La selección se basa en un enfoque estacionario e induce nichos en la población GA-P [158]. Se consideran dos operadores de cruce diferentes, cruce intra-nicho y cruce inter-nicho, dependiendo de si los padres que van a cruzarse codifican la misma consulta o no. La parte GA se cruza usando el cruce BLX-α [58], mientras que la mutación se lleva a cabo mediante el operador no uniforme de Michalewicz [133]. Los operadores clásicos de PG son aplicados

a las partes de GP.

Por otro lado, en [40], presentan un nuevo algoritmo evolutivo, un híbrido entre enfriamiento simulado y PG, para extender la propuesta de Kraft y otros. De igual forma que en [39], una consulta se codifica almacenando la estructura de consulta en la parte de expresión, y los pesos de los términos en la cadena de valores. El operador de vecino —llamado *macromutación* en [159]— genera una solución vecina a partir del individuo actual mediante un cambio aleatorio realizado bien en la parte expresional, o bien en la cadena de valores.

En ambos trabajos, se propone, además, una extensión para adaptar el umbral de recuperación (como en [157]). En todos los casos, los resultados obtenidos son bastantes significativos, mejorando en un alto grado la propuesta de Kraft y otros en las colecciones de prueba consideradas (entre ellas, Cranfield).

En [41], se desarrolla un nuevo proceso IQBE para derivar automáticamente consultas Booleanas extendidas para SRI Difusos de un conjunto de documentos relevantes proporcionados por el usuario. Esta aproximación permite generar simultáneamente diversas consultas con distintos balances de precisión y exhaustividad en una misma ejecución. Se basan en algoritmos evolutivos avanzados, GA-P, especialmente diseñados para trabajar con problemas multiobjetivo por medio de una técnica multiobjetivo basada en Pareto.

El rendimiento del algoritmo se prueba sobre la clásica colección de Cranfield, usando una filosofía clásica, obteniéndose unos resultados prometedores.

Propuestas de Boughanem y otros

Todas las propuestas de Boughanem y otros [20],[21],[22] están basadas en el uso de técnicas genéticas para la resolución de problemas multimodales en el área de la RI. Las principales características de estos modelos son el uso de operadores genéticos basados en el conocimiento, en vez de los clásicos operadores ciegos, y el uso de técnicas de nichos.

En [20], los autores definen un AG para RI que emplea operadores basados en conocimiento y guiados por una heurística para la resolución de problemas multimodales relevantes. Miden el efecto de las probabilidades de cruce y mutación, del tamaño de la

población, y comparan los operadores basados en conocimiento contra los operadores ciegos. El objetivo es encontrar un conjunto optimal de documentos que mejor representen las necesidades del usuario.

En [21],[22], presentan una aproximación genética que combina los resultados de múltiples evaluaciones de una consulta. Los componentes del AG se describen a continuación:

- Cada individuo genético es una consulta y cada gen es un término índice o un concepto. La población se organiza en diferentes nichos.
- La función de adaptación mide la efectividad de la consulta durante la fase de recuperación.
- El esquema de selección está basado en una variante del esquema usual de selección por ruleta. El operador de cruce está basado en la ponderación de los términos y el operador de mutación explora las ocurrencias de los términos en los documentos relevantes para expandir y/o volver a pondera la consulta.

Además, el AG en [20] usa operadores ciegos y cruce basado en la co-ocurrencia de los términos; mientras que los algoritmos en [21], [22] usan métodos de mezcla.

Varios experimentos de diferente tipo realizados sobre las colecciones TREC validan sus propuestas. En los tres trabajos, los resultados presentados demuestran la efectividad del enfoque genético en el rendimiento de la evaluación múltiple de consultas, así como el interesante uso del dominio del conocimiento para desarrollar operadores genéticos y de los nichos para mejorar la recuperación.

2.2.4.- Aprendizaje de Funciones de Similitud para Sistemas de Recuperación de Información de Espacio Vectorial

El objetivo se centra en usar un AE para generar una medida de similitud para un SRI vectorial personalizada a las necesidades del usuario. Es una nueva filosofía de retroalimentación de relevancia ya que se adapta la función de emparejamiento y no la consulta. Se han propuesto dos variantes diferentes en la literatura especializada:

- Combinación lineal de funciones de similitud ya existentes [142].
- Aprendizaje automático de nuevas medidas de similitud [59] [61].

A continuación describimos en detalle la propuesta de Fan y otros [61] por ser un enfoque que aprende consultas persistentes, basado en AE, uno de los objetivos de esta tesis.

Propuesta de Fan y otros

En [61], Fan y otros proponen un modelo basado en dos etapas para la creación de perfiles de usuario; considerando un perfil como la combinación de una consulta persistente que representa necesidades de información que perduran en el tiempo, y una función de ranking personalizada que estima la relevancia de la nueva información. Cada una de las etapas del modelo se centra en una componente del perfil, como se describe a continuación.

La primera etapa se encarga de construir automáticamente las consultas persistentes utilizando la fórmula del Valor de Selección de Robertson (RSV) [147] como método de construcción, debido a su rendimiento y a que presenta menos dependencia del proceso de emparejamiento [60].

El objetivo de la segunda etapa es obtener la función de ranking “óptima” mediante un proceso adaptativo. Los autores proponen utilizar un algoritmo de PG, donde cada posible función es codificada como un árbol.

Para los experimentos, usan dos conjuntos diferentes de datos, el corpus AP con una amplio dominio de documentos con una longitud media de 450 palabras, y los 10 GB de datos web usados en TREC, con más de un millón de documentos web; y comparan el nuevo modelo con los sistema *Okapi BM25* [150] y con una máquina de soporte vectorial [124], obteniendo resultados muy prometedores.

2.2.5.- Diseño de perfiles de usuario para la Recuperación de Información en Internet.

Los SRIs están limitados por la falta de personalización en la representación de las necesidades del usuario. Una cuestión importante en esta situación es la construcción de perfiles de usuario que mantengan asociados la información previamente recuperada junto con las necesidades previas del usuario. En [131][119][50], se muestran tres propuestas en las que están involucrados perfiles de usuario y AGs.

En [131], se propone un agente que modele las necesidades de información de un usuario a la hora de realizar búsquedas en la red por medio de un proceso adaptativo basado en un AG con genes difusos. El AG representa el conocimiento que se tiene sobre las preferencias y permite retroalimentación por parte del usuario. La teoría de conjuntos difusos se incluye para trabajar con la imprecisión presente tanto en la preferencias del usuario, como en la evaluación de los documentos recuperados. Este sistema es un enfoque viable para aprender la necesidades de información del usuario y mantenerlas al día utilizando sólo la retroalimentación por relevancia que proporcionada por el usuario.

En [119], Larsen y otros presentan un esquema que permite trabajar con el conocimiento, basado en la experiencia, que se tiene sobre las preferencias del usuario representándolo como un perfil. Éste se genera a partir de un proceso de filtrado en vez de un proceso de recuperación donde no se consideraba información alguna sobre el usuario. De esta forma, los autores filtran la colección de documentos usando la información recuperada gracias a la primera consulta y generan el perfil, usando un AG para encontrar los términos más discriminatorios, p.e., aquellos que permiten al sistema discernir entre documentos relevantes e irrelevantes, que se seleccionan y almacenan como parte del perfil para ser usados en consultas futuras.

En [50], Chen y otros proponen un sistema adaptativo de consultas flexibles (ASQ) para mejorar la eficacia de los perfiles de usuario. ASQ consta de dos componentes principales: un sistema de consultas flexibles on-line y un mecanismo de aprendizaje off-line basado en un AG. Este sistema proporciona resultados personalizados integrando la información sobre una imagen y los perfiles de usuario empleando una técnica de agregación basada en lógica difusa. El AG se usa para mejorar los perfiles a través de la información proporcionada por el usuario. Los experimentos demuestran que la eficacia de la recuperación se incrementa de

manera significativa.

2.2.6.- Otras aplicaciones.

Otras áreas en las que se han aplicado los AE son:

- Recuperación de imágenes [51][109][169].
- Clasificación de páginas web [126].
- Agentes de búsqueda en Internet [9][105][163][178][179].

2.3.- Lógica Difusa e Información Lingüística.

El concepto de Lógica Difusa fue presentado por Zadeh en los años 60 [186], como un medio para representar y manipular datos que no eran precisos. Puede verse, por tanto, como una extensión de los sistemas de lógica clásica que proporciona un marco de trabajo para tratar con el problema de la representación del conocimiento en un entorno de incertidumbre e imprecisión.

La importancia de la Lógica Difusa proviene del hecho de que la mayoría de las formas de razonar de los humanos son aproximadas, es decir, llevan asociada incertidumbre e imprecisión.

2.3.1.- Introducción

La Lógica Difusa, como su nombre indica, es una lógica, alternativa a aquella con la que hemos trabajado siempre, que pretende introducir un grado de incertidumbre en las sentencias que califica. Para entender esto, observemos cómo funciona la teoría de conjuntos clásica.

Supongamos el conjunto de números naturales {0,1,...10}. Si intentamos agrupar "números inferiores o iguales a 5" en un subconjunto, el proceso de razonamiento de la lógica a la que estamos acostumbrados será:

```
Es cierto que 0 <= 5 ? SI (no cabe duda)
Es cierto que 6 <= 5 ? NO (no cabe duda)
```

proceso que repetiríamos para los diez números del conjunto inicial, hasta obtener el subconjunto deseado, {0,1,2,3,4,5}. La lógica tradicional funciona a la perfección.

El inconveniente de esta lógica es que en la vida real nos encontramos frecuentemente con criterios de clasificación que no son tan tajantes como el ejemplo anterior. Por ejemplo, dado un conjunto de personas, se las intenta agrupar según su altura. Las personas no son sólo altas o bajas; la mayoría pertenecen a grupos de altura intermedia. La gente suele ser "mas bien alta" o "mediana". Casi nunca las calificamos con rotundidad, porque el lenguaje que usamos nos permite introducir modificadores que añaden imprecisión: un poco, mucho, algo...

Como la lógica tradicional es bivaluada (solo admite dos valores: o el elemento pertenece al conjunto o no pertenece, sin más), se ve maniatada para agrupar al anterior conjunto de personas. Las personas serán altas o bajas. La solución que presentará la lógica de siempre será definir un umbral de pertenencia (por ejemplo, un valor que todo el mundo considera que, de ser alcanzado o superado, la persona en cuestión puede llamarse alta). Si dicho umbral es 1.80, todas las personas que midan 1.80 o más serán altas, mientras que las otras serán bajas. Según esta manera de pensar, alguien que mida 1.79 será tratado igual que otro que mida 1.60, ya que ambos han merecido el calificativo de personas 'bajas'.

Si dispusiéramos de una herramienta para caracterizar las alturas de forma que las transiciones entre las que son altas y las que no lo son fueran suaves, estaríamos reproduciendo la realidad mucho más fielmente. Por ejemplo, cojamos el grupo de personas 'altas'. Es evidente que en la realidad hay unos puntos de cruce donde las personas dejan de ser altas para ser consideradas medianas, de forma que el concepto de 'alto' decrece linealmente con la altura. Asignando una función lineal para caracterizar el concepto 'alto' en lugar de definir un sólo umbral de separación estamos dando mucha más información acerca de los elementos. Esta función, como veremos, se llamará función de pertenencia.

Se pueden extraer dos conclusiones:

- En la lógica convencional hay una perdida inherente de información acerca de los elementos de los conjuntos siempre que el criterio de clasificación sea vago (no tenga umbrales definidos).
- La imprecisión puede ser introducida de forma lingüística (es decir, usando palabras y modificadores de nuestro lenguaje.)

Por lo tanto, la Lógica Difusa puede interpretarse como un superconjunto de la tradicional lógica Booleana, que ha sido extendida para manejar el concepto de “parcialmente verdadero” (valores de verdad entre “absolutamente verdadero” y “absolutamente falso”).

2.3.2.- Conceptos Básicos

Conjuntos difusos

La Lógica Difusa se fundamenta en el concepto de conjunto difuso [186] y, al igual que se verifica una estrecha relación entre la lógica clásica y el concepto de subconjunto, así también

ocurre con la Lógica Difusa y la teoría de conjuntos difusos.

Para un universo U, se define un subconjunto difuso A de U como un conjunto de pares de la forma

$$A = \{<x, \mu_A(x)> \mid x \in U\}$$

donde $\mu_A(x)$ es la función de pertenencia del conjunto A, que generalmente toma valores en el intervalo [0,1].

Función de pertenencia

La función característica χ_B de un conjunto clásico (B) determina que elementos del universo de discurso (U) pertenecen a B. Puesto que en los conjuntos clásicos los elementos pertenecen o no, χ_B asignará a cada elemento de U, un elemento del conjunto {0,1}.

La función de pertenencia μ_A de un conjunto difuso (A) también asigna a cada elemento del universo un valor de pertenencia al conjunto pero, en vez de considerar esta pertenencia como absoluta, la considera gradual. Por lo tanto, podemos definir la función de pertenencia como aquella aplicación que asocia a cada elemento de un conjunto difuso el grado con que pertenece a dicho conjunto.

La función Figura 2.1 muestra la descripción y la gráfica de una función de pertenencia trapezoidal.

$$\mu_A(x,a,b,\alpha,\beta) = \begin{cases} 0, & x < \alpha \\ \dfrac{x-\alpha}{a-\alpha}, & \alpha \leqslant x \leqslant a \\ 1, & a \leqslant x \leqslant b \\ \dfrac{\beta - x}{\beta - b}, & b \leqslant x \leqslant \beta \\ 0, & x > \beta \end{cases}$$

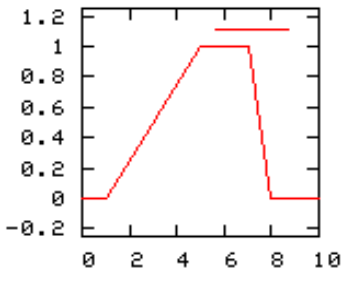


Figura 2.1: Función de pertenencia trapezoidal

Soporte

Se define el soporte de un conjunto difuso A en el universo U, como el conjunto formado por todos los elementos de U cuyo grado de pertenencia a A es distinto de 0.

$$supp(A) = \{x \in U, \mu_A(x) > 0\}$$

Altura

Se define la altura de un conjunto difuso A como el mayor grado de pertenencia de todos los elementos de dicho conjunto.

$$h(A) = max\{\mu_A(x) | x \in U\}$$

α-corte

El α-corte de un conjunto difuso A es un conjunto formado por todos los elementos del universo U cuyos grados de pertenencia en A son mayores o iguales que el valor $\alpha \in [0,1]$.

$$\alpha_A = \{x \in U \mid \mu_A(x) > \alpha\}$$

Conjunto de niveles

Se denomina conjunto de niveles de un conjunto difuso, y se representa como L(A), al conjunto de grados de pertenencia de sus elementos.

$$L(A) = \{ a \mid \mu_A(x)=a, x \in U \}$$

2.3.3.- Operaciones con conjuntos difusos

Las operaciones lógicas que se pueden establecer entre conjuntos difusos son la intersección, la unión y el complemento, igual que las que usamos en lógica bivaluada. Mientras que el resultado de operar dos conjuntos clásicos es un nuevo conjunto clásico, las mismas operaciones con conjuntos difusos nos darán como resultado otros conjuntos también difusos.

En Lógica Difusa, hay muchas maneras de definir estas operaciones. Cualquier operación que cumpla las condiciones de una t-norma puede ser usada para hacer la intersección, igual que cualquier t-conorma puede ser empleada para unir conjuntos difusos. Las t-conormas especifican un conjunto de condiciones que deben reunir aquellas operaciones que deseen ser

usadas para unir conjuntos, mientras que las t-normas hacen lo propio para las intersecciones.

La Tabla 2.1 muestra las propiedades que deben cumplir los dos familias de funciones y algunos ejemplos de funciones de cada familia.

	Propiedades	***Ejemplos***
T-Normas *T: [0,1] x [0,1] --> [0,1]* $\mu_{A\cap B}(x)=T[\mu_A(x),\mu_B(x)]$	**Conmutativa**: T(a,b)=T(b,a) **Asociativa:** T(a,T(b,c))=T(T(a,b),c) **Monotonía**: T(a,b) > = T(c,d) si a>=c, y b>=d **Condiciones frontera** T(a,1)=a	**Intersección estándar** T(a,b)=min(a,b) **Producto algebraico** T(a,b)=a · b **Intersección drástica** $T(a,b)=\begin{cases} a & si \quad b=1 \\ b & si \quad a=1 \\ 0 & en\ otro\ caso \end{cases}$
T-Conormas *S: [0,1] x [0,1] --> [0,1]* $\mu_{A\cup B}(x)=S[\mu_A(x),\mu_B(x)]$	**Conmutativa**: S(a,b)=S(b,a) **Asociativa:** S(a,S(b,c))=S(S(a,b),c) **Monotonía**: S(a,b) > = S(c,d) si a>=c, y b>=d **Condiciones frontera** S(a,0)=a	**Unión estándar** S(a,b)=max(a,b) **Suma algebraica** S(a,b)=a+b -a · b **Unión drástica** $S(a,b)=\begin{cases} a & si \quad b=0 \\ b & si \quad a=0 \\ 1 & en\ otro\ caso \end{cases}$

Tabla 3: T-normas y T-conormas

Una característica importante de este tipo de funciones es son parejas duales. Cada t-norma tiene asociada su correspondiente t-conorma y viceversa.

Estas operaciones se definen de las siguiente manera:

- Complemento: $\mu_{\sim A}(x)=1-\mu_A(x)$
- Intersección: $A \cap B=\{(x,\mu_{A\cap B}), \mu_{A\cap B}(x)=T[\mu_A(x),\mu_B(x)]\}$
- Unión: $A \cup B=\{(x,\mu_{A\cup B}), \mu_{A\cup B}(x)=S[\mu_A(x),\mu_B(x)]\}$

En las figuras 2.2 y 2.3 podemos ver algunas de estas operaciones de forma gráfica.

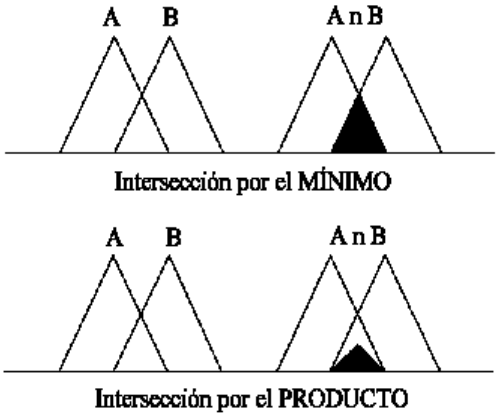


Figura 2.2: Intersección de conjuntos difusos

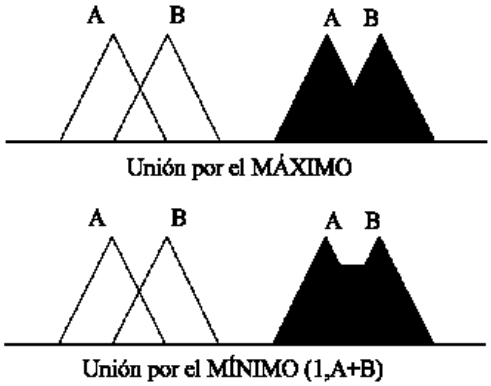


Figura 2.3: Unión de conjuntos difusos

2.3.4.- Información Lingüística

El uso de información lingüística es una técnica apropiada para tratar con los aspectos cualitativos de los problemas que modela los valores lingüísticos por medios de variables lingüísticas [187]. Puesto que las palabras son menos precisas que los números, el concepto de variable lingüística es útil para proveer una medida que sirva para caracterizar de manera aproximada los fenómenos que son demasiado complejos o están mal definidos para ser sensibles a una descripción por medio de los términos cuantitativos convencionales. Su uso es beneficioso porque presenta un marco de trabajo más flexible para representar la información de manera más directa y conveniente cuando no es posible expresarla con exactitud. Así, la imposición de tener que cuantificar un concepto cualitativo es eliminada y el sistema se simplifica.

DEFINICIÓN DE VARIABLE LINGÜÍSTICA

Una variable lingüística [187] se caracteriza por una quíntupla *(L,H(L),U,G,M)* en la que

- *L* es el nombre de la variable.
- *H(L)* (o simplemente *H*) denota el conjunto de términos of *L*, por ejemplo, el conjunto de nombres de los valores lingüísticos de *L*, siendo cada valor una variable difusa expresada genéricamente como *X* que toma valores en el universo de discurso.

- *U* es el universo de discurso el cual está asociado con la variable base *u*;
- *G* es una regla sintáctica (normalmente con forma de gramática) para la generación de los nombres de los valores de *L*.
- *M* es la regla semántica encargada de dar significado a cada *L*, *M(X)*, el cual es un subconjunto difuso de *U*.

Por ejemplo, consideremos la variables lingüística *edad*, es decir, *L = edad*, con *U* = [0,100] y la variable base $u \in U$. El conjunto de términos asociados con edad podría ser *H(L) = {joven, maduro, viejo}* donde cada término en *H(edad)* es el nombre de un valor lingüístico de *edad*. El significado *M(X)* de una etiqueta $H \in H(edad)$ se define como la restricción *H(u)* sobre la variable base *u* impuesta sobre el nombre de *H*. Por lo tanto *M(X)* es un conjunto difuso de *U* cuya función de pertenencia *H(u)* representa la semántica del nombre *H*. La Figura 2.4 ilustra este concepto.

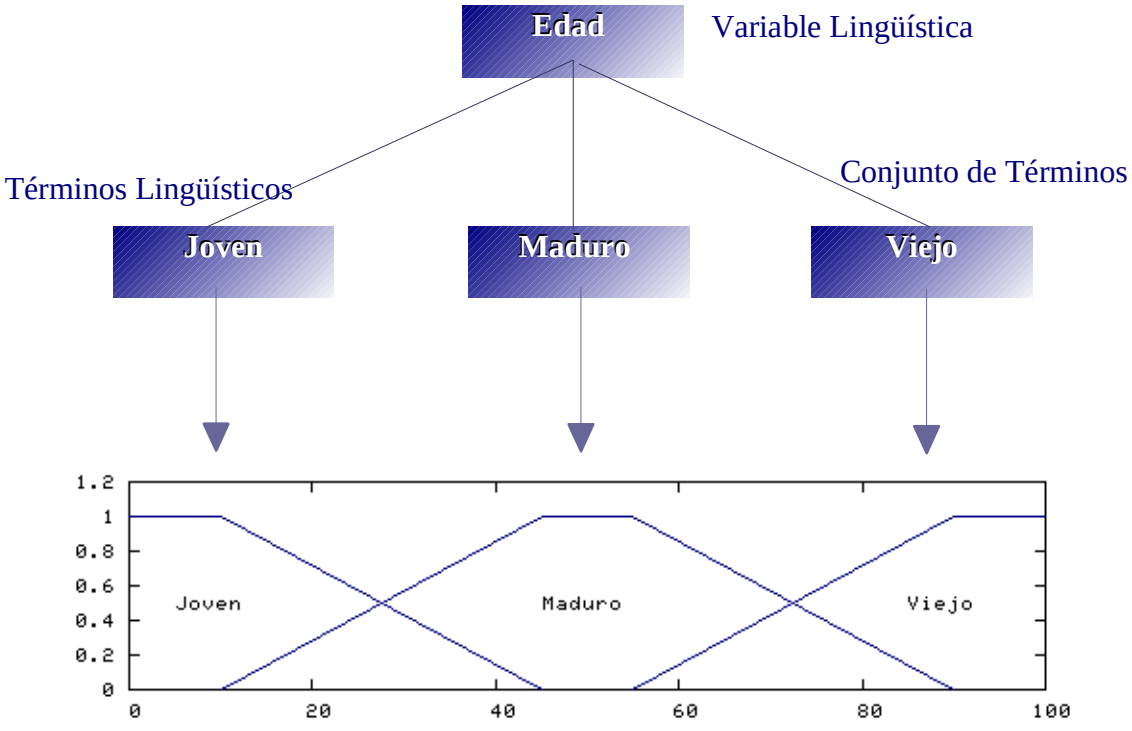


Figura 2.4: Ejemplo de una variable lingüística

2.3.5.- Enfoque Lingüístico para la Resolución de Problemas

El enfoque lingüístico se aplica cuando las variables que intervienen en un problema son de carácter lingüístico en vez de numérico [187]. Por ejemplo, en situaciones donde intervienen los individuos, los cuales usan más bien descriptores lingüísticos que numéricos para dar sus opiniones. Con ello, se consigue modelar de forma más directa y apropiada gran cantidad de problemas reales, ya que nos permite representar la información de los individuos (casi siempre poco precisa) de manera muy aproximada a como inicialmente ellos se expresan.

En cualquier aplicación que use el enfoque lingüístico, para resolver un problema en particular, hay que tomar dos decisiones [89]:

- Elección del conjunto de términos y su semántica.
- Elección del operador de agregación de información lingüística.

La elección del conjunto de términos lingüísticos y su semántica para representar la información lingüística es el primer objetivo que hay que satisfacer en cualquier enfoque lingüístico para resolver un problema concreto. Desde un punto de vista práctico, podemos encontrar dos posibilidades para elegir la descripción adecuada del conjunto de términos y su semántica:

- ***Un enfoque lingüístico clásico***. La primera posibilidad define el conjunto de términos lingüísticos por medio de una gramática libre de contexto, y el significado de los términos viene representado por número difusos que a su vez son descritos por funciones de pertenencia basadas en ciertos parámetros o reglas semánticas [187][171].
- ***Un enfoque lingüístico ordinal***. La segunda define el conjunto de términos por medio de una estructura ordenada de términos lingüísticos, y el significado de los términos se deriva de su propia estructura ordenada, la cual podrá estar o no simétricamente distribuida en el intervalo [0,1] [88][172][184].

En ambos casos, un aspecto importante que es necesario analizar con el fin de establecer la descripción de una variable lingüística es la granularidad de la incertidumbre, por ejemplo,

el nivel de discriminación entre los diferentes grados de la incertidumbre. En otras palabras, la cardinalidad del conjunto de términos lingüísticos usado para expresar la información lingüística. Esta cardinalidad debe ser lo suficientemente pequeña para no imponer una precisión inútil a los usuarios, y lo suficientemente rica para permitir una discriminación de las valoraciones en un número limitado de grados. Valores típicos de cardinalidad, usados en los modelos lingüísticos, son valores impares, tales como 7 or 9, con un límite superior de 11 o no más de 13. Estos valores clásicos de cardinalidad parecen chocar con la observación de Miller sobre el hecho de que los humanos no son capaces de manejar razonablemente más de 7 ± 2 niveles diferentes de cuantificación [134]. En el enfoque lingüístico clásico, no es fácil mantener bajo control la granularidad de la incertidumbre debido a que la gramática puede generar una lista de descriptores muy amplia, lo que incluiría valores inadecuados de cardinalidad (por ejemplo, muy altos). Sin embargo, en el enfoque lingüístico ordinal, se puede controlar este aspecto ya que permite escoger el conjunto de descriptores. Éstos suelen ser pocos, aunque significativos y útiles.

2.3.6.- Aplicación de la Lógica Difusa a la Interpretación de los Pesos de un SRI

Como comentamos en la sección 1.3.4, el modelo Booleano extendido surgió al aplicar las técnicas difusas al modelo Booleano. Una de la extensiones realizadas fue la consideración de pesos numéricos en las consultas con diferentes semánticas, lo que permite al usuario cuantificar la “importancia subjetiva” de los requisitos de la selección.

En las siguientes secciones vamos a describir tres interpretaciones existentes para los pesos.

- ***Importancia relativa***; permite al usuario expresar la importancia de cada término en la consulta [13][144][156].
- ***Umbral***; considera los pesos como umbrales, premiando al documento cuyo grado de pertenencia para el término *t* sea mayor o igual que el grado de pertenencia del término en la consulta pero permitiendo algún valor de coincidencia parcial cuando el grado de pertenencia del documento es menor que el umbral[144][24].

- ***Documento perfecto***; especifica que la descripción difusa de la consulta representa qué descripción ideal difusa del documento debería darse para satisfacerla [25][15].

2.3.6.1.- Semántica de importancia relativa

Define los pesos de las consultas como medidas de la importancia relativa de cada término con respecto a los demás en la consulta. El usuario asignará pesos de importancia relativa a cada término dentro de la consulta para establecer una dominancia de unos sobre otros a la hora de efectuar la recuperación. Así, un término con peso 0 debe considerarse completamente irrelevante, sin ningún efecto sobre el cálculo del RSV.

Con esta semántica, una consulta puramente Booleana puede interpretarse como la formulación de una petición en la cual todos los términos son igualmente relevantes para el usuario con un peso implícito de valor 1. Inicialmente, esta semántica fue formalizada dentro de la teoría de conjuntos difusos definiendo la función de evaluación E como un producto [144]:

$$E(d,<t,w>) = w \cdot F(d,t)$$

Sin embargo, existe un problema con el operador Y. Al modelarse como el mínimo, es dominado por el término de menor peso y genera inconsistencia con la semántica de importancia relativa, que requiere que el término con menor peso contribuya en menor medida en el RSV. La decisión se basará en el término menos importante, que es justo lo contrario de lo que desea el usuario.

Para solventar con este problema, se propuso el uso de las t-normas e implicaciones en la definición de la función E para efectuar la recombinación de pesos [13],[156]. De este modo, se hará depender la definición de la función E, que recombina el peso y el término, del operador Y u O que ligue el término en cuestión en la consulta compuesta. Si en la consulta aparece una disyunción aplicaremos una t-norma mientras que si se trata de una conjunción aplicaremos una t-conorma. En la Tabla 2.2 vemos la forma que tomarían las funciones para las parejas empleadas, cada una de las cuales implementa un modelo de recuperación diferente.

t-norma	***implicaciones***
Mínimo min(x,y)	Dicene max(1-x,y)
Lukasiewicz Max{x+y-1,0}	Lukasiewicz Min{1-x+y,1}
Fodor $\begin{cases} min(x,y) & si\ x+y>1 \\ 0 & en\ otro\ caso \end{cases}$	Gödel $\begin{cases} 1 & si\ x \leqslant y\ 1 \\ y & en\ otro\ caso \end{cases}$
Fodor $\begin{cases} min(x,y) & si\ max(x,y)=1 \\ 0 & en\ otro\ caso \end{cases}$	Fodor $\begin{cases} 1 & si\ x \leqslant y\ 1 \\ max(1-x,y) & en\ otro\ caso \end{cases}$

Tabla 4: Parejas de t-normas e implicaciones

Otro aspecto de la semántica de importancia relativa que puede causar cierta dificultad en el contexto de teoría de conjuntos difusos es el concepto de *complemento*. Una consulta Booleana divide la colección de documentos D en dos subconjuntos: los que son recuperados y lo que no. Como consecuencia de esto, la consulta con negación debería hacer la misma división pero con los nombres cambiados. Esto no sucede cuando introducimos pesos de "importancia relativa": si una consulta *q* se asocia a un peso *w* que actúa como factor multiplicativo, su función de evaluación es $w \cdot E^*(d,q)$. La función de evaluación para la consulta negada NOT *q* con un peso asociado *w* es $w \cdot (1-E^*(d,q))$. Obsérvese que ambas funciones no son complementarias.

2.3.6.2.- Semántica de umbral

Diferentes autores [16],[114],[144] han propuesto una solución alternativa para superar los problemas que plantea la semántica de importancia relativa. Se trata de las semánticas de umbral para consultas ponderadas: un peso *w* asociado a un término de la consulta requiere que los documentos se evalúen chequeando sus grados de importancia *F(d,t)* contra el umbral *w*. Dicho de otra forma, los pesos indicarán un umbral que tendrá que ser superado para que el documento se considere relevante a una consulta. De este modo, cuanto más pequeño sea el umbral, mayor será el número de documentos recuperados. Un umbral cero significará

recuperar todos los documentos sobre el tópico.

La consulta Booleana se interpretará como el caso en el que todos los pesos están asociados con un peso implícito con valor cero. Así, la formulación de consultas con pesos acotados es un criterio más radical que el uso de pesos de importancia relativa para especificar los documentos deseados.

Con esta semántica, el peso actuará como un selector sobre el subconjunto de documentos que contienen el término, es decir, definirá el modelo de documento mínimamente aceptable. Además, los pesos no indican ninguna relación entre los términos de la consulta. Cada peso especifica un requisito que ha de ser satisfecho por cada F(d,t).

2.3.6.3.- Semántica de Documento Perfecto

Dentro de esta tercera semántica, la consulta se va a interpretar como la especificación de una colección perfecta de documentos o como el conjunto ideal de documentos que satisfacen la necesidades del usuario [25]. Cuando la consulta se expresa de una Forma Normal Disyuntiva (DNF), cada disyunción identifica una clase de documentos perfectos o ideales, o dicho de otro modo, la consulta (con sus pesos) será una representación o patrón del documento óptimo que el usuario quiere recuperar.

Por tanto, el proceso de evaluación de la consulta debe seleccionar todos los documentos que sean los más similares al ideal como sea posible. Este matiz implica que el usuario debe ser capaz de especificar los documentos ideales de una forma precisa y compatible a la representación del documento. De este modo, el peso del término de la consulta tiene la misma semántica que el peso del término índice. Será un término índice perfecto que especificará el grado de relación ideal que los documentos deseados deben tener respecto al término especificado.

En el modelo de Bordogna, Carrara y Pasi [15], el peso w actúa como una restricción sobre los términos índice del documento, F(d,t), requiriendo que estén lo más cerca posible a w. Según sea w alto o bajo, el usuario declara su interés sobre documentos que tratan mucho o no mucho acerca de un término. De esta forma, una consulta Booleana se interpretará como un caso en el que los términos no negados deben tener un peso *w* igual a 1 y los términos negados un peso *w* igual a 0.

2.4.- SRI Lingüísticos

Los SRI son sistemas que nos permiten identificar de forma eficiente y eficaz aquellos documentos en una colección que mejor se adecuan a las necesidades de información de un usuario, expresadas mediante una consulta. Como hemos visto en el Capítulo 1, un SRI está compuesto por tres componentes principales:

- *Base de datos*: almacena los documentos y la representación de su contenido (términos índice).
- *Subsistema de consulta*: permite a los usuarios formular sus consultas por medio del lenguaje de consulta.
- *Subsistema de evaluación*: evalúa el grado en el que los documentos satisfacen los requisitos expresados en las consultas, tras lo cual asigna a cada documento un valor de relevancia (RSV).

El subsistema de consulta soporta la interacción usuario-SRI y, por lo tanto, debería tener en cuenta la imprecisión y la vaguedad típica de la comunicación humana. Este aspecto puede ser modelado introduciendo pesos en el lenguaje de consulta. Muchos autores han propuesto SRI ponderados usando la Teoría de Conjuntos Difusos para su modelado [13][15][18][24][23][25][115][136][180]), asumiendo el uso de pesos numéricos (valores en [0,1]). Sin embargo, el uso de lenguajes de consulta basados en pesos numéricos fuerza al usuario a cuantificar conceptos cualitativos (como “importancia”), ignorando el hecho de que muchos usuarios no son capaces de proporcionar exactamente sus necesidades de información si las expresan de forma cuantitativa, pero sí si lo hacen de forma cualitativa.

Parece más natural, por lo tanto, caracterizar el contenido de los documentos asociando descriptores lingüísticos los términos de una consulta, tales como “importante” o “muy importante”, en vez de valores numéricos. En este sentido, se han propuesto varios modelos de RI lingüísticos que usan un enfoque difuso-lingüístico [187] para modelar los pesos de las consultas y la relevancia de los documentos.

En las secciones siguientes describiremos brevemente algunos de los SRI Lingüísticos que se han propuesto en la literatura.

2.4.1.- SRI Lingüístico Ponderado basado en un Enfoque Clásico.

En [17], Bordogna y Pasi proponen una extensión lingüística de un SRI Booleano ponderado, formalizado con la Teoría de los Conjuntos Difusos.

Basado en el enfoque clásico, en este SRI las consultas se definen como generalizaciones de las consultas de los modelos ponderados. Con este fin, las autoras introducen descriptores lingüísticos: a)en el lenguaje de consulta, para permitir expresar la importancia que un término debe tener en los documentos que se recuperen, b)en el mecanismo de clasificación, para etiquetar los documentos recuperados en clases de equivalencia.

Base de datos

Los documentos se representan de forma difusa, de manera que el significado de un término que describe a un documento se puede expresar como un número en el rango [0,1], conocido como término índice ponderado.

La representación de los documentos se basa en la noción de conjunto difuso, en el que la transición de pertenencia o no pertenencia de un elemento es gradual en vez de abrupta. Un documento se tratará como un conjunto difuso de términos, *M(d)*:

$$M(d) = \{t, \mu_d(t) \mid t \in T\}$$

$\mu_{d(t)}$ indica como de significativo es el término *t* en la representación del documento *d.*

Subsistema de consultas

Una consulta es cualquier expresión Booleana legítima donde los componentes atómicos son pares <t,q> que pertenecen al conjunto *TQ = T x T(Importancia)*; *t* es un elemento del conjunto *T* de términos, y *q* es el valor de la variable lingüística, *Importancia*, cualificando la importancia que el término *t* debe tener en los documentos que se recuperen.

El conjunto TQ* de consultas legítimas se define por medio las siguientes reglas sintácticas:

1. $\forall$ <t, q> $\in$TQ $\rightarrow$ <t,q> $\in$TQ*

2. $\forall$ Q1,Q2 $\in$TQ* $\rightarrow$ Q1 $\wedge$ Q2 $\in$TQ*

3. $\forall$ Q1,Q2 $\in$TQ* $\rightarrow$ Q1 $\vee$ Q2 $\in$TQ*

4. $\forall$ Q $\in$TQ* $\rightarrow$ $\neg$Q $\in$TQ*

5. Sólo se pueden obtener consultas difusas legítimas aplicando las reglas 1-4.

La importancia de los pesos asociados a los términos y la relevancia de los documentos se modela mediante variables lingüísticas, concretamente las variables *Importancia* y *Relevancia*. De esta forma, con la consulta <*t,importante*> el usuario indica quiere recuperar aquellos documentos en los que el concepto expresado por *t* tenga importancia considerable.

Subsistema de evaluación

Las consultas se evalúan por medio de la función *E**, asignando un RSV a cada documento. Esta función, *E**, está basada en la evaluación de los componentes atómicos y de sus conectores lógicos Booleanos y se define sobre la base de la función *E*, que evalúa un par <*t,q*> computando el grado de compatibilidad del peso numérico del término índice *F(d,t)* con el valor lingüístico *q* para cada documento $d \in D$.

El procedimiento de evaluación comienza evaluando los componentes atómicos, y continúa en un proceso de abajo a arriba ("*bottom-up*"), hasta que toda la consulta es evaluada.

La función E, determina como de bien satisface un documento d los requerimientos expresados por <t,q> y se define como:

E: D x TQ $\rightarrow$ [0,1]

$$E(d,\langle t,q\rangle)=\begin{cases}\mu_q(F(d,t)) & F(d,t)\neq 0\\ 0 & F(d,t)=0\end{cases}$$

La función *E*: D* x *TQ** $\rightarrow$ *[0,1]*, que evalúa las consultas $Q \in TQ^*$, se define extendiendo la función *E* mediante la aplicación recursiva de las siguientes reglas:

$$E^*(d,Q) = E(d,<t,q>) \quad \text{donde } Q = <t,q>$$

$$E^*(d,Q1 \wedge q2) = \min(E^*(d,Q1),E^*(d,Q2))$$

$$E^*(d,Q1 \vee Q2) = \max(E^*(d,Q1),E^*(d,Q2))$$

$$E^*(d, \neg Q) = 1-E^*(d,Q)$$

donde $Q,Q1,Q2 \in TQ^*$.

2.4.2.- SRI Lingüístico Ponderado con Doble Semántica basado en un Enfoque Clásico

En [114], Kraft y otros proponen una extensión lingüística de un modelo de recuperación Booleano ponderado, formalizándolo dentro de la teoría de conjuntos difusos. Reemplazan los pesos numéricos por descriptores lingüísticos que especifican el grado de importancia de los términos. Estos descriptores lingüísticos se manejan como especificaciones de un umbral lingüístico, liberando al usuario de la tarea de elegir un umbral numérico para representar la importancia deseada del término. Además, la extensión que proponen no es simétrica, ya que los documentos en los que aparece el término con un valor inferior al peso en la consulta son tratados de forma diferente a aquellos en los que parece el término con un valor superior al del peso en la consulta.

En el modelo propuesto, la consulta <*t, importante*> es sinónimo de la consulta <*t, al menos importante*>, que expresa mucho más claro el hecho de que los documentos que se desean son aquellos que tienen un valor de *F* lo mas alto posible.

Para establecer esta nueva interpretación, los autores empiezan analizando la semántica de umbral para pesos numéricos y terminan proponiendo una nueva función para la evaluación de los términos individuales, definida de la siguiente forma:

$$\mu_{imp}(F) = \max_{w \in [i,j]} g(F(d,t),w) = \begin{cases} \frac{(1+i)}{2} x e^{k(F-i)^2} & \text{para } F < i \\ \frac{(1+F)}{2} & \text{para } i \le F \le j \\ (1+j)x(\frac{1}{2} + \frac{(F-j)}{4}) & \text{para } F > j \end{cases}$$

donde i y j, con i < j, son dos umbrales que delimitan el rango en el que satisface la restricción *importante*.

Los operadores Booleanos Y y O se modelan mediante los operadores mínimo y máximo.

2.4.3.- SRI Lingüístico Ponderado que usa Cuantificadores Lingüísticos para definir los Operadores de Agregación

Las autoras exponen que la mayoría de las extensiones del lenguaje de consulta Booleano no permiten elegir los operadores de agregación de información, cuando sería muy interesante contar con diversos operadores de agregación con un comportamiento intermedio entre el operador Y (*todos*) y el operador O (*al menos 1*), que mejorarían el poder expresivo del lenguaje de consulta.

En [18], Bordogna y Pasi definen un lenguaje de consulta en el que el criterio de agregación lingüístico puede ser especificado por el usuario. Dentro de un marco de trabajo de decisión multicriterio, adoptan la noción de cuantificadores lingüísticos para definir operadores de agregación.

A partir de esto, definen un nuevo lenguaje de consulta en el que la agregación de los requerimientos se especifica a través de cuantificadores lingüísticos. Con esto, las autoras permiten que el usuario pueda solicitar que se satisfagan no sólo *todos* (agregación Y) o *al menos uno* (agregación O) de los criterios de la consulta, sino que también pueda pedir que se satisfagan "*al menos n*", "*mas de k*", o "*al menos unos pocos*".

Además, definen un operador "*y posiblemente*" que permite el uso de jerarquías en la agregación de criterios de selección con el fin de poder expresar prioridades. Con este operador, Bordogna y Pasi solucionan el problema que presentan los lenguajes de consulta convencionales para expresar "opcionalidad" en la selección de criterios.

Para definir los cuantificadores lingüísticos en el lenguaje de consulta, hacen uso de la familia de operadores de agregación de información ponderada ordenada (OWA), propuestos por Yager en [183]. En concreto, definen los siguientes operadores:

```
"todo", "al menos k", "al menos la mitad", "la mayoría
de", "al menos unos pocos", "más de"
```

El subsistema de evaluación actúa de acuerdo a las siguientes reglas:

(1) $E^*(d,s) = E(d,s)$, donde $s \in S$.

(2) $E^*(d,cuantificador(q_1,...,q_n)) = \text{OWA}_{\text{cuantificador}}(E^*(d,q_1),...,E^*(d,q_n))$.

(3) $E^*(d, \text{NO } q) = 1 - E^*(d,q)$.

(4) $E^*(d,q_1 \text{ “y posiblemente” } q_2) = (E^*(d,q1) \wedge (1 - \text{“}al\ menos\text{”}\ \alpha(\text{poss}[E^*(d,q_2)/E^*(d,q_1)])))$

$\vee(E^*(d,q_1) \wedge E^*(d,q_2))$ en donde $q, q_1, q_2, ..., q_n \in Q$.

2.4.4.- SRI Lingüístico Ordinal Monoponderado

En [19], las autoras proponen un modelo lingüístico de RI caracterizado por usar una representación de los documentos en la que cada par documento-término tiene asociado un grado lingüístico que indica lo significativo que es el término en el documento. Además, el lenguaje de consulta asocia pesos lingüísticos a los términos que indican cómo de importantes son y utiliza cuantificadores lingüísticos como operadores de agregación.

El modelo propuesto es un modelo ordinal donde las etiquetas lingüísticas se definen en una escala ordinal con un orden total. Los pesos lingüísticos de la consulta pueden especificarse con tres semánticas diferentes: importancia relativa , umbral o significado ideal, definidas sobre las variables *“Significado”*, *“Umbral”* e *“Importancia”*, respectivamente.

Por su parte, los operadores de agregación se definen como cuantificadores lingüísticos en un entorno ordinal. A su vez, dichos cuantificadores se definen como operadores OWA ordinales que agregan un conjunto de etiquetas ordinales, dando como resultado una etiqueta definida en la variable “*Relevancia*”.

Definición de consultas lingüísticas ponderadas

Para las autoras, una consulta q se define por medio de una gramática libre de contexto $\{T_q,N_q,P_q,S_q\}$, en la que T_q es el alfabeto definido como todos los términos índice *T*, las etiquetas y los operadores de agregación; N_q es el conjunto de símbolos no terminales; P_q es el conjunto de reglas de producción; y S_q es el símbolo de inicio.

Definición de la evaluación de las consultas lingüísticas

La función E* evalúa una consulta *q* contra un documento *d* produciendo una etiqueta en escala "Relevancia". Bordogna y Pasi indican que su definición depende estrictamente de la función de evaluación de los átomos <*t,lq*> (función E), y de los operadores de agregación usados para combinarlos. Por su parte, la definición de la función E dependerá de la interpretación de los pesos. La definición de la función E* es la siguiente:

Si $q = <t,lq>$ entonces $E^*(d,q)=E(d,<t,lq>)$

Si q = NO q entonces E*(d, NO q) = $\text{Etiqueta}_{\text{Relevancia}}(\neg_{\text{Relevancia}} E^*(d,q))$

1. Se usa la semántica ideal o de umbral :

Si $q = cuantificador(<t_1,lq_1>,...,<t_n,lq_n>)$ entonces
$E^*(d,q)=OWA_{cuantificador}\ (E(d,q_1),...,E(d,q_n))$

2. Se usa la semántica de importancia relativa:

Si $q = cuantificador(<t_1,lq_1>,...,<t_n,lq_n>)$ entonces
$E^*(d,q)=OWA_{cuantificador}\ (E(d,q_1)^{lq1},...,E(d,q_n)^{lqn})$

2.4.5.- SRI Lingüístico Ordinal Multiponderado

En [95], Herrera-Viedma propone un SRI lingüístico basado en el enfoque ordinal. Las principales aportaciones son la ponderación de los términos de una consulta con varios pesos, cada uno con una semántica diferente, y el poder expresar restricciones cuantitativas y cualitativas.

Definición de la Base de Datos

El autor considera una base de datos construida automáticamente, que almacena un conjunto finito de documentos $D = \{d_1, ..., d_m\}$ con su representación $R(D) = \{Rd_1,..., Rd_m\}$, y un conjunto finito de términos índice $T = \{t1,..., tl\}$.

Definición del Subsistema de Consulta

El autor propone un subsistema de consulta con un lenguaje basado en consultas Booleanas ponderadas con pesos lingüísticos, para expresar las necesidades de los usuarios. Con este lenguaje, cada consulta es una combinación de términos índice ponderados conectados mediante los operadores lógicos Booleanos Y, O y NO.

El conjunto Q de consultas legítimas se define por medio las siguientes reglas sintácticas:

1. $\forall\ q=<t,c^1,c^2,c^3>, \in TxS^3 \rightarrow q \in Q.$
2. $\forall\ p,q \in Q \rightarrow q \wedge p \in Q.$
3. $\forall\ p,q \in Q \rightarrow q \vee p \in Q.$
4. $\forall\ q \in Q \rightarrow \neg(q) \in Q.$
5. Sólo se pueden obtener consultas difusas legítimas aplicando las reglas 1-4.

Los términos de las consultas pueden ponderarse de acuerdo a tres pesos lingüísticos diferentes simultáneamente. Al igual que en [17], el autor usa la variable lingüística Importancia para expresar los pesos lingüísticos, pero definiéndola con un enfoque ordinal. Cada peso modela una semántica diferente: c^1, la semántica de umbral simétrica, c^2, la semántica cuantitativa y c^3 la semántica de importancia relativa [13][180].

Definición del Subsistema de Evaluación

Herrera-Viedma presenta un método de evaluación constructivo “bottom-up” que satisface el criterio de separabilidad [25][180], a la vez que soporta todas las semánticas consideradas. Sus principales características son:

a) Los valores RSV obtenidos son valores lingüísticos de la variable Relevancia.
b) Considera sólo los términos que aparecen en las consultas.
c) Las consultas son preprocesadas y trasformadas en consultas en forma normal disyuntiva (DNF) o conjuntiva (CNF).
d) El subsistema distingue tres niveles de evaluación: i) evaluación de los átomos,

ii) evaluación de las subexpresiones, y iii) evaluación de la consulta completa.

e) La semántica cuantitativa y la de umbral simétrico se aplican en la evaluación individual de los átomos.
f) Primero se aplica la semántica de umbral y después la cuantitativa.
g) El subsistema distingue dos tipos de conectivas lógicas: i) conectivas ponderadas, que establecen relaciones entre los átomos de una subexpresión, ii) conectivas no ponderadas, que establecen relaciones entre las subexpresiones.
h) Considera que la semántica de importancia no tiene sentido en consultas con un único átomo.
i) Las consultas tendrán más de una subexpresión y cada subexpresión más de un átomo.
j) La semántica de importancia se aplica en la evaluación de las subexpresiones Booleanas.
k) Las conectivas lógicas ponderadas Y y O se modelan por medio de los operadores de agregación lingüísticos, *LWC* y *LWD* [88], respectivamente.
l) Solo los átomos estarán negados, al expresarse las consultas en forma normal.
m) Las conectivas lógicas no ponderadas se modelan con los operadores MIN y MAX, respectivamente.

De acuerdo con lo anterior, el subsistema evalúa una consulta mediante los cinco pasos siguientes:

1. Preprocesamiento de la consulta.
2. Evaluación de los átomos con respecto a la semántica de umbral.
3. Evaluación de los átomos con respecto a la semántica cuantitativa.
4. Evaluación de las subexpresiones y modelado de la semántica de importancia.
5. Evaluación de la consulta completa.

2.4.6.- SRI Lingüístico Ordinal Multiponderado en Dos Elementos

En [96], Herrera-Viedma presenta un SRI lingüístico cuyo lenguaje de consulta permite al usuario ponderar dos elementos de una consulta simultáneamente: los términos y las subexpresiones.

Definición de la Base de Datos

El autor considera la misma base que ya consideró en el SRI descrito en sección anterior. Una base de datos construida automáticamente, que almacena un conjunto finito de documentos $D = \{d_1, ..., d_m\}$ con su representación $R(D) = \{Rd_1,..., Rd_m\}$, y un conjunto finito de términos índice $T = \{t1,..., tl\}$.

Definición del Subsistema de Consulta

El autor propone un subsistema de consulta que soporta consultas lingüísticas ponderadas simultáneamente en dos elementos, términos y subexpresiones. Cada consulta se expresa como una combinación de términos índice conectados mediante los operadores lógicos Y, O y NO. Además, las consultas estarán expresadas en forma normal, conjuntiva o disyuntiva.

Los pesos asociados a los términos y a las subexpresiones se representan con valores lingüísticos ordinales de la variable *Importancia* para modelar la correspondiente semántica (aunque con interpretaciones diferentes):

- Semántica de umbral simétrica (pesos asociados a los términos).
- Semántica de importancia (pesos asociados a las subexpresiones).

El conjunto Q de consultas legítimas se define por medio las siguientes reglas sintácticas:

1. $\forall q^1 = \langle t_i \mid \neg t_i, c_i \rangle \rightarrow q^1 \in Q,$
2. $\forall q^2 \ \wedge_{k-1}^{n \geq 2} = q_k^1 \rightarrow q^2 \in Q,$
3. $\forall q^3 \ \vee_{k-1}^{n \geq 2} = q_k^1 \rightarrow q3 \in Q,$
4. $\forall q^4 \ \vee_{p-1}^{m \geq 2} = (q_p^2 \mid q_p^1, cp) \rightarrow q^4 \in Q,$
5. $\forall q^5 \ \wedge_{p-1}^{m \geq 2} = (q_p^3 \mid q_p^1, cp) \rightarrow q^5 \in Q,$
6. Sólo se pueden obtener consultas legítimas aplicando las reglas 1-5.

donde c_i es el peso asociado al término t_i, que modela la semántica de umbral; c_p es el valor lingüístico ordinal asociado a la subexpresión q_p^2 o al átomo q_p^1, que modela la semántica de importancia.

Definición del Subsistema de Evaluación

Herrera-Viedma presenta un método de evaluación constructivo "bottom-up" que satisface el criterio de separabilidad [25][180] y soporta la ponderación en dos elementos. Actúa en dos pasos:

1. Los documentos se evalúan de acuerdo a su relevancia únicamente respecto a los átomos, asignándoseles un RSV parcial.
2. Los documentos se evalúan respecto a las combinaciones Booleanas de componentes atómicos (el RSV parcial), y este proceso se repite hasta que la consulta esté totalmente evaluada. Al finalizar, a cada documento se le asignará un valor RSV respecto a la consulta.

El subsistema presenta las siguientes características:

a) Los valores RSV obtenidos son valores lingüísticos de la variable Relevancia.
b) Considera solo los términos que aparecen en las consultas.
c) La semántica de umbral simétrica se aplica en la evaluación de los átomos.
d) Como las consultas están en forma CNF o DNF, solo los átomos estarán negados.
e) La semántica de importancia no tiene sentido en consultas formadas por una única subexpresión.
f) El subsistema distingue dos tipos de conectivas lógicas: i) conectivas no ponderadas, que establecen relaciones simples entre los átomos de una subexpresión, ii) conectivas ponderadas, que establecen relaciones de importancia entre las subexpresiones.
g) La semántica de importancia asociada las subexpresiones se modela cuando se aplican las conectivas ponderadas Y y O en la evaluación de la consulta. Estas conectivas se modelan por medio de los operadores de agregación LWC y LWD, respectivamente.
h) Las conectivas Y y O no ponderadas se modelan a través de los operadores difusos MIN y MAX, respectivamente.

El subsistema de evaluación actúa de acuerdo a las siguientes reglas

$1.E(d_j,q^1) = g(d_j,q^1).$

$2.E(d_j,q^2) = MIN(E(d_j,q_1^1), \ldots, E(d_j,q_n^1)).$

$3.E(d_j,q^3) = MAX(E(d_j,q_1^1), \ldots, E(d_j,q_n^1)).$

$4.E(d_j,q^4) = LWD([(c_1,E(d_j,q_1^h)),\ldots,(c_m,E(d_j,q_h^m)],h \in \{1,2\}$

$$5.E(d_j,q^5) = LWC([(c_1,E(d_j,q_1^h)),..,(c_m,E(d_j,q_h^m)], h \in \{1,3\}$$

2.4.7.- Inconvenientes

Las distintas propuestas de SRI Lingüísticos revisadas en las secciones anteriores permiten valorar, mediante variables lingüísticas, diferentes aspectos que encontramos en la actividad de un SRI como, por ejemplo, la relevancia de los documentos, la importancia de los términos de la consulta, etc. Sin embargo, presentan algunas limitaciones:

- Normalmente, la mayoría de los lenguajes basados en consultas difusas [13][17][24][114] no permiten a los usuarios construir consultas en las que los elementos se ponderen de acuerdo a varias semánticas simultáneamente.

- En muchos SRI Lingüísticos, las entradas y la salida se valoran sobre el mismo conjunto de etiquetas *S* [96][95]. Sin embargo, esta forma de expresar la entrada y la salida no es conveniente ya que, por un lado, se reducen las posibilidades de comunicación entre usuario y sistema; y, por otro, puesto que se están representando diferentes conceptos, parece lógico usar diferentes conjuntos de etiquetas para modelarlos.

En el Capítulo 4 nos planteamos solucionar estas carencias, presentando un modelo de SRI Lingüístico que use diferentes conjuntos de etiquetas con diferente granularidad y/o semántica para representar los diferentes tipos de información que pueden aparecer en el proceso de recuperación de información.

3.- UN ALGORITMO PG MULTIOBJETIVO PARA EL APRENDIZAJE AUTOMÁTICO DE CONSULTAS PERSISTENTES

Un usuario puede satisfacer una necesidad de información que le surge en un determinado momento sin más que sentarse delante de un sistema y teclear la consulta que la represente. Entonces, si el sistema contara con alguna información acerca de las preferencias del usuario, podría ayudarlo en la formulación de la consulta o emplear ese conocimiento para filtrar la información que se obtenga inicialmente. Sin embargo, los SRI están limitados por la falta de personalización en la representación de las necesidades del usuario. Por tanto, una cuestión importante es la construcción de perfiles que representen estas necesidades.

En el presente capítulo, proponemos un algoritmo evolutivo multiobjetivo basado en PG que permita aprender de forma automática los perfiles que representan las necesidades de información de un usuario. Para dotar de mayor expresividad a los perfiles, proponemos representarlos como consultas clásicas de RI (consultas persistentes [61]), formuladas mediante un modelo de RI Booleano, en vez de como la clásica estructura “bag of words”.

En primer lugar, haremos una pequeña introducción, justificando la caracterización de los perfiles como consultas clásicas de RI y el uso de un enfoque evolutivo multiobjetivo para aprender las consultas persistentes. Seguidamente, repasaremos los algoritmos evolutivos multiobjetivo, las diferentes familias de este tipo de algoritmos y una propuesta multiobjetivo existente para el aprendizaje de consultas Booleanas extendidas. Posteriormente, mostraremos los principales componentes de la propuesta de Smith y Smith [165], la cual será nuestro punto de referencia, al constituir el algoritmo básico de PG para el aprendizaje de consultas Booleanas. Terminaremos describiendo nuestra propuesta y realizando un análisis de los resultados obtenidos.

3.1.- Justificación

¿Por qué representar los perfiles como consultas?

Como se comentó en el Capítulo 1, el filtrado de información (FI) es un proceso de búsqueda de información donde las necesidades de información del usuario perduran en el tiempo [84][138]. Este tipo de necesidades de información del usuario se representan por medio de un "*perfil*".

Existen dos formas típicas de representar las necesidades a largo plazo de un usuario [61]:

- **Enfoque explícito "Bag of Words**": consiste en un conjunto de palabras clave que representan los intereses del usuario.
- **Enfoque explícito de Categorías**: los intereses se identifican seleccionándolos de un conjunto de categorías predefinidas.

Aunque cualquiera de los dos enfoques trabajaría correctamente para ciertos tópicos clásicos o para usuarios con un conocimiento específico del sistema, la realidad es que, fuera de estos escenarios, las cosas no funcionan tan bien.

En el primer enfoque, los usuarios se encuentran con la dificultad de tener que seleccionar las palabras adecuadas para representar sus necesidades y comunicarse, así, con el sistema. Es lo que clásicamente se conoce como el "*problema del vocabulario*", en la interacción hombre-ordenador [76]. Además, para obtener resultados satisfactorios, el perfil no debe ser demasiado amplio, ya que provocaría que el sistema de filtrado recuperase demasiados documentos irrelevantes; ni demasiado específico, lo que originaría una perdida de información valiosa.

El segundo enfoque presenta principalmente tres inconvenientes: i) las categorías pueden no ser suficientemente precisas; ii) los usuarios pueden necesitar mucho tiempo para encontrar, si es que la encuentran, la categoría o subcategoría que represente sus necesidades; iii) puede haber discrepancia entre el usuario y el sistema sobre la categoría en la que clasificar una información.

En definitiva, son necesarias nuevas formas de representar las necesidades de información de un usuario.

Belkin y Croft sugirieron que las técnicas de RI podían ser aplicadas con éxito en el filtrado de información [7]. De esta forma, el perfil puede ser representado mediante una consulta clásica de RI, las llamadas "*consultas persistentes*" [61].

Representar el perfil como una consulta persistente, en lugar de como la clásica estructura de "bag of words", nos proporciona:

- Más flexibilidad, al poder usar cualquier modelo de RI para formularla: Booleano, Booleano extendido, Lingüístico, ...
- Más expresividad, al usar términos y operadores Booleanos (Y, O, NO) para representar las necesidades de información, lo que es más interpretable para el ser humano.

¿Por qué utilizar Algoritmos Evolutivos Multiobjetivo?

Como vimos en la Sección 2.1.3, los AEs se han aplicado en la resolución de un gran número de problemas dentro del marco de la RI, siendo uno de los más numerosos el aprendizaje de consultas. La facilidad de los AEs para generar consultas con distintas estructuras (consultas Booleanas, Booleanas extendidas, lingüísticas, etc.) los hace muy adecuados para generar perfiles representados como consultas.

Normalmente, la aplicación de los AEs en el área de la RI se ha basado en la combinación de los dos criterios habitualmente considerados en el enfoque algorítmico de RI, precisión y exhaustividad [5][153][155][176], en una función de adaptación simple mediante un esquema ponderado. Ahora bien, estos dos criterios están inversamente relacionados [27], esto es, cuando la precisión sube, la exhaustividad normalmente baja y viceversa (véase la Sección 1.4). Por tanto, parece claro que la optimización de estos criterios es un problema multiobjetivo.

Este tipo de problemas se caracterizan por el hecho de tener que optimizar simultáneamente diferentes objetivos [30]. Por eso, no existe una única mejor solución para resolver el problema. En un típico problema multiobjetivo de optimización tendremos un conjunto de soluciones que serán superiores al resto cuando se consideran todos los objetivos. A este conjunto de soluciones se le conoce como "Pareto" y a las soluciones contenidas en él, que son mejores que el resto, se las conoce como soluciones no dominadas o Pareto-

optimales. En cambio, al resto de soluciones no incluidas en el conjunto Pareto se las denomina soluciones dominadas. Mientras que ninguna de las soluciones del Pareto sea absolutamente mejor que las otras soluciones no dominadas, todas ellas son igualmente aceptables para satisfacer los objetivos planteados.

Por tanto, si combinamos los AEs con un enfoque multiobjetivo, seremos capaces de obtener varias consultas (perfiles) con diferente balance de precisión y exhaustividad, para un determinado problema de RI (necesidad concreta de información de un usuario), en una sola ejecución.

3.2.- Preliminares

En esta sección estudiaremos la aplicación del enfoque multiobjetivo al área del los AEs y describiremos un enfoque concreto de un AE multiobjetivo para el aprendizaje de consultas. Los conceptos básicos de la optimización multiobjetivo y las técnicas clásicas para la resolución de estos problemas se pueden repasar en el Apéndice A.2

3.2.1.- Algoritmos Evolutivos Multiobjetivo

Una diferencia importante entre un método clásico de optimización y búsqueda y un AE es que en este último se procesa una población completa de soluciones en cada iteración (generación), en lugar de una única solución. Esta característica les da a los AEs una gran ventaja para su uso en la resolución de problemas de optimización multiobjetivo.

Uno de los fines de un procedimiento de optimización multiobjetivo es encontrar tantas soluciones Pareto-optimales como sea posible. Como decíamos anteriormente, los AEs trabajan en cada generación con una población de soluciones, por lo que, en teoría, podría ser posible hacer algunos cambios en el AE básico para conseguir una población completa de soluciones Pareto-optimales en una única ejecución.

Si conseguimos esto, eliminaremos los usos repetitivos de un método de optimización simple para encontrar soluciones Pareto-optimales diferentes en cada ejecución. De igual modo, se eliminará también la necesidad de utilizar algunos parámetros tales como vectores de pesos, vectores objetivo, etc. que sirven para transformar un problema de optimización multiobjetivo en uno mono-objetivo de manera que cada representación de estos vectores está asociada con una solución Pareto-optimal particular del problema (véase el Apéndice A.2).

Por otra parte, la población de un AE se puede utilizar para enfatizar todas las soluciones no dominadas y, al mismo tiempo, mantener un conjunto de soluciones diferentes usando un operador de nichos, de manera que se encuentren y se mantengan múltiples soluciones de buena calidad.

Después de algunas generaciones, este proceso puede conducir a la población a converger cerca de la frontera optimal (el frente) del Pareto con una buena distribución de soluciones a lo largo de ésta.

MOTIVACIÓN PARA ENCONTRAR SOLUCIONES PARETO-OPTIMALES

Antes de enumerar los distintos algoritmos multiobjetivo que estudiaremos, vamos a reflexionar sobre las ventajas de encontrar múltiples soluciones Pareto-optimales.

Para determinar una única solución para un problema multiobjetivo podemos utilizar distintas aproximaciones, dependiendo de la información sobre el problema de la que se disponga en un principio. Tendremos las siguientes posibilidades:

- *Basada en preferencia o "a priori"* → Si se conoce la preferencia exacta de cada objetivo dentro del problema, en otras palabras, se conoce el vector de pesos en el cual estamos interesados, no es necesario buscar múltiples soluciones. Un método clásico basado en pesos sería suficiente para encontrar la solución óptima correspondiente.

- *Aproximación ideal* → Normalmente, no se está seguro de la importancia exacta de los distintos objetivos. En este caso, es mejor encontrar primero un conjunto de soluciones Pareto-optimales y después escoger una solución de entre ellas usando alguna información de más alto nivel. Este método permite dar una perspectiva

general de las posibles soluciones optimales antes de elegir una concreta y también permite escoger una solución de acuerdo al grado de importancia deseado para cada objetivo.

- ☞ *Aproximación "a posteriori"* ➔ Para conseguir una perspectiva de soluciones Pareto-optimales distinta, podemos usar la aproximación "a priori" de la siguiente manera. Primero elegimos un conjunto de vectores de pesos, después construimos un problema de optimización monobjetivo para cada uno, encontrando la correspondiente solución óptima. Todas estas soluciones formarán el conjunto de soluciones Pareto-optimales resultante.

La aproximación ideal es la mejor estrategia en la práctica. Los métodos evolutivos que describiremos a continuación estarán basados en ella.

3.2.2.- Tipos de Algoritmos Evolutivos Multiobjetivo

Podemos distinguir entre dos tipos de AEs multiobjetivo: elitistas y no elitistas. Por un lado, los algoritmos elitistas mantienen las mejores soluciones de cada generación, llamadas elite, en la siguiente generación, de forma que se fomenta la permanencia de estas soluciones en generaciones sucesivas. Por el contrario, los AEs no elitistas no mantienen estas soluciones elite, de forma que la optimización se logra mediante la aplicación de los operadores genéticos tradicionales sin tener en cuenta las mejores soluciones de generaciones anteriores. Las dos subsecciones siguientes están dedicadas a repasar algunos de los enfoques existentes en cada familia.

3.2.2.1.- Algoritmos evolutivos multiobjetivo no-elitistas

Vector Evaluated Genetic Algorithm (VEGA)

VEGA es el AG multiobjetivo (AGMO) más simple de todos los existentes y constituye una extensión sencilla de un AG básico mono-objetivo para optimización multiobjetivo [161], [160]. La idea básica del algoritmo es dividir aleatoriamente la población en tantas subpoblaciones, de igual tamaño, como número de objetivos tengamos que optimizar. A cada individuo de las subpoblaciones se le asignará un fitness basado en una función objetivo

diferente, de manera que cada función objetivo se usará para evaluar a algunos miembros de la población global. A continuación, un operador de muestreo aleatorio simple [79] actúa sobre cada subpoblación basándose en su función objetivo asociada. Este hecho enfatiza las buenas soluciones individuales para cada objetivo. Finalmente, se llevan a cabo el cruce y la mutación de individuos como en cualquier AG.

Para intentar establecer soluciones que sean buenas para distintos objetivos, se permite el cruce entre individuos de distintas subpoblaciones.

Por otro lado, con objeto de mantener diversidad, en estudios posteriores se incluyeron dos modificaciones en el algoritmo VEGA, "*Non-Dominated Selection Heuristic*" y "*Mate Selection Heuristic*".

Weight-Based Genetic Algorithm

Como su nombre sugiere, en el WBGA [83] cada función objetivo se multiplica por un peso (normalizado, no negativo). En el cromosoma de este AG, se representan tanto las variables de decisión como sus pesos asociados. Para calcular la función de evaluación de la solución codificada en el cromosoma, se pondera cada uno de los objetivos con sus pesos asociados y se suman para llegar al resultado final.

A diferencia de otras aproximaciones con vectores de pesos que se hacen para algoritmos multiobjetivo, en este caso cada individuo mantiene un vector de pesos distinto que evoluciona junto con el resto de la solución encontrando en cada ejecución del algoritmo múltiples soluciones Pareto-optimales.

De este modo, lo más importante en WBGA es mantener la diversidad en el vector de pesos entre los miembros de la población. Esto se puede hacer de dos maneras distintas, utilizando una función de sharing o un vector de evaluación.

Multiple Objective Genetic Algorithm (MOGA)

El algoritmo MOGA [68] introdujo por primera vez el uso de una clasificación de no-dominancia de la población. Se diferencia de los AGs clásicos en la manera de asignar el valor de adaptación a cada individuo.

En primer lugar, se clasifica la población asignándole un rango de acuerdo a un criterio de no-dominancia. Este rango se calculará sumándole 1 al número de soluciones que dominan a cada una de las soluciones de la población, de manera que las soluciones no-dominadas tendrán rango 1 y el rango mayor será como mucho igual al tamaño de la población M. A continuación, se ordena la población de mayor a menor rango y se asigna un fitness adaptado por interpolación lineal en esta ordenación desde los individuos con rango 1 hasta los individuos con mayor rango. Después se promedia el fitness adaptado de las soluciones con igual rango para que todas tengan la misma importancia en la población, de modo que es con esta adaptación con la que se realiza la selección para el posterior cruce y mutación sobre la población.

En nuevas versiones del algoritmo [68], se introdujeron técnicas de proporción de nichos entre individuos con el mismo rango con objeto de mantener la diversidad entre las soluciones. Para conseguir una distribución lo mas uniforme posible de las soluciones en la frontera Pareto-optimal, se propone una versión del algoritmo con la técnica de nichos cuya innovación radica en la manera de determinar el parámetro σ (el radio de nicho) [68]. Este parámetro indica la distancia máxima existente entre dos soluciones para que sean consideradas pertenecientes al mismo nicho. Normalmente, es un parámetro fijado por el usuario pero en esta versión se propone ir recalculándolo en cada generación teniendo en cuenta el espacio ocupado por las soluciones y el número total de soluciones de la población que serán las que al final quedaran repartidas uniformemente por el Pareto.

Non Dominated Sorting Genetic Algorithm (NSGA)

En este algoritmo [168], se mantienen los dos propósitos principales de un algoritmo de optimización multiobjetivo (obtener muchas soluciones no dominadas, bien distribuidas en el frente del Pareto) mediante un esquema de asignación de fitness que prefiere las soluciones no dominadas y una técnica de compartición de nichos (sharing) que conserva la diversidad.

El primer paso del NSGA es ordenar la población siguiendo un criterio de no dominancia, para lo que se pueden usar muchas técnicas. Una de ellas se basa en ir agrupando la población en frentes de soluciones no dominadas, excluyendo progresivamente las soluciones de los frentes ya calculados. De esta manera, en un principio se calculará el primer frente de las soluciones no dominadas, luego se extraerán las soluciones de dicho frente para obtener las

soluciones no dominadas existentes en la población restante, las cuales formarán el segundo frente, y así sucesivamente. Procediendo de este modo, ninguna solución es mejor para todos los objetivos que otra solución del mismo frente.

Una vez ordenada la población de esta forma, comenzamos con la asignación de fitness, que se hará de manera que cada frente obtenga un fitness adaptado menor que el del frente anterior. En primer lugar, se asigna el mayor valor al primer frente (un valor teórico prefijado por el usuario) y posteriormente se usa una técnica de compartición de nichos (sharing) sobre las variables de decisión de este frente. El frente siguiente recibe un valor de adaptación algo menor que el mas pequeño asignado al frente anterior y así sucesivamente.

Después de asignar el fitness, sólo resta realizar la selección (proporcional en los métodos clásicos, aunque en implementaciones recientes se ha utilizado la selección por torneo, como veremos a continuación), el cruce y la mutación.

Niched-Pareto Genetic Algorithm (NPGA)

Basado igual que el NSGA en el concepto de dominancia, se diferencia de éste en el operador de selección, ya que usa la técnica de selección por torneo binario [102].

Durante la selección por torneo binario, se van eligiendo aleatoriamente dos soluciones de la población de padres, que pasan a ser comparadas con un subconjunto de soluciones elegidas también aleatoriamente de la población y de un tamaño fijado previamente.

Si una de las dos soluciones elegidas domina a todas las del subconjunto y la otra es dominada al menos por una solución de este subconjunto, la primera es seleccionada al considerarse la ganadora del torneo. En otro caso, si ambos individuos son dominados por al menos una solución de la subpoblación o no son dominados por ninguna, se comparan con la población actual de descendientes (la cual se encuentra en construcción). Cada solución de las aspirantes a ser seleccionadas como padres se incluye en la población de descendientes obtenidos hasta el momento y se calcula la cuenta de su nicho, es decir, se cuenta cuántas de las soluciones de la población de descendientes están dentro de su radio de nicho. El torneo lo ganará la solución que menor cuenta de nicho tenga. Cada vez que se seleccionan dos padres aplicando el proceso anterior, se cruzan y se introducen los dos descendientes obtenidos en la nueva población hasta que ésta se complete.

3.2.2.2.- Algoritmos evolutivos multiobjetivo elitistas

Un esquema elitista favorece la elite de una población, dándole la oportunidad de pasar directamente a la siguiente generación. Para el caso de la optimización genética monobjetivo, existen varias formas de introducir el elitismo. De igual modo, en estos problemas es fácil identificar cuáles son las soluciones elite a introducir.

Pero, ¿qué ocurre cuando trabajamos con problemas multiobjetivo? El problema que se presenta es identificar cuáles serán las soluciones elite cuando tenemos más de una función objetivo. Para ello, se utiliza un sistema de ranking. Así, una solución puede ser evaluada como buena o mala en base a su rango de no dominancia en la población.

Por otro lado, es necesario considerar que el elitismo puede introducirse en mayor o menor grado pero siempre se habrá de tener cuidado pues un alto grado de elitismo puede reducir la diversidad de las soluciones en la población.

Pasamos a describir una serie de AGMOs que intentan introducir este elitismo en un grado controlado para evitar problemas como los antes mencionados.

Elitist Non-Dominated Sorting Genetic Algorithm (NSGA II)

NSGA II [52] es un algoritmo muy completo ya que, no sólo incorpora una estrategia de preservación de la elite, sino que usa además un mecanismo explícito para preservar la diversidad. Aunque este algoritmo no tiene muchas coincidencias con el NSGA original, sus autores decidieron mantener el nombre.

NSGA II trabaja con una población de descendientes Q_t que se crea usando la población padre P_t. Ambas poblaciones (Q_t y P_t) se combinan para formar una sola llamada R_t, de tamaño $2 \cdot M$, de la que se busca extraer el frente del Pareto.

Es entonces cuando se realiza una ordenación sobre los individuos no dominados para clasificar la población R_t. Aunque esto supone un mayor esfuerzo si lo comparamos únicamente con la ordenación del conjunto Q_t, permite una comprobación global de las soluciones no dominadas que pertenecen tanto a la población de descendientes como a la de los padres.

Una vez que la ordenación de los individuos no dominados termina, la nueva generación se forma con soluciones de los diferentes frentes de no dominados, tomando de uno de los frentes cada vez. Se comienza con el mejor frente de individuos no dominados y se continua con las soluciones del segundo frente, después con el tercero, etc. Como el tamaño de R_t es 2·M, no todos los frentes pueden pertenecer a la nueva población, ya que ésta tiene sólo tamaño M. Todos los frentes de soluciones que no pasan a la nueva población se eliminan directamente.

En lugar de descartar arbitrariamente algunos miembros del último frente, sería prudente usar una estrategia de nichos para mantener los miembros del mismo que se encuentran en las zonas menos pobladas. Una estrategia como la descrita no afecta al procedimiento que sigue el algoritmo, sobre todo en las primeras generaciones. Esto es porque, en fases tempranas del proceso evolutivo, existen muchos frentes en la población combinada. Por tanto, es probable que soluciones de muchos frentes no dominados de buena calidad estén ya incluidas en la nueva población antes de completar su tamaño M.

Sin embargo, durante las ultimas etapas de la simulación, es probable que la mayoría de las soluciones de la población se encuentren en el mejor frente de no dominadas. También es probable que el número de soluciones en el primer frente de no dominadas de la población combinada R_t (de tamaño 2·M) sea mayor que M.

Es en ese momento cuando el algoritmo anterior asegura la selección de un conjunto diverso de soluciones de este frente mediante el método de nichos. Cuando la población entera converge a la frontera Pareto-optimal, el algoritmo continúa, de forma que se asegure la mejor distribución entre las soluciones.

Strength Pareto Evolutionary Algorithm (SPEA)

Este algoritmo [193], [194] introduce elitismo por el mantenimiento explicito de una población externa P'. Esta población almacena un número fijo de soluciones no dominadas encontradas desde el comienzo de la simulación.

En cada generación, las nuevas soluciones no dominadas encontradas se comparan con la población externa existente y se preservan las soluciones no dominadas resultantes. Además, SPEA usa estas soluciones elite para participar en las operaciones genéticas con la población

actual con la esperanza de influenciar a la población para conducirla hacia buenas regiones en el espacio de búsqueda.

El algoritmo comienza con una población P_0 de tamaño M que se crea de forma aleatoria y una población externa P'_0 que en principio se encuentra vacía y tiene un máximo de capacidad M'. En cada generación t, las mejores soluciones no dominadas (pertenecientes al primer frente no dominado) de la población P_t se copian en la población externa P'_t.

A partir de ese momento, las soluciones dominadas existentes en la población externa se van borrando de la misma. De esta forma, soluciones elite encontradas previamente que son ahora dominadas por una nueva solución elite, son eliminadas de la población externa. Lo que queda en la población externa son las mejores soluciones no dominadas de una población combinada que contiene soluciones elite viejas y nuevas. Sin embargo, si este proceso continúa durante muchas generaciones, existe el peligro de superpoblar la población externa con soluciones no dominadas.

Para restringir el crecimiento de la población externa, su tamaño se limita a M', esto es, cuando el tamaño de la población externa es menor que M', se mantienen todas las soluciones elite en ella. Sin embargo, cuando el tamaño excede M', no todas las soluciones elite se pueden introducir en la población externa. Sólo se mantendrán aquellas que se encuentren menos "apiñadas" en el frente de no dominados.

Una vez que las nuevas soluciones elite se guardan para la siguiente generación, el algoritmo vuelve a la población actual y utiliza operadores genéticos para obtener una nueva población. El primer paso es asignar un fitness a cada solución de la población. Además de asignar el fitness a los miembros de la población actual, también se le asigna a los de la población externa. De hecho, SPEA asigna en primer lugar un fitness S_i a cada miembro de la población externa, proporcional al número de individuos de la población que una solución externa i domina. En otras palabras, asigna un mejor fitness a una solución elite que domine más soluciones en la población actual.

Con estos valores del fitness, se aplica un procedimiento de selección por torneo binario en la población formada por $P_t \cup P_t'$. En este proceso se enfatizarán las soluciones elite de la población externa. Al finalizar el torneo, se aplican los operadores de cruce y mutación sobre los individuos seleccionados para crear una nueva población de tamaño M.

Pareto Archived Evolution Strategy (PAES)

Es un AE multiobjetivo [110] que usa una estrategia de evolución (EE), en su forma más simple, una EE-(1+1). Dicha estrategia usa la mutación de un único padre para crear un solo hijo, por lo que es un algoritmo de búsqueda por entornos, métodos que según la experiencia, dan muy buenos resultados en algunos tipos de problemas.

El funcionamiento del algoritmo es el siguiente: primero, se genera aleatoriamente la composición del padre, el cual se muta para dar lugar a un único descendiente. A continuación, ambas soluciones se comparan y la ganadora pasa a ser el padre en la siguiente generación. Para hallar la ganadora entre estas dos soluciones, se utiliza un archivo de tamaño limitado con los mejores individuos encontrados hasta el momento. Inicialmente, dicho archivo está vacío. De este modo, se realiza una comparación entre el padre y el hijo siguiendo criterios de dominancia. Así, si el padre domina al hijo, éste no se acepta y se procede a otra mutación sobre el padre. En cambio, si el hijo domina al padre, el hijo se introduce en el archivo y pasa a ser el padre de la siguiente generación.

La dificultad aparece cuando ambas soluciones son no dominadas. En tal caso, se compara el descendiente con el archivo actual de soluciones no dominadas, de manera que se incluye en el mismo si domina a algún miembro del archivo (eliminando a ese miembro), o si pertenece al frente de soluciones no-dominadas del archivo (sólo si existe un espacio disponible). Para distinguir quién será el padre en la próxima generación entre el padre actual y el descendiente o para eliminar soluciones cuando el archivo esté lleno, se evalúa la distribución de soluciones en el espacio de búsqueda, de manera que las que están en zonas menos abarrotadas tengan mayor preferencia.

Para dar al algoritmo PAES una perspectiva global, se introduce el concepto de estrategia de evolución multi-miembro con el (1+λ)-PAES, donde una solución padre se muta λ veces, y el (μ+λ)-PAES, donde cada uno de los μ padres y λ descendientes se comparan con el archivo actual.

3.2.3.- Evaluación de la Calidad de un Algoritmo Evolutivo Multiobjetivo

Como venimos mencionado, los AEs multiobjetivo basados en el Pareto se caracterizan

por devolver muchas soluciones distintas a un problema determinado con diferentes objetivos a cumplir, igualmente válidas entre sí. Esto plantea las dos preguntas siguientes:

1. ¿cómo podemos medir la calidad del algoritmo en lo que respecta a la generación de estas soluciones?

2. ¿cómo podemos escoger una o varias de estas soluciones para resolver nuestro problema?

Dedicaremos las dos subsecciones siguientes a describir la forma en que podemos responder ambas preguntas.

3.2.3.1.- Métricas para la medición de la calidad de los Paretos

En el caso de la optimización multiobjetivo, la definición de calidad es sustancialmente más compleja que para la optimización de problemas mono-objetivo, ya que la optimización en sí implica varios objetivos. Un conjunto Pareto de soluciones no dominadas debería cumplir los siguientes requisitos:

1. Estar compuesto por un número alto de soluciones.

2. Estar compuesto por un número alto de soluciones distintas.

3. Minimizar la distancia existente con respecto al frente del Pareto óptimo para el problema en cuestión (es decir, el conjunto real de soluciones de dicho problema, el cual es a veces desconocido, como ocurre en nuestro caso).

4. Presentar una buena distribución de las soluciones que lo componen (en el mejor de los casos, una distribución uniforme). La evaluación de este criterio puede basarse en el uso de una métrica.

5. Maximizar la extensión del frente no dominado (Pareto) obtenido, es decir, para cada objetivo, las soluciones no dominadas deben cubrir la mayor amplitud de valores posible.

En la literatura, podemos encontrar varios intentos para formalizar las definiciones anteriores (o partes de ellas) mediante métricas cuantitativas (véase [192]). Por ejemplo, los dos primeros criterios son sencillos de medir, basta con contar tanto el número de soluciones

que componen el frente del Pareto derivado, como el número de soluciones distintas existentes entre ellas.

En el contexto de las investigaciones sobre convergencia al frente del Pareto óptimo (criterio 3), varios autores han considerado la distancia del conjunto de soluciones no dominadas generado al Pareto óptimo [69], de la misma manera que la función M_1 que introduciremos posteriormente. La distribución no se tenía en cuenta, porque el interés no estaba en este aspecto. Sin embargo, en estudios comparativos, la distancia por sí sola no es suficiente para la evaluación de la calidad del Pareto y, consecuentemente, del comportamiento del algoritmo que lo derivó, ya que frentes con distribuciones muy distintas pueden tener la misma distancia al frente del Pareto óptimo.

De este modo, queda claro el hecho de que este criterio no es suficiente por sí solo y, además, presenta el problema de que es necesario conocer el frente real del Pareto para poder aplicarlo, cosa que no es posible en muchas ocasiones.

En [194], Zitzler y Thiele presentaron dos métricas complementarias para evaluar la calidad de los conjuntos de soluciones no dominadas generados por AEMOs basadas en los criterios 4 y 5 mostrados anteriormente. Por un lado, se tiene en cuenta la distribución de las soluciones no dominadas en el espacio genotípico o en el espacio objetivo (métrica M_2). Por otro, se considera el tamaño del área ocupada por dichas soluciones en el frente del Pareto (métrica M_3).

Pasamos a describir la composición de las tres últimas métricas mencionadas. Sea **S** el frente del Pareto óptimo para el problema a resolver, *S'* el conjunto de soluciones no dominadas generado por el AE multiobjetivo, a y a' dos soluciones pertenecientes respectivamente a los dos conjuntos anteriores, δ un parámetro de vecindad y sea *d* una medida de distancia:

1. La función M_1 proporciona la distancia media al conjunto Pareto óptimo **S**:

$$M_1(S') := \frac{1}{|S'|} \sum_{a' \in S'} min\{d(a', a); \quad a \in S\}$$

Lógicamente, cuanto menor sea el valor de M_1, menor será la distancia existente entre

ambos conjuntos de soluciones no dominadas y mejor será la calidad del Pareto derivado.

2. La función M_2 tiene en cuenta la distribución de las soluciones del conjunto Pareto derivado con respecto al número de soluciones no dominadas que lo componen (|*S'*|):

$$M_2(S') := \frac{1}{|S'-1|}\sum_{a' \in S'} |\{b' \in S'; d(a',b') > \delta\}|$$

Nótese como, para cada solución del conjunto *a'*, se contabiliza cuántas de las soluciones restantes están a una distancia mayor de δ de ella. Finalmente, se calcula el valor medio de la suma de la cuenta correspondiente a cada solución. De este modo, el valor de M_2 está definido en [0,|*S'*-1|] y el Pareto generado será tanto mejor cuanto mayor sea dicho valor para un parámetro de vecindad adecuado. Por ejemplo, el valor M_2=|*S'*-1| indica que, para cada solución del Pareto, no existe ninguna otra solución a una distancia menor de δ de ella.

3. Por último, la función M_3 considera la extensión del frente descrito por *S´*:

$$M_3(S') := \sqrt{\sum_{i=1}^{n} max\{d(a'_i, b'_i); a', b' \in S'\}}$$

Así, M_3 mide la distancia máxima en cada dimensión para determinar el área que ocupa el Pareto.

Las métricas anteriores están definidas en el espacio genotípico, es decir, en el espacio de las soluciones al problema. Análogamente, existen tres métricas M_1*, M_2*, M_3* definidas sobre el espacio objetivo, es decir, sobre el espacio de vectores de valores de los objetivos. Sean ***Y*** e *Y'* los conjuntos de los vectores objetivo que corresponden a **S** y *S'*, respectivamente; *p* y *p'* vectores objetivos de dichos conjuntos, y $\delta^* > 0$ y d^* como antes, tenemos:

$$M_1^x(Y') := \frac{1}{|Y'|}\sum_{p' \in Y'} min\{d^x(p', p); \quad p \in Y\}$$

$$M_2^x(Y') := \frac{1}{|Y'-1|}\sum_{p' \in Y'} |\{q' \in S'; d^x(p',q') > \delta\}|$$

$$M_3^x(Y') := \sqrt{\sum_{i=1}^{n} max\{d^x(p'_i, q'_i); p', q' \in Y'\}}$$

Nótese que, en el caso de problemas con dos objetivos, el valor de M_3^* equivale a la distancia existente entre las dos soluciones más extremas:

$$M_3^x(S') := d'; \quad d' = max\{d(a',b')\}, \quad a', b' \in S'$$

En nuestro caso, en el que ambos objetivos, exhaustividad y precisión, están definidos en [0,1], los dos vectores objetivo más extremos serían el (0,0) y el (1,1). Por tanto, la distancia entre ellos sería $\sqrt{2}$ y $M_3^* \in [0,1.4142]$.

De los comentarios realizados en esta sección se puede deducir el hecho de que es difícil definir métricas de calidad para conjuntos de soluciones no dominadas y de que, probablemente, no es posible definir una sola métrica que incluya todos los criterios deseados de una forma coherente. Por esta razón, en nuestro estudio consideraremos todas las métricas que podamos.

3.2.4.- Un Algoritmo GA-P Multiobjetivo para el Aprendizaje de Consultas Booleanas Extendidas

Antes de exponer nuestra propuesta, vamos a describir una contribución previa que incorpora las técnicas multiobjetivo basadas en el Pareto a un algoritmo de PG para el aprendizaje de consultas en RI.

En [41], Cordón y otros presentan la ampliación de una técnica para aprender consultas Booleanas extendidas para SRI difusos mediante un AE multiobjetivo basado en el Pareto. Para ello, toman como base un algoritmo GA-P básico similar al presentado en [39] y lo

extienden para transformarlo en un AE multiobjetivo para obtener múltiples consultas diferentes en una sola ejecución. El AE multiobjetivo no elitista considerado es el MOGA de Fonseca y Fleming [68].

Los componentes del algoritmo GA-P propuesto son:

- **Esquema de representación**: La parte de la expresión (parte GP) codifica la composición de la consulta, términos y operadores lógicos, y la cadena de valores (parte GA) representa los pesos de los términos.

- **Operadores genéticos**: En el caso del cruce, las partes GA se cruzan mediante el operador de cruce BLX-α, mientras que las partes GP utilizan el cruce habitual. En lo que respecta a los operadores de mutación, en la parte GA emplean la mutación uniforme de Michalewicz y en la parte GP los dos operadores usados por Kraft en [116], mutación por subárbol aleatorio y por cambio de un término por otro.

- **Generación de la población inicial:** El primer individuo se obtiene mediante la generación de un árbol aleatorio que represente una consulta con la longitud máxima predefinida compuesta por términos elegidos de entre los existentes en los documentos relevantes iniciales proporcionados por el usuario, y con todos los pesos de los términos fijados a 1. El resto de los individuos se generarán del mismo modo pero considerando pesos aleatorios en el intervalo [0,1].

- **Esquema de selección**: El modelo de evaluación se basa en el esquema generacional clásico e implica los siguientes cuatro pasos:

  1. A cada individuo se le asigna un rango igual al número de individuos que lo dominan más uno (los individuos no dominados reciben rango 1).

  2. Se ordena la población de menor a mayor rango.

  3. A cada individuo se le asigna un valor de fitness que depende de su orden en la población, concretamente, el inverso de su orden.

  4. Se promedia la asignación de fitness de cada bloque (grupo de individuos con el mismo rango) entre los individuos contenidos en dicho bloque.

Una vez calculados los valores finales de adaptación, se aplica un mecanismo de selección por torneo binario, lo que introduce la máxima diversidad posible en la población.

Con el fin de evitar la presión selectiva que puede producir este mecanismo de selección, y en consecuencia la convergencia prematura que podría ocasionar, los autores utilizan un esquema de nichos. Para ello, hacen uso de la métrica Euclídea para medir la cercanía entre la eficacia de dos preguntas diferentes en el espacio. El sharing lo hacen, por tanto, en el espacio objetivo.

3.3.- Algoritmo de Programación Genética Mono-objetivo de Smith y Smith

Como se ha comentado en el en capítulo 2 de esta memoria, en [165], *Smith y Smith* propusieron un algoritmo de IQBE para aprender automáticamente consultas Booleanas (términos unidos por operadores Booleanos) para SRI Booleanos. Se basaba en un algoritmo de PG cuyas componentes se describen a continuación[1]:

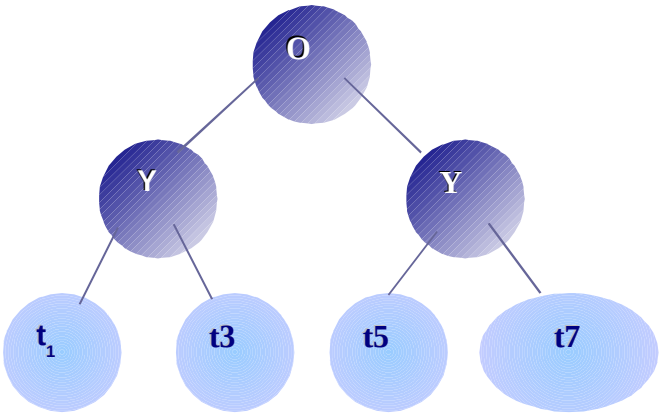


Figura 3.1: Representación de una consulta en forma de árbol

- **Esquema de codificación**: las consultas Booleanas se codifican en árboles de expresión, donde los nodos terminales son los términos de la consulta y los nodos internos los operados Booleanos Y y O. Gracias a esta representación, los árboles son siempre binarios (es decir, todo nodo tiene dos hijos), lo que facilita su manejo. La

1 Debemos señalar que algunas de estas componentes han sido alteradas de la definición inicial propuesta por Smith y Smith buscando mejorar el rendimiento del algoritmo. La implementación realizada en esta memoria ha variado la composición de algunos elementos tales como el mecanismo de selección y el operador de mutación. Naturalmente, hemos respetado los aspectos básicos de dicho algoritmo para poder establecer comparaciones realistas.

Figura 3.1 muestra un ejemplo de una consulta codificada en un árbol de este tipo.

- **Esquema de selección**: el algoritmo evoluciona según el esquema generacional clásico, donde se crea una población intermedia a partir de la población actual mediante el mecanismo de selección por Torneo binario. Igualmente, se aplica el modelo elitista que preserva el mejor individuo de la generación anterior en la población actual.

- **Operadores genéticos**: se considerará el operador de cruce convencional de la PG [113], el cual se basa en la selección aleatoria de una arista en cada padre y el intercambio de los dos subárboles que penden de estas aristas entre ambos padres. Por otro lado, se seleccionarán aleatoriamente las tres posibilidades siguientes para el operador de mutación:

    a) Cambio aleatorio de un término de la consulta por otro.
    b) Cambio aleatorio de un operador por otro.
    c) Negación de un término.

- **Generación de la población inicial**: la población inicial se genera mediante la selección aleatoria de los términos incluidos en el conjunto de documentos relevantes proporcionados por el usuario, teniendo más probabilidad de ser seleccionados aquellos que aparecen en más documentos.

- **Función de adaptación:** Se maximiza la siguiente función:

$$F = \alpha \cdot P + \beta \cdot E$$

donde la precisión P y la exhaustividad E se calculan de la forma que se muestra en el Apéndice A.1, mientras que α y β son los pesos que ponderan la relación entre ambos factores.

3.4.- Algoritmo de Programación Genética Multiobjetivo para el Aprendizaje de Consultas Persistentes

En el presente capítulo, proponemos un algoritmo evolutivo multiobjetivo basado en PG que permita aprender de forma automática los perfiles que representan las necesidades de información de un usuario. Para dotar de mayor expresividad a los perfiles, proponemos representarlos como consultas clásicas de RI (consultas persistentes [61]), formuladas mediante un modelo de RI Booleano, en vez de como una estructura "bag of words".

Como hemos venido comentado a lo largo de esta memoria, Belkin y Croft sugirieron que las técnicas de RI podían ser aplicadas con éxito en el FI [7]. De esta forma, los perfiles pueden ser representados mediante consulta clásicas de RI (consultas persistentes [61]) y, además, las técnicas de formulación de consultas, tales como retroalimentación por relevancia o el IQBE, se pueden aplicar al proceso de FI.

Así mismo, en [165], Smith y Smith propusieron un proceso IQBE basado en un algoritmo de PG para aprender automáticamente consultas Booleanas, que hemos repasado en la sección anterior. En vista de lo sugerido por Belkin y Croft, podemos interpretar que lo que aprende son perfiles representados como consultas Booleanas, englobándolo así dentro de nuestro marco de trabajo.

Puesto que el objetivo de este capítulo es presentar un algoritmo que aprenda consultas persistentes, derivadas por un SRI Booleano, utilizando un enfoque multiobjetivo, parece muy adecuado utilizar como base para nuestro algoritmo el propuesto por Smith y Smith, extendiéndolo con un enfoque multiobjetivo basado en el Pareto. Esto nos permitirá obtener varios perfiles con diferente balance de precisión y exhaustividad en una sola ejecución.

En primer lugar, veremos los componentes comunes con el algoritmo mono-objetivo de Smith y Smith, introduciendo, a continuación, el algoritmo multiobjetivo considerado para extenderlo.

3.4.1.- Componentes Comunes

Esquema de representación: Será el mismo empleado en el algoritmo de Smith y Smith de la sección 3.3, codificando la estructura de la consulta en un árbol donde las hojas son lo términos y los nodos internos los operadores (véase la figura 3.1).

Esquema de selección: Como en el algoritmo de Smith y Smith analizado en la Sección 3.3, el modelo de evolución se basa en el esquema generacional clásico, junto con la selección elitista. La población intermedia se crea a partir de la actual mediante la selección por torneo binario.

Operadores genéticos: Se consideran los mismos que en el caso de Smith y Smith; el cruce habitual en PG y el cambio de un término u operador elegido aleatoriamente por otro, como operador de mutación.

Generación de la población inicial: La población inicial se genera mediante la selección aleatoria de los términos incluidos en el conjunto de documentos relevantes proporcionados por el usuario, teniendo más probabilidad de ser seleccionados aquellos que aparecen en más documentos.

3.4.2.- Esquema Multiobjetivo Considerado

Inicialmente se utilizo el MOGA de Fonseca y Flemming [68] como AE multiobjetivo basado en Pareto para ser incorporado al algoritmo base de Smith y Smith, obteniéndose unos resultados muy favorables, que mejoraban los obtenidos con el algoritmo original. Sin embargo, al incorporar los componentes del MOGA se pierde el concepto de elitismo; no hay forma de asegurar la permanencia de la mejor solución porque, como se ha visto, no existe una única mejor solución, sino un conjunto de ellas. Como solución a esto se decidió diseñar un nuevo modelo evolutivo que usara una población externa, donde se almacenan soluciones no dominadas encontradas a lo largo de la búsqueda, como el SPEA y el NSGA II (véase la sección 3.2.2.2).

Adaptándonos a los nuevos modelos, más potentes que el MOGA al mantener el concepto de elitismo, hemos considerado el SPEA [193], [194], como AE multiobjetivo para ser incorporado al algoritmo GP básico.

Como vimos en la Sección 3.2.2.2, el elitismo se introduce por el mantenimiento explícito de un población externa P'. Esta población almacena un número fijo de soluciones no dominadas encontradas desde el comienzo de la simulación.

En cada generación, las nuevas soluciones no dominadas encontradas se comparan con la población externa existente y se preservan las soluciones no dominadas resultantes. Además,

SPEA usa estas soluciones élite para participar en las operaciones genéticas con la población actual con la esperanza de influenciar a la población para conducirla hacia buenas regiones en el espacio de búsqueda.

Se inicia la simulación con una población inicial (P), generada de la forma mencionada en la sección anterior, y una población elitista (P_e), formada por los individuos no dominados de ésta. A partir de este momento, en cada generación del algoritmo se selecciona una población intermedia, P_aux, a partir de las mejores soluciones existentes, tanto en la población elitista, P_e, como en la población actual, P. El primer paso para seleccionar las soluciones que compondrán la población auxiliar es asignar un fitness a cada solución, tanto de la población actual como de la población élite. Los fitness asociados a cada solución son los descritos a continuación y se miden en términos de minimizar:

- **Elementos de la población élite**: se asigna un fitness S_i a cada miembro, proporcional al número de individuos de la población actual que dicha solución domina:

$$S_i = \frac{n_i}{(M+1)},$$

  donde n_i es el número de soluciones dominadas por el individuo *i* en la población actual y *M* es el tamaño de la población actual.

- **Elementos de la población actual**: se asigna a cada a cada miembro un fitness, F_j, igual a la suma de los fitness de los elementos de la población élite que lo dominen más uno.

$$F_j = 1 + \sum_{i \in P_e \; e \; i \; domina \; j} S_j$$

Con estos valores del fitness, se aplica un procedimiento de selección usando torneo binario, en la población formada por P $\cup$ P_e. En este proceso se enfatizarán las soluciones elite de la población externa. Al finalizar el torneo, se aplican los operadores de cruce y mutación sobre los individuos seleccionados para crear una nueva población de tamaño N.

Una vez creada la nueva población actual, P, las soluciones no dominadas existentes en ésta se copian en la población élite, P_e, eliminando de esta última las soluciones dominadas y duplicadas. De esta forma, soluciones élite encontradas previamente que son ahora

dominadas por una nueva solución élite son eliminadas de la población externa. Lo que queda en la población externa son las mejores soluciones no dominadas de una población combinada que contiene soluciones élite viejas y nuevas.

Para restringir el crecimiento de la población élite, se limita su tamaño N_e, esto es, cuando el tamaño de la población elitista es menor que N_e, se mantienen todas las soluciones élite en ella, pero cuando lo supera, se reduce el conjunto a tamaño N_e mediante técnicas de clustering, quedándose con la solución del cluster que tenga mínima distancia al resto (la más cercana al centro).

En las figuras 3.2 y 3.3 se pueden ver los esquemas del AE multiobjetivo SPEA y del algoritmo de clustering, respectivamente.

Algunas consideraciones

- Consideraremos que dos soluciones son diferentes cuando lo sean en el espacio de objetivos.
- Utilizaremos el algoritmo de clustering en el espacio de objetivos como sugieren los autores.

```
P <--  Generación población inicial
P_e <-- No dominados de P

Para cada generación
    Asignar fitness a P y P_e
    P_aux <-- Selección_Torneo (P∪P_e)
    P <-- Cruce_Mutación (P_aux)
    P_e <--P_e ∪ No_dominados(P);
    P_e <-- No_dominados_y_no_repetidos (P_e);
    Si |P_e| > N_e
          Clustering
    fin_sin
fin_para
```

Figura 3.2: Esquema AE Multiobjetivo SPEA

```
Asignar cada elemento a un cluster
Mientras |P_e| > N_e
   Para cada par de clusters,
      Calcular la distancia entre
      clusters como la distancia media entre todos
      sus elementos
```

$$D_{ij} = \frac{1}{|C_i| \cdot |C_j|} \cdot \sum_{i \in C_i, j \in C_j} d(i,j)$$

```
   fin_para
   Combinar los dos clusters cuya distancia entre
   ellos sea mínima.
fin_mientras
Elegir una solución de cada cluster, la de mínima
distancia media al resto de elementos del cluster.
```

Figura 3.3: Algoritmo de Clustering

3.5.- Experimentación y Análisis de Resultados

Como se puede observar en la sección anterior, nuestra propuesta se basa en el algoritmo IQBE de Smith y Smith [165], pero ampliándolo con un enfoque multiobjetivo basado en Pareto (AE multiobjetivo SPEA). Por tanto, parece coherente realizar experimentos, no sólo con nuestra propuesta, sino también con el algoritmo mono-objetivo de Smith y Smith, que es el que pretendemos mejorar.

Esta sección se encarga de describir los experimentos realizados y los resultados obtenidos por cada uno de los dos algoritmos, el de Smith y Smith, y el nuestro. Los experimentos se realizarán sobre las bases documentales Cranfield y CACM, utilizando el entorno experimental considerado en el Apéndice A.1.

3.5.1.- Algoritmo Mono-Objetivo de Smith y Smith

El algoritmo de Smith y Smith se ha ejecutado 5 veces, cada una de ellas con una combinación diferente de los coeficientes de ponderación, α y β, utilizados en la función F. Los valores considerados para los distintos parámetros del algoritmo se recogen en la Tabla 3.1.

Parámetros	Valores
Tamaño de la población	800
Número de evaluaciones	50000
Tamaño del torneo	2
Probabilidad de Cruce	0,8
Probabilidad de Mutación (por cromosoma)	0,2
Tamaño límite para el árbol de consulta	20 nodos
Probabilidad de escoger un término relevante	0.8
Probabilidad de negar un término	0.3

Tabla 5: Valores de parámetros considerados para el algoritmo de Smith y Smith

Las diferentes combinaciones de coeficientes de ponderación utilizadas han sido: (1.2, 0.8), (1.1, 0.9), (1, 1), (0.9, 1.1), y (0.8, 1.2).

Estas cinco combinaciones representan diferentes escenarios de aprendizaje en los que se se potencia en mayor o menor medida un criterio frente al otro. Así, las dos primeras combinaciones dan mayor importancia a la precisión frente a la exhaustividad, mientras que las dos últimas potencian la exhaustividad frente a la precisión. La tercera combinación, considera que ambos criterios son igual de importantes.

3.5.1.1.- Resultados obtenidos con la colección Cranfield

Cada una de las tablas siguientes (Tablas de la 3.2 a la 3.18) presenta los resultados obtenidos para cada una de las diecisiete consultas de Cranfield consideradas (véase el Apéndice A.1), sobre el conjunto de entrenamiento (parte izquierda de la tabla) y sobre el conjunto de prueba (parte derecha).

Dentro de cada tabla, las filas corresponden a cada una de las cinco ejecuciones realizadas con una combinación diferente de coeficientes de ponderación; y de izquierda a derecha, las columnas recogen los datos siguientes:

i) Número de ejecución.

ii) Valores correspondientes a la evaluación sobre el conjunto de entrenamiento de la mejor consulta persistente generada por el algoritmo: Número de nodos en el árbol que representa la consulta, Precisión, Exhaustividad, Valor de la función de adaptación, Número de documentos relevantes recuperados y Número total de documentos recuperados.

iii) Valores correspondientes a la evaluación sobre el conjunto de prueba de la mejor consulta persistente generada por el algoritmo: Número de nodos en el árbol que representa la consulta, Precisión, Exhaustividad, Valor de la función de adaptación, Número de documentos relevantes recuperados y Número total de documentos recuperados .

C. 1	*ENTRENAMIENTO*						*PRUEBA*					
	Nodos	**P**	**E**	**Fitness**	**#rel**	**#rec**	**Nodos**	**P**	**E**	**Fitness**	**#rel**	**#rec**
1	19	0.900	0.643	1.594	9	10	9	0.429	0.200	0.674	3	7
2	19	1.000	0.286	1.357	4	4	5	0.000	0.000	0.000	0	1
3	19	1.000	0.357	1.357	5	5	7	0.026	0.067	0.092	1	39
4	17	0.118	1.000	1.206	14	119	15	0.039	0.333	0.402	4	129
5	19	0.060	1.000	1.248	14	223	15	0.040	0.600	0.752	9	225

Tabla 6: Resultados del algoritmo de Smith y Smith para la consulta 1 de Cranfield

C. 2	*ENTRENAMIENTO*						*PRUEBA*					
	Nodos	**P**	**E**	**Fitness**	**#rel**	**#rec**	**Nodos**	**P**	**E**	**Fitness**	**#rel**	**#rec**
1	19	1.000	0.417	1.533	5	5	9	0.000	0.000	0.000	0	3
2	19	1.000	0.500	1.550	6	6	17	0.167	0.077	0.253	1	6
3	19	1.000	0.167	1.167	2	2	11	0.018	1.000	1.018	13	722
4	19	0.218	1.000	1.000	12	55	19	0.300	0.923	1.285	12	40
5	19	0.048	1.000	1.000	12	220	19	0.046	1.000	1.237	13	283

Tabla 7: Resultados del algoritmo de Smith y Smith para la consulta 2 de Cranfield

C. 3	*ENTRENAMIENTO*						*PRUEBA*					
	Nodos	**P**	**E**	**Fitness**	**#rel**	**#rec**	**Nodos**	**P**	**E**	**Fitness**	**#rel**	**#rec**
1	19	1.000	1.000	2.000	4	4	7	0.294	1.000	1.153	5	17
2	19	1.000	1.000	2.000	4	4	13	0.625	1.000	1.587	5	8
3	19	1.000	1.000	2.000	4	4	5	0.714	1.000	1.714	5	7
4	19	1.000	1.000	2.000	4	4	19	0.625	1.000	1.663	5	8
5	19	1.000	1.000	2.000	4	4	11	0.625	1.000	1.700	5	8

Tabla 8: Resultados del algoritmo de Smith y Smith para la consulta 3 de Cranfield

C. 7	*ENTRENAMIENTO*						*PRUEBA*					
	Nodos	**P**	**E**	**Fitness**	**#rel**	**#rec**	**Nodos**	**P**	**E**	**Fitness**	**#rel**	**#rec**
1	19	1.000	0.667	1.733	2	2	17	0.000	0.000	0.000	0	1
2	19	1.000	1.000	2.000	3	3	13	0.000	0.000	0.000	0	1
3	19	1.000	1.000	2.000	3	3	17	0.000	0.000	0.000	0	3
4	19	1.000	1.000	2.000	3	3	9	0.000	0.000	0.000	0	1
5	19	1.000	1.000	2.000	3	3	15	0.000	0.000	0.000	0	0

Tabla 9: Resultados del algoritmo de Smith y Smith para la consulta 7 de Cranfield

C. 8	*ENTRENAMIENTO*						*PRUEBA*					
	Nodos	**P**	**E**	**Fitness**	**#rel**	**#rec**	**Nodos**	**P**	**E**	**Fitness**	**#rel**	**#rec**
1	19	1.000	0.333	1.467	2	2	13	0.000	0.000	0.000	0	0
2	19	1.000	0.500	1.550	3	3	17	0.000	0.000	0.000	0	1
3	19	1.000	0.500	1.500	3	3	17	0.000	0.000	0.000	0	5
4	19	0.800	0.667	1.453	4	5	13	0.222	0.333	0.567	2	9
5	19	0.833	0.833	1.667	5	6	17	0.000	0.000	0.000	0	4

Tabla 10: Resultados del algoritmo de Smith y Smith para la consulta 8 de Cranfield

C. 11	*ENTRENAMIENTO*						*PRUEBA*					
	Nodos	**P**	**E**	**Fitness**	**#rel**	**#rec**	**Nodos**	**P**	**E**	**Fitness**	**#rel**	**#rec**
1	19	1.000	1.000	2.000	4	4	0	0.000	0.000	0.000	0	0
2	19	1.000	1.000	2.000	4	4	7	0.000	0.000	0.000	0	1
3	19	1.000	1.000	2.000	4	4	7	1.000	0.250	1.250	1	1
4	19	1.000	1.000	2.000	4	4	3	0.006	1.000	1.105	4	690
5	19	1.000	0.750	1.700	3	3	17	0.000	0.000	0.000	0	1

Tabla 11: Resultados del algoritmo de Smith y Smith para la consulta 11 de Cranfield

C. 19	*ENTRENAMIENTO*						*PRUEBA*					
	Nodos	**P**	**E**	**Fitness**	**#rel**	**#rec**	**Nodos**	**P**	**E**	**Fitness**	**#rel**	**#rec**
1	19	1.000	1.000	2.000	5	5	17	0.105	0.800	0.766	4	38
2	19	1.000	1.000	2.000	5	5	9	0.014	0.600	0.555	3	222
3	19	1.000	0.800	1.800	4	4	15	0.500	0.200	0.700	1	2
4	19	1.000	1.000	2.000	5	5	11	0.000	0.000	0.000	0	1
5	19	1.000	1.000	2.000	5	5	11	0.000	0.000	0.000	0	4

Tabla 12: Resultados del algoritmo de Smith y Smith para la consulta 19 de Cranfield

C. 23	*ENTRENAMIENTO*						*PRUEBA*					
	Nodos	**P**	**E**	**Fitness**	**#rel**	**#rec**	**Nodos**	**P**	**E**	**Fitness**	**#rel**	**#rec**
1	19	1.000	0.312	1.450	5	5	15	0.000	0.000	0.000	0	6
2	19	1.000	0.312	1.381	5	5	19	0.400	0.118	0.546	2	5
3	19	1.000	0.312	1.312	5	5	15	0.024	0.941	0.965	16	678
4	19	0.037	1.000	1.133	16	433	19	0.027	0.647	0.736	11	410
5	19	0.039	1.000	1.232	16	406	19	0.027	0.647	0.798	11	409

Tabla 13: Resultados del algoritmo de Smith y Smith para la consulta 23 de Cranfield

C. 26	*ENTRENAMIENTO*						*PRUEBA*					
	Nodos	**P**	**E**	**Fitness**	**#rel**	**#rec**	**Nodos**	**P**	**E**	**Fitness**	**#rel**	**#rec**
1	19	1.000	1.000	2.000	3	3	11	0.000	0.000	0.000	0	2
2	19	1.000	0.667	1.700	2	2	17	0.000	0.000	0.000	0	0
3	19	1.000	0.667	1.667	2	2	13	0.000	0.000	0.000	0	0
4	19	0.750	1.000	1.775	3	4	15	0.000	0.000	0.000	0	0
5	19	1.000	1.000	2.000	3	3	19	0.000	0.000	0.000	0	0

Tabla 14: Resultados del algoritmo de Smith y Smith para la consulta 26 de Cranfield

C. 38	*ENTRENAMIENTO*						*PRUEBA*					
	Nodos	**P**	**E**	**Fitness**	**#rel**	**#rec**	**Nodos**	**P**	**E**	**Fitness**	**#rel**	**#rec**
1	19	1.000	0.600	1.680	3	3	17	0.000	0.000	0.000	0	2
2	19	1.000	0.800	1.820	4	4	15	0.000	0.000	0.000	0	3
3	19	1.000	0.400	1.400	2	2	9	0.000	0.000	0.000	0	1
4	19	1.000	1.000	2.000	5	5	17	0.167	0.167	0.333	1	6
5	19	0.833	1.000	1.867	5	6	15	0.125	0.167	0.300	1	8

Tabla 15: Resultados del algoritmo de Smith y Smith para la consulta 38 de Cranfield

C. 39	*ENTRENAMIENTO*						*PRUEBA*					
	Nodos	**P**	**E**	**Fitness**	**#rel**	**#rec**	**Nodos**	**P**	**E**	**Fitness**	**#rel**	**#rec**
1	19	1.000	0.571	1.657	4	4	11	0.000	0.000	0.000	0	2
2	19	1.000	0.571	1.614	4	4	9	0.000	0.000	0.000	0	2
3	19	1.000	0.571	1.571	4	4	13	0.000	0.000	0.000	0	4
4	19	1.000	0.429	1.371	3	3	19	0.143	0.143	0.286	1	7
5	19	1.000	0.571	1.486	4	4	13	0.000	0.000	0.000	0	43

Tabla 16: Resultados del algoritmo de Smith y Smith para la consulta 39 de Cranfield

C. 40	*ENTRENAMIENTO*						*PRUEBA*					
	Nodos	**P**	**E**	**Fitness**	**#rel**	**#rec**	**Nodos**	**P**	**E**	**Fitness**	**#rel**	**#rec**
1	19	1.000	0.833	1.867	5	5	11	0.000	0.000	0.000	0	4
2	19	1.000	0.833	1.850	5	5	9	0.029	0.143	0.161	1	34
3	19	1.000	0.500	1.500	3	3	15	0.000	0.000	0.000	0	1
4	19	0.261	1.000	1.335	6	23	19	0.050	0.143	0.202	1	20
5	19	0.250	1.000	1.400	6	24	19	0.067	0.143	0.225	1	15

Tabla 17: Resultados del algoritmo de Smith y Smith para la consulta 40 de Cranfield

C. 47	*ENTRENAMIENTO*						*PRUEBA*					
	Nodos	**P**	**E**	**Fitness**	**#rel**	**#rec**	**Nodos**	**P**	**E**	**Fitness**	**#rel**	**#rec**
1	19	1.000	0.429	1.543	3	3	7	0.000	0.000	0.000	0	1
2	19	1.000	0.571	1.614	4	4	9	0.000	0.000	0.000	0	0
3	19	1.000	0.429	1.429	3	3	7	0.000	0.000	0.000	0	1
4	19	1.000	0.714	1.686	5	5	19	0.000	0.000	0.000	0	3
5	19	0.750	0.857	1.629	6	8	17	0.364	0.500	0.891	4	11

Tabla 18: Resultados del algoritmo de Smith y Smith para la consulta 47 de Cranfield

C. 73	*ENTRENAMIENTO*						*PRUEBA*					
	Nodos	**P**	**E**	**Fitness**	**#rel**	**#rec**	**Nodos**	**P**	**E**	**Fitness**	**#rel**	**#rec**
1	19	1.000	0.600	1.680	6	6	17	0.000	0.000	0.000	0	1
2	19	1.000	0.300	1.370	3	3	15	0.000	0.000	0.000	0	1
3	19	1.000	0.400	1.400	4	4	17	0.000	0.000	0.000	0	4
4	19	1.000	0.400	1.340	4	4	15	0.500	0.091	0.550	1	2
5	19	0.169	1.000	1.336	10	59	15	0.120	0.545	0.751	6	50

Tabla 19: Resultados del algoritmo de Smith y Smith para la consulta 73 de Cranfield

C. 157	*ENTRENAMIENTO*						*PRUEBA*					
	Nodos	**P**	**E**	**Fitness**	**#rel**	**#rec**	**Nodos**	**P**	**E**	**Fitness**	**#rel**	**#rec**
1	19	1.000	0.300	1.440	6	6	5	0.000	0.000	0.000	0	2
2	19	1.000	0.300	1.370	6	6	9	0.000	0.000	0.000	0	2
3	19	1.000	0.350	1.350	7	7	9	0.250	0.050	0.300	1	4
4	19	0.046	1.000	1.141	20	438	17	0.025	0.600	0.682	12	483
5	19	0.056	1.000	1.245	20	354	19	0.029	0.500	0.624	10	339

Tabla 20: Resultados del algoritmo de Smith y Smith para la consulta 157 de Cranfield

C. 220	*ENTRENAMIENTO*						*PRUEBA*					
	Nodos	**P**	**E**	**Fitness**	**#rel**	**#rec**	**Nodos**	**P**	**E**	**Fitness**	**#rel**	**#rec**
1	19	1.000	0.500	1.600	5	5	7	0.333	0.100	0.480	1	3
2	19	1.000	0.500	1.550	5	5	15	0.017	0.900	0.829	9	526
3	19	1.000	0.500	1.500	5	5	15	0.500	0.200	0.700	2	4
4	19	0.143	1.000	1.229	10	70	19	0.075	0.600	0.728	6	80
5	19	0.244	1.000	1.395	10	41	15	0.037	0.800	0.990	8	216

Tabla 21: Resultados del algoritmo de Smith y Smith para la consulta 220 de Cranfield

C. 225	*ENTRENAMIENTO*						*PRUEBA*					
	Nodos	**P**	**E**	**Fitness**	**#rel**	**#rec**	**Nodos**	**P**	**E**	**Fitness**	**#rel**	**#rec**
1	19	1.000	0.583	1.667	7	7	15	0.333	0.077	0.462	1	3
2	19	1.000	0.250	1.325	3	3	7	0.200	0.077	0.289	1	5
3	19	1.000	0.250	1.250	3	3	13	0.019	1.000	1.019	13	688
4	19	0.064	1.000	1.158	12	187	19	0.021	0.308	0.357	4	195
5	19	0.034	1.000	1.227	12	355	17	0.017	0.462	0.567	6	353

Tabla 22: Resultados del algoritmo de Smith y Smith para la consulta 225 de Cranfield

ANÁLISIS DE RESULTADOS

A continuación, analizaremos los resultados presentados en las tablas anteriores. En primer lugar, comentaremos los resultados para dos grupos de consultas diferenciados (con más de 20 documentos relevantes y con menos de 15, véase la Tabla 3.19) de acuerdo a la efectividad de la recuperación. Seguidamente, analizaremos el tamaño de las mejores consultas persistentes aprendidas y el tiempo que el algoritmo tardó en generarlas. Estos dos puntos se analizarán independientemente del número de documentos relevantes que tenga asociado la consulta.

Nº de documentos relevantes	*Consultas*
más de 20	1, 2, 23, 73, 157, 220, 225
menos de 15	3, 7, 8, 11, 19, 26, 38, 39, 40, 47

Tabla 23: Grupos de consultas de Cranfield

Eficacia de las consultas con más de 20 documentos relevantes

Al evaluar las mejores consultas persistentes aprendidas sobre el conjunto de entrenamiento observamos que, en tres de las cinco ejecuciones (aquellas con un valor de α mayor o igual a 1) de cada consulta, se consiguen recuperar únicamente documentos relevantes (precisión de 1), aunque este número es reducido (menos de la mitad), lo que supone valores de exhaustividad por debajo de 0.5. En cambio, en las consultas persistentes aprendidas en las dos últimas ejecuciones, el comportamiento es el inverso, pues recuperan todos los documentos relevantes, pero junto con un gran número de documentos irrelevantes.

La consulta 23 (Tabla 13) es un claro ejemplo. En las tres primeras ejecuciones se recuperan 5 documentos, todos relevantes, consiguiendo una precisión de 1 y una exhaustividad de 0.312, al sólo recuperarse 5 de los 16 documentos relevantes asociados a la misma. En las ejecuciones restantes, por el contrario, se consigue recuperar el total de documentos relevantes, pero junto con unos 400 documentos irrelevantes, lo que supone valores muy bajos de precisión.

En lo que se refiere a la capacidad de las consultas persistentes de recuperar nuevos documentos relevantes, podemos decir que se consiguen recuperar muchos documentos relevantes aunque junto con bastantes más irrelevantes, lo que se traduce en valores altos de exhaustividad y valores bajos de precisión. Este comportamiento dificulta el acceso del

usuario a nuevos documentos que se adapten a sus necesidades de información.

Cabe destacar que el peor comportamiento lo presenta la consulta 73, en la que el algoritmo no es capaz de recuperar ningún documento relevante nuevo con 3 de las cinco consultas persistentes aprendidas. En nuestra opinión, esto se puede deber a la gran diversidad de términos índice en los documentos relevantes para esta consulta y, por lo tanto, al hecho de que a los términos existentes en los documentos del conjunto de entrenamiento les resulta más difícil describir de forma apropiada los documentos relevantes del conjunto de prueba.

Eficacia de las consultas con menos de 15 documentos relevantes

El menor número de juicios de relevancia de estas consultas (entre 6 y 15) hace más fácil recuperar todos los documentos relevantes. De ahí que la mayoría de las ejecuciones los recuperen todos o, en proporción, más de los que recuperaban las consultas del apartado anterior, consiguiéndose habitualmente valores de exhaustividad por encima de 0.5 y, en muchos casos, incluso iguales a 1. Además, únicamente se recuperan documentos relevantes (precisión igual a 1) o, a lo sumo, unos pocos documentos irrelevantes, siendo la consulta 40 la que recupera más documentos irrelevantes (17 y 18 en sus dos últimas ejecuciones), números que están muy lejos de los 400 que recupera la consulta 23.

Así, por ejemplo, la consulta 3 (Tabla 8) genera consultas persistentes capaces de recuperar todos los documentos relevantes y ninguno irrelevante en todas las ejecuciones Consigue de esta forma una precisión y una exhaustividad de 1.

La eficacia de recuperación sobre el conjunto prueba es bastante mala en general. De hecho, la mayoría de las consultas no consiguen recuperar ningún documento relevante o, a lo sumo, uno o dos. Esto puede deberse, como ya comentamos antes, a la diversidad de términos índice en los documentos relevantes para estas consultas, lo que impide a los términos existentes en los documentos del conjunto de entrenamiento describir apropiadamente los documentos relevantes del conjunto de prueba.

Mejor consulta

La consulta que deriva las consultas persistentes con mejor comportamiento es la 3, la cual tiene asociados 9 documentos relevantes. Su eficacia sobre el conjunto de entrenamiento,

como comentamos anteriormente, es impecable, recuperando únicamente todos los documentos relevantes. Por su parte, la eficacia sobre el conjunto de prueba no tiene nada que envidiar a la de entrenamiento, si tenemos en cuenta el comportamiento del resto de las consultas. En cada una de las ejecuciones se consigue recuperar el conjunto completo de documentos relevantes (un valor de exhaustividad de 1), sin que esto conlleve la reducción de la precisión (valores por encima de 0.6 excepto en la primera ejecución). A modo de ejemplo, la Figura 3.4 muestra la estructura de la consulta generada en la tercera ejecución.

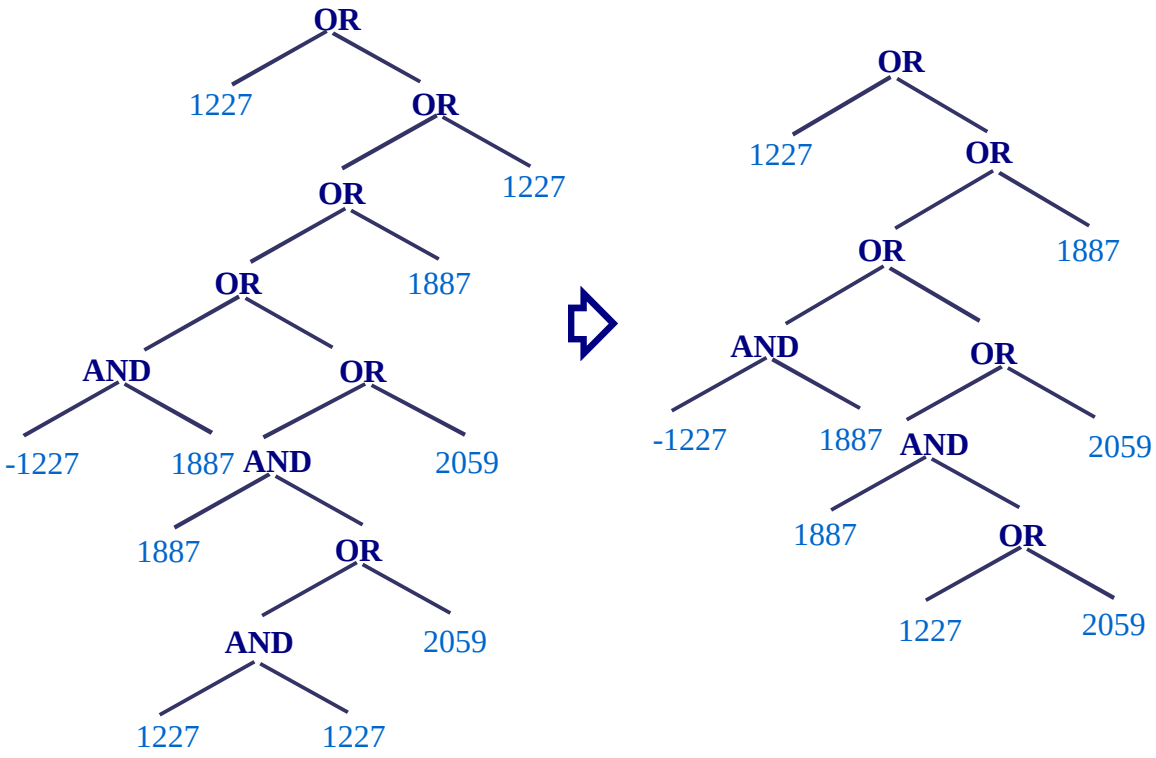


Figura 3.4: Mejor consulta persistente generada por el algoritmo de Smith y Smith sobre la colección Cranfield (Ejecución 3 de la consulta 3)

Tamaño de los árboles

En todas las ejecuciones, la población ha convergido hasta presentar consultas de gran tamaño, lo que permite obtener mejor eficacia en la recuperación. Así, en la mayoría de los casos, la mejor consulta persistente aprendida está compuesta por 19 nodos, el tamaño máximo permitido.

El tamaño de las consultas traducidas varía, puesto que no todos los términos presentes en el conjunto de entrenamiento lo están también en el conjunto de prueba.

Tiempo de ejecución

Las Tablas 3.20 y 3.21 muestran los tiempos de ejecución del algoritmo de Smith y Smith para cada una de las diecisiete consultas de Cranfield, así como las medias y las desviaciones típicas. Los tiempos aparecen expresados en minutos y segundos, y las ejecuciones más rápidas están marcadas en negrita.

El algoritmo de Smith y Smith es muy robusto en este aspecto. La media ronda el minuto y 35 segundos en las diecisiete consultas ejecutadas. Además, la desviación típica es muy reducida en todos los casos, no superando nunca los 3 segundos, lo que nos indica la poca variación existente. La ejecución más rápida la encontramos en la consulta 3, ejecución 3, con un tiempo de 1 minuto y 30 segundos, mientras que la más lenta corresponde a la consulta 40, también en la ejecución 3, con 1 minuto y 43 segundos.

	C. 1	***C. 2***	***C. 3***	***C. 7***	***C. 8***	***C. 11***	***C. 19***	***C. 23***
Ej. 1	01:36	01:35	01:33	01:38	01:37	01:37	01:35	01:41
Ej. 2	01:37	01:33	01:30	01:37	01:36	01:32	01:36	01:42
Ej. 3	01:36	01:35	**01:30**	**01:35**	01:37	01:37	01:37	01:42
Ej. 4	01:36	01:36	01:31	01:37	**01:35**	01:31	**01:35**	01:41
Ej. 5	01:36	**01:34**	01:30	01:36	01:35	**01:31**	01:36	**01:39**
Media	01:36	01:35	01:31	01:37	01:36	01:34	01:36	01:41
Desv.	00:01	00:01	00:01	00:01	00:01	00:03	00:01	00:01

Tabla 24: Tiempos de ejecución del algoritmo de Smith y Smith para la colección Cranfield

	C. 26	*C. 38*	*C. 39*	*C. 40*	*C. 47*	*C. 73*	*C. 157*	*C. 220*	*C. 225*
Ej. 1	01:36	01:37	01:35	01:39	**01:35**	01:35	01:38	01:41	01:39
Ej. 2	01:36	01:37	**01:33**	01:39	01:36	01:33	01:40	01:39	01:41
Ej. 3	01:38	01:39	01:36	01:43	01:38	01:34	01:41	01:39	**01:37**
Ej. 4	**01:34**	**01:34**	01:35	01:42	01:37	01:36	**01:36**	01:41	01:37
Ej. 5	01:35	01:36	01:33	**01:37**	01:35	**01:33**	01:39	**01:38**	01:40
Media	01:36	01:37	01:34	01:40	01:36	01:34	01:39	01:40	01:39
Desv.	00:01	00:02	00:01	00:02	00:01	00:01	00:02	00:01	00:02

Tabla 25: Tiempos de ejecución del algoritmo de Smith y Smith para la colección Cranfield

3.5.1.2.- Resultados obtenidos con la colección CACM

Cada una de las siguientes tablas (Tablas de la 3.22 a la 3.39) presenta los resultados obtenidos para cada una de las dieciocho consultas de CACM consideradas, sobre el conjunto de entrenamiento (parte izquierda de la tabla) y sobre el conjunto de prueba (parte derecha).

Dentro de cada tabla, las filas corresponden a cada una de las cinco ejecuciones realizadas con una combinación diferente de coeficientes de ponderación. Las tablas presentan la misma estructura que las mostradas en la sección anterior.

C. 4	*ENTRENAMIENTO*						*PRUEBA*					
	Nodos	**P**	**E**	**Fitness**	**#rel**	**#rec**	**Nodos**	**P**	**E**	**Fitness**	**#rel**	**#rec**
1	19	1.000	0.667	1.733	4	4	11	0.000	0.000	0.000	0	1
2	19	1.000	0.667	1.700	4	4	11	0.500	0.167	0.700	1	2
3	19	1.000	0.833	1.833	5	5	7	0.000	0.000	0.000	0	1
4	17	1.000	0.833	1.817	5	5	7	0.000	0.000	0.000	0	0
5	19	1.000	0.667	1.600	4	4	19	0.000	0.000	0.000	0	1

Tabla 26: Resultados del algoritmo de Smith y Smith para la consulta 4 de CACM

C. 7	*ENTRENAMIENTO*						*PRUEBA*					
	Nodos	**P**	**E**	**Fitness**	**#rel**	**#rec**	**Nodos**	**P**	**E**	**Fitness**	**#rel**	**#rec**
1	19	1.000	0.643	1.714	9	9	5	0.200	0.071	0.297	1	5
2	19	1.000	0.357	1.421	5	5	13	1.000	0.143	1.229	2	2
3	191	1.000	0.643	1.643	9	9	13	0.143	0.071	0.214	1	7
4	19	1.000	0.571	1.529	8	8	15	0.261	0.429	0.706	6	23
5	19	0.800	0.857	1.669	12	15	11	0.545	0.429	0.951	6	11

Tabla 27: Resultados del algoritmo de Smith y Smith para la consulta 7 de CACM

C. 9	*ENTRENAMIENTO*						*PRUEBA*					
	Nodos	**P**	**E**	**Fitness**	**#rel**	**#rec**	**Nodos**	**P**	**E**	**Fitness**	**#rel**	**#rec**
1	19	1.000	1.000	2.000	4	4	15	0.238	1.000	1.086	5	21
2	19	1.000	1.000	2.000	4	4	17	0.000	0.000	0.000	0	1
3	19	1.000	1.000	2.000	4	4	17	0.500	0.400	0.900	2	4
4	19	1.000	1.000	2.000	4	4	19	0.500	0.400	0.890	2	4
5	19	1.000	1.000	2.000	4	4	19	0.000	0.000	0.000	0	8

Tabla 28: Resultados del algoritmo de Smith y Smith para la consulta 9 de CACM

C. 10	*ENTRENAMIENTO*						*PRUEBA*					
	Nodos	**P**	**E**	**Fitness**	**#rel**	**#rec**	**Nodos**	**P**	**E**	**Fitness**	**#rel**	**#rec**
1	19	1.000	0.529	1.624	9	9	7	0.200	0.056	0.284	1	5
2	19	1.000	0.471	1.524	8	8	17	0.600	0.167	0.810	3	5
3	19	1.000	0.529	1.529	9	9	17	0.000	0.000	0.000	0	2
4	19	0.354	1.000	1.419	17	48	17	0.326	0.833	1.210	15	46
5	19	0.014	1.000	1.211	17	1255	17	0.009	0.667	0.808	12	1269

Tabla 29: Resultados del algoritmo de Smith y Smith para la consulta 10 de CACM

C. 14	*ENTRENAMIENTO*						*PRUEBA*					
	Nodos	**P**	**E**	**Fitness**	**#rel**	**#rec**	**Nodos**	**P**	**E**	**Fitness**	**#rel**	**#rec**
1	19	1.000	0.500	1.600	11	11	9	0.429	0.136	0.623	3	7
2	19	0.537	1.000	1.490	22	41	11	0.513	0.909	1.382	20	39
3	17	0.524	1.000	1.524	22	42	3	0.525	0.955	1.480	21	40
4	19	0.524	1.000	1.571	22	42	15	0.014	1.000	1.112	22	1601
5	19	0.524	1.000	1.619	22	42	15	0.512	0.955	1.555	21	41

Tabla 30: Resultados del algoritmo de Smith y Smith para la consulta 14 de CACM

C. 19	*ENTRENAMIENTO*						*PRUEBA*					
	Nodos	**P**	**E**	**Fitness**	**#rel**	**#rec**	**Nodos**	**P**	**E**	**Fitness**	**#rel**	**#rec**
1	19	1.000	0.800	1.840	4	4	7	0.000	0.000	0.000	0	8
2	19	1.000	1.000	2.000	5	5	9	0.017	0.667	0.619	4	236
3	19	1.000	0.800	1.800	4	4	11	0.000	0.000	0.000	0	0
4	19	1.000	1.000	2.000	5	5	9	0.135	0.833	1.038	5	37
5	19	1.000	1.000	2.000	5	5	11	0.000	0.000	0.000	0	0

Tabla 31: Resultados del algoritmo de Smith y Smith para la consulta 19 de CACM

C. 24	*ENTRENAMIENTO*						*PRUEBA*					
	Nodos	**P**	**E**	**Fitness**	**#rel**	**#rec**	**Nodos**	**P**	**E**	**Fitness**	**#rel**	**#rec**
1	19	1.000	1.000	2.000	6	6	9	0.000	0.000	0.000	0	0
2	19	1.000	0.833	1.850	5	5	7	0.000	0.000	0.000	0	0
3	19	1.000	1.000	2.000	6	6	13	0.000	0.000	0.000	0	1
4	19	1.000	1.000	2.000	6	6	11	0.004	0.857	0.946	6	1583
5	19	1.000	0.833	1.800	5	5	17	0.000	0.000	0.000	0	1

Tabla 32: Resultados del algoritmo de Smith y Smith para la consulta 24 de CACM

C. 25	*ENTRENAMIENTO*						*PRUEBA*					
	Nodos	**P**	**E**	**Fitness**	**#rel**	**#rec**	**Nodos**	**P**	**E**	**Fitness**	**#rel**	**#rec**
1	19	1.000	0.320	1.456	8	8	11	0.016	1.000	0.819	26	1603
2	19	1.000	0.480	1.532	12	12	17	0.333	0.154	0.505	4	12
3	19	1.000	0.360	1.360	9	9	13	0.111	0.038	0.150	1	9
4	17	1.000	0.280	1.208	7	7	15	0.018	1.000	1.116	26	1430
5	19	0.077	1.000	1.262	25	324	19	0.070	0.846	1.072	22	313

Tabla 33: Resultados del algoritmo de Smith y Smith para la consulta 25 de CACM

C. 26	*ENTRENAMIENTO*						*PRUEBA*					
	Nodos	**P**	**E**	**Fitness**	**#rel**	**#rec**	**Nodos**	**P**	**E**	**Fitness**	**#rel**	**#rec**
1	17	1.000	0.667	1.733	10	10	3	0.000	0.000	0.000	0	4
2	19	1.000	0.800	1.820	12	12	7	0.100	0.067	0.170	1	10
3	19	1.000	0.800	1.800	12	12	3	0.000	0.000	0.000	0	4
4	19	1.000	0.867	1.853	13	13	15	0.467	0.467	0.933	7	15
5	19	0.012	1.000	1.210	15	1209	19	0.009	0.733	0.887	11	1221

Tabla 34: Resultados del algoritmo de Smith y Smith para la consulta 26 de CACM

C. 27	*ENTRENAMIENTO*						*PRUEBA*					
	Nodos	**P**	**E**	**Fitness**	**#rel**	**#rec**	**Nodos**	**P**	**E**	**Fitness**	**#rel**	**#rec**
1	19	1.000	0.571	1.657	8	8	13	0.000	0.000	0.000	0	4
2	19	0.917	0.786	1.715	11	12	7	0.333	0.133	0.487	2	6
3	19	0.923	0.857	1.780	12	13	11	0.333	0.200	0.533	3	9
4	19	0.096	1.000	1.186	14	146	19	0.073	0.533	0.653	8	109
5	19	0.016	1.000	1.213	14	879	19	0.011	0.600	0.729	9	845

Tabla 35: Resultados del algoritmo de Smith y Smith para la consulta 27 de CACM

C. 40	*ENTRENAMIENTO*						*PRUEBA*					
	Nodos	**P**	**E**	**Fitness**	**#rel**	**#rec**	**Nodos**	**P**	**E**	**Fitness**	**#rel**	**#rec**
1	19	1.000	0.600	1.680	3	3	13	0.000	0.000	0.000	0	1
2	19	1.000	0.800	1.820	4	4	13	0.000	0.000	0.000	0	2
3	19	1.000	0.800	1.800	4	4	7	0.000	0.000	0.000	0	4
4	19	1.000	0.800	1.780	4	4	13	0.000	0.000	0.000	0	1
5	19	1.000	0.600	1.520	3	3	7	0.000	0.000	0.000	0	1

Tabla 36: Resultados del algoritmo de Smith y Smith para la consulta 40 de CACM

C. 42	*ENTRENAMIENTO*						*PRUEBA*					
	Nodos	**P**	**E**	**Fitness**	**#rel**	**#rec**	**Nodos**	**P**	**E**	**Fitness**	**#rel**	**#rec**
1	19	1.000	0.800	1.840	8	8	5	0.000	0.000	0.000	0	2
2	19	1.000	0.800	1.820	8	8	7	0.286	0.154	0.453	2	7
3	19	1.000	0.800	1.800	8	8	9	0.333	0.154	0.487	2	6
4	19	1.000	0.500	1.450	5	5	17	0.538	0.538	1.077	7	13
5	19	0.008	1.000	1.207	10	1201	5	0.103	0.615	0.821	8	78

Tabla 37: Resultados del algoritmo de Smith y Smith para la consulta 42 de CACM

C. 43	*ENTRENAMIENTO*						*PRUEBA*					
	Nodos	**P**	**E**	**Fitness**	**#rel**	**#rec**	**Nodos**	**P**	**E**	**Fitness**	**#rel**	**#rec**
1	19	1.000	0.500	1.600	10	10	0	0.000	0.000	0.000	0	0
2	19	1.000	0.550	1.595	11	11	1	1.000	0.048	1.143	1	1
3	19	1.000	0.550	1.550	11	11	7	0.400	0.095	0.495	2	5
4	19	0.016	1.000	1.114	20	1286	19	0.010	0.619	0.690	13	1343
5	19	0.016	1.000	1.212	20	1283	17	0.014	0.857	1.040	18	1284

Tabla 38: Resultados del algoritmo de Smith y Smith para la consulta 43 de CACM

C. 45	*ENTRENAMIENTO*						*PRUEBA*					
	Nodos	**P**	**E**	**Fitness**	**#rel**	**#rec**	**Nodos**	**P**	**E**	**Fitness**	**#rel**	**#rec**
1	19	1.000	0.538	1.631	7	7	5	0.000	0.000	0.000	0	2
2	17	1.000	0.385	1.446	5	5	7	0.286	0.154	0.453	2	7
3	19	1.000	0.538	1.538	7	7	9	0.333	0.154	0.487	2	6
4	17	1.000	0.615	1.577	8	8	17	0.538	0.538	1.077	7	1
5	19	0.181	1.000	1.344	13	72	5	0.103	0.615	0.821	8	78

Tabla 39: Resultados del algoritmo de Smith y Smith para la consulta 45 de CACM

C. 58	*ENTRENAMIENTO*						*PRUEBA*					
	Nodos	**P**	**E**	**Fitness**	**#rel**	**#rec**	**Nodos**	**P**	**E**	**Fitness**	**#rel**	**#rec**
1	19	1.000	0.533	1.627	8	8	5	0.008	0.867	0.703	13	1592
2	19	1.000	0.533	1.580	8	8	5	0.000	0.000	0.000	0	1
3	19	1.000	0.733	1.733	11	11	1	0.000	0.000	0.000	0	1
4	19	1.000	0.533	1.487	8	8	1	0.000	0.000	0.000	0	1
5	19	0.013	1.000	1.210	15	1177	19	0.012	0.933	1.130	14	1173

Tabla 40: Resultados del algoritmo de Smith y Smith para la consulta 58 de CACM

C. 59	*ENTRENAMIENTO*						*PRUEBA*					
	Nodos	**P**	**E**	**Fitness**	**#rel**	**#rec**	**Nodos**	**P**	**E**	**Fitness**	**#rel**	**#rec**
1	19	0.875	0.667	1.583	14	16	15	0.203	0.636	0.753	14	69
2	19	0.875	0.667	1.562	14	16	19	0.750	0.545	1.316	12	16
3	19	0.833	0.714	1.548	15	18	15	0.786	0.500	1.286	11	14
4	19	0.015	1.000	1.114	21	1389	17	0.013	0.864	0.962	19	1421
5	19	0.016	1.000	1.213	21	1333	17	0.015	0.909	1.103	20	1357

Tabla 41: Resultados del algoritmo de Smith y Smith para la consulta 59 de CACM

C. 60	*ENTRENAMIENTO*						*PRUEBA*					
	Nodos	**P**	**E**	**Fitness**	**#rel**	**#rec**	**Nodos**	**P**	**E**	**Fitness**	**#rel**	**#rec**
1	19	1.000	0.692	1.754	9	9	11	0.008	0.857	0.695	12	1550
2	19	1.000	0.538	1.585	7	7	7	0.000	0.000	0.000	0	2
3	19	1.000	0.692	1.692	9	9	13	0.125	0.071	0.196	1	8
4	19	1.000	0.615	1.577	8	8	7	0.009	1.000	1.108	14	1602
5	19	0.214	0.923	1.279	12	56	19	0.140	0.500	0.712	7	50

Tabla 42: Resultados del algoritmo de Smith y Smith para la consulta 60 de CACM

C. 61	*ENTRENAMIENTO*						*PRUEBA*					
	Nodos	**P**	**E**	**Fitness**	**#rel**	**#rec**	**Nodos**	**P**	**E**	**Fitness**	**#rel**	**#rec**
1	19	1.000	0.667	1.733	10	10	3	0.500	0.125	0.700	2	4
2	19	1.000	0.600	1.640	9	9	11	1.000	0.125	1.212	2	2
3	19	1.000	0.667	1.667	10	10	13	0.009	0.875	0.884	14	1570
4	19	0.700	0.933	1.657	14	20	15	0.227	0.312	0.548	5	22
5	19	1.000	0.533	1.440	8	8	13	0.015	0.125	0.162	2	134

Tabla 43: Resultados del algoritmo de Smith y Smith para la consulta 61 de CACM

ANÁLISIS DE RESULTADOS

A continuación, analizaremos los resultados presentados en las tablas anteriores. Al igual que con la colección Cranfield, en primer lugar comentaremos los resultados de cada grupo de consultas (Tabla 3.40), de acuerdo a la efectividad de la recuperación. Seguidamente, analizaremos el tamaño de las mejores consultas persistentes aprendidas y el tiempo que el algoritmo tardó en generarlas. Estos dos puntos se analizarán independientemente del número de documentos relevantes que tenga asociados la consulta.

Nº de documentos relevantes	*Consultas*
más de 30	10, 14, 25, 43, 59 y 61
entre 21 y 30	7, 26, 27, 42, 45, 58 y 60
menos de 15	4, 9, 19, 24 y 40

Tabla 44: Grupos de consultas de CACM

Eficacia de las consultas con más de 30 documentos relevantes

Al evaluar las mejores consultas persistentes aprendidas sobre el conjunto de entrenamiento, observamos un comportamiento similar al descrito para las consultas de Cranfield con más de 20 documentos relevantes. Podemos diferenciar dos tipos de ejecuciones, aquellas en las que la consulta persistente derivada recupera muy pocos documentos irrelevantes e incluso ninguno (ejecuciones con valores de precisión cercanos a 1), junto con algo más de la mitad de los documentos relevantes asociados a la consulta en cuestión; y ejecuciones que generan consultas persistentes que sí recuperan todos los documentos relevantes, pero que también recuperan un montón de basura (documentos irrelevantes), lo que provoca que los valores de precisión se reduzcan considerablemente; en muchos casos sin alcanzar ni siquiera el 0.1.

Las ejecuciones de la consulta 43 (Tabla 38) presentan ambos comportamientos. Por ejemplo, la primera ejecución deriva una consulta persistente que recupera únicamente la mitad de los documentos relevantes, obteniéndose una precisión de 1 y una exhaustividad de 0.5. En cambio, con la cuarta ejecución se recuperan los 20 documentos posibles, aunque junto con 1263 documentos más, consiguiéndose una precisión de 0.016 y una exhaustividad de 1.

En lo que se refiere a la capacidad de las consultas persistentes de recuperar nuevos documentos relevantes, los comportamientos más generalizados son: i) recuperación de mucho ruido junto con los documentos relevantes, lo que dificulta el acceso del usuario a nuevos documentos que se adapten a sus necesidades de información; y ii) recuperación de muy pocos documentos, comportamiento que, aunque facilita el acceso de los usuarios a los nuevos documentos, no proporciona mucha información debido al bajo número de documentos relevantes que se obtienen. Por lo tanto, ambos comportamientos son poco deseables.

Eficacia de las consultas que presentan entre de 21 y 30 documentos relevantes

El comportamiento de estas consultas es muy parecido al descrito en el apartado anterior. En la evaluación sobre el conjunto de entrenamiento, encontramos los dos tipos de ejecuciones mencionados anteriormente, aquellas que derivan consultas persistentes que consiguen recuperar todos los documentos relevantes pero también muchos irrelevantes; y ejecuciones en las que las consultas generadas recuperan únicamente, o prácticamente sólo, documentos relevantes, pero no todos. Los valores de exhaustividad asociados a estas consultas son ligeramente más altos que los conseguidos por sus homólogas del apartado anterior, puesto que al haber menos documentos relevantes, resulta más fácil recuperarlos.

La consulta 42 es un ejemplo de esto, al recuperar 8 documentos relevantes de los 10 posibles, en tres de las cinco ejecuciones, obteniéndose una precisión de 1 y una exhaustividad de 0.8.

Por otro lado, cuando las consultas generadas son evaluadas en el conjunto de prueba, las tablas muestran que se sigue sin conseguir un equilibrio entre precisión y exhaustividad, evolucionando, incluso, hacia un comportamiento ligeramente peor. Esto se puede observar en el hecho de que se derivan un mayor número de consultas persistentes que no son capaces de recuperar ningún documento relevante, como puede verse en la Tabla 43, en la que tres de las cinco ejecuciones no consiguen recuperar ningún documento relevante.

Eficacia de las consultas que presentan menos de 15 documentos relevantes

Conforme disminuye el número de juicios de relevancia de las consultas es más fácil recuperar todos los documentos relevantes. Prueba de esto es el hecho de que la mayoría de las ejecuciones de este bloque recuperan todos los documentos relevantes o, en proporción, más de los que recuperaban las consultas de los apartados anteriores, consiguiéndose valores de exhaustividad por encima de 0.6 y, en muchos casos, incluso iguales a 1. Además, únicamente se recuperan documentos relevantes (precisión igual a 1).

Así, por ejemplo, la consulta 9 (Tabla 28) genera consultas persistentes capaces de recuperar todos, y solamente, los documentos relevantes en todas las ejecuciones, obteniendo los valores máximos de precisión y exhaustividad.

La eficacia de recuperación sobre el conjunto de prueba es bastante mala en general. De hecho, la mayoría de las consultas no consiguen recuperar ningún documento relevante o, a lo

sumo, uno o dos; y las que recuperan más documentos relevantes lo hacen junto a un número considerable de irrelevantes, lo que dificulta el acceso a esas nuevas fuentes de información.

Mejor consulta

Algunos de los mejores resultados obtenidos corresponden a la consulta 14, la cual tiene asociados 22 documentos relevantes, tanto en el conjunto de entrenamiento como en el de prueba. Sobre el conjunto de entrenamiento, las consultas persistentes generadas recuperan todos los documentos relevantes, junto con otros 20 irrelevantes, lo que no supone un ruido excesivo, si lo comparamos con otras consultas. Además, es una de las consultas que presenta mejor comportamiento sobre el conjunto de prueba, encontrando cierto equilibrio entre la precisión y la exhaustividad. Se consiguen recuperar casi todos los documentos relevantes (un valor de exhaustividad cercano a 1), sin que esto conlleve la reducción de la precisión (valores alrededor de 0.5 excepto la cuarta ejecución). La Figura 3.5 muestra la estructura de consulta persistente generada en la ejecución 3.

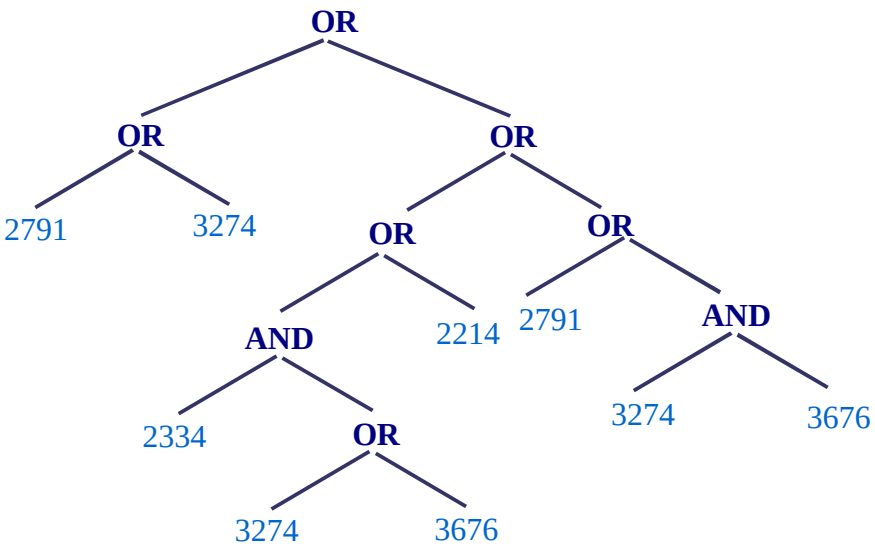


Figura 3.5; Mejor consulta persistente generada por el algoritmo de Smith y Smith sobre la colección CACM (Ejecución 3 de la consulta 14)

Tamaño de los árboles

En lo que respecta a este aspecto, encontramos de nuevo el mismo comportamiento que en Cranfield: el algoritmo de Smith y Smith converge hasta presentar consultas de gran

tamaño para mejorar la efectividad de la recuperación.

Tiempo de ejecución

Las Tablas 3.41 y 3.42 muestran los tiempos de ejecución del algoritmo de Smith y Smith para cada una de las dieciocho consultas de CACM, así como las medias y las desviaciones típicas. Los tiempos aparecen expresados en minutos y segundos, y las ejecuciones más rápidas están marcadas en negrita.

Al igual que en la experimentación con Cranfield, el algoritmo de Smith y Smith es muy robusto en lo que se refiere a este parámetro. Los valores se mueven alrededor de los 6 minutos y 50 segundos, 7 minutos, y 7 minutos y 10 segundos, para las consultas con menos de 15, entre 21 y 30 y con más de 30 documentos relevantes, respectivamente. La ejecución más rápida la encontramos en la consulta 9, ejecución 4, con un tiempo de 6 minutos y 49 segundos, mientras que la más lenta corresponde a la consulta 59 en la ejecución 5, con 7 minutos y 25 segundos.

	C. 4	***C. 7***	***C. 9***	***C. 10***	***C. 14***	***C. 19***	***C. 24***	***C. 25***	***C. 26***
Ej. 1	06:53	06:59	06:57	**06:52**	07:09	**06:56**	06:55	07:09	**06:51**
Ej. 2	06:56	06:58	06:56	06:56	07:11	06:58	06:57	**07:04**	06:52
Ej. 3	**06:53**	06:55	06:54	06:57	07:12	06:57	06:58	07:07	06:55
Ej. 4	06:56	07:00	**06:49**	06:58	07:14	07:01	06:59	07:11	06:52
Ej. 5	07:00	**06:54**	06:51	07:00	**07:08**	06:59	**06:54**	07:13	06:55
Media	06:56	06:57	06:53	06:57	07:11	06:58	06:57	07:09	06:53
Desv.	00:03	00:03	00:03	00:03	00:02	00:02	00:02	00:03	00:02

Tabla 45: Tiempos de ejecución del algoritmo de Smith y Smith para la colección CACM

	C. 27	*C. 40*	*C. 42*	*C. 43*	*C. 45*	*C. 58*	*C. 59*	*C. 60*	*C. 61*
Ej. 1	07:00	07:02	07:01	07:17	**06:52**	06:58	07:14	**06:54**	07:02
Ej. 2	06:57	**06:58**	06:58	07:18	06:59	**06:55**	07:10	06:57	**07:00**
Ej. 3	07:01	07:00	07:02	07:20	07:00	06:59	07:23	06:58	07:06
Ej. 4	07:01	06:59	07:01	07:19	07:02	06:56	**07:10**	06:58	07:06
Ej. 5	**06:57**	06:59	**06:58**	**07:14**	06:59	06:57	07:25	06:58	07:08
Media	06:59	07:00	07:00	07:18	06:58	06:57	07:16	06:57	07:04
Desv.	00:02	00:02	00:02	00:02	00:04	00:02	00:07	00:02	00:03

Tabla 46: Tiempos de ejecución del algoritmo de Smith y Smith para la colección CACM

3.5.1.3.- Resumen

En general, sea cual sea la colección, encontramos consultas con dos tipos de comportamiento diferentes:

a) Consultas que recuperan todos o casi todos los documentos relevantes, pero a costa de recuperar muchos irrelevantes (exhaustividad cercana a 1 y precisión muy baja).

b) Consultas que recuperan únicamente documentos relevantes, o junto con muy pocos irrelevantes (precisión cercana o igual a 1).

En el segundo tipo de consultas, el valor de la exhaustividad se verá influenciado por el número de documentos relevantes que tenga asociado la consulta. Cuanto menor sea este número, más posibilidades hay de recuperar todos los documentos y viceversa.

Respecto a la capacidad para recuperar nuevos documentos relevantes no utilizados en el proceso de aprendizaje, hay que decir que los resultados son bastante pobres. En su mayoría, exceptuando un par de casos, las consultas o bien recuperan mucho ruido junto con los documentos relevantes, lo que dificulta el acceso del usuario a nuevos documentos que se adapten a sus necesidades de información; o bien recuperan muy pocos documentos, comportamiento que, aunque facilita el acceso de los usuarios a los nuevos documentos, no proporciona mucha información debido al bajo número de documentos relevantes que se obtienen.

Además, las consultas con un menor número de documentos relevantes parecen tener una gran diversidad de términos índice en estos documentos, lo que impide a los términos existentes en los documentos del conjunto de entrenamiento describir apropiadamente los documentos relevantes del conjunto de prueba. De ahí, en nuestra opinión, que estas consultas no sean capaces de recuperar ningún documento relevante nuevo.

Queda claro que este algoritmo no consigue derivar consultas persistentes con un buen equilibrio entre precisión y exhaustividad, salvo en algunos casos. El problema se encuentra en la función de adaptación que utiliza, la cual combina mediante un esquema ponderado la precisión y exhaustividad. Puesto que ambos criterios son contrapuestos [27], el fin de la precisión es la ausencia de ruido, mientras que en la exhaustividad se intenta evitar el silencio; parece complicado poder optimizarlos los dos de forma conjunta con un esquema ponderado, como demuestran los resultados obtenidos.

Con nuestra propuesta pretendemos optimizar de forma individual ambos criterios con el fin de generar consultas persistentes con un mejor equilibrio entre precisión y exhaustividad, además de obtener varias consultas persistentes distintas en una única ejecución. En las siguientes secciones presentamos los resultados que hemos obtenido con el algoritmo propuesto.

3.5.2.- Algoritmo de PG Multiobjetivo

Hemos ejecutado nuestra propuesta (SPEA-PG) 5 veces, cada una de ellas con una semilla diferente, para cada una de las 35 consultas consideradas. Los valores empleados para los distintos parámetros del algoritmo se recogen en la Tabla 3.43.

Parámetros	Valores
Tamaño de la población	800
Número de evaluaciones	50000
Tamaño de la población elitista	50
Tamaño del torneo	2
Probabilidad de Cruce	0,8
Probabilidad de Mutación (por cromosoma)	0,2
Tamaño límite para el árbol de consulta	20 nodos
Probabilidad de escoger un término relevante	0.8
Probabilidad de negar un término	0.3

Tabla 47: Valores de parámetros considerados para el algoritmo SPEA-PG

La sección incluye cuatro subsecciones, dedicadas, la mitad, a la experimentación realizada con las diecisiete consultas seleccionadas de Cranfield y, la otra mitad, a las dieciocho de CACM. Las dos subsecciones correspondientes a cada colección están dedicadas, respectivamente, al análisis de los Paretos obtenidos y de un conjunto de consultas representativas extraídas de dichos Paretos.

MÉTRICAS UTILIZADAS PARA MEDIR LA CALIDAD DE LOS PARETOS

De las cinco métricas mencionadas en la Sección 3.2.3.1, haremos uso de cuatro, abandonando únicamente la métrica de similitud con el Pareto óptimo por su imposibilidad de aplicación en nuestro caso al no conocerse éste. En concreto, trabajaremos con el número de soluciones contenidas en los conjuntos Pareto derivados y el número de éstas que son distintas *con respecto a los valores de los objetivos*[2]. En el caso de las métricas M_2 y M_3, trabajaremos también en el espacio objetivo (es decir, emplearemos M_2^* y M_3^*), que es donde nos interesa que estén bien distribuidas las consultas persistentes aprendidas, para poder obtener la mayor variedad posible de balances exhaustividad-precisión.

2 En realidad, mostraremos el número de soluciones no dominadas que se obtengan, valor que será igual al número de soluciones diferentes respecto a los valores de los objetivos. Esta generalización es correcta, puesto que los Paretos proporcionados por el algoritmo SPEA están formados por los individuos de la población elitista que, como se comentó en la Sección 3.4, sólo contiene soluciones Pareto-optimales diferentes.

Conjunto de Soluciones Representativas del Pareto

Una vez especificado el modo de evaluar la calidad de los frentes de los Paretos generados por un AE, es necesario establecer un mecanismo para poder seleccionar un grupo de consultas que sean representativas de dicho conjunto. La idea sería escoger una serie de consultas que cubran lo mejor posible todo el frente del Pareto; lo primero que se nos ocurre, es elegir todo el Pareto, pero en ocasiones sería un número demasiado excesivo.

Una buena solución consiste en seleccionar cinco consultas, una en cada uno de los siguientes intervalos de precisión: [0.0, 0.2], (0.2, 0.4], (0.4, 0.6], (0.6, 0.8], (0.8, 1.0], de manera que se cubre todo el frente del Pareto. Sin embargo, puede ocurrir que no existan consultas en alguno de los intervalos (el frente no esté bien distribuido). En ese caso, se elige otra consulta, primero del intervalo siguiente y, en caso de que tampoco exista, de cualquiera de los otros intervalos. Nunca habrá dos soluciones iguales, en el espacio de decisiones y dentro de un mismo intervalo, se elegirá la solución con mayor precisión.

El conjunto de soluciones representativas se obtiene de la unificación de los Paretos correspondiente a cada una de las ejecuciones. En la Figura 3.6 se puede ver de manera gráfica el proceso seguido.

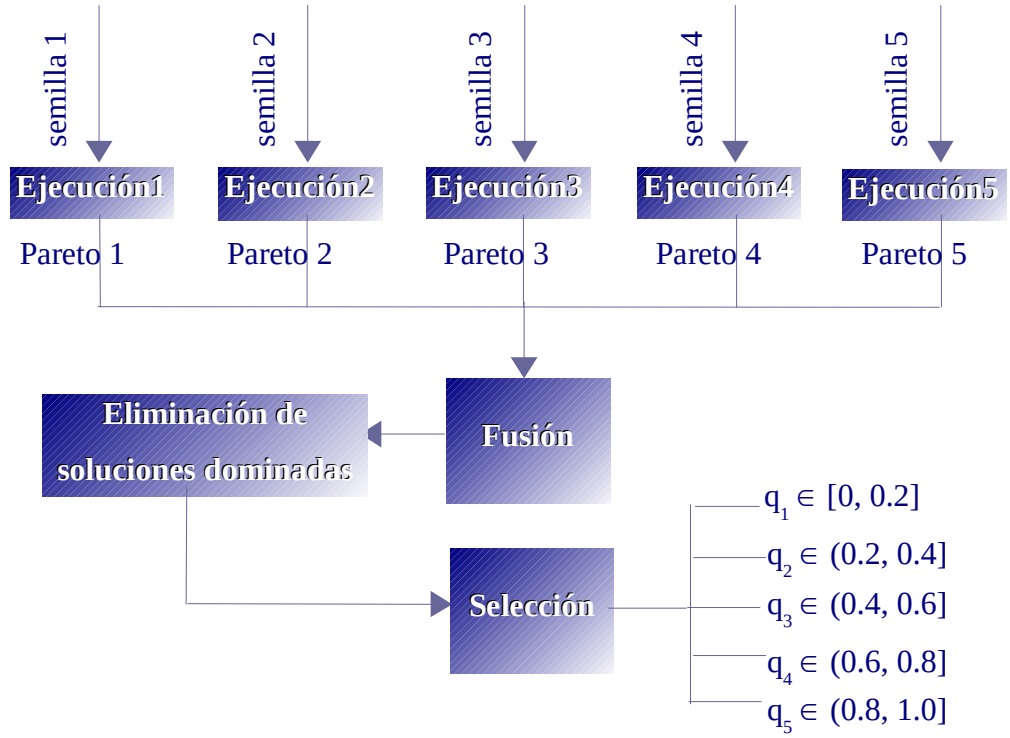


Figura 3.6: Selección del conjunto de consultas representativas

3.5.2.1.- Análisis de los Paretos obtenidos en la experimentación realizada con Cranfield

Las Tablas 3.44 a 3.60 recogen las estadísticas de los Paretos generados en las 5 ejecuciones del algoritmo SPEA-PG efectuadas sobre las 17 consultas de Cranfield. De izquierda a derecha, las columnas se corresponden con las cuatro métricas comentadas anteriormente: *N. Sol.* (Número de soluciones contenidas en el Pareto, igual al número de soluciones distintas existentes en dicho conjunto, en el espacio objetivo), M_2^* (valor medio obtenido para la métrica de distribución del Pareto en el espacio objetivo) y M_3^* (valor medio obtenido para la métrica de extensión del Pareto en el espacio objetivo). Finalmente, las dos últimas filas de cada tabla contienen las medias y las desviaciones típicas.

Posteriormente, la Tabla 3.61 resume los resultados de las diecisiete tablas anteriores, mostrando los valores medios y las desviaciones típicas en cada caso.

C. 1	*N_sol*	M_2^*	M_3^*
Ej. 1	7	3.167	1.106
Ej. 2	7	3.500	1.135
Ej. 3	6	3.000	1.112
Ej. 4	7	3.333	1.139
Ej. 5	5	2.250	1.081
Media	6.400	3.050	1.114
Desv.	0.358	0.194	0.009

Tabla 48: Estadísticas de los Paretos generados por el algoritmo SPEA-PG en la consulta 1 de Cranfield

C. 2	*N_sol*	M_2^*	M_3^*
Ej. 1	4	2.000	1.049
Ej. 2	7	3.167	1.113
Ej. 3	6	2.600	1.088
Ej. 4	6	3.000	1.076
Ej. 5	6	2.600	1.070
Media	5.800	2.673	1.079
Desv.	0.438	0.180	0.009

Tabla 49: Estadísticas de los Paretos generados por el algoritmo SPEA-PG en la consulta 2 de Cranfield

C. 3	*N_sol*	M_2^*	M_3^*
Ej. 1	1	0	0
Ej. 2	1	0	0
Ej. 3	1	0	0
Ej. 4	1	0	0
Ej. 5	1	0	0
Media	1	0	0
Desv.	0	0	0

Tabla 50: Estadísticas de los Paretos generados por el algoritmo SPEA-PG en la consulta 3 de Cranfield

C. 7	*N_sol*	M_2^*	M_3^*
Ej. 1	1	0.000	0.000
Ej. 2	1	0.000	0.000
Ej. 3	1	0.000	0.000
Ej. 4	2	1.000	0.856
Ej. 5	1	0.000	0.000
Media	1.200	0.200	0.171
Desv.	0.179	0.179	0.153

Tabla 51: Estadísticas de los Paretos generados por el algoritmo SPEA-PG en la consulta 7 de Cranfield

C. 8	*N_sol*	M_2^*	M_3^*
Ej. 1	3	1.500	0.816
Ej. 2	2	1.000	0.753
Ej. 3	2	1.000	0.556
Ej. 4	3	1.500	0.913
Ej. 5	3	1.500	0.977
Media	2.600	1.300	0.803
Desv.	0.219	0.110	0.065

Tabla 52: Estadísticas de los Paretos generados por el algoritmo SPEA-PG en la consulta 8 de Cranfield

C. 19	*N_sol*	M_2^*	M_3^*
Ej. 1	1	0.000	0.000
Ej. 2	1	0.000	0.000
Ej. 3	1	0.000	0.000
Ej. 4	1	0.000	0.000
Ej. 5	1	0.000	0.000
Media	1.000	0.000	0.000
Desv.	0.000	0.000	0.000

Tabla 54: Estadísticas de los Paretos generados por el algoritmo SPEA-PG en la consulta 19 de Cranfield

C. 11	*N_sol*	M_2^*	M_3^*
Ej. 1	1	0.000	0.000
Ej. 2	1	0.000	0.000
Ej. 3	1	0.000	0.000
Ej. 4	1	0.000	0.000
Ej. 5	1	0.000	0.000
Media	1.000	0.000	0.000
Desv.	0.000	0.000	0.000

Tabla 53: Estadísticas de los Paretos generados por el algoritmo SPEA-PG en la consulta 11 de Cranfield

C. 23	*N_sol*	M_2^*	M_3^*
Ej. 1	9	4.250	1.211
Ej. 2	8	3.714	1.181
Ej. 3	9	4.000	1.182
Ej. 4	8	3.714	1.142
Ej. 5	9	4.000	1.215
Media	8.600	3.936	1.186
Desv.	0.219	0.091	0.012

Tabla 55: Estadísticas de los Paretos generados por el algoritmo SPEA-PG en la consulta 23 de Cranfield

C. 26	*N_sol*	M_2^*	M_3^*
Ej. 1	1	0.000	0.000
Ej. 2	1	0.000	0.000
Ej. 3	1	0.000	0.000
Ej. 4	1	0.000	0.000
Ej. 5	1	0.000	0.000
Media	1.000	0.000	0.000
Desv.	0.000	0.000	0.000

Tabla 56: Estadísticas de los Paretos generados por el algoritmo SPEA-PG en la consulta 26 de Cranfield

C. 38	*N_sol*	M_2^*	M_3^*
Ej. 1	1	0.000	0.000
Ej. 2	1	0.000	0.000
Ej. 3	2	1.000	0.606
Ej. 4	1	0.000	0.000
Ej. 5	1	0.000	0.000
Media	1.200	0.200	0.121
Desv.	0.179	0.179	0.108

Tabla 57: Estadísticas de los Paretos generados por el algoritmo SPEA-PG en la consulta 38 de Cranfield

C. 39	*N_sol*	M_2^*	M_3^*
Ej. 1	2	1.000	0.518
Ej. 2	3	1.500	0.886
Ej. 3	2	1.000	0.665
Ej. 4	3	1.500	0.935
Ej. 5	2	1.000	0.518
Media	2.400	1.200	0.704
Desv.	0.219	0.110	0.079

Tabla 58: Estadísticas de los Paretos generados por el algoritmo SPEA-PG en la consulta 39 de Cranfield

C. 40	*N_sol*	M_2^*	M_3^*
Ej. 1	1	0.000	0.000
Ej. 2	1	0.000	0.000
Ej. 3	2	1.000	0.902
Ej. 4	1	0.000	0.000
Ej. 5	1	0.000	0.000
Media	1.200	0.200	0.180
Desv.	0.179	0.179	0.161

Tabla 59: Estadísticas de los Paretos generados por el algoritmo SPEA-PG en la consulta 40 de Cranfield

C. 47	*N_sol*	M_2^*	M_3^*
Ej. 1	3	1.500	0.713
Ej. 2	2	1.000	0.822
Ej. 3	2	1.000	0.518
Ej. 4	2	1.000	0.518
Ej. 5	1	0.000	0.000
Media	2.000	0.900	0.514
Desv.	0.283	0.219	0.126

Tabla 60: Estadísticas de los Paretos generados por el algoritmo SPEA-PG en la consulta 47 de Cranfield

C. 73	*N_sol*	M_2^*	M_3^*
Ej. 1	4	2.000	0.863
Ej. 2	3	1.500	0.903
Ej. 3	6	3.000	1.142
Ej. 4	4	2.000	0.940
Ej. 5	5	2.500	1.063
Media	4.400	2.200	0.982
Desv.	0.456	0.228	0.047

Tabla 61: Estadísticas de los Paretos generados por el algoritmo SPEA-PG en la consulta 73 de Cranfield

C. 157	*N_sol*	M_2^*	M_3^*
Ej. 1	8	3.714	1.094
Ej. 2	13	5.750	1.182
Ej. 3	10	4.444	1.135
Ej. 4	12	5.455	1.172
Ej. 5	10	4.556	1.189
Media	10.600	4.784	1.154
Desv.	0.780	0.328	0.016

Tabla 62: Estadísticas de los Paretos generados por el algoritmo SPEA-PG en la consulta 157 de Cranfield

C. 220	*N_sol*	M_2^*	M_3^*
Ej. 1	4	2.000	0.983
Ej. 2	3	1.500	0.894
Ej. 3	4	2.000	1.007
Ej. 4	3	1.500	0.782
Ej. 5	3	1.500	1.003
Media	3.400	1.700	0.934
Desv.	0.219	0.110	0.039

Tabla 63: Estadísticas de los Paretos generados por el algoritmo SPEA-PG en la consulta 220 de Cranfield

C. 225	*N_sol*	M_2^*	M_3^*
Ej. 1	6	3.000	1.155
Ej. 2	6	2.800	1.124
Ej. 3	6	2.800	1.117
Ej. 4	7	3.500	1.295
Ej. 5	4	2.000	1.063
Media	5.800	2.820	1.151
Desv.	0.438	0.216	0.035

Tabla 64: Estadísticas de los Paretos generados por el algoritmo SPEA-PG en la consulta 225 de Cranfield

	N_sol		M_2^*		M_3^*	
	Media	***Desviación***	***Media***	***Desviación***	***Media***	***Desviación***
C. 1	6.400	0.358	3.050	0.194	1.114	0.009
C. 2	5.800	0.438	2.673	0.180	1.079	0.009
C. 3	1.000	0.000	0.000	0.000	0.000	0.000
C. 7	1.200	0.179	0.200	0.179	0.171	0.153
C. 8	2.600	0.219	1.300	0.110	0.803	0.065
C. 11	1.000	0.000	0.000	0.000	0.000	0.000
C. 19	1.000	0.000	0.000	0.000	0.000	0.000
C. 23	8.600	0.219	3.936	0.091	**1.186**	0.012
C. 26	1.000	0.000	0.000	0.000	0.000	0.000
C. 38	1.200	0.179	0.200	0.179	0.121	0.108
C. 39	2.400	0.219	1.200	0.110	0.704	0.079
C. 40	1.200	0.179	0.200	0.179	0.180	0.161
C. 47	2.000	0.283	0.900	0.219	0.514	0.126
C. 73	4.400	0.456	2.200	0.228	0.982	0.047
C. 157	**10.600**	0.7800	**4.784**	0.328	1.154	0.016
C. 220	3.400	0.219	1.700	0.110	0.934	0.039
C. 225	5.800	0.438	2.800	0.216	1.151	0.035

Tabla 65: Estadísticas de los Paretos generados por el algoritmo SPEA-PG en las consultas de Cranfield

Análisis de resultados

Podemos comenzar destacando el hecho de que se cumple el principal objetivo de nuestra propuesta, obtener varias consultas persistentes con diferente balance precisión-exhaustividad en una única ejecución, ya que los frentes de los Paretos obtenidos están bastante bien distribuidos como demuestran los altos valores de las métricas M_2^* y M_3^*.

La mayoría de las consultas generan un número de consultas persistentes con diferente balance de precisión-exhaustividad proporcional al número de documentos relevantes que tienen asociadas (aquellos casos en que se proporciona un mayor número de documentos, un mayor número de consultas persistentes forman el frente del Pareto). Entre las consultas con menor número de documentos relevantes, hay algunas que sólo consiguen derivar una consulta persistente (p.e. las consultas 3, 11, 19 y 26), lo que se traduce en un Pareto que

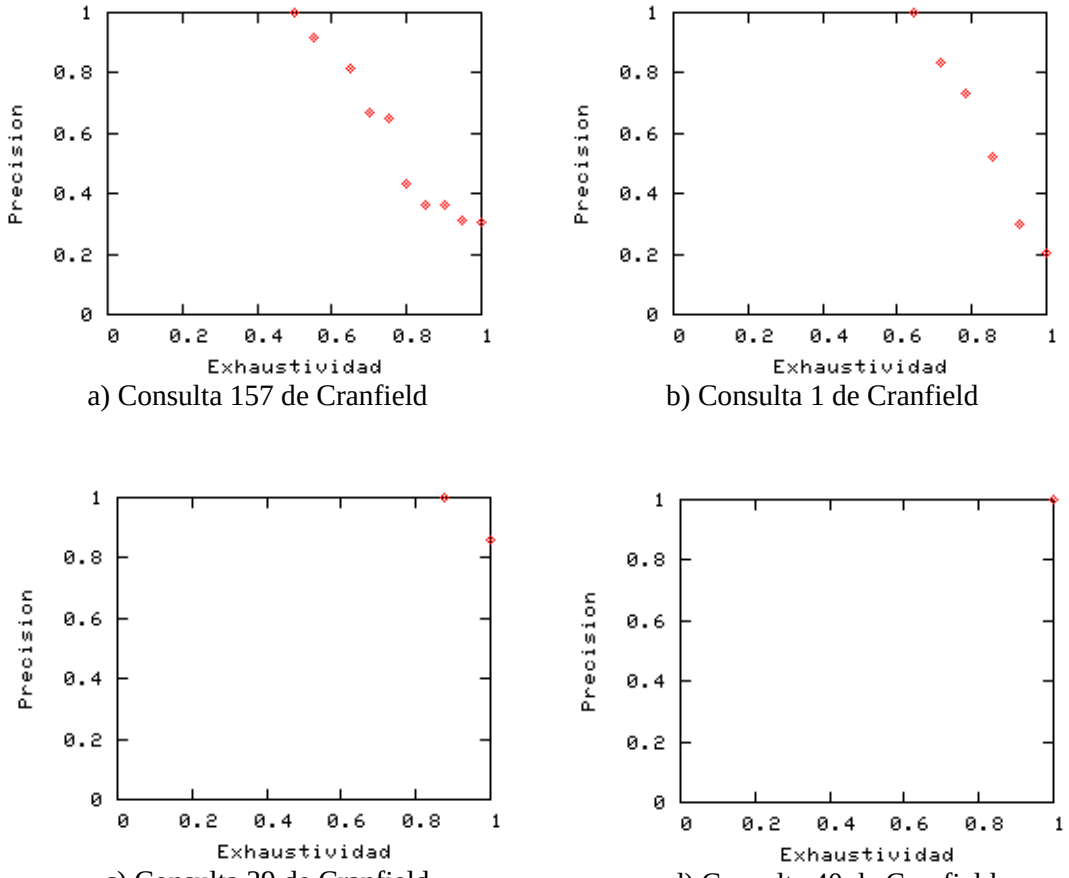
cubre un único punto del espacio en vez de una zona del mismo. Lógicamente, esta única solución se localiza en la esquina superior derecha del espacio de búsqueda, es decir, presenta una precisión y una exhaustividad de 1 (Figura 3.7, apartado d), con lo que no se pueden encontrar más soluciones no dominadas al satisfacerse plenamente los dos objetivos.

Los valores de la métricas M_2^* y M_3^* son también muy apropiados, destacando especialmente los de esta última, bastante cercanos a 1.4142, el máximo valor posible. Esto nos muestra como los frentes de los Paretos generados cubren una zona amplia del espacio. Obviamente, estas métricas no tienen sentido cuando el frente del Pareto está formado por una única solución, puesto que ésta no se puede comparar con ninguna otra, obteniéndose valores de 0 en ambas métricas para este tipo de consultas.

Por otro lado, las desviaciones típicas se encuentran alrededor de 0.5, 0.2 y 0.05 para el número de soluciones diferentes, M_2^* y M_3^*, respectivamente, lo que indica un comportamiento homogéneo del algoritmo.

Haciendo un análisis algo más pormenorizado, la consulta 157 (Tabla 62) es la que obtiene la media más alta en cuanto a soluciones distintas presentes en el frente del Pareto, alrededor de 11. Además, es esta misma consulta la que obtiene una mejor media en cuanto a la distribución de las soluciones en el frente del Pareto con un 4.784, junto con un 1.154 para M_3^*, muy cercano al máximo de la distancia euclídea (1.4142), que es la que se encarga de medir la calidad de esta métrica.

Como ejemplo, la Figura 3.7 muestra los frentes de los Paretos obtenidos para las consultas 157, 1, 39 y 40, compuestos, respectivamente, por 10, 6, 2 y 1 consultas persistentes. Recordemos que el eje X representa los valores de exhaustividad y el eje Y los de precisión. Como se hizo en [192], los Paretos obtenidos por las cinco ejecuciones realizadas para cada consulta se han fusionado y las soluciones dominadas se han eliminado del conjunto unificado antes de imprimir la curva.



a) Consulta 157 de Cranfield

b) Consulta 1 de Cranfield

c) Consulta 39 de Cranfield

d) Consulta 40 de Cranfield

Figura 3.7: Frentes de los Paretos derivados por el algoritmo SPEA-PG para las consultas 157, 1, 39 y 40 de Cranfield

3.5.2.2.- Análisis de los Paretos obtenidos en la experimentación realizada con CACM

Las Tablas 3.62 a 3.79 recogen las estadísticas de los Paretos generados en las 5 ejecuciones del algoritmo SPEA-PG efectuadas sobre las 18 consultas de CACM. Por otro lado, la Tabla 3.80 recoge las estadísticas en media de los Paretos generados para CACM.

C. 4	*N_sol*	M_2^*	M_3^*
Ej. 1	2	1.000	0.408
Ej. 2	1	0.000	0.000
Ej. 3	1	0.000	0.000
Ej. 4	2	1.000	0.556
Ej. 5	1	0.000	0.000
Media	1.400	0.409	0.289
Desv.	0.219	0.219	0.165

Tabla 66: Estadísticas de los Paretos generados por el algoritmo SPEA-PG en la consulta 4 de CACM

C. 7	*N_sol*	M_2^*	M_3^*
Ej. 1	5	2.250	0.935
Ej. 2	5	2.250	0.806
Ej. 3	5	2.250	0.943
Ej. 4	3	1.500	0.604
Ej. 5	3	1.000	0.565
Media	4.200	1.850	0.771
Desv.	0.438	0.230	0.072

Tabla 67: Estadísticas de los Paretos generados por el algoritmo SPEA-PG en la consulta 7 de CACM

C. 9	*N_sol*	M_2^*	M_3^*
Ej. 1	1	0.000	0.000
Ej. 2	1	0.000	0.000
Ej. 3	1	0.000	0.000
Ej. 4	1	0.000	0.000
Ej. 5	1	0.000	0.000
Media	1.000	0.000	0.000
Desv.	0.000	0.000	0.000

Tabla 68: Estadísticas de los Paretos generados por el algoritmo SPEA-PG en la consulta 9 de CACM

C. 10	*N_sol*	M_2^*	M_3^*
Ej. 1	7	3.167	1.065
Ej. 2	8	3.714	1.077
Ej. 3	5	2.250	0.931
Ej. 4	6	2.800	1.069
Ej. 5	6	2.600	1.006
Media	6.400	2.906	1.030
Desv.	0.456	0.224	0.025

Tabla 69: Estadísticas de los Paretos generados por el algoritmo SPEA-PG en la consulta 10 de CACM

C. 14	*N_sol*	M_2^*	M_3^*
Ej. 1	3	1.000	0.877
Ej. 2	4	1.667	0.927
Ej. 3	6	2.200	0.911
Ej. 4	7	2.333	0.975
Ej. 5	6	2.600	0.982
Media	5.200	1.960	0.934
Desv.	0.657	0.254	0.018

Tabla 70: Estadísticas de los Paretos generados por el algoritmo SPEA-PG en la consulta 14 de CACM

C. 19	*N_sol*	M_2^*	M_3^*
Ej. 1	1	0.000	0.000
Ej. 2	1	0.000	0.000
Ej. 3	1	0.000	0.000
Ej. 4	2	1.000	0.981
Ej. 5	1	0.000	0.000
Media	1.200	0.200	0.196
Desv.	0.179	0.179	0.175

Tabla 71: Estadísticas de los Paretos generados por el algoritmo SPEA-PG en la consulta 19 de CACM

C. 24	*N_sol*	M_2^*	M_3^*
Ej. 1	2	1.000	0.645
Ej. 2	1	0.000	0.000
Ej. 3	2	1.000	0.556
Ej. 4	1	0.000	0.000
Ej. 5	1	0.000	0.000
Media	1.400	0.400	0.240
Desv.	0.219	0.219	0.132

Tabla 72: Estadísticas de los Paretos generados por el algoritmo SPEA-PG en la consulta 24 de CACM

C. 25	*N_sol*	M_2^*	M_3^*
Ej. 1	12	5.273	1.223
Ej. 2	9	4.125	1.109
Ej. 3	9	4.000	1.018
Ej. 4	10	4.667	1.231
Ej. 5	11	5.100	1.171
Media	10.200	4.663	1.168
Desv.	0.522	0.227	0.024

Tabla 73: Estadísticas de los Paretos generados por el algoritmo SPEA-PG en la consulta 25 de CACM

C. 26	*N_sol*	M_2^*	M_3^*
Ej. 1	2	1.000	0.574
Ej. 2	3	1.500	0.672
Ej. 3	2	1.000	0.429
Ej. 4	4	1.667	0.691
Ej. 5	2	1.000	0.501
Media	2.600	1.233	0.573
Desv.	0.358	0.130	0.044

Tabla 74: Estadísticas de los Paretos generados por el algoritmo SPEA-PG en la consulta 26 de CACM

C. 27	*N_sol*	M_2^*	M_3^*
Ej. 1	6	2.600	1.026
Ej. 2	5	2.000	0.953
Ej. 3	4	2.000	0.852
Ej. 4	4	1.667	0.925
Ej. 5	5	2.500	0.978
Media	4.800	2.153	0.947
Desv.	0.335	0.155	0.026

Tabla 75: Estadísticas de los Paretos generados por el algoritmo SPEA-PG en la consulta 27 de CACM

C. 40	*N_sol*	M_2^*	M_3^*
Ej. 1	1	0.000	0.000
Ej. 2	2	1.000	0.758
Ej. 3	2	1.000	0.606
Ej. 4	1	0.000	0.000
Ej. 5	1	0.000	0.000
Media	1.400	0.400	0.273
Desv.	0.219	0.219	0.151

Tabla 76: Estadísticas de los Paretos generados por el algoritmo SPEA-PG en la consulta 40 de CACM

C. 42	*N_sol*	M_2^*	M_3^*
Ej. 1	3	1.500	0.697
Ej. 2	2	1.000	0.922
Ej. 3	2	1.000	0.437
Ej. 4	3	1.500	0.942
Ej. 5	5	2.500	1.111
Media	3.000	1.500	0.822
Desv.	0.490	0.245	0.104

Tabla 77: Estadísticas de los Paretos generados por el algoritmo SPEA-PG en la consulta 42 de CACM

C. 43	*N_sol*	M_2^*	M_3^*
Ej. 1	6	2.800	1.082
Ej. 2	8	3.174	1.154
Ej. 3	7	3.000	1.155
Ej. 4	9	4.125	1.087
Ej. 5	8	3.714	1.108
Media	7.600	3.471	1.117
Desv.	0.456	0.221	0.014

Tabla 78: Estadísticas de los Paretos generados por el algoritmo SPEA-PG en la consulta 43 de CACM

C. 45	*N_sol*	M_2^*	M_3^*
Ej. 1	3	1.500	0.739
Ej. 2	5	2.500	1.047
Ej. 3	4	2.000	1.046
Ej. 4	6	2.800	1.204
Ej. 5	4	2.000	0.855
Media	4.400	2.160	0.978
Desv.	0.456	0.201	0.073

Tabla 79: Estadísticas de los Paretos generados por el algoritmo SPEA-PG en la consulta 45 de CACM

C. 58	*N_sol*	M_2^*	M_3^*
Ej. 1	6	2.600	1.035
Ej. 2	7	3.167	1.108
Ej. 3	7	3.333	1.097
Ej. 4	6	2.800	1.130
Ej. 5	8	3.571	1.088
Media	6.800	3.094	1.092
Desv.	0.334	0.158	0.014

Tabla 80: Estadísticas de los Paretos generados por el algoritmo SPEA-PG en la consulta 58 de CACM

C. 59	*N_sol*	M_2^*	M_3^*
Ej. 1	8	3.857	1.207
Ej. 2	8	3.571	1.159
Ej. 3	7	3.000	1.062
Ej. 4	9	4.125	1.113
Ej. 5	11	5.000	1.238
Media	8.600	3.911	1.156
Desv.	0.607	0.295	0.028

Tabla 81: Estadísticas de los Paretos generados por el algoritmo SPEA-PG en la consulta 59 de CACM

C. 60	*N_sol*	M_2^*	M_3^*
Ej. 1	3	1.000	0.767
Ej. 2	3	1.500	0.796
Ej. 3	6	3.000	0.959
Ej. 4	3	1.500	0.685
Ej. 5	5	2.250	0.847
Media	4.000	1.850	0.811
Desv.	0.566	0.313	0.041

Tabla 82: Estadísticas de los Paretos generados por el algoritmo SPEA-PG en la consulta 60 de CACM

C. 61	*N_sol*	M_2^*	M_3^*
Ej. 1	4	1.667	0.966
Ej. 2	3	1.500	0.954
Ej. 3	3	1.500	0.803
Ej. 4	3	1.500	0.885
Ej. 5	4	1.333	0.641
Media	3.400	1.500	0.850
Desv.	0.219	0.047	0.053

Tabla 83: Estadísticas de los Paretos generados por el algoritmo SPEA-PG en la consulta 61 de CACM

	N_sol		M_2^*		M_3^*	
	Media	***Desviación***	***Media***	***Desviación***	***Media***	***Desviación***
C. 4	1.400	0.219	0.400	0.219	0.289	0.165
C. 7	4.200	0.438	1.850	0.230	0.771	0.072
C. 9	1.000	0.000	0.000	0.000	0.000	0.000
C. 10	6.400	0.456	2.906	0.224	1.030	0.025
C. 14	5.200	0.657	1.960	0.254	0.934	0.018
C. 19	1.200	0.179	0.200	0.179	0.196	0.175
C. 24	1.400	0.219	0.400	0.219	0.240	0.132
C. 25	**10.200**	0.522	**4.633**	0.227	**1.168**	0.024
C. 26	2.600	0.358	1.233	0.130	0.573	0.044
C. 27	4.800	0.335	2.153	0.155	0.947	0.026
C. 40	1.400	0.219	0.400	0.219	0.273	0.151
C. 42	3.000	0.490	1.500	0.254	0.822	0.104
C. 43	7.600	0.456	3.471	0.221	1.117	0.014
C. 45	4.400	0.456	2.160	0.201	0.978	0.073
C. 58	6.800	0.335	3.094	0.158	1.092	0.014
C. 59	8.600	0.607	3.911	0.295	1.156	0.028
C. 60	4.000	0.566	1.850	0.313	0.811	0.041
C. 61	3.400	0.219	1.500	0.047	0.850	0.053

Tabla 84: Estadísticas de los Paretos generados por el algoritmo SPEA-PG en las consultas de CACM

Análisis de resultados

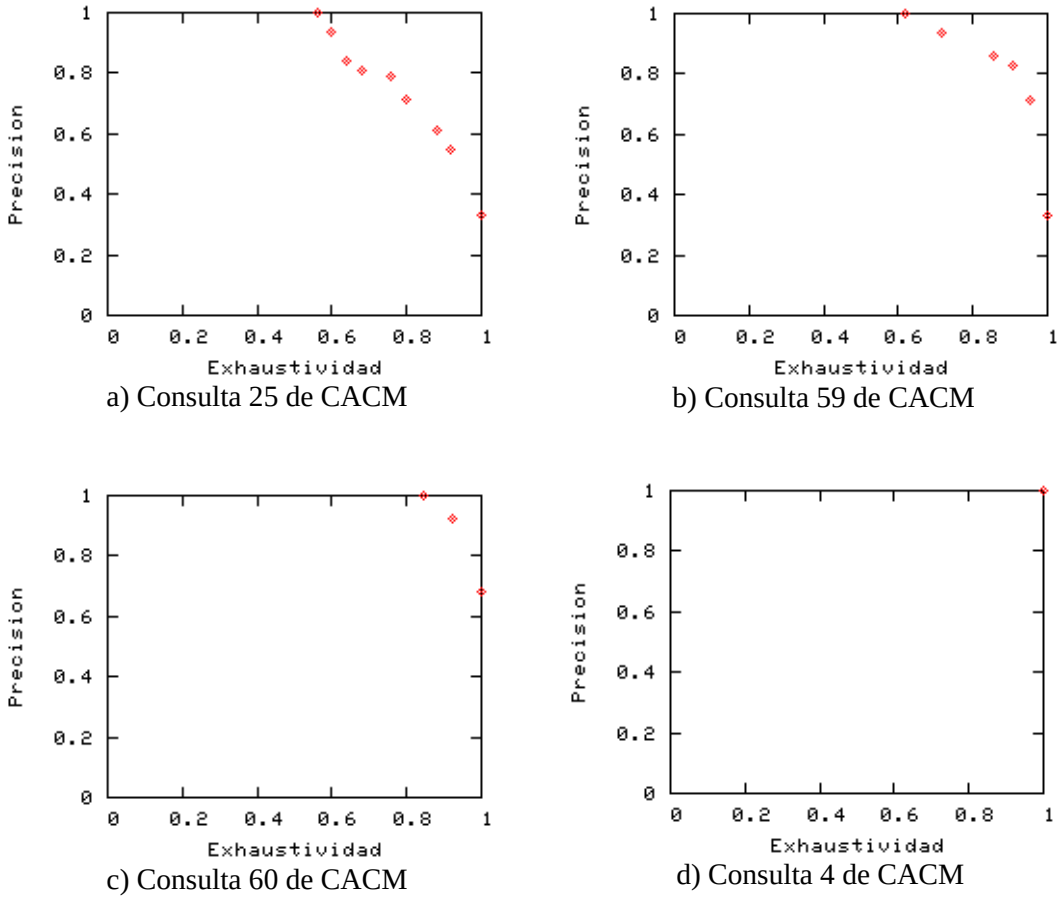
Las conclusiones son similares al caso anterior en lo que respecta a la calidad de los Paretos. De hecho, los números obtenidos son bastante parecidos a los de la experimentación de Cranfield, lo que muestra la robustez del algoritmo.

Las desviaciones típicas se encuentran alrededor de 0.5, 0.25 y 0.07 para el número de soluciones diferentes, M_2^* y M_3^*, respectivamente, lo que indica un comportamiento homogéneo del algoritmo.

Como podemos observar en la Tabla 84, la consulta 25 acapara los mejores valores de las métricas. Así, presenta la media de Paretos con mayor número de soluciones no dominadas con un total de 10.200, el valor más alto en la distribución de soluciones en el frente del

Pareto (4.633) y la mejor media respecto a la extensión del frente no dominado con un total 1.168, cercano al resultado óptimo deseado (1.4142).

La Figura 3.8 muestra los frentes del Pareto generados para las consulta 25, 59, 60 y 4, con 9, 5, 3, y 1 consultas persistentes, respectivamente. El eje X representa los valores de exhaustividad y el eje Y los de precisión. Al igual que en la experimentación de Cranfield, los Paretos obtenidos por las cinco ejecuciones realizadas para cada consulta se han fusionado y, antes de imprimir la curva, las soluciones dominadas se han eliminado de dicho conjunto.



a) Consulta 25 de CACM

b) Consulta 59 de CACM

c) Consulta 60 de CACM

d) Consulta 4 de CACM

Figura 3.8: Frentes de los Paretos derivados por el algoritmo SPEA-PG sobre las consultas 25, 59, 60 y 4 de CACM

NÚMERO DE SOLUCIONES EN LOS FRENTES DE LOS PARETOS DERIVADOS

Al analizar el número de soluciones presentes en los frente de los Paretos con diferente balance entre precisión y exhaustividad, llama la atención que este número puede ser un poco pequeño en comparación con el tamaño de la población elitista. La principal razón de este comportamiento se encuentra en el modo de determinar la similitud entre un par de soluciones (consultas persistentes).

Dos soluciones pueden ser iguales en el espacio de objetivos o en el espacio de decisiones. En RI, si trabajamos en el espacio de objetivos, dos soluciones serán iguales cuando lo sean sus valores de precisión y exhaustividad, independientemente de su estructura. Sin embargo, si trabajamos en el espacio de decisiones, dos soluciones serán iguales cuando coincidan tanto su valores de precisión y exhaustividad como su representación (p.e., composición de la consulta).

En un primer momento, decidimos trabajar en el espacio de objetivos, apoyados en la recomendación de los autores del algoritmo SPEA [193], [194] de utilizar el algoritmo de clustering en este espacio. Sin embargo, tras varias ejecuciones, nos dimos cuenta de que al utilizar este criterio se eliminaban muchas soluciones optimales, cuya diferencia con las mantenidas en la población elitista era únicamente el árbol de la consulta, reduciéndose considerablemente el número de individuos en la población elitista.

Intentando aumentar el número de individuos en la población elitista, optamos por trabajar en el espacio de decisiones, tanto a la hora de aplicar el algoritmo de clustering como a la hora de determinar si dos soluciones eran iguales, haciendo uso de una distancia entre árboles, *distancia de edición* [123].

Aunque la medida aumentó el número de soluciones en la población elitista, este aumento fue desmesurado, provocando que el tiempo de ejecución se multiplicase por diez, sin que se obtuviese una mejora significativa. Finalmente, decidimos mantener la opción original, dejando para trabajos futuros la búsqueda de nuevas funciones de similitud entre árboles de expresión.

3.5.2.3.- Análisis de las consultas representativas del Pareto en la experimentación realizada con Cranfield

Las Tablas 3.81 a 3.97 muestran la eficacia de la recuperación de las consultas seleccionadas del Pareto fusionado obtenido tras combinar los Paretos asociados a cada una de las 5 ejecuciones realizadas, para cada una de las consultas de Cranfield.

Dentro de cada tabla, y de izquierda a derecha, las columnas recogen los datos siguientes:

i) Número de ejecución.

ii) Valores correspondientes a la evaluación sobre el conjunto de entrenamiento de las consultas persistentes: Número de nodos en el árbol que representa la consulta, Precisión, Exhaustividad, Número de documentos relevantes recuperados y Número total de documentos recuperados.

iii) Valores correspondientes a la evaluación sobre el conjunto de prueba de las consultas persistente: Número de nodos en el árbol que representa la consulta, Precisión, Exhaustividad, Número de documentos relevantes recuperados y Número total de documentos recuperados.

C. 1	***ENTRENAMIENTO***					***PRUEBA***				
	Nodos	***P***	***E***	***#rel***	***#rec***	***Nodos***	***P***	***E***	***#rel***	***#rec***
1	19	0.206	1.000	14	68	15	0.100	0.467	7	70
2	19	0.302	0.929	13	43	19	0.212	0.467	7	33
3	19	0.522	0.857	12	23	15	0.333	0.400	6	18
4	19	0.733	0.786	11	15	13	0.400	0.267	4	10
5	19	1.000	0.643	9	9	15	1.000	0.067	1	1

Tabla 85: Eficacia de la recuperación de las consultas persistentes seleccionadas para la consulta 1 de Cranfield

C. 2	*ENTRENAMIENTO*					*PRUEBA*				
	Nodos	*P*	*E*	*#rel*	*#rec*	*Nodos*	*P*	*E*	*#rel*	*#rec*
1	19	0.833	0.833	10	12	15	0.333	0.308	4	12
2	19	0.343	1.000	12	35	17	0.382	1.000	13	34
3	19	1.000	0.583	7	7	15	0.500	0.308	4	8
4	19	0.733	0.917	11	15	19	0.455	0.385	5	11
5	19	1.000	0.583	7	7	15	0.333	0.077	1	3

Tabla 86: Eficacia de la recuperación de las consultas persistentes seleccionadas para la consulta 2 de Cranfield

C. 3	*ENTRENAMIENTO*					*PRUEBA*				
	Nodos	*P*	*E*	*#rel*	*#rec*	*Nodos*	*P*	*E*	*#rel*	*#rec*
1	13	1.000	1.000	4	4	9	0.667	0.800	4	6
2	19	1.000	1.000	4	4	15	1.000	0.200	1	1
3	9	1.000	1.000	4	4	5	0.500	1.000	5	10
4	15	1.000	1.000	4	4	13	0.333	0.200	1	3
5	19	1.000	1.000	4	4	17	0.714	1.000	5	7

Tabla 87: Eficacia de la recuperación de las consultas persistentes seleccionadas para la consulta 3 de Cranfield

C. 7	*ENTRENAMIENTO*					*PRUEBA*				
	Nodos	*P*	*E*	*#rel*	*#rec*	*Nodos*	*P*	*E*	*#rel*	*#rec*
1	17	1.000	1.000	3	3	15	0.000	0.000	0	1
2	19	1.000	1.000	3	3	17	0.000	0.0000	0	2
3	19	1.000	1.000	3	3	17	0.000	0.000	0	2
4	19	1.000	1.000	3	3	17	0.000	0.0000	0	0

Tabla 88: Eficacia de la recuperación de las consultas persistentes seleccionadas para la consulta 7 de Cranfield

C. 8	ENTRENAMIENTO					PRUEBA				
	Nodos	P	E	#rel	#rec	Nodos	P	E	#rel	#rec
1	19	1.000	0.833	5	5	17	0.000	0.000	0	2
2	19	1.000	0.833	5	5	17	0.100	0.167	1	10
3	19	0.857	1.000	6	7	15	0.000	0.000	0	5

Tabla 89: Eficacia de la recuperación de las consultas persistentes seleccionadas para la consulta 8 de Cranfield

C. 11	ENTRENAMIENTO					PRUEBA				
	Nodos	P	E	#rel	#rec	Nodos	P	E	#rel	#rec
1	19	1.000	1.000	4	4	11	0.039	0.750	3	77
2	19	1.000	1.000	4	4	9	0.005	0.750	3	665
3	19	1.000	1.000	4	4	11	0.000	0.000	0	2
4	19	1.000	1.000	4	4	5	0.000	0.000	0	0
5	19	1.000	1.000	4	4	15	0.000	0.000	0	0

Tabla 90: Eficacia de la recuperación de las consultas persistentes seleccionadas para la consulta 11 de Cranfield

C. 19	ENTRENAMIENTO					PRUEBA				
	Nodos	P	E	#rel	#rec	Nodos	P	E	#rel	#rec
1	17	1.000	1.000	5	5	13	0.000	0.000	0	0
2	19	1.000	1.000	5	5	11	0.250	0.200	1	4
3	17	1.000	1.000	5	5	7	0.000	0.000	0	3
4	17	1.000	1.000	5	5	13	0.000	0.000	0	1
5	19	1.000	1.000	5	5	17	0.000	0.000	0	2

Tabla 91: Eficacia de la recuperación de las consultas persistentes seleccionadas para la consulta 19 de Cranfield

C.23	ENTRENAMIENTO					PRUEBA				
	Nodos	P	E	#rel	#rec	Nodos	P	E	#rel	#rec
1	19	0.424	0.875	14	33	19	0.182	0.235	4	22
2	19	0.258	1.000	16	62	19	0.169	0.706	12	71
3	19	0.481	0.812	13	27	19	0.393	0.647	11	28
4	19	0.733	0.688	11	15	15	0.333	0.176	3	9
5	19	1.000	0.500	8	8	19	0.400	0.235	4	10

Tabla 92: Eficacia de la recuperación de las consultas persistentes seleccionadas para la consulta 23 de Cranfield

C. 26	ENTRENAMIENTO					PRUEBA				
	Nodos	P	E	#rel	#rec	Nodos	P	E	#rel	#rec
1	19	1.000	1.000	3	3	11	0.000	0.000	0	23
2	19	1.000	1.000	3	3	17	0.000	0.000	0	3
3	19	1.000	1.000	3	3	17	0.000	0.000	0	1
4	19	1.000	1.000	3	3	17	0.000	0.000	0	0
5	19	1.000	1.000	3	3	17	0.000	0.000	0	1

Tabla 93: Eficacia de la recuperación de las consultas persistentes seleccionadas para la consulta 26 de Cranfield

C. 38	ENTRENAMIENTO					PRUEBA				
	Nodos	P	E	#rel	#rec	Nodos	P	E	#rel	#rec
1	19	1.000	1.000	5	5	17	0.000	0.000	0	5
2	19	1.000	1.000	5	5	15	0.000	0.000	0	4
3	19	1.000	1.000	5	5	15	0.000	0.000	0	4
4	19	1.000	1.000	5	5	13	0.000	0.000	0	3

Tabla 94: Eficacia de la recuperación de las consultas persistentes seleccionadas para la consulta 38 de Cranfield

C. 39	*ENTRENAMIENTO*					*PRUEBA*				
	Nodos	*P*	*E*	*#rel*	*#rec*	*Nodos*	*P*	*E*	*#rel*	*#rec*
1	17	1.000	0.857	6	6	15	0.250	0.143	1	4
2	19	1.000	0.857	6	6	15	0.062	0.143	1	16
3	17	1.000	0.857	6	6	9	0.000	0.000	0	1
4	19	0.875	1.000	7	8	17	0.200	0.143	1	5
5	19	0.875	1.000	7	8	11	0.000	0.000	0	6

Tabla 95: Eficacia de la recuperación de las consultas persistentes seleccionadas para la consulta 39 de Cranfield

C. 40	*ENTRENAMIENTO*					*PRUEBA*				
	Nodos	*P*	*E*	*#rel*	*#rec*	*Nodos*	*P*	*E*	*#rel*	*#rec*
1	19	1.000	1.000	6	6	15	0.071	0.143	1	14
2	19	1.000	1.000	6	6	11	0.000	0.000	0	6
3	19	1.000	1.000	6	6	9	0.000	0.000	0	2
4	19	1.000	1.000	6	6	13	0.000	0.000	0	2

Tabla 96: Eficacia de la recuperación de las consultas persistentes seleccionadas para la consulta 40 de Cranfield

C. 47	*ENTRENAMIENTO*					*PRUEBA*				
	Nodos	*P*	*E*	*#rel*	*#rec*	*Nodos*	*P*	*E*	*#rel*	*#rec*
1	19	1.000	1.000	7	7	19	0.333	0.125	1	3

Tabla 97: Eficacia de la recuperación de las consultas persistentes seleccionadas para la consulta 47 de Cranfield

C. 73	ENTRENAMIENTO					PRUEBA				
	Nodos	P	E	#rel	#rec	Nodos	P	E	#rel	#rec
1	17	1.000	0.800	8	8	15	0.182	0.182	2	11
2	19	0.818	0.900	9	11	13	0.053	0.182	2	38
3	19	0.556	1.000	10	18	15	0.167	0.182	2	12

Tabla 98: Eficacia de la recuperación de las consultas persistentes seleccionadas para la consulta 73 de Cranfield

C. 157	ENTRENAMIENTO					PRUEBA				
	Nodos	P	E	#rel	#rec	Nodos	P	E	#rel	#rec
1	19	0.303	1.000	20	66	19	0.167	0.500	10	60
2	19	0.362	0.850	17	47	15	0.200	0.450	9	45
3	19	0.432	0.800	16	37	17	0.250	0.400	8	32
4	19	0.667	0.700	14	21	17	0.300	0.300	6	20
5	19	1.000	0.500	10	10	19	0.444	0.200	4	9

Tabla 99: Eficacia de la recuperación de las consultas persistentes seleccionadas para la consulta 157 de Cranfield

C. 220	ENTRENAMIENTO					PRUEBA				
	Nodos	P	E	#rel	#rec	Nodos	P	E	#rel	#rec
1	19	1.000	0.800	8	8	19	0.400	0.200	2	5
2	19	1.000	0.800	8	8	17	0.143	0.200	2	14
3	19	0.900	0.900	9	10	19	0.200	0.100	1	5
4	19	0.900	0.900	9	10	17	0.143	0.200	2	14
5	19	0.588	1.000	10	17	17	0.250	0.100	1	4

Tabla 100: Eficacia de la recuperación de las consultas persistentes seleccionadas para la consulta 220 de Cranfield

C. 225	ENTRENAMIENTO					PRUEBA				
	Nodos	P	E	#rel	#rec	Nodos	P	E	#rel	#rec
1	19	0.423	0.917	11	26	17	0.167	0.231	3	18
2	19	0.203	1.000	12	59	19	0.053	0.231	3	57
3	19	0.455	0.833	10	22	15	0.143	0.154	2	14
4	19	0.750	0.750	9	12	17	0.231	0.231	3	13
5	19	1.000	0.667	8	8	15	0.047	0.308	4	85

Tabla 101: Eficacia de la recuperación de las consultas persistentes seleccionadas para la consulta 225 de Cranfield

Análisis de resultados

A continuación, analizaremos los resultados presentados en las tablas anteriores. En primer lugar, comentaremos los resultados de cada grupo de consultas (consultas con más de 20 documentos relevantes y consultas que presentan menos de 15 documentos relevantes, Tabla 23) de acuerdo a la efectividad de la recuperación. Seguidamente, analizaremos el tamaño de las mejores consultas persistentes aprendidas y el tiempo que el algoritmo tardó en generarlas. Estos dos puntos se analizarán independientemente del número de documentos relevantes que tenga asociados la consulta.

Eficacia de las consultas con más de 20 documentos relevantes

Al unificar los Paretos obtenidos en cada una de las ejecuciones correspondientes a una consulta y seleccionar un conjunto representativo, de acuerdo al procedimiento descrito en la Sección 3.5.2, observamos como dicho conjunto se compone de diversas consultas persistentes con diferente equilibrio precisión-exhaustividad, objetivo de esta propuesta.

La diversidad de soluciones se mueve entre consultas persistentes que son capaces de recuperar únicamente documentos relevantes, alrededor de la mitad, y consultas persistentes que representan perfectamente las necesidades del usuario y recuperan todos documentos relevantes, aunque junto con basura (documentos irrelevantes). Las soluciones intermedias (precisión y exhaustividad entre 0.7 y 0.8) son las que mejor balance precisión-exhaustividad presentan, recuperando casi todos los documentos relevantes, sin que dicha recuperación vaya

acompañada de mucho ruido.

Si nos fijamos en la Tabla 3.81, correspondiente a la consulta 1, podemos ver como la mejor consulta es la que ocupa la cuarta posición. Ésta recupera 15 documentos, de los cuales 11 son relevantes, obteniéndose una precisión de 0.733 y una exhaustividad de 0.786.

La eficacia de estas consultas a la hora de recuperar nuevos documentos que se adapten a la necesidades de un usuario (evaluación sobre el conjunto de prueba) es prometedora. Todas las consultas recuperan alrededor de 5 documentos relevantes, incluso algunas recuperan la totalidad de ellos. Además, la precisión se mantiene por encima de 0.3, lo que indica que el acceso a los nuevos documentos, por parte de los usuarios, no se verá excesivamente entorpecido por una maraña de documentos irrelevantes.

Eficacia de las consultas con menos de 15 documentos relevantes

Como ya vimos en la Sección 3.5.2.1, al analizar los Paretos obtenidos por consultas con pocos documentos relevantes asociados, los frentes de dichos Paretos estaban formados por un número muy reducido de soluciones (una o dos) diferentes en el espacio objetivo, en su mayoría soluciones con una precisión y exhaustividad iguales a 1. En consecuencia, al fusionar los Paretos, parece normal que todas las soluciones incluidas en él presenten dichos valores, independientemente de que tengan o no la misma estructura. De ahí, que casi la totalidad de las consultas persistentes con menos de 15 documento relevantes mostradas en las tablas anteriores tengan valores máximos de precisión y exhaustividad. Además, en algunas consultas, el número de soluciones diferentes, no sólo en el espacio de objetivos sino también en el espacio de decisiones, es inferior a cinco, lo que no permite seleccionar un conjunto completo de cinco soluciones.

En contraste con lo anterior, el comportamiento de estas consultas deja bastante que desear al ser evaluadas sobre el conjunto de prueba. Dejando a un lado la consulta 3, que presenta uno de los mejores comportamientos, el resto de consultas no consiguen recuperar ningún documentos que se adapte a las necesidades de información del usuario.

Mejor consulta

Podemos distinguir dos consultas, aunque cada una desde un punto de vista: la que mayor variedad de soluciones con diferente balance de precisión-exhaustividad consigue y la que mejor se comporta en el proceso de obtener nuevos documentos.

La primera a la que nos referimos es la consulta 1. En la Tabla 85 podemos ver como las cinco soluciones presentan diferente equilibrio entre los criterios de precisión y exhaustividad. Comienzan con valores bajos de precisión (0.206) y altos de exhaustividad (1.00) y van aumentándose y disminuyéndose, respectivamente, hasta llegar a una solución con unos valores de precisión y exhaustividad de 1.00 y 0.643.

La segunda consulta es la número 3. Esta consulta no consigue un espectro variado de soluciones sino, al contrario, todas la consultas persistente obtenidas se concentran en un único punto, precisión y exhaustividad iguales a 1. Sin embargo, y a diferencia del resto de soluciones con este comportamiento, al ser evaluada sobre los documentos de prueba, los resultados obtenidos son muy favorables, como muestra la Tabla 87. En tres de las cinco soluciones seleccionadas consigue recuperar prácticamente la totalidad de documentos relevantes, con apenas documentos irrelevantes.

Tamaño de los árboles

En todas las ejecuciones, excepto dos de la consulta 3, la población ha convergido hasta presentar consultas de gran tamaño, lo que permite obtener mejor eficacia en la recuperación. Así, en la mayoría de los casos, las consultas persistentes aprendidas están compuestas por 19 nodos, el tamaño máximo permitido.

Tiempo de ejecución

Las Tablas 3.98 y 3.99 muestran los tiempos de ejecución del algoritmo SPEA-PG para cada una de las diecisiete consultas de Cranfield, así como las medias y las desviaciones típicas. Los tiempos aparecen expresados en minutos y segundos, y las ejecuciones más rápidas están marcadas en negrita.

.

	C. 1	*C. 2*	*C. 3*	*C. 7*	*C. 8*	*C. 11*	*C. 19*	*C. 23*
Ej. 1	01:36	01:35	01:33	01:38	01:37	01:37	01:35	01:41
Ej. 2	01:37	**01:33**	01:30	01:37	01:36	01:32	01:36	01:42
Ej. 3	01:36	01:35	01:31	**01:35**	01:37	01:37	01:37	01:42
Ej. 4	**01:36**	01:36	**01:30**	01:37	**01:35**	01:31	**01:35**	01:41
Ej. 5	01:36	01:34	01:30	01:36	01:35	**01:31**	01:36	**01:39**
Media	01:36	01:35	01:31	01:37	01:36	01:34	01:36	01:41
Desv.	00:00	00:01	00:01	00:01	00:01	00:03	00:01	00:01

Tabla 102: Tiempos de ejecución del algoritmo SPEA-PG para la colección Cranfield

	C. 26	*C. 38*	*C. 39*	*C. 40*	*C. 47*	*C. 73*	*C. 157*	*C. 220*	*C. 225*
Ej. 1	01:36	01:37	01:35	01:39	**01:35**	01:35	01:38	01:41	01:39
Ej. 2	01:36	01:37	**01:33**	01:39	01:36	01:33	01:40	01:39	01:41
Ej. 3	01:38	01:39	01:36	01:42	01:38	01:34	01:41	01:39	**01:37**
Ej. 4	**01:34**	**01:34**	01:35	01:42	01:36	01:36	**01:36**	01:41	01:37
Ej. 5	01:35	01:36	01:33	**01:37**	01:35	**01:33**	01:39	**01:38**	01:40
Media	01:36	01:37	01:34	01:40	01:36	01:34	01:39	01:40	01:39
Desv.	00:01	00:02	00:01	00:02	00:01	00:01	00:02	00:01	00:02

Tabla 103: Tiempos de ejecución del algoritmo SPEA-PG para la colección Cranfield

La media ronda el minuto y 37 segundos en las diecisiete consultas ejecutadas. Además, la desviación típica es muy reducida en todos los casos, no superando nunca los 3 segundos, lo que nos indica la poca variación existente. La ejecución más rápida la encontramos en la consulta 3, ejecución 4, con un tiempo de 1 minuto y 30 segundos, mientras que la más lenta corresponde a la consulta 40, también en la ejecución 4, con 1 minuto y 42 segundos

3.5.2.4.- Análisis de las consultas representativas del Pareto en la experimentación realizada con CACM

Las Tablas 3.100 a 3.117 muestran la eficacia de la recuperación de las consultas seleccionadas del Pareto fusionado, obtenido tras combinar los Paretos asociados a cada una de las 5 ejecuciones realizadas, para cada una de las consultas de Cranfield.

C. 4	***ENTRENAMIENTO***					***PRUEBA***				
	Nodos	***P***	***E***	***#rel***	***#rec***	***Nodos***	***P***	***E***	***#rel***	***#rec***
1	19	1.000	1.000	6	6	17	0.000	0.000	0	4
2	17	1.000	1.000	6	6	15	1.000	0.333	2	2
3	17	1.000	1.000	6	6	13	0.500	0.167	1	2

Tabla 104: Eficacia de la recuperación de las consultas persistentes seleccionadas para la consulta 4 de CACM

C. 7	***ENTRENAMIENTO***					***PRUEBA***				
	Nodos	***P***	***E***	***#rel***	***#rec***	***Nodos***	***P***	***E***	***#rel***	***#rec***
1	19	1.000	0.857	12	12	17	0.231	0.214	3	13
2	17	1.000	0.857	12	12	13	0.231	0.214	3	13
3	19	0.867	0.929	13	15	17	0.375	0.429	6	16
4	19	0.867	0.929	13	15	13	0.353	0.429	6	17
5	19	0.824	1.000	14	17	17	0.009	0.714	10	1133

Tabla 105: Eficacia de la recuperación de las consultas persistentes seleccionadas para la consulta 7 de CACM

C. 9	*ENTRENAMIENTO*					*PRUEBA*				
	Nodos	*P*	*E*	*#rel*	*#rec*	*Nodos*	*P*	*E*	*#rel*	*#rec*
1	17	1.000	1.000	4	4	15	0.008	0.400	2	245
2	17	1.000	1.000	4	4	15	0.000	0.000	0	1
3	15	1.000	1.000	4	4	13	0.000	0.000	0	4
4	17	1.000	1.000	4	4	13	0.500	0.400	2	4
5	9	1.000	1.000	4	4	7	0.667	0.400	2	3

Tabla 106: Eficacia de la recuperación de las consultas persistentes seleccionadas para la consulta 9 de CACM

C. 10	*ENTRENAMIENTO*					*PRUEBA*				
	Nodos	*P*	*E*	*#rel*	*#rec*	*Nodos*	*P*	*E*	*#rel*	*#rec*
1	19	0.500	0.941	16	32	11	0.255	0.667	12	47
2	19	0.486	1.000	17	35	11	0.378	0.778	14	37
3	19	0.556	0.882	15	27	13	0.333	0.556	13	30
4	19	0.765	0.765	13	17	13	0.368	0.389	7	19
5	19	1.000	0.647	11	11	11	0.333	0.167	3	9

Tabla 107: Eficacia de la recuperación de las consultas persistentes seleccionadas para la consulta 10 de CACM

C. 14	*ENTRENAMIENTO*					*PRUEBA*				
	Nodos	*P*	*E*	*#rel*	*#rec*	*Nodos*	*P*	*E*	*#rel*	*#rec*
1	19	0.618	0.955	21	34	13	0.500	0.818	18	36
2	19	0.595	1.000	22	37	13	0.514	0.864	19	37
3	19	0.595	1.000	22	37	13	0.014	1.000	22	1563
4	19	0.625	0.909	20	32	15	0.571	0.909	20	35
5	19	1.000	0.636	14	14	13	0.525	0.955	21	40

Tabla 108: Eficacia de la recuperación de las consultas persistentes seleccionadas para la consulta 14 de CACM

C. 19	*ENTRENAMIENTO*					*PRUEBA*				
	Nodos	*P*	*E*	*#rel*	*#rec*	*Nodos*	*P*	*E*	*#rel*	*#rec*
1	19	1.000	1.000	5	5	13	0.250	0.167	1	4
2	17	1.000	1.000	5	5	9	0.000	0.000	0	0
3	19	1.000	1.000	5	5	17	0.000	0.000	0	1
4	19	1.000	1.000	5	5	11	0.000	0.000	0	0

Tabla 109: Eficacia de la recuperación de las consultas persistentes seleccionadas para la consulta 19 de CACM

C. 24	*ENTRENAMIENTO*					*PRUEBA*				
	Nodos	*P*	*E*	*#rel*	*#rec*	*Nodos*	*P*	*E*	*#rel*	*#rec*
1	19	1.000	1.000	6	6	11	0.000	0.000	0	0
2	19	1.000	1.000	6	6	17	0.000	0.000	0	2
3	17	1.000	1.000	6	6	13	0.000	0.000	0	0

Tabla 110: Eficacia de la recuperación de las consultas persistentes seleccionadas para la consulta 24 de CACM

C. 25	*ENTRENAMIENTO*					*PRUEBA*				
	Nodos	*P*	*E*	*#rel*	*#rec*	*Nodos*	*P*	*E*	*#rel*	*#rec*
1	19	0.611	0.880	22	36	15	0.364	0.462	12	33
2	19	0.333	1.000	25	75	19	0.232	0.615	16	69
3	19	0.548	0.920	23	42	19	0.382	0.500	13	34
4	19	0.792	0.760	19	24	15	0.294	0.192	5	17
5	19	1.000	0.560	14	14	15	0.100	0.038	1	10

Tabla 111: Eficacia de la recuperación de las consultas persistentes seleccionadas para la consulta 25 de CACM

C. 26	*ENTRENAMIENTO*					*PRUEBA*				
	Nodos	*P*	*E*	*#rel*	*#rec*	*Nodos*	*P*	*E*	*#rel*	*#rec*
1	15	1.000	0.933	14	14	11	0.529	0.600	9	17
2	19	0.938	1.000	15	16	13	0.333	0.333	5	15

Tabla 112: Eficacia de la recuperación de las consultas persistentes seleccionadas para la consulta 26 de CACM

C. 27	*ENTRENAMIENTO*					*PRUEBA*				
	Nodos	*P*	*E*	*#rel*	*#rec*	*Nodos*	*P*	*E*	*#rel*	*#rec*
1	19	1.000	0.786	11	11	11	0.667	0.133	2	3
2	19	0.800	0.857	12	15	19	0.308	0.267	4	13
3	19	0.722	0.929	13	18	17	0.231	0.200	3	13
4	19	0.560	1.000	14	25	13	0.158	0.200	3	19

Tabla 113: Eficacia de la recuperación de las consultas persistentes seleccionadas para la consulta 27 de CACM

C. 40	*ENTRENAMIENTO*					*PRUEBA*				
	Nodos	*P*	*E*	*#rel*	*#rec*	*Nodos*	*P*	*E*	*#rel*	*#rec*
1	19	1.000	1.000	5	5	17	0.000	0.000	0	5
2	19	1.000	1.000	5	5	13	0.000	0.000	0	3
3	19	1.000	1.000	5	5	15	0.200	0.200	1	5

Tabla 114: Eficacia de la recuperación de las consultas persistentes seleccionadas para la consulta 40 de CACM

C. 42	*ENTRENAMIENTO*					*PRUEBA*				
	Nodos	*P*	*E*	*#rel*	*#rec*	*Nodos*	*P*	*E*	*#rel*	*#rec*
1	19	1.000	0.900	9	9	5	0.143	0.091	1	7
2	19	1.000	0.900	9	9	7	0.400	0.182	2	5
3	19	0.909	1.000	10	11	5	0.111	0.091	1	9

Tabla 115: Eficacia de la recuperación de las consultas persistentes seleccionadas para la consulta 42 de CACM

C. 43	*ENTRENAMIENTO*					*PRUEBA*				
	Nodos	*P*	*E*	*#rel*	*#rec*	*Nodos*	*P*	*E*	*#rel*	*#rec*
1	19	0.317	1.000	20	63	19	0.183	0.524	11	60
2	19	0.395	0.850	17	43	17	0.013	1.000	21	1603
3	19	0.358	0.950	19	53	13	0.280	0.667	14	50
4	19	0.727	0.800	16	22	19	0.375	0.429	9	24
5	19	1.000	0.650	13	13	9	0.333	0.095	2	6

Tabla 116: Eficacia de la recuperación de las consultas persistentes seleccionadas para la consulta 43 de CACM

C. 45	*ENTRENAMIENTO*					*PRUEBA*				
	Nodos	*P*	*E*	*#rel*	*#rec*	*Nodos*	*P*	*E*	*#rel*	*#rec*
1	17	1.000	0.769	10	10	9	0.000	0.000	0	0
2	19	1.000	0.769	10	10	13	0.438	0.538	7	16
3	19	0.923	0.923	12	13	11	0.500	0.538	7	14
4	19	0.684	1.000	13	19	11	0.333	0.538	7	21

Tabla 117: Eficacia de la recuperación de las consultas persistentes seleccionadas para la consulta 45 de CACM

C. 58	*ENTRENAMIENTO*					*PRUEBA*				
	Nodos	*P*	*E*	*#rel*	*#rec*	*Nodos*	*P*	*E*	*#rel*	*#rec*
1	19	0.667	0.800	12	18	7	0.009	1.000	15	1602
2	19	0.395	1.000	15	38	11	0.171	0.467	7	41
3	19	0.591	0.867	13	22	11	0.154	0.267	4	26
4	19	0.786	0.733	11	14	9	0.211	0.267	4	19
5	19	1.000	0.533	8	8	11	0.364	0.267	4	11

Tabla 118: Eficacia de la recuperación de las consultas persistentes seleccionadas para la consulta 58 de CACM

C. 59	*ENTRENAMIENTO*					*PRUEBA*				
	Nodos	*P*	*E*	*#rel*	*#rec*	*Nodos*	*P*	*E*	*#rel*	*#rec*
1	19	0.826	0.905	19	23	13	0.619	0.591	13	21
2	19	0.333	1.000	21	63	17	0.274	0.773	17	62
3	17	0.938	0.714	15	16	13	0.800	0.545	12	15
4	19	0.714	0.952	20	28	15	0.577	0.682	15	26
5	19	1.000	0.619	13	13	11	0.833	0.227	5	6

Tabla 119: Eficacia de la recuperación de las consultas persistentes seleccionadas para la consulta 59 de CACM

C. 60	*ENTRENAMIENTO*					*PRUEBA*				
	Nodos	*P*	*E*	*#rel*	*#rec*	*Nodos*	*P*	*E*	*#rel*	*#rec*
1	19	1.000	0.846	11	11	13	0.083	0.071	1	12
2	19	1.000	0.846	11	11	11	0.083	0.071	1	12
3	15	1.000	0.846	11	11	9	0.143	0.143	2	14
4	19	0.923	0.923	12	13	15	0.077	0.071	1	13
5	19	0.684	1.000	13	19	13	0.300	0.429	6	20

Tabla 120: Eficacia de la recuperación de las consultas persistentes seleccionadas para la consulta 60 de CACM

C. 61	*ENTRENAMIENTO*					*PRUEBA*				
	Nodos	***P***	***E***	***#rel***	***#rec***	***Nodos***	***P***	***E***	***#rel***	***#rec***
1	19	0.875	0.933	14	16	15	0.455	0.312	5	11
2	19	0.929	0.867	13	14	9	0.600	0.188	3	5
3	19	0.875	0.933	14	16	17	0.500	0.250	4	8
4	19	0.789	1.000	15	19	13	0.455	0.312	5	11
5	19	1.000	0.800	12	12	13	0.750	0.188	3	4

Tabla 121: Eficacia de la recuperación de las consultas persistentes seleccionadas para la consulta 61 de CACM

Análisis de resultados

A continuación, analizaremos los resultados presentados en las tablas anteriores. Al igual que con la colección Cranfield, en primer lugar comentaremos los resultados de cada grupo de consultas (consultas con más de 30 documentos relevantes, consultas que presentan entre 21 y 30 documentos relevantes, y consultas con menos de 15 documentos relevantes, Tabla 44) de acuerdo a la efectividad de la recuperación. Seguidamente, analizaremos el tamaño de las mejores consultas persistentes aprendidas y el tiempo que el algoritmo tardó en generarlas. Estos dos puntos se analizarán independientemente del número de documentos relevantes que tenga asociados la consulta.

Eficacia de las consultas con más de 30 documentos relevantes

Igual que en la colección Cranfield, el conjunto de soluciones seleccionado de acuerdo al procedimiento descrito en la Sección 3.5.2 se compone de diversas consultas persistentes con diferente equilibrio precisión-exhaustividad, objetivo de esta propuesta.

La diversidad de soluciones se mueve entre consultas persistentes que son capaces de recuperar únicamente documentos relevantes, alrededor de la mitad, y consultas persistentes que representan perfectamente las necesidades del usuario y recuperan todos documentos relevantes, aunque junto con basura (documentos irrelevantes). Las soluciones intermedias (precisión y exhaustividad entre 0.7 y 0.8) son las que mejor balance precisión-exhaustividad presentan, recuperando casi todos los documentos relevantes y disminuyendo el ruido.

Si nos fijamos en la Tabla 3.117, correspondiente a la consulta 61, podemos ver como la mejor consulta es la que ocupa la tercera posición. Ésta recupera 16 documentos, de los cuales 14 son relevantes, obteniéndose una precisión de 0.875 y una exhaustividad de 0.933.

La eficacia de estas consultas a la hora de recuperar nuevos documentos que se adapten a la necesidades de un usuario (evaluación sobre el conjunto de prueba) es prometedora, consiguiendo mejores resultados aquellas consultas que presentaban valores intermedios de precisión y exhaustividad.

Los valores de precisión y exhaustividad son bastante buenos. Excepto en dos ocasiones en los que la recuperación de documentos relevantes va acompañada de mucho ruido, el resto mantienen la precisión alrededor de 0.5.

Eficacia de las consultas que presentan entre 21 y 30 documentos relevantes

Estas consultas tienen un comportamiento muy similar al descrito en la sección anterior, aunque empieza a verse como los conjuntos de consultas seleccionadas tienden a estar formados cada vez más por soluciones situadas en la misma zona del espacio. En concreto, son consultas persistentes con valores de precisión y exhaustividad cercanas a 1. Además, aparecen consultas cuyos Paretos unificados no tienen ni siquiera cinco soluciones diferentes en el espacio de objetivos y decisiones (Tabla 115) .

En lo que respecta a la evaluación sobre el conjunto de prueba, siguen siendo las soluciones intermedias las que obtienen mejores resultados. Además, las soluciones con valores muy extremos en cualquiera de los objetivos, sobre todo las pertenecientes a los Paretos con menos de cinco soluciones diferentes, se comportan de manera pésima al ser evaluadas sobre nuevos documentos, empezando a apreciarse un comportamiento que será generalizado en las consultas con menos de 15 documentos relevantes.

Eficacia de las consultas con menos de 15 documentos relevantes

Como ya vimos en la Sección 3.5.2.2 al analizar los Paretos obtenidos por consultas con pocos documentos relevantes asociados, los frentes de dichos Paretos estaban formados por un número muy reducido de soluciones (una o dos) diferentes en el espacio objetivo, en su mayoría soluciones con una precisión y exhaustividad iguales a 1. En consecuencia, al

fusionar los Paretos, parece normal que todas las soluciones incluidas en ellos presenten dichos valores, independientemente de que tengan o no la misma estructura. De ahí, que la totalidad de las consultas persistentes mostradas en las tablas anteriores con menos de 15 documentos relevantes, presenten valores máximos de precisión y exhaustividad. Además, en algunas consultas, el número de soluciones diferentes, no sólo en el espacio de objetivos, sino también en el espacio de decisiones, es inferior a cinco, lo que no permite seleccionar un conjunto completo de cinco soluciones.

En contraste con lo anterior, el comportamiento de estas consultas deja bastante que desear al ser evaluadas sobre el conjunto de prueba. Ninguna de las consultas consigue recuperar ningún documento que se adapte a las necesidades de información del usuario, o a lo sumo 1, igual que ocurría con las consultas de Cranfield con menos de 15 documentos relevantes.

Mejor consulta

La consulta que presentan una mayor diversidad en las soluciones que forman el frente del Pareto es la consulta 25. En la Tabla 111 podemos ver como las cinco soluciones presentan diferente equilibrio entre los criterios de precisión y exhaustividad. Comienzan con valores bajo de precisión (0.333) y altos de exhaustividad (1.00) y van aumentando y disminuyendo, respectivamente, hasta la solución con valores de precisión y exhaustividad iguales a 1.00 y 0.560.

Tamaño de los árboles

En lo que respecta a este aspecto, encontramos de nuevo el mismo comportamiento que en Cranfield: el algoritmo converge hasta presentar consultas de gran tamaño para mejorar la efectividad de la recuperación.

Tiempo de ejecución

Las Tablas 3.118 y 3119 muestran los tiempos de ejecución del algoritmo SPEA-PG para cada una de las dieciocho consultas de CACM, así como las medias y las desviaciones típicas. Los tiempos aparecen expresados en minutos y segundos, y las ejecuciones más rápidas están marcadas en negrita.

	C. 4	*C. 7*	*C. 9*	*C. 10*	*C. 14*	*C. 19*	*C. 24*	*C. 25*	*C. 26*
Ej. 1	06:53	06:59	06:57	**06:52**	**07:06**	**06:56**	06:55	07:09	**06:51**
Ej. 2	06:56	06:58	06:56	06:56	07:11	06:58	06:57	**07:04**	06:52
Ej. 3	**06:53**	06:55	06:54	06:57	07:12	06:57	06:58	07:07	06:57
Ej. 4	06:56	07:00	**06:49**	06:58	07:14	07:01	06:59	07:11	06:51
Ej. 5	07:00	**06:54**	06:51	07:00	07:08	06:59	**06:54**	07:13	06:55
Media	06:56	06:57	06:53	06:57	07:10	06:58	06:57	07:09	06:53
Desv.	00:03	00:03	00:03	00:03	00:03	00:02	00:02	00:03	00:03

Tabla 122: Tiempos de ejecución del algoritmo SPEA-PG para la colección CACM

	C. 27	*C. 40*	*C. 42*	*C. 43*	*C. 45*	*C. 58*	*C. 59*	*C. 60*	*C. 61*
Ej. 1	07:00	07:02	07:01	07:17	**06:52**	06:58	07:14	**06:54**	07:02
Ej. 2	06:57	**06:58**	06:58	07:18	06:59	**06:55**	07:10	06:57	**07:00**
Ej. 3	07:02	07:00	07:02	07:20	07:00	06:59	07:23	06:58	07:06
Ej. 4	07:01	06:59	07:01	07:19	07:02	06:56	**07:10**	06:58	07:06
Ej. 5	**06:57**	06:59	**06:58**	**07:14**	06:59	06:57	07:25	06:58	07:08
Media	06:59	07:00	07:00	07:18	06:58	06:57	07:16	06:57	07:04
Desv.	00:02	00:02	00:02	00:02	00:04	00:02	00:07	00:02	00:03

Tabla 123: Tiempos de ejecución del algoritmo SPEA-PG para la colección CACM

Los valores se mueven alrededor de los 7 minutos, con desviaciones típicas que sólo superan los 3 segundos en dos ocasiones (consultas 47 y 59). La ejecución más rápida la encontramos en la consulta 9, ejecución 5, con un tiempo de 6 minutos y 49 segundos, mientras que la más lenta corresponde a la consulta 59 en la ejecución 5, con 7 minutos y 25 segundos.

3.5.2.5.- Resumen

Globalmente, podemos destacar los siguientes aspectos en los experimentos realizados con el algoritmo SPEA-PG sobre las bases Cranfield y CACM.

- El número de soluciones con diferente balance entre precisión y exhaustividad presentes en los frentes de los Paretos es pequeño en comparación con el tamaño de la población elitista. Este comportamiento se debe a que consideramos que dos soluciones son iguales si lo son en el espacio de objetivos, independientemente de la estructura que tengan.

- Consultas con un mayor número de documentos relevantes asociados consiguen Paretos con un mayor número de soluciones y, además, cubren una zona más amplia del espacio. Muestra de esto, aparte de los valores de las métricas, son los conjuntos de soluciones obtenidos. En ellos podemos observar como las consultas con más documentos relevantes derivan en una sola ejecución varias consultas con diferente balance precisión-exhaustividad, mientras que, conforme disminuye el número de documentos relevantes, los conjuntos presentan menor diversidad de consultas persistentes, la mayoría de ellas con valores muy altos de precisión y exhaustividad.

- La eficacia de las consultas para obtener nuevos documentos relevantes, no conocidos durante el proceso de aprendizaje, es mayor para aquellas consultas que, en vez de presentar valores extremos de los criterios, consiguen equilibrarlos.

- Existe sobreaprendizaje en las consultas con muy pocos documentos relevantes. Consiguen muy buenos resultados sobre los conjuntos de entrenamiento, pero al evaluarlas sobre nuevos conjuntos, presentan un comportamiento pésimo.

3.5.3.- Algoritmo Mono-objetivo *versus* Algoritmo Multiobjetivo

El principal objetivo del uso de un enfoque multiobjetivo para el aprendizaje automático de consultas persistentes, ha quedado patente en los resultado analizados en la sección anterior, al generarse varias consultas persistentes con diferente balance de precisión y exhaustividad en una única ejecución.

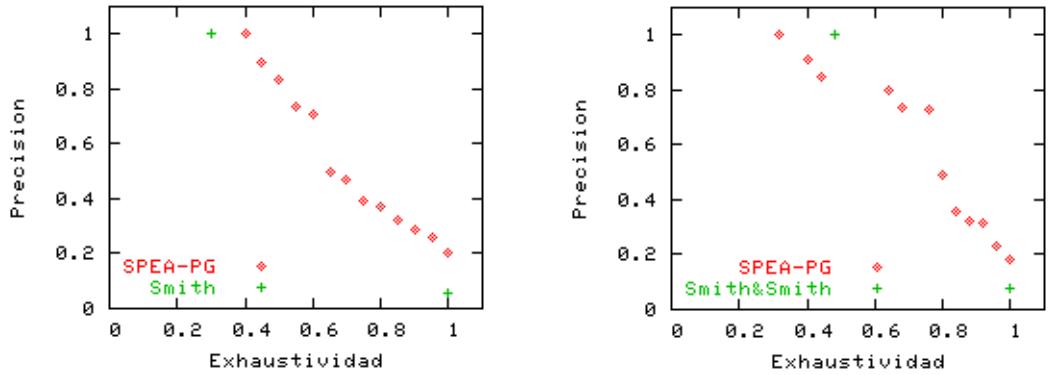
Para terminar el capítulo, vamos a comparar los resultados obtenidos por nuestro algoritmo frente a los obtenidos por el algoritmo de Smith y Smith [165], citando resultados concretos.

NÚMERO DE SOLUCIONES OBTENIDAS

Al trabajar con una optimización independiente de los dos criterios, nuestra propuesta multiobjetivo para aprender consultas persistentes representadas como consultas Booleanas clásicas, consigue un mayor equilibrio entre precisión y exhaustividad en las consultas persistentes. Además, en una sola ejecución, se generan varias consultas persistentes con diferente balance precisión-exhaustividad, frente a la única que genera el algoritmo mono-objetivo.

Aunque nos encontramos casos en los que el número de soluciones diferentes se reduce a una sola, (p.e., las consultas 3, 7, 11 y 19 de Cranfield, o las consultas 9 y 19 de CACM), lo que equipara nuestro algoritmo al algoritmo mono-objetivo en lo que a número de consultas persistentes generadas por ejecución se refiere, esta solución se localiza en la esquina superior derecha del espacio de búsqueda presentando, lógicamente, una precisión y exhaustividad de 1.

En el resto de de casos, en una única ejecución se consiguen diversas consultas que se localizan a lo largo del espacio de búsqueda, consiguiéndose mayor diversidad que con el algoritmo de Smith y Smith. Así por ejemplo, la segunda ejecución de la consulta 157 de Cranfield y la primera ejecución de la consulta 25 de CACM generan 13 y 12 consultas persistes, respectivamente, frente a la única que deriva el algoritmo mono-objetivo de Smith y Smith. La Figura 3.9 muestra la distribución de las soluciones para cada una de estas ejecuciones, así como la mejor solución en exhaustividad y precisión generada por Smith y Smith.



a) Consulta 157 de Cranfield b) Consulta 25 de CACM

Figura 3.9: Distribución de las soluciones generadas por SPEA-PG y Smith & Smith

EFICACIA DE LOS ALGORITMOS

En el análisis de resultados correspondiente a cada algoritmo, hemos distinguido dos y tres grupos de consultas, para las colecciones Cranfield y CACM, respectivamente. Globalmente, podemos realizar una única división, consultas con menos de 15 documentos relevantes y consultas con más de 20 documentos relevantes.

Consultas con más de 20 documentos relevantes

Centrándonos en las consultas con más de 20 documentos relevantes, vemos que nuestro algoritmo recupera en media muchos más documentos relevantes que el algoritmo de Smith y Smith, reduciendo, además, la cantidad de basura que acompaña a la recuperación. Si nos fijamos en la mejores soluciones en precisión y exhaustividad generadas por cada algoritmo vemos como nuestra propuesta multiobjetivo obtiene mejores resultados que el algoritmo mono-objetivo (véanse las Tablas 3.120 y 3.121). A muestra de ejemplo, para la consulta 157 de Cranfield, las dos soluciones más extremas obtenidas por el algoritmo SPEA-PG presentan un par precisión-exhaustividad de (1, 0.5) y (0.33, 1) respectivamente, mientras que las homólogas generadas por Smith y Smith se quedan por debajo con valores de (1, 0.35) y (0.056, 1). De manera similar en la consulta 43 de CACM, los valores de las consultas persistentes derivadas por el algoritmo multiobjetivo ((1, 0.650) y (0.317, 1)) son mayores que los de las soluciones del algoritmo mono-objetivo ((1, 0.550) y (0.016, 1)).

Las Tablas 3.120 y 3.121 muestran las medias y desviaciones típicas correspondientes al número de documentos relevantes recuperados y el número total de recuperados, para las consultas 157 de Cranfield y 43 de CACM, en las que observamos que: i) el número de documentos relevantes recuperados es mayor en las consultas persistentes derivadas por el enfoque multiobjetivo; ii) el número de documentos recuperados en total es mayor para las consultas persistentes generadas por el algoritmo mono-objetivo, y por lo tanto el ruido adiciona; iii) el SPEA-PG presentan desviaciones típicas más pequeñas, lo que presupone un comportamiento más homogéneo de este algoritmo.

Cranfield 157	***Doc. Relevantes Recuperados***		***Documentos Recuperados***	
	Media	***Desviación***	***Media***	***Desviación***
SPEA-PG	16.75	2.1	36.2	19.6
Smith & Smith	11.8	6.7	162.2	192.73

Tabla 124: Comparativa de SPEA-PG y Smith&Smith para la consulta 157 de Cranfield

CACM 43	***Doc. Relevantes Recuperados***		***Documentos Recuperados***	
	Media	***Desviación***	***Media***	***Desviación***
SPEA-PG	17	2.45	38.8	18.72
Smith & Smith	14.4	4.5	520.2	624.05

Tabla 125: Comparativa de SPEA-PG y Smith&Smith para la consulta 43 de CACM

Al evaluar las consultas persistentes en nuevas bases de datos, el algoritmo de Smith y Smith presenta un doble comportamiento, como ya se comento anteriormente, por un lado algunas consultas persistentes consiguen recuperar muchos documentos relevantes, pero junto con mucho ruido, lo que dificulta al usuario la tarea de obtener nueva información; y por otro lado, aparecen bastantes consultas persistentes que no son capaces de recuperar ningún documento relevantes lo que tampoco es deseable. A diferencia de Smith y Smith, nuestra propuesta multiobjetivo, no presenta ningún de los dos comportamientos, salvo en ocasiones contadas, mejorándose de esta manera los resultados.

Sirvan como ejemplo de esto las consultas 157 y 58 de Cranfield y CACM, respectivamente.

Consulta 157 de Cranfield. El algoritmo de Smith y Smith deriva dos consultas que no son capaces de recuperar nada, una que que recupera un único documento relevantes y otras dos que aunque recupera la mitad de los documentos relevantes lo hace junto con más de la mitad de los documentos irrelevantes (~473). Sin embargo, SPEA-PG obtiene cinco consultas que recuperan algo menos de la mitad de los documentos, pero con un máximo de 50 documentos relevantes en el peor caso.

Consulta 58 de CACM. Con el algoritmo mono-objetivo se generan tres consultas persistentes que no recuperan ningún documento relevante, mientras que otras dos lo recupera prácticamente todo, tanto documentos relevantes como irrelevante (14 de 1173 en el mejor caso). En frente de esto, cuatro de las cinco consultas derivadas por nuestra propuesta multiobjetivo recuperan alrededor de 5 documentos relevantes, sin que el máximo número de irrelevantes supere los 35. Por supuesto, nuestra propuesta no es perfecta, como demuestra la quinta consulta persistente derivada, que aunque recupera todos los documentos relevantes, esto también es extensible a los irrelevantes.

Consultas con menos de 15 documentos relevantes

En lo que se refiere a consultas con pocos documentos relevantes, el comportamiento de ambos algoritmos es bastante similar.

Al evaluar las consultas persistentes sobre los conjuntos de documentos utilizados para el aprendizaje, ambos algoritmos recuperan únicamente documentos relevantes, diferenciándose en la cantidad de éstos que recuperan. Concretamente, nuestro algoritmo SPEA-PG recupera casi todos los documentos relevantes, superando ligeramente al de Smith y Smith como demuestran los resultados correspondientes a la consulta 4 de CACM, para la cual todas las consultas persistentes generadas por nuestra propuesta recuperan todo los documentos relevantes presentes (precisión y exhaustividad iguales a 1), mientras que las cinco consultas persistentes derivadas por el algoritmo mono-objetivo, aunque recuperan únicamente documentos relevantes, no lo recuperan todos.

Si la evaluación se hace sobre conjuntos de documentos no conocidos en el proceso de aprendizaje, ambos algoritmos se comportan bastante mal, al no ser capaces en muchas ocasiones de recuperar ningún documento relevante. No obstante, en los casos en los que sí que se recuperan documentos relevantes, el ruido asociado es menor.

En resumen, nuestro algoritmo aparte de obtener mejores resultados en la evaluación sobre nuevos documentos, recupera menos documentos basura, lo que facilita el acceso de los usuarios a esta nueva información.

TIEMPO DE EJECUCIÓN

Por último, la utilización de un enfoque multiobjetivo no influye en el tiempo de ejecución de los algoritmos, como demuestran los resultados obtenidos. Los tiempos empleados son los mismos para ambos algoritmos, con lo que el objetivo de nuestra propuesta queda reforzado, al no sólo generar varias consultas persistentes con diferente balance precisión-exhaustividad, sino al realizarlo en el mismo tiempo en que el algoritmo mono-objetivo genera una única solución.

4.- UN MODELO DE SRI DIFUSO-LINGÜÍSTICO BASADO EN INFORMACIÓN LINGÜÍSTICA MULTIGRANULAR

Las distintas propuestas de SRI Lingüísticos revisadas en el Capítulo 2 permiten valorar, mediante variables lingüísticas [187], diferentes aspectos que encontramos en la actividad de un SRI como, por ejemplo, la relevancia de los documentos, la importancia de los términos de la consulta, etc. Sin embargo, presentan algunas limitaciones:

a) Normalmente, la mayoría de los lenguajes basados en consultas difusas [13][17][24][114] no permiten a los usuarios construir consultas en las que los elementos se ponderen de acuerdo a varias semánticas simultáneamente.

b) En muchos SRI Lingüísticos, las entradas y la salida se valoran sobre el mismo conjunto de etiquetas S [96][95], reduciendo las posibilidades de comunicación entre usuario y sistema.

En el presente capítulo, nos planteamos el uso del Modelado Lingüístico Difuso Multigranular (MLDM) con el fin de dotar de más flexibilidad y facilidad de representación a las consultas instantáneas en el proceso de RI. Para ello, proponemos un modelo de SRI Lingüístico Multigranular que use diferentes conjuntos de etiquetas con diferente granularidad y/o semántica para representar los distintos tipos de información que puedan aparecer en el proceso de recuperación.

En primer lugar, haremos una introducción planteando de forma más detenida el problema, mostrando como el MLDM se puede usar para el manejo de información cualitativa en la formulación de consultas como una forma de facilitar a los usuarios la expresión de sus necesidades de información. A continuación, describiremos brevemente algunos conceptos básicos, pero importantes, para el desarrollo de nuestra propuesta. Seguidamente, pasaremos a mostrar la estructura de nuestro SRI basado en información lingüística multigranular, núcleo de este capítulo. Finalmente, expondremos una serie de ejemplos del funcionamiento del SRI propuesto.

4.1.- Introducción

Las fases más cruciales en la RI son dos, la formulación de las necesidades de información y la selección de la información que las satisface. Ambas están impregnadas de subjetividad e imprecisión. De hecho, en muchos casos prácticos, no se puede proporcionar una descripción precisa de las necesidades de información, ya que no existe a priori una idea exacta de lo que se está buscando. Además, por otro lado, es importante conocer en que medida la información recuperada satisface una necesidad de información.

Los modelos de RI Booleanos no tienen en cuenta los problemas causados por la imprecisión y la subjetividad. De hecho, los documentos se representan como si fuesen conjuntos matemáticos de términos y las consultas como combinaciones Booleanas de términos, convirtiéndolas en las únicas unidades conceptuales para tratar con información. La principal limitación de estos sistemas es que no permiten indicar en las consultas la importancia de los términos en los documentos que se están buscando, ni ordenar los documentos recuperados de acuerdo a los juicios de relevancia [155][31].

Como vimos en el Capítulo 2, se han llevado a cabo varias extensiones sobre el mismo, con el fin de solucionar algunas de estas limitaciones. La Teoría de Conjuntos Difusos [186] se ha empleado como herramienta para tal propósito, especialmente por su habilidad para tratar con la imprecisión y la incertidumbre en el proceso de RI. Así, las extensiones se realizan en los siguientes aspectos principales:

- La representación de los documentos se convierte en conjuntos difusos definidos sobre el universo de términos, y los términos se transforman en conjuntos difusos definidos sobre el universo de los documentos tratados, introduciendo así un grado de relevancia (relación) entre un documento y un término.

- Se consideran pesos numéricos en las consultas con diferentes semánticas (podemos encontrar una revisión de éstas en [16] y en la Sección 2.3.6), permitiendo así al usuario cuantificar la "importancia subjetiva" de los requisitos de selección y por tanto tener más medios para expresar sus necesidades.

- Puesto que la relevancia de un documento a una consulta no siempre es total o inexistente, se introduce el grado de relevancia del documento, RSV.

- Se han propuesto distintas alternativas de operadores para modelar las conectivas

Booleanas.

Los pesos asociados a los términos índice se obtienen normalmente por medio de un procedimiento de indización automático en el que no existe interacción entre el usuario y el sistema [153]. Por lo tanto, parece razonable usar valores cuantitativos en la representación del contenido del documento. Sin embargo, en los otros niveles de representación (nivel de consulta y de evaluación), sí interviene la interacción usuario-sistema y, por tanto, se deberá tener en cuenta la posibilidad de usar valores cualitativos, típicos en la comunicación humana.

Los lenguajes de consulta basados en pesos numéricos fuerzan al usuario a cuantificar de forma cuantitativa conceptos (tales como "i*mportancia*"), ignorando el hecho de que muchos usuarios no se encuentran capacitados para proporcionar, de manera precisa, sus necesidades de información si las expresan de forma cuantitativa, pero sí, si lo hacen de forma cualitativa. En realidad, parece más natural caracterizar el contenido de los documentos mediante la asociación explícita de un descriptor lingüístico a un término en una consulta, tales como "*importante*" o "*muy importante*", en vez de un valor numérico. De manera similar, los SRI son más amigables si los niveles de relevancia estimados para los documentos son suministrados mediante un valor lingüístico (por ejemplo, se podrían usar términos lingüísticos, tales como "*relevante*", "*muy relevante*"), en vez de a través de una puntuación.

Por lo tanto, el uso de variables lingüísticas [187] para representar la información de entrada y salida en el proceso de recuperación de los SRI mejora la interacción entre el usuario y el sistema (véanse los enfoques de SRI Lingüísticos repasados en la Sección 2.4).

Usando variables lingüísticas, cada peso es un valor de la variable lingüística "*Importancia*", mientras que el RSV asociado a cada documento tras la evaluación de una consulta es un valor de la variable lingüística "*Relevancia*". En muchos SRI Lingüísticos, estas dos variables se valoran sobre el mismo conjunto de etiquetas S [96][95], es decir, se usa el mismo conjunto de etiquetas para expresar la entrada y la salida del SRI. Sin embargo, esta forma de expresar la entrada y la salida no es conveniente ya que, por un lado, se reducen las posibilidades de comunicación entre usuario y sistema; y, por otro, puesto que ambas variables lingüísticas representan diferentes conceptos, parece lógico usar diferentes conjuntos de etiquetas para modelarlas; utilizando para ello, por ejemplo, un modelado lingüístico multigranular [90]. En definitiva, nos referimos a usar conjuntos de etiquetas con diferente granularidad y/o semántica para representar los diferentes tipos de información que

pueden aparecer en el proceso de recuperación de información.

En respuesta a lo anterior, el objetivo de este capítulo se centra en presentar un SRI que usa información lingüística multigranular para expresar la entrada y la salida del sistema, empleando las variables lingüísticas usadas habitualmente (*Importancia* y *Relevancia*), pero valoradas sobre conjuntos de etiquetas con diferente granularidad o semántica; así como un método para procesar este tipo de información.

4.2.- Conceptos Básicos

En esta sección vamos a repasar algunas herramientas para el procesamiento de información lingüística de las que haremos uso en el desarrollo de nuestro modelo de SRI Lingüístico Multigranular.

4.2.1.- Enfoque Lingüístico Ordinal

Como comentamos en el Capítulo 2, un enfoque lingüístico-difuso es una técnica apropiada para tratar con los aspectos cualitativos de los problemas. Este enfoque modela los valores lingüísticos por medios de variables lingüísticas [187]. Su aplicación introduce un entorno de trabajo más flexible para representar la información de forma más directa y adecuada cuando no es posible expresarla de manera exacta.

El enfoque lingüístico ordinal es un tipo especial del enfoque lingüístico que facilita el MLDM [19][88][95]. Este tipo de enfoque se define considerando un conjunto de etiquetas finito y totalmente ordenado, $S = \{s_i, i = 0, ..., T\}$, en el sentido usual ($s_i \geq s_j$ si $i \geq j$) y con cardinalidad impar (7 o 9). El término central representa una incertidumbre de "aproximadamente 0.5" y el resto de términos están situados simétricamente a ambos lados de él [12].

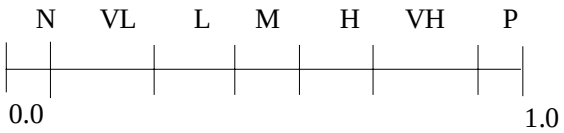


Figura 4.1: Conjunto ordenado de siete términos distribuidos simétricamente

La semántica de este conjunto de etiquetas se establece a partir de la propia estructura de

orden del conjunto, considerando que cada etiqueta del par (s_i, s_{T-i}) es igualmente informativa. En algunos enfoques [88][96][95], la semántica se completa asociando a las etiquetas números difusos definidos en el intervalo [0,1]. Estos números se describen por medio de funciones de pertenencia trapezoidales representadas por una cuádrupla $(a_i,b_i,\alpha_i,\beta_i)$ (los dos primeros parámetros indican el intervalo en el que el valor de pertenencia es 1.0, mientras que el tercer y cuarto parámetro indican el ancho izquierdo o derecho, respectivamente, del soporte).

En la Figura 4.2 se puede ver, a modo de ejemplo, un conjunto de nueve etiquetas junto con su semántica.

```
N  = (0.00 0.00 0.00 0.00)      H  = (0.63 0.80 0.05 0.06)
EL = (0.01 0.02 0.02 0.05)      VH = (0.78 0.92 0.06 0.05)
VL = (0.10 0.18 0.06 0.05)      EH = (0.98 0.99 0.05 0.01)
L  = (0.22 0.36 0.05 0.06)      T = (1.00 1.00 0.00 0.00)
M  = (0.41 0.58 0.09 0.07)
```

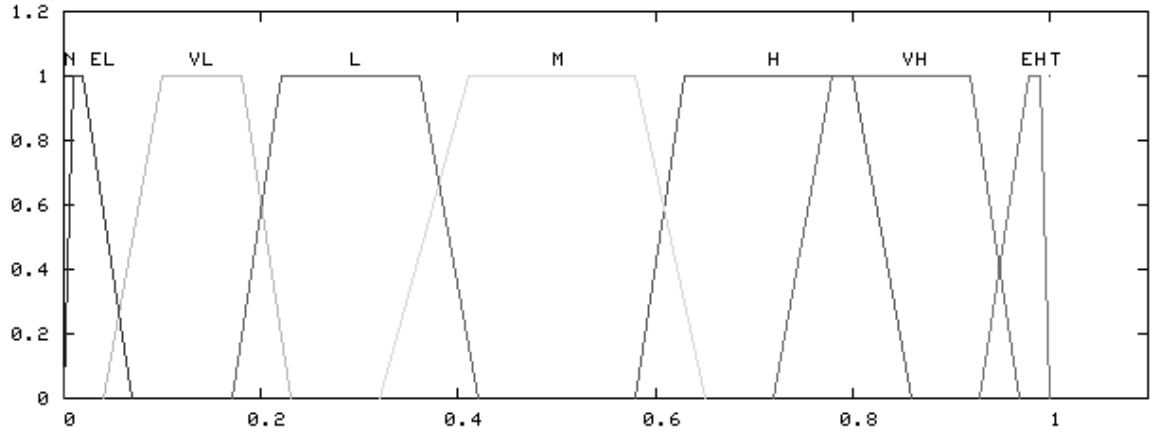


Figura 4.2: Conjunto de nueve etiquetas con su semántica

Además, hace falta definir un modelo computacional para combinar la información lingüística. El modelo se establece definiendo una serie de operadores:

- ☞ Negación: **Neg**$(s_i) = s_j$, $j = T-i$
- ☞ Comparación
  - Maximización: **MAX**$(s_i,s_j) = s_i$ si $s_i >= s_j$
  - Minimización: **MIN** $(s_i, s_j) = s_i$ si $s_i <= s_j$
- ☞ Operadores de agregación lingüística [91][92].

4.2.2.- Información Lingüística Multigranular

Un parámetro importante que es necesario determinar es la "*granularidad de la incertidumbre*", por ejemplo, la cardinalidad del conjunto de términos lingüísticos usado para expresar la información lingüística. Esta cardinalidad debe ser lo suficientemente pequeña para no imponer una precisión inútil a los usuarios, y lo suficientemente rica para permitir una discriminación de las valoraciones en un número limitado de grados. Valores típicos de cardinalidad, usados en los modelos lingüísticos, como dijimos antes, son valores impares, tales como 7 o 9 [134], en ocasiones excepcionales, dependiendo del contexto, con un límite superior de 11 o no más de 13, donde el término medio representa una valoración de "*aproximadamente 0.5*", y el resto de los términos están situados simétricamente alrededor de él [12].

Como comentamos al comienzo del capítulo, nuestro SRI trabajará con información lingüística multigranular, es decir, manejará simultáneamente varios conjuntos de etiquetas con diferente granularidad o semántica. Existen dos razones que hacen necesario el uso de este tipo de información:

- ☞ Si en el proceso de RI aparecen diferentes conceptos, parece lógico que sean valorados mediante conjuntos de etiquetas diferentes.
- ☞ Por otro lado, en función del grado de incertidumbre que un usuario presente al cualificar un determinado fenómeno, el conjunto de etiquetas elegido para representar su conocimiento tendrá más o menos términos. Por lo tanto, si son varios los usuarios que cualifican un fenómeno, serán necesarios varios conjuntos de etiquetas, cada uno con una granularidad diferente en función del grado de incertidumbre que tenga el

usuario sobre el fenómeno.

Para el manejo de información lingüística multigranular existen diferentes propuestas [90][173], que podemos tomar como base para modelar los procesos de RI de nuestro SRI Lingüístico Multigranular.

4.3.- Un Modelo de SRI basado en Información Lingüística Multigranular

En esta sección presentamos un modelo de SRI Lingüístico que acepta consultas Booleanas ponderadas con pesos lingüísticos y proporciona como salida documentos valorados por su RSV, expresándolos a través de información lingüística multigranular. Sus principales características son:

- Las consultas Booleanas ponderadas y los valores RSV lingüísticos asociados con los documentos se valoran sobre conjuntos de etiquetas con diferente granularidad y/o semántica.
- Los términos presentes en la consulta se pueden ponderar simultáneamente con tres pesos lingüísticos diferentes, asociados con tres semánticas diferentes: semántica de umbral simétrico, semántica de importancia relativa y semántica cuantitativa.
- Los operadores Booleanos se modelan de manera flexible utilizando el operador OWA propuesto por Yager en [183].
- Los documentos recuperados se clasifican en clases de relevancia, identificadas por valores lingüísticos ordinales valorados sobre un conjunto de etiquetas diferente de los asociados con los pesos de las consultas.

4.3.1.- Base de Datos

Consideramos un conjunto de documentos $D = \{d_1, ..., d_m\}$ representados mediante un conjunto de términos índice $T = \{t_1,...,t_l\}$, que describen el contenido de los documentos. Se define una función de indización numérica $F: D \times T \rightarrow [0,1]$, que asigna un peso numérico entre 0 y 1 a cada par documento-término (d_j,t_i).

Así, $F(d_j,t_i)$ es un peso numérico que indica cómo de significativo es el término t_i en la descripción del documento d_j. En los casos extremos, si $F(d_j,t_i)=0$, el documento d_j no está representado en modo alguno por el término t_i, mientras que si $F(d_j,t_i)=1$, el documento está perfectamente representado por los conceptos indicados por el término en cuestión. Usando valores numéricos en el intervalo [0,1], la función F puede ponderar los términos índice de acuerdo a su importancia en la descripción del contenido de un documento, con el fin de mejorar la recuperación de documentos.

4.3.2.- Definición de Consultas Lingüísticas Multigranulares

Las consultas se expresan como combinaciones de términos índice ponderados, conectados por los operadores lógicos Y ($\wedge$), O ($\vee$), y NO ($\neg$). Cada término puede ser ponderado simultáneamente con varios pesos [96][95]. De manera particular, proponemos que cada término se pueda ponderar simultáneamente o no con tres pesos asociados con diferentes semánticas. De esta forma, el sistema proporciona mayor apoyo a los usuarios para que especifiquen sus necesidades.

4.3.2.1.- Semánticas de los pesos

Asignando pesos a las consultas, los usuarios especifican restricciones sobre los documentos que el SRI debe satisfacer a la hora de la recuperación. Se pueden imponer dos tipos de restricciones sobre los documentos que se quieren recuperar [95]:

- **Restricciones cualitativas**: este tipo de restricciones se imponen cuando se quieren expresar criterios que afectan a la cualidad de la representación de los documentos que deseamos que se recuperen. Por ejemplo, restricciones que deberán satisfacer los términos índice que aparecen en la representación de los documentos recuperados.

- **Restricciones cuantitativas**: hablamos de restricciones cuantitativas cuando los pesos de la consulta expresan criterios que afectan a la cantidad de documentos que se recuperarán. Por ejemplo, restricciones para ser satisfechas por el número de documentos que serán recuperados.

Normalmente, la mayoría de los lenguajes basados en consultas difusas [13][17][24][114] no permiten a los usuarios construir consultas en las que los elementos se puedan pondar de

acuerdo a varias semánticas simultáneamente.

Sin embargo, en algunas situaciones, un usuario podría querer ver *unos pocos* documentos (restricción cuantitativa) en los que el concepto expresado por el término índice t_i tiene *mucha importancia* (restricción cualitativa). Con el fin de poder contemplar estas situaciones, en [96][95], se proponen diferentes lenguajes de consulta que permiten al usuario construir consultas en las que los elementos se pueden ponderar de acuerdo a varias semánticas simultáneamente.

En esta memoria, al igual que en [95], proponemos utilizar tres semánticas, dos cualitativas y una cuantitativa, para representar el significado de los pesos asociados a un término de una consulta. Las tres semánticas que hemos elegido son consistentes y complementarias unas con las otras; consistentes, porque las necesidades de información expresadas por cualquiera de las semánticas no se contradicen con las expresadas con las otras semánticas; y complementarias, porque el usuario tiene herramientas suficientes para expresar gran parte de sus necesidades de información utilizando las semánticas.

SEMÁNTICA DE UMBRAL SIMÉTRICO [95]

Asociando pesos (*w*) como umbrales a los términos de una consulta. el usuario requiere que los documentos se evalúen chequeando sus grados de importancia *F(d,t)* contra el peso umbral *w*. Dicho de otra forma, los pesos indicarán un umbral que tendrá que ser superado para que el documento se considere relevante para una consulta.

Normalmente, este tipo de semántica premia al documento cuyo grado de pertenencia para el término t sea mayor que el grado de pertenencia del término en la consulta (peso umbral) pero permitiendo algún valor de coincidencia parcial cuando el grado de pertenencia del documento es menor que el umbral .

Una semántica de umbral simétrico es un tipo especial de semántica de umbral que admite que un usuario pueda indicar la presencia o ausencia de los pesos en la formulación de las consultas ponderadas. Es simétrica respecto a la etiqueta que representa el valor de umbral medio, presentando el comportamiento normal para los valores situado a la derecha del umbral medio (pesos de presencia) y el comportamiento contrario para los valores que se encuentran a la izquierda (pesos de ausencia o pesos de presencia con valores bajos).

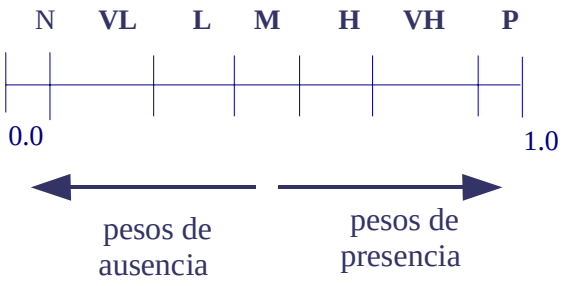


Figura 4.3: Semántica de umbral simétrico

SEMÁNTICA DE IMPORTANCIA RELATIVA [13][180]

Define los pesos de las consultas como medida de la importancia relativa de cada término con respecto a los demás en la consulta. El usuario asignará pesos de importancia relativa a cada término dentro de la consulta para establecer una dominancia de unos sobre otros a la hora de efectuar la recuperación. En el conjunto final de documentos recuperados, los términos más importantes deben aportar más documentos, o los documentos en los que aparezcan los términos más importantes deben pesar más. En la práctica, esto significa que el usuario quiere que el RSV de un documento este dominado por los términos de mayor peso.

SEMÁNTICA CUANTITATIVA

Define los pesos de las consultas como una medida de la cantidad de documentos relevantes para cada término de una consulta que el usuario quiere considerar en la computación del conjunto final de documentos que se recuperarán. Asociando pesos de este tipo a los términos de una consulta, el usuario quiere que se le muestre un conjunto de documentos, al que habrán contribuido en mayor medida los términos con mayor peso (aportando un mayor número de documentos pertinentes). En la práctica, el uso de esta semántica cuantitativa presenta dos consecuencias beneficiosas en comparación con los sistemas clásicos existentes:

- ☞ Los RSV se calculan usando un número reducido de documentos. Estos documentos se han determinando teniendo en cuenta los pesos cuantitativos asociados a los términos de una consulta. Con este peso, un usuario puede elegir aquellos documentos que mejor

se ajusten a los conceptos representados por el término, la mayoría de los documentos que se ajusten al concepto, algunos de los documentos, etc. Por lo tanto, podemos refinar los documentos de salida del SRI. En nuestro caso, como veremos en las secciones siguientes, esta semántica nos ayudará a disminuir el número de clases de relevancia en las que los documentos se pueden clasificar, como salida del SRI.

- ☞ Un control flexible del número total de documentos a recuperar que se realiza término a término.

4.3.2.2.- Reglas para la formulación de consultas lingüísticas ponderadas multigranulares

Formalmente, en [17] se define una consulta Booleana cuyos términos tienen asociado un peso lingüístico y su correspondiente semántica, como cualquier expresión Booleana legítima cuyos componentes atómicos son parejas $<t_i,c_i>$ que pertenecen al conjunto, T *x H(Importancia)*; t_i es un elemento del conjunto T de términos, y c_i es un valor de la variable lingüística, *Importancia.*

En nuestro caso, cada término podrá ser ponderado de acuerdo a tres semánticas diferentes, incluso simultáneamente. Al igual que en [95], usamos la variable *Importancia* para modelar cada una de las semánticas, pero con interpretaciones diferentes. Por ejemplo, un término t_i que tiene asociado un valor *"Alto"* en el peso relacionado con la semántica de umbral, significa que el usuario espera que se recuperen aquellos documentos en los que el término t_i tenga al menos una importancia alta. Sin embargo, el mismo término t_i, con un valor para el peso cuantitativo de "*Alto*", indica que el usuario quiere un conjunto de documentos en el que el término t_i contribuya con un alto número de documentos. De igual forma, si el valor "*Alto*" se asocia al peso de importancia relativa, el usuario reseña que el significado del término t_i debe tener una alta importancia en el computo del conjunto de documentos recuperados.

El problema del modelo propuesto en [95] es que diferentes pesos lingüísticos asociados con un mismo término son valorados en el mismo conjunto de etiquetas, S. Para resolverlo, proponemos representar los pesos lingüísticos usando información lingüística multigranular, p.e., considerando conjuntos de etiquetas con diferentes cardinalidades y/o semánticas para valorar los pesos asociados con las tres semánticas, S^1, S^2 y S^3, respectivamente.

De acuerdo con lo anterior, para nosotros una consulta es cualquier expresión Booleana legítima cuyos componentes atómicos (átomos) son cuádruplas $<t_i,ci^1,ci^2,ci^3>$ que pertenecen al conjunto $T\ x\ S^1\ x\ S^2\ x\ S^3$; donde $t_i \in T$, y $c_i^1 \in S^1$, $c_i^2 \in S^2$ y $c_i^3 \in S^3$ son valores de la variable lingüística "Importancia", modelando las semánticas de umbral, cuantitativa y de importancia relativa, respectivamente.

El conjunto Q de consultas legítimas se define por medio las siguientes reglas sintácticas:

1. $\forall\ q=<t,c^1,c^2,c^3>, \in TxS^1xS^2xS^3 \rightarrow q \in Q.$
2. $\forall\ p,q \in Q \rightarrow q \wedge p \in Q.$
3. $\forall\ p,q \in Q \rightarrow q \vee p \in Q.$
4. $\forall\ q \in Q \rightarrow \neg(q) \in Q.$
5. Todas las consultas lingüísticas multigranulares legítimas son aquellas obtenidas por las reglas 1-4.

De igual modo que [25], suponemos que un término puede aparecer varias veces en la misma consulta, y por tanto, el subsistema de consulta debe aceptar la posibilidad de que un mismo término tenga asociados diferentes vectores de tres pesos.

4.3.3.- Evaluación de las Consultas Lingüísticas Multigranulares.

El objetivo del subsistema de evaluación consiste en evaluar el grado en el que las representaciones de los documentos satisfacen los requisitos expresados en una consulta ponderada, de acuerdo a tres semánticas diferentes. Normalmente, el método de evaluación para consultas Booleanas actúa por medio de un proceso constructivo que recorre el árbol que representa una consulta de abajo a arriba (bottom-up), es decir, se evalúan primero los átomos, luego las combinaciones de átomos y así sucesivamente hasta que la consulta completa esté evaluada.

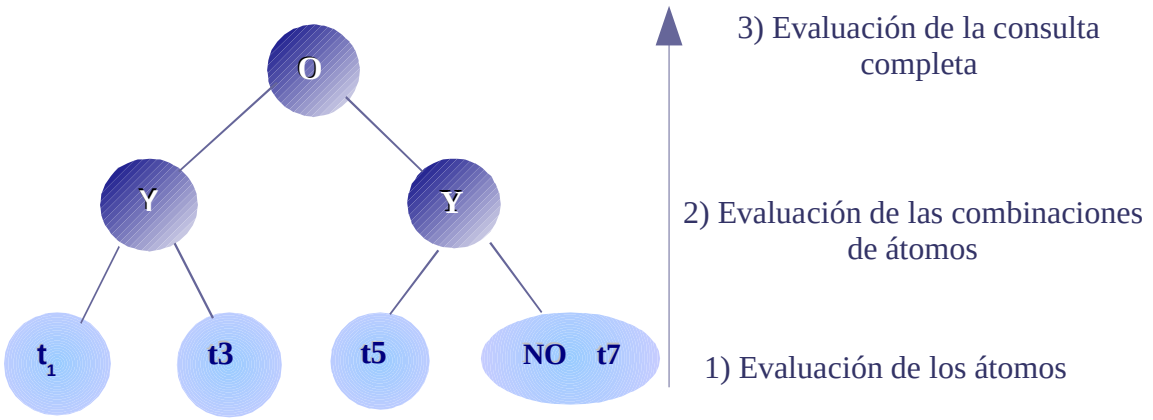


Figura 4.4: Método bottom-up para evaluar consultas Booleanas

De forma similar, en esta sección proponemos un método de evaluación bottom-up para procesar consultas lingüística ponderadas multigranulares. Este método evalúa el grado en el que las representaciones de los documentos satisfacen los requisitos expresados en una consulta, de acuerdo a las tres semánticas asociadas con los pesos y manejando satisfactoriamente la información lingüística multigranular. Además, dado que el concepto de relevancia es diferente del concepto de importancia, usaremos un conjunto de etiquetas S' diferentes de los usados para expresar las consultas (S^1, S^2 y S^3), para proveer el valor de relevancia de los documentos.

Para manejar los pesos lingüísticos multigranulares asociados a los términos de las consultas, hemos desarrollado un procedimiento basado en una herramienta de manejo de información lingüística multigranular definida en [90]. Este procedimiento actúa unificando la información lingüística multigranular antes de procesar las consultas. Para ello, tenemos que elegir un conjunto de etiquetas como base para la representación uniforme de la información, BLTS, (en inglés, *basic linguistic term set*), y transformar (haciendo uso de funciones de transformación) toda la información multigranular en el conjunto unificado de etiquetas BLTS. En nuestro caso, la elección del conjunto BLTS es fácil de determinar: debe coincidir con el conjunto de etiquetas usado para expresar la salida del SRI (grados de relevancia de los documentos), p.e., BLTS = S'.

El método que proponemos evalúa una consulta en los cinco pasos siguientes:

1. Preprocesamiento de la consulta.
2. Evaluación de los átomos con respecto a la semántica de umbral.
3. Evaluación de los átomos con respecto a la semántica cuantitativa.
4. Evaluación de las subexpresiones y modelado de la semántica de importancia.
5. Evaluación de la consulta completa.

4.3.3.1.- Preprocesamiento de la consulta

La consulta que representa las necesidades de información del usuario se preprocesa para ponerla en forma normal conjuntiva (CNF) o disyuntiva (DNF), de forma que cada subexpresión Booleana tenga más de dos átomos. Las consultas con un único término se mantienen en su forma original. De acuerdo con lo anterior, una consulta q_w con I subexpresiones y N átomos, puede tener cualquiera de las formas mostradas en las Figuras 4.5 y 4.6, p.e., como un árbol O/Y-Ponderado o como un árbol Y/O-Ponderado.

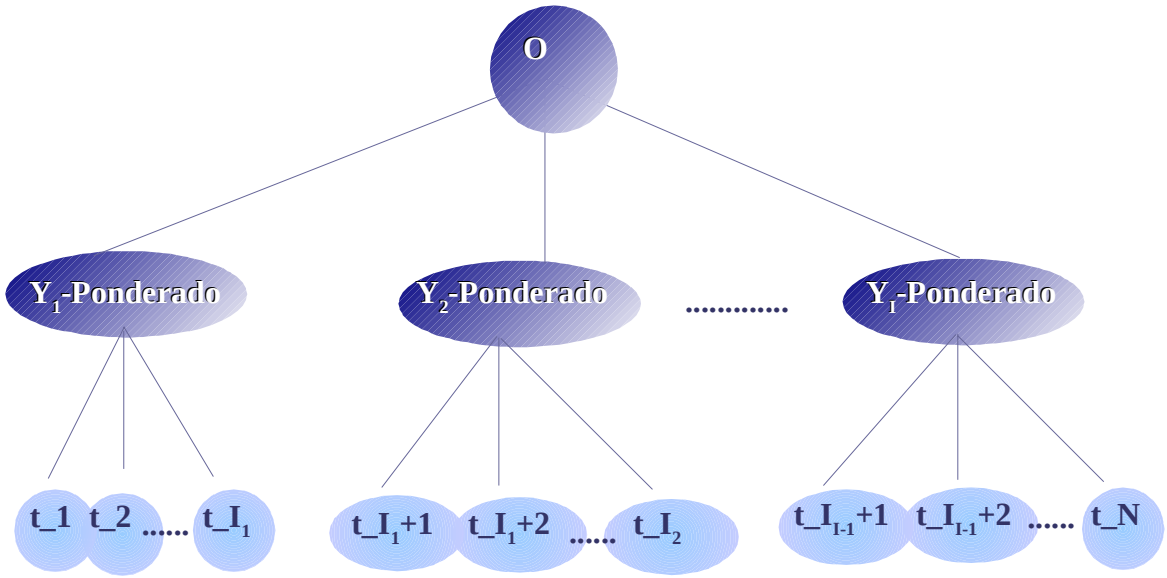


Figura 4.5: Consulta en forma normal disyuntiva (DNF)

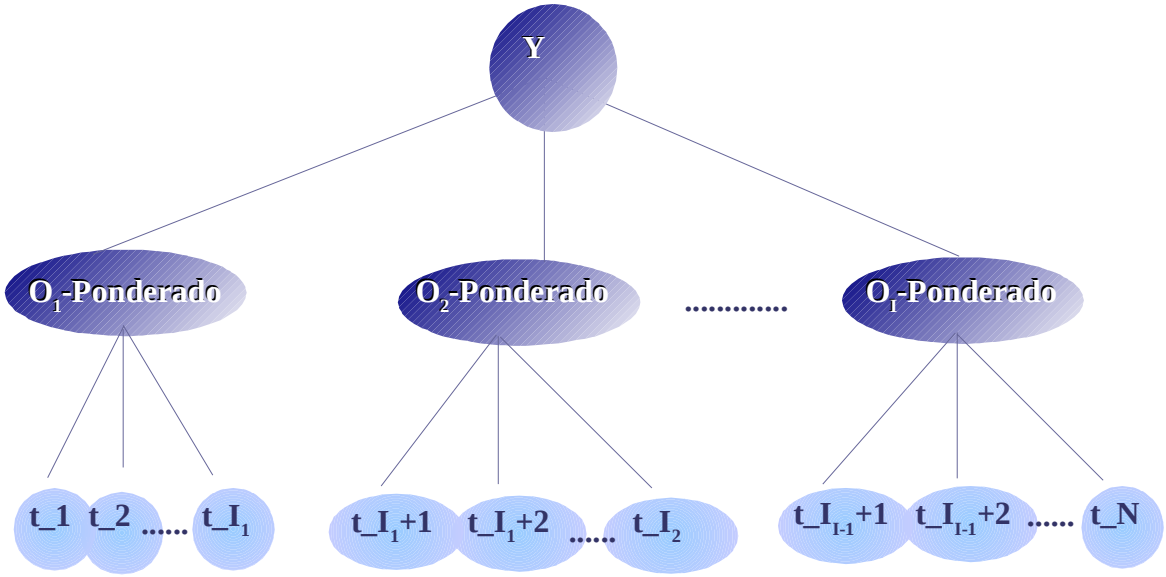


Figura 4.6: Consulta en forma normal conjuntiva (CNF)

4.3.3.2.- Evaluación de los átomos respecto a la semántica de umbral simétrico.

Cuando el usuario utiliza la semántica de umbral simétrico, está buscando aquellos documentos con una presencia mínimamente aceptable de un término en su representación o, por el contrario, documentos con una presencia máxima aceptable de un término en su representación [96][95]. Así, por ejemplo, cuando un usuario pregunta por documentos en los que el concepto representado por un término t_i aparezca con el valor "*Alta Importanc*ia", no espera que se rechace un documento con un grado de importancia $F(d,t_i)$ mayor que "*Alta*". Por el contrario, si el usuario solicita documentos en los que el concepto representado por el término t_i aparezca con el valor "*Baja Importancia*", no querrá que se rechace un documento cuyo grado de importancia $F(d,t_i)$ sea menor que "*Baja*".

Por lo tanto, ante un átomo $<t_i,c_i^1,c_i^2,c_i^3>$, los pesos que indican la presencia de un término en un documento $c_i^1 \geq s^1_{T/2}$ (p.e., *Alto* o *Muy Alto*) serán tratados de forma diferente a los pesos que indican la ausencia de un término en un documento $c_i^1 < s^1_{T/2}$ (p.e. *Bajo*, *Muy Bajo*). Si $c_i^1 \geq s^1_{T/2}$, la necesidad representada por el átomo $<t_i,c_i^1,c_i^2,c_i^3>$ es sinónima de la representada por el átomo $<t_i$, al menos $c_i^1,c_i^2,c_i^3>$, lo que significa que los documentos que se desean recuperar son aquellos con un grado de importancia $F(d,t_i)$ tan alto como sea posible. En cambio, y si $c_i^1 < s^1_{T/2}$, el átomo inicial es sinónimo de $<t_i$, como máximo $c_i^1,c_i^2,c_i^3>$,

indicando que los documentos deseados son aquellos con un grado de importancia tan bajo como sea posible.

Esta interpretación se define por medio de una función de similitud parametrizada y lingüística g^1: D x T x S^1 --> S^1 [96]. Dado un átomo $<t_i, c_i^1, c_i^2, c_i^3>$ y un documento $d_j \in D$, g^1 calcula el RSV lingüístico de d_j, $RSV_j^{i,1}$, determinando cómo de bien satisface el peso del término índice $F(d_j,t_i)$ la necesidad de información expresada por el peso lingüístico c_i^1 según la siguiente expresión:

$$RSV_j^{i,1} = g^1(d_j, t_i, c_i^1) = \begin{cases} s^1_{min\{\alpha+\beta, T\}} & \text{si} \quad s^1_{T/2} \le s^1_b \le s^1_a \\ s^1_{max\{0,\ \alpha-\beta\}} & \text{si} \quad s^1_{T/2} \le s^1_b \ \text{ y } \ s^1_a < s^1_b \\ \text{Neg}(s^1_{max\{0,\ \alpha-\beta\}}) & \text{si} \quad s^1_a \le s^1_b < s^1_{T/2} \\ \text{Neg}(s^1_{min\{\alpha+\beta, T\}}) & \text{si} \quad s^1_b < s^1_{T/2} \ \text{ y } \ s^1_b < s^1_a \end{cases}$$

donde

i) $s_b^1 = ci^1$.

ii) s_a^1 es el peso lingüístico del término índice obtenido como s_a^1 = Etiqueta($F(d_j,t_i)$), siendo Etiqueta: [0,1] $\rightarrow S^1$ una función que asigna un etiqueta del conjunto S^1 a un valor numérico r del intervalo [0,1] de acuerdo a la siguiente expresión:

$$Etiqueta(r) = \text{Sup}_q\{s_q^1 \in S^1 : \mu_{s_q^1}(r) = \text{Sup}_v\{\mu_{s_v^1}(r)\}\}$$

iii) B es un valor que penaliza o recompensa el $RSV_j^{i,1}$, valor con el fin de satisfacer o no el átomo $<t_i, c_i^1, c_i^2, c_i^3>$, y puede definirse de forma independiente, por ejemplo como $B = 1$, o dependiendo de la cercanía entre Etiqueta($F(d_j,t_i)$) y c_i^1, por ejemplo como $B = round(2(|b-a|)/T)$.

4.3.3.3.- Evaluación de los átomos con respecto a la semántica cuantitativa

En este paso se vuelven a evaluar los documentos para ver como satisfacen las necesidades de información representadas por los términos individuales de la consulta, pero considerando las restricciones impuestas por la semántica cuantitativa.

El uso de este tipo de pesos se interpreta de la siguiente manera: cuando un usuario establece cierto número de documentos para un término, expresándolo como un peso lingüístico modelado con la semántica cuantitativa, el conjunto de documentos que se recupere debe tener un número mínimo de documentos que satisfagan la compatibilidad o función de pertenencia asociada con el significado de la etiqueta usada como peso. Además, estos documentos deben ser aquellos que mejor satisfagan las restricciones impuestas por la semántica de umbral al término.

Es importante señalar que, en un contexto de RI difuso, el uso de una semántica de umbral implica el establecimiento de restricciones sobre la función de pertenencia que caracteriza el conjunto difuso de documentos recuperados por un término índice, mientras que el uso de una semántica cuantitativa implica el establecimiento de restricciones sobre el soporte de dicho conjunto difuso.

Por lo tanto, dado un átomo $<t_i, c_i^1, c_i^2, c_i^3>$ y asumiendo que $RSV_j^{i,1} \in S^1$ representa la evaluación para d_j, según la semántica de umbral, modelamos la interpretación de la semántica cuantitativa por medio de un función lingüística de similitud, g^2, la cual se define entre el $RSV_j^{i,1}$ y el peso lingüístico $c_i^2 \in S^2$. Entonces, la evaluación del átomo $<t_i, c_i^1, c_i^2, c_i^3>$ con respecto a la semántica cuantitativa asociada con c_i^2 para un documento d_j, llamada $RSV_j^{i,1,2} \in S^1$, se obtiene por medio de la función lingüística de similitud g^2:D x S^1 x $S^2 \rightarrow S^1$ de la siguiente manera:

$$RSV_j^{i,1,2} = g^2(RSV_j^{i,1}, c_i^2, d_j) = \begin{cases} s_0^1 & \text{si} \quad d_j \notin B^S \\ RSV_j^{i,1} & \text{si} \quad d_j \in B^S \end{cases}$$

donde B^S es el conjunto de documentos tales que $B^S \in$ Soporte(M), siendo M un subconjunto difuso de documento obtenido $M = \{(d_1, \text{RSV}_1^{i,1}), ..., (d_m, \text{RSV}_m^{i,1})\}$ al aplicar el algoritmo de la Figura 4.7.

(1) $K = \#\mathrm{Supp}(M)$
(2) REPETIR
$M^K = \{s_q \in S : \mu_{s_q}(K/m) = \mathrm{Sup}_v\{\mu_{s_v}(K/m)\}\}.$
$s^K = \mathrm{Sup}_q\{s_q \in M^K\}.$
$K = K - 1.$
(3) HASTA $((c_i^2 \in M^{K+1})$ O $(c_i^2 \geq s^{K+1}))$.
(4) $B^S = \{d_{\sigma(1)}, \ldots, d_{\sigma(K+1)}\}$, tal que $RSV^{i,1}_{\sigma(h)} \leq RSV^{i,1}_{\sigma(l)}$, $\forall\, l \leq h$

Figura 4.7: Algoritmo para calcular el conjunto B^S

De acuerdo a la función g^2, la aplicación de la semántica cuantitativa consiste en reducir el número de documentos que serán considerados por el subsistema de evaluación, para el término t_i, en las siguientes etapas. Así, asignando pesos cuantitativos cercanos a s_0, un usuario muestra sus preferencias considerando los documentos más representativos en M, mientras que si asigna pesos cercanos a s_T no hace distinción alguna entre los documentos existentes en M.

Observación: Aunque en esta etapa trabajamos con diferentes dominios lingüísticos, si embargo no hay necesidad de agregación de información lingüística multigranular. En consecuencia, no es necesario utilizar el conjunto de etiquetas BLTS.

4.3.3.4.- Evaluación de las subexpresiones y modelado de la semántica de importancia relativa

La aplicación de la semántica de importancia relativa en un término simple no tiene significado. Por tanto, en esta etapa evaluamos la relevancia de los documentos respecto a las subexpresiones compuestas de más de dos átomos.

Dada una subexpresión q_v con $I \geq 2$ átomos, cada documento tiene asociado un RSV $RSV_j^{i,1,2} \in S^1$ parcial, respecto a cada átomo $<t_i, c_i^1, c_i^2, c_i^3>$ de la subexpresión q_v. La evaluación de un documento d_j respecto a la subexpresión completa q_v implica la agregación de los grados de relevancia parciales $\{RSV_j^{i,1,2} \in S^1,\ i = 1, ..., I\}$, ponderados por medio de los respectivos grados asociados a la semántica de importancia relativa $\{c_i^3 \in S^3,\ i = 1, ..., I\}$. Por lo tanto, como $S^1 \neq S^3$, es necesario desarrollar un procedimiento de agregación de

información lingüística multigranular. Como comentamos anteriormente, para poder hacerlo, lo primero es elegir un conjunto de etiquetas BLTS sobre el que se unificará la información lingüística. En este caso, elegimos como conjunto BLTS el conjunto S', conjunto usado normalmente para evaluar el grado de relevancia de los documentos (RSV), de forma que cada valor lingüístico se transforma a un valor del conjunto S'. Este cambio se realiza por medio de la siguiente función de transformación:

Definición [90]: Sean $A = \{l_0, ..., l_p\}$ y $S' = \{s'_0, ..., s'_m\}$ dos conjuntos de etiquetas, tal que $m \geq p$. Una función de transformación multigranular, $\tau_{AS'}$ se define como $\tau_{AS'} \rightarrow F(S')$:

$$\tau_{AS'}(l_i) = \{(s'_k, \alpha_k^i) / k \in \{0, ..., m\}\}, \quad \forall l_i \in A,$$
$$\alpha_k^i = max_y min\{\mu_{l_i}, \mu_{c'_k}(y)\},$$

donde F(S') es el conjunto de conjuntos difuso definidos en S', y μ_{li} (y) y $\mu_{s'k}(y)$ son las funciones de pertenencia que describen los conjuntos difusos asociados a los términos l_i y s'_k, respectivamente.

En consecuencia, el resultado de $\tau_{AS'}$ para cualquier valor de A es un conjunto difuso definido en el conjunto BLTS, S'. Usando las funciones de transformación $\tau_{S1S'}$ y $\tau_{S3S'}$, transformamos los valores lingüísticos $RSV_j^{i,1,2} \in S^1$, $i = 1, ..., I\}$ y $\{c_i^3 \in S^3, i = 1, ..., I\}$ en valores lingüísticos de S', respectivamente. Por lo tanto, los valores RSV^j_i y c^3_i se representan como conjuntos difusos definidos sobre S' y caracterizados por las siguientes expresiones:

$$(1)\ \tau_{S^1S'}(RSV_j^{i,1,2}) = [(s'_0, \alpha_0^{i,j}), ..., (s'_m, \alpha_m^{i,j})], \text{ y}$$
$$(2)\ \tau_{S^2S'}(c_i^3) = [(s'_0, \alpha_0^i), ..., (s'_m, \alpha_m^i)], \text{ respectivamente.}$$

En cada subexpresión q_v encontramos que los átomos pueden estar combinados usando los operadores Booleanos Y o O, dependiendo de la forma normal en la que esté la consulta. Las restricciones impuestas por la semántica de importancia relativa se aplicarán utilizando los operadores de agregación usados para modelar dichos operadores Booleanos. Estos operadores de agregación nos garantizan que, cuanto más importante sea un término en una consulta, más influirá en el cálculo del RSV. Para conseguir esto, dichos operadores de agregación deben llevar a cabo dos actividades [88]:

1. La transformación de la información lingüística ponderada de acuerdo a los grados de importancia por medio de la función de transformación *h*.

2. La agregación de la información transformada por medio de un operador de agregación de información lingüística no ponderada, *f*.

La función de transformación (h) depende del tipo de agregación (f) que se lleve a cabo sobre la información ponderada. En [182], Yager discutió el efecto de los grados de importancia sobre los tipo de agregación "MAX" (usado para modelar el operador O) y "MIN" (usado para modelar el operador Y) y sugirió una clase de funciones de transformación (h) en cada tipo de agregación. Para la agregación MIN, sugirió usar una familia de t-conormas que actuasen sobre la información ponderada y sobre la negación del grado de importancia, mientras que para la agregación MAX, propuso una familia de t-normas que actuasen sobre la información y el grado.

EL OPERADOR OWA

El OWA [183] es un operador de agregación de información (en inglés, *Ordered Weighted Averaging)* que tiene en cuenta el orden de las valoraciones que van a ser agregadas.

Definición:

Sea $A = \{a_1,...,a_m\}$, $a_k \in [0,1]$ un conjunto de valoraciones que se quieren agregar, el operador OWA, ϕ, se define como:

$$\phi(a_1,...,a_m) = W \cdot B^T$$

donde W es un vector de pesos y B una permutación de los elementos de A, que se definen de la siguiente manera:

$$W = [w_1,...w_m], \quad w_i \in [0,1] \text{ y } \sum_i w_i = 1$$

$$B = B = \sigma(A) = \{a_{\sigma(1)}, \ ..., \ a_{\sigma(m)}\} a_{\sigma(j)} \leq a_{\sigma(i)} \ \forall \ i \leq j$$

El operador OWA es un operador "and-or" [183], lo que le permite modelar los operadores MAX y MIN de manera flexible. Con el fin de poder clasificar los operadores

OWA según estuviesen más cerca de un comportamiento "y" o de un comportamiento "o", Yager introdujo una "medida de orness" [183], asociada con el vector W:

$$orness(W) = \frac{1}{m-1}\sum_{i=1}^{m}(m-i)w_i$$

Fijado un vector de pesos W, el operador OWA estará más cercano al operador clásico O cuanto más cercana a uno sea la medida *orness*; mientras que estará más cerca del operador clásico Y, cuanto más cercana a cero sea la mencionada medida. Matemáticamente, un operador OWA cuyo vector de pesos contenga la mayoría de los valores distintos de cero en las primeras posiciones, se comportará de manera similar a un operador O (*orness* (W) > 0.5), mientras que si estos valores se sitúan en las últimas posiciones el comportamiento sera similar al de un operador Y (*orness* (W) ≤0.5).

Siguiendo las ideas consideradas antes, hemos optado por utilizar los operadores OWA $\phi 1$ (con orness(W) ≤ 0.5) y $\phi 2$ (con orness(W) > 0.5) con el fin de modelar los operadores Y y O, respectivamente. Por consiguiente, cuando h= $\phi 1$, f = max(Neg(peso),valor), y cuando h = $\phi 2$, f = min(peso, valor).

Con estos operadores, dado un documento d_j, evaluamos su relevancia respecto a la subexpresión q_v, RSV_j^v, como $RSV_j^v = [(s'_0,\alpha_0^v), ..., (s'_m, \alpha_m^v)]$, donde

(1) Si q_v es una subexpresión conjuntiva, entonces

$$\alpha_k^v = \phi^1(max((1-\alpha_k^1), \alpha_k^{1j}), \ldots, max((1-\alpha_k^I), \alpha_k^{Ij}))$$

(2) Si q_v es una subexpresión disyuntiva, entonces

$$\alpha_k^v = \phi^2(min(\alpha_k^{1,}\alpha_k^{1j}), \ldots, min(\alpha_k^I, \alpha_k^{Ij}))$$

Como comentamos anteriormente, el hecho de utilizar los operadores OWA para modelar las conectivas Y y O nos permite trabajar con un concepto de computación más flexible en el subsistema de evaluación.

4.3.3.5.- Evaluación de la consulta completa

Llegados a este punto, a cada documento d_j se le asigna un RSV_j, como resultado de su evaluación respecto a la consulta completa. El valor RSV_j final de cada documento se alcanza combinando los RSV parciales obtenidos al evaluar el documento respecto a cada una de las subexpresiones que componen la consulta usando, otra vez, los operadores OWA ϕ^1 y ϕ^2 para modelar de manera flexible las conectivas Y y O, respectivamente.

Por lo tanto, dado un documento d_j, evaluamos su relevancia con respecto a la consulta q como $RSV_j = [(s'_0,\beta_0^j), ..., (s'_m, \beta_m^j)]$, donde:

(1) Si q está en forma normal conjuntiva:

$$B_k^j = \phi^1(\alpha_k^{1,} \ldots, \alpha_k^V)$$

(2) Si q está en forma normal disyuntiva:

$$B_k^j = \phi^2(\alpha_k^{1,} \ldots, \alpha_k^V)$$

siendo V el número de subexpresiones de q.

Observación: *Sobre consultas negadas.* Hay que destacar que, si la consulta está en forma CNF o forma DNF, sólo será necesario definir el operador de negación a nivel de átomo. Esto simplifica la definición del de negación. Al igual que en [95], la evaluación de un documento d_j para un átomo negado $<\neg(t_i), c_i^1, c_i^2, c_i^3>$ se obtiene negando el peso del término índice $F(t_i,d_j)$, es decir, se calcula g^1 utilizando el valor lingüístico Etiqueta(1-$F(t_i,d_j)$).

4.3.3.6.- Salida del SRI

Al final de la evaluación de una consulta que representa las necesidades de información de un usuario, cada documento d_j se caracteriza por un conjunto difuso definido en S', su RSV_j. Por supuesto, la respuesta de un SRI donde la relevancia de cada documento se expresa como un conjunto difuso no es fácil de entender, y lo es menos el trabajar con ella. Para solucionar este problema, la salida de nuestro SRI vendrá dada por medio de clases

lingüísticas de relevancia, al igual que en [96][95]. Además, dentro de cada clase de relevancia, establecemos un orden de los documentos usando para ello un grado de confianza asociado a cada documento.

Esto lo logramos calculando una etiqueta $s' \in S'$ para cada documento d_j, que representa su clase de relevancia. Diseñamos un proceso simple de aproximación lingüística en S' usando una medida de similitud, p.e, la distancia Euclídea. Cada etiqueta $s'_k \in S'$ se representa como un conjunto difuso definido en S', p.e., $\{(s'_0, 0), ..., (s'_k,1),..., (s'_m,0)\}$. Calculamos s^j como

$$s^j = MAX\{s'_l \mid Conf(s'_l, RSV_j) = min_k\{Conf(s'_k, RSV_j)\}\}$$

donde $Conf(s'_k, RSV_j) \in [0,1]$ es el grado de confianza asociado a d_j definido como

$$Conf(s'_k, RSV_j) = \sqrt{\sum_{i=0}^{k-1} (\beta_i^j)^2 + (\beta_k^j - 1)^2 + \sum_{i=k+1}^{m} (\beta_i^j)^2}$$

4.3.4.- Ejemplo de Aplicación

En esta sección, presentamos un ejemplo de funcionamiento del SRI Lingüístico Multigranular que hemos propuesto en las secciones anteriores.

4.3.4.1.- Base de datos

Consideramos una pequeña base de datos con un conjunto de siete documentos $D= \{d_1, ...,d_7\}$, representados por medio de un conjunto de diez términos índice $T = \{t_1,...,t_{10}\}$. Los documentos son indizados haciendo uso de una función de indexación F, la cual les asigna los siguientes pesos a cada uno:

$d_1 = 0.7/t_5 + 0.4/t_6 + 1/t_7$

$d_2 = 1.0/t_4 + 0.6/t_5 + 0.8/t_6 + 0.9/t_7$

$d_3 = 0.5/t_2 + 1.0/t_3 + 0.8/t_4$

$d_4 = 0.9/t_4 + 0.5/t_6 + 1.0/t_7$

$d_5 = 0.7/t_3 + 1/t_4 + 0.4/t_5 + 0.8/t_9 + 0.6/t_{10}$

$d_6 = 1.0/t_5 + 0.99/t_6 + 0.8/t_7$

$d_7 = 0.8/t_5 + 0.02/t_6 + 0.8/t_7 + 0.9/t_8$

De igual forma, consideramos los siguientes cuatro conjuntos de etiquetas con diferente cardinalidad y semántica para valorar los pesos asociados con cada una de las semánticas descritas en la sección 4.3.2.1 (umbral, cuantitativa y importancia relativa) y los RSVs, respectivamente:

$S^1 = \{$

$MI = (0.00\ 0.00\ 0.00\ 0.25)$ $VL = (0.25\ 0.25\ 0.25\ 0.15)$

$L = (0.40\ 0.40\ 0.15\ 0.10)$ $M = (0.50\ 0.50\ 0.10\ 0.10)$

$MU = (0.60\ 0.60\ 0.10\ 0.15)$ $VM = (0.75\ 0.75\ 0.15\ 0.25)$

$MA = (1.00\ 1.00\ 0.25\ 0.00)$

$\}$

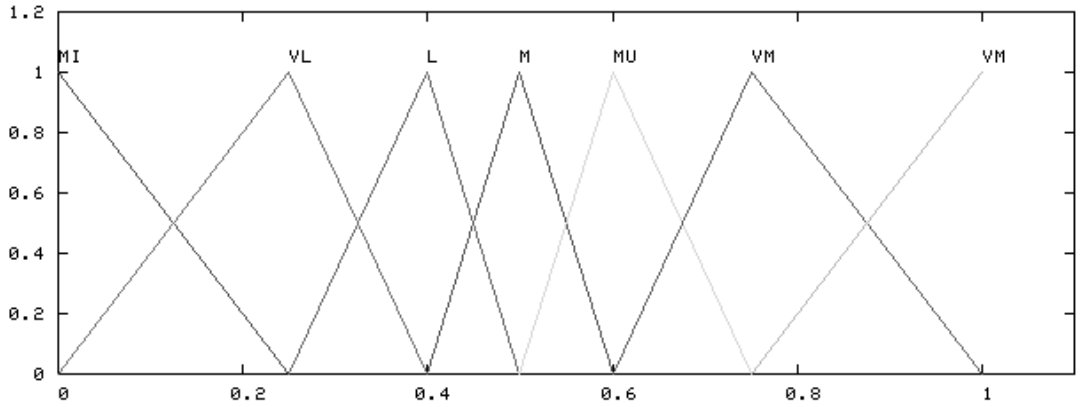


Figura 4.8: Conjunto S^1 de siete etiquetas para modelar la semántica de umbral simétrico

$S^2 = \{$

$N = (0.00\ 0.00\ 0.00\ 0.25)$ $L = (0.25\ 0.25\ 0.25\ 0.25)$

$M = (0.50\ 0.50\ 0.25\ 0.25)$ $H = (0.75\ 0.75\ 0.25\ 0.25)$

$T = (1.00\ 1.00\ 0.25\ 0.00)$

$\}$

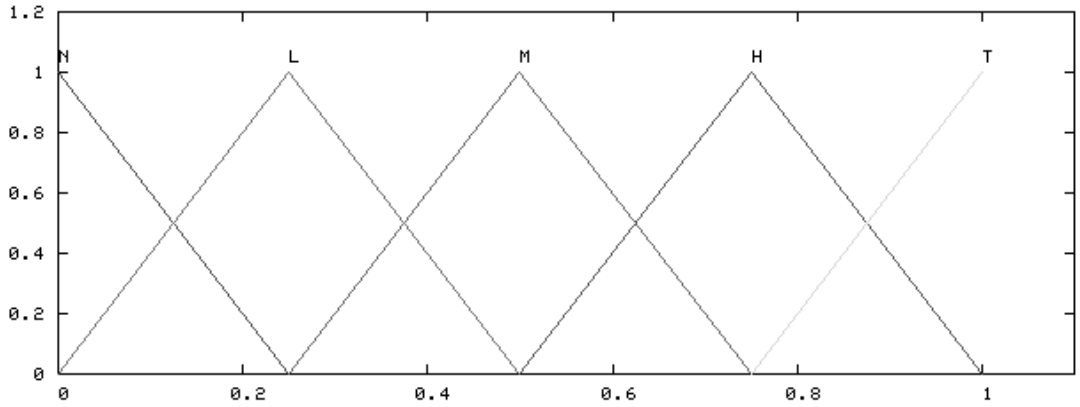


Figura 4.9: Conjunto S^2 de cinco etiquetas para modelar la semántica cuantitativa

S^3 ={

N = (0.00 0.00 0.00 0.00) EL = (0.01 0.02 0.02 0.05)

VL= (0.10 0.18 0.06 0.05) L = (0.22 0.36 0.05 0.06)

M = (0.41 0.58 0.09 0.07) H = (0.63 0.80 0.05 0.06)

VH = (0.78 0.92 0.06 0.05) EH = (0.98 0.99 0.05 0.01)

T = (1.00 1.00 0.00 0.00)

}

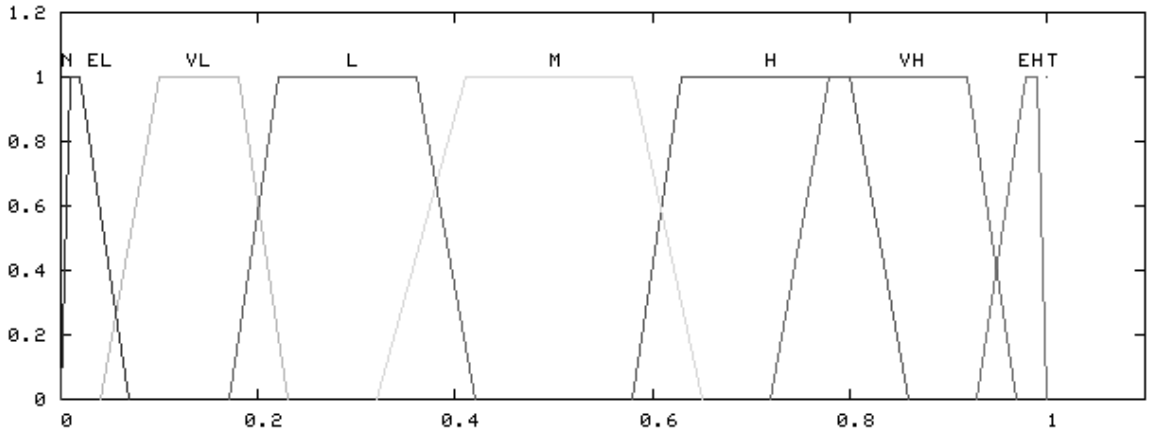


Figura 4.10: Conjunto S^3 de nueve etiquetas para modelar la semántica de importancia relativa

S' ={

N = (0.00 0.00 0.00 0.00) EL = (0.01 0.02 0.02 0.05)

VL= (0.10 0.18 0.06 0.05) ML = (0.22 0.30 0.05 0.06

L = (0.31 0.36 0.05 0.06) M = (0.41 0.58 0.09 0.07)

H = (0.63 0.70 0.05 0.06) MH = (0.71 0.80 0.05 0.06)

VH = (0.78 0.92 0.06 0.05) EH = (0.98 0.99 0.05 0.01)

T = (1.00 1.00 0.00 0.00)

}

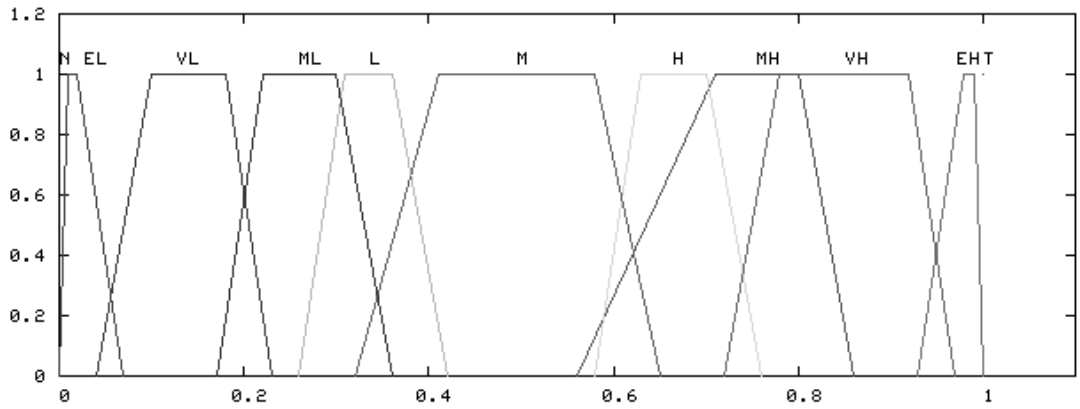


Figura 4.11: Conjunto S' de once etiquetas para modelar los RSVs finales de los documentos.

Finalmente, consideramos las necesidades de un usuario formuladas a través de la siguiente consulta:

$$q = ((t_5, MU, L, VH) \wedge (t_6, L, L, VL)) \vee (t_7, MU, L, H)$$

El proceso aplicado por el SRI se muestra en los siguientes apartados.

PREPROCESAMIENTO DE LA CONSULTA

La consulta q está en forma normal disyuntiva (DNF), pero presenta una subexpresión con un único átomo. En consecuencia, q debe ser preprocesada y transformada a forma normal de manera que todas las subexpresiones que la compongan tengan al menos dos átomos. La consulta equivalente tras la transformación sería la que se muestra a continuación y estaría expresada en forma normal conjuntiva (CNF):

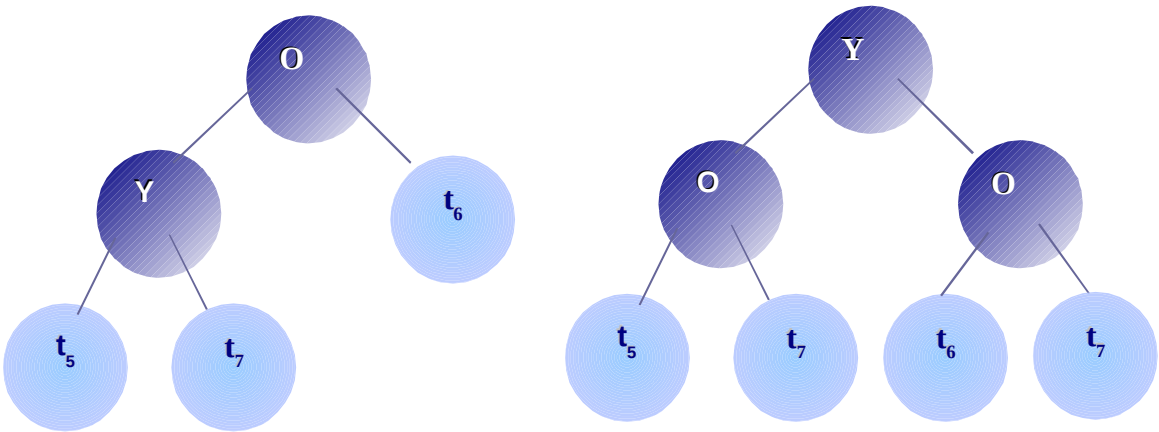


Figura 4.12: Consulta antes y después del preprocesamiento

$$q' = ((t_5,MU,L,VH) \vee (t_7,MU,L,H)) \wedge ((t_6,L,L,VL) \vee (t_7,MU,L,H))$$

EVALUACIÓN DE LOS ÁTOMOS CON RESPECTO A LA SEMÁNTICA DE UMBRAL SIMÉTRICO

Lo primero es obtener la representación lingüística de los documentos usando la función de traducción *Etiqueta*:

$$d_1 = VM/t_5 + L/t_6 + MA/t_7$$

$$d_2 = MA/t_4 + MU/t_5 + VM/t_6 + MA/t_7$$

$$d_3 = M/t_2 + MA/t_3 + VM/t_4$$

$$d_4 = MA/t_4 + M/t_6 + MA/t_7$$

$$d_5 = VM/t_3 + MA/t_4 + L/t_5 + VM/t_9 + MU/t_{10}$$

$$d_6 = MA/t_5 + MA/t_6 + VM/t_7$$

$$d_7 = VM/t_5 + MI/t_6 + VM/t_7 + MA/t_8$$

En la sección 4.3.3.2, comentamos que el parámetro *B*, de la función g^1, podía definirse de forma independiente, por ejemplo como $B = 1$, o dependiendo de la cercanía entre Label($F(d_j,t_i)$) y c_i^1, por ejemplo como $B = round(2(|b-a|)/T)$. En este ejemplo consideramos la última opción. La evaluación de los átomos de acuerdo a la semántica de umbral simétrico modelada mediante la función g^1 es la siguiente:

$$\{RSV_1^{5,1} = VM,\ RSV_2^{5,1} = MU,\ RSV_5^{5,1} = VL,\ RSV_6^{5,1} = MA\ RSV_7^{5,1} = VM\}$$

$$\{RSV_1^{6,1} = MU,\ RSV_2^{6,1} = MI,\ RSV_4^{6,1} = M,\ RSV_6^{6,1} = MI\ RSV_7^{6,1} = MA\}$$

$$\{RSV_1^{7,1} = MA,\ RSV_2^{7,1} = MA,\ RSV_4^{7,1} = MA,\ RSV_6^{7,1} = VM\ RSV_7^{7,1} = VM\}$$

EVALUACIÓN DE LOS ÁTOMOS CON RESPECTO A LA SEMÁNTICA CUANTITATIVA

La evaluación de los átomos de la consulta de acuerdo a la semántica cuantitativa modelando por la función g^2 es:

$$\{RSV_1^{5,1,2} = VM,\ RSV_6^{5,1,2} = MA\}$$

$$\{RSV_1^{6,1,2} = MU,\ RSV_7^{6,1,2} = MA\}$$

$$\{RSV_1^{7,1,2} = MA,\ RSV_2^{7,1,2} = MA\}$$

Como se puede apreciar, el uso de esta semántica decrementa el número de documentos asociados que se considerarán para cada término.

EVALUACIÓN DE LAS SUBEXPRESIONES Y MODELADO DE LA SEMÁNTICA DE IMPORTANCIA RELATIVA

Los resultados de la función de transformación $\tau_{S1S'}$ aplicada sobre los $RSV_j^{i,1,2}$ son los que se muestran a continuación:

$$\tau_{S1S'}(MU) = \{(s_0,0),\ (s_1,0),\ (s_2,0),\ (s_3,0),\ (s_4,0),\ (s_5,0.88),\ (s_6,0.85),\ (s_7,0.45),\ (s_8,\ 0.14),\ (s_9,0),\ (s_{10},0)\}$$

$$\tau_{S1S'}(VM) = \{(s_0,0),\ (s_1,0),\ (s_2,0),\ (s_3,0),\ (s_4,0),\ (s_5,0.23),\ (s_6,0.76),\ (s_7,1),\ (s_8,0.90),\ (s_9,0.23),\ (s_{10},0)\}$$

$$\tau_{S1S'}(MA) = \{(s_0,0),\ (s_1,0),\ (s2,0),\ (s_3,0),\ (s_4,0),\ (s_5,0),\ (s_6,0.032),(s_7,0.35),(s_8,0.73),(s_9,0.96),\ (s_{10},1)\}$$

$$\tau_{S1S'}(MI) = \{(s_0,1),\ (s_1,0.96),\ (s_2,0.68),\ (s_3,0.27),\ (s_4,0),\ (s_5,0),\ (s_6,0),\ (s_7,0),\ (s_8,0),\ (s_9,0),\ (s_{10},0)\}$$

y los de la función $\tau_{S3S'}$ aplicada sobre el grado de importancia relativa de los términos c_i^3 son:

$$\tau_{S3S'}(VH) = \{(s_0,0),\ (s_1,0),\ (s_2,0),\ (s_3,0),\ (s_4,0),\ (s_5,0),\ (s_6,0.33),\ (s_7,1.0),\ (s_8,1.0),\ (s_9,0.4),\ (s_{10},0)\}$$

$$\tau_{S3S'}(H) = \{(s_0,0),\ (s_1,0),\ (s_2,0),\ (s_3,0),\ (s_4,0),\ (s_5,0.58),\ (s_6,1.0),\ (s_7,1.0),\ (s_8,\ 1.0),\ (s_9,0),\ (s_{10},0)\}$$

$$\tau_{S3S'}(VL) = \{(s_0,0),\ (s_1,0.27),\ (s_2,1.0),\ (s_3,0.6),\ (s_4,0),\ (s_5,0),\ (s_6,0),\ (s_7,0),\ (s_8,0),\ (s_9,0),\ (s_{10},0)\}$$

La consulta q' tiene dos subexpresiones y cada una de ellas presenta dos átomos:

$$q'^1 = ((t_5, MU, L, VH) \vee (t_7, MU, L, H))$$

$$q'^2 = ((t_6, L, L, VL) \vee (t_7, MU, L, H))$$

Cada subexpresión está en forma normal disyuntiva (DNF), por lo que debemos utilizar un operador OWA con ϕ^2 con *orness (W)* > 0.5 (por ejemplo, *W*=[1,0]).

Evaluación de la subexpresión q'^{1}

$RSV_1^1 = \{(s_0,0), (s_1,0), (s_2,0), (s_3,0), (s_4,0), (s_5,0), (s_6,0.33), (s_7,1.0), (s_8,0.9), (s_9,0.23), (s_{10},0)\}$

$RSV_2^1 = \{(s_0,0), (s_1,0), (s_2,0), (s_3,0), (s_4,0), (s_5,0), (s_6,0.032), (s_7,0.35), (s_8,0.73), (s_9,0), (s_{10},0)\}$

$RSV_6^1 = \{(s_0,0), (s_1,0), (s_2,0), (s_3,0), (s_4,0), (s_5,0), (s_6,0.032), (s_7,0.35), (s_8,0.73), (s_9,0.4), (s_{10},0)\}$

Evaluación de la subexpresión q'^{2}

$RSV_1^2 = \{(s_0,0), (s_1,0), (s_2,0), (s_3,0), (s_4,0), (s_5,0), (s_6,0.032), (s_7,0.35), (s_8,0.73), (s_9,0), (s_{10},0)\}$

$RSV_2^2 = \{(s_0,0), (s_1,0.27), (s_2,0.67), (s_3,0.26), (s_4,0), (s_5,0), (s_6,0.032), (s_7,0.35), (s_8,0.73), (s_9,0), (s_{10},0)\}$

$RSV_7^2 = \{(s_0,0), (s_1,0), (s_2,0), (s_3,0), (s_4,0), (s_5,0), (s_6,0), (s_7,0), (s_8,0), (s_9,0), (s_{10},0)\}$

EVALUACIÓN DE LA CONSULTA COMPLETA

La evaluación del documento con respecto a la consulta completa lo obtenemos usando un operador OWA ϕ^1 con *orness (W)* ≤ 0.5 (por ejemplo, *W*=[0.4,0.6]):

$RSV_1 = \{(s_0,0), (s_1,0), (s_2,0), (s_3,0), (s_4,0), (s_5,0), (s_6,0.15), (s_7,0.61), (s_8,0.8), (s_9,0.092), (s_{10},0)\}$

RSV_2 = {(s_0,0), (s_1,0.11), (s_2,0.27), (s_3,0.10), (s_4,0), (s_5,0), (s_6,0.032), (s_7,0.35), (s_8,0.73), (s_9,0), (s_{10},0)}

RSV_6 = {(s_0,0), (s_1,0), (s_2,0), (s_3,0), (s_4,0), (s_5,0), (s_6,0.013), (s_7,0.14), (s_8,0.3), (s_9,0.16), (s_{10},0)}

RSV_7 = {(s_0,0), (s_1,0), (s_2,0), (s_3,0), (s_4,0), (s_5,0), (s_6,0), (s_7,0), (s_8,0), (s_9,0), (s_{10},0)}

SALIDA DEL SRI

Para terminar, calculamos una etiqueta $s_j \in S'$ para cada documento d_j, que representará la clase de relevancia a la que pertenece, obteniéndose la salida final del sistema. Además, la etiqueta s_j lleva asociado su valor de la medida Conf, que nos permite distinguir entre documentos cuando estos pertenezcan a la misma clase de relevancia.

{(d_1,(VH,0.665)), (d_2,(VH,0.539)), (d_6,(VH,0.731))}

5.- UN ALGORITMO GENÉTICO MULTIOBJETIVO PARA EL APRENDIZAJE DE CONSULTAS PERSISTENTES REPRESENTADAS COMO CONSULTAS LINGÜÍSTICAS

En los capítulos previos hemos visto como los modelos lingüísticos dotaban de más flexibilidad y facilidad de representación a las consultas instantáneas en el proceso de RI, y como la caracterización de perfiles de usuario como consultas clásicas de RI (consultas persistentes) conseguía una mayor expresividad en los mismos.

En este capítulo, proponemos combinar los resultados obtenidos en los Capítulos 3 y 4. Para ello, nos planteamos usar consultas lingüísticas como consultas persistentes (perfiles), lo que hará más interpretables los perfiles de usuario; y presentamos un AG multiobjetivo basado en el enfoque IQBE [49] que permita aprenderlas automáticamente a partir de un conjunto de documentos proporcionados por el usuario, solucionando lo que se conoce como "*problema del vocabulario*" en la interacción hombre-ordenador [76].

En primer lugar, haremos una pequeña introducción, planteando de forma más extendida el por qué caracterizar los perfiles de usuario como consultas lingüísticas. A continuación, consideraremos la estructura que tiene nuestra propuesta (esquema de codificación, operadores genéticos, etc.) para pasar posteriormente a realizar una análisis pormenorizado de los resultados obtenidos en la experimentación realizada sobre las colecciones Cranfield y CACM. Finalmente, compararemos nuestra propuesta con un algoritmo para generar perfiles de usuario basado en el modelo espacio vectorial [60].

5.1.- Introducción

Como hemos venido comentado a lo largo de esta memoria, las consultas persistentes pueden representarse por medio de consultas clásicas usando cualquier modelo de RI (Booleano, Booleano extendido, ...). En concreto, en el Capítulo 3, las representábamos mediante consultas Booleanas, consiguiendo una mayor expresividad que utilizando la habitual "bag of words".

Sin embargo, obtener buenos resultados en un proceso de búsqueda de información depende de la habilidad del usuario para expresar sus necesidades de información mediante una consulta. Se ha demostrado que, a menudo, el usuario no tiene una imagen clara de lo que está buscando y solo puede representar sus necesidades de información de forma imprecisa y vaga, encontrándonos con la situación conocida como, "fuzzy-querying" [137].

Los lenguajes de consulta flexibles pueden ayudar a solucionar este problema gracias a su capacidad de personalización. Un lenguaje de estas características es aquel que permite expresar la necesidades de información subjetivas de forma simple y aproximada [141].

En este sentido, se han propuesto varios modelos de RI lingüísticos que usan un enfoque difuso-lingüístico [187] para modelar los pesos de las consultas y la relevancia de los documentos (véase la Sección 2.4). Sin embargo, estos modelos presentan una serie de problemas a los que hemos intentado dar solución mediante un nuevo SRI Lingüístico basado en Información Lingüística Multigranular (Capítulo 4).

En el capítulo actual nos planteamos usar consultas lingüísticas generadas por el SRI Lingüístico propuesto en el Capítulo 4 para representar las consultas persistentes. Pensamos que esta forma de modelar los perfiles como consultas persistentes flexibles mejorará sustancialmente la interpretabilidad de las consultas persistentes obtenidas y la eficacia en la recuperación.

Por otro lado, a pesar de que se pueden utilizar diferentes tipos de consultas para representar las consultas persistentes (perfiles), al usuario le suele resultar muy difícil formular la consulta, debido a la dificultad para seleccionar las palabras adecuadas para comunicarse con el sistema, lo que clásicamente se conoce como "*problema del vocabulario*" en la interacción hombre-ordenador [76]. Por esta razón, se han aplicado técnicas de

aprendizaje automático en la construcción de “perfiles implícitos” [60][138]. En concreto, el sistema aprende de manera automática el perfil a partir de un conjunto de documentos proporcionado por el usuario.

Siguiendo esta filosofía, en este capítulo presentamos un algoritmo evolutivo IQBE multiobjetivo para derivar de manera automática consultas persistentes representadas como consultas lingüísticas. El proceso de IQBE puede utilizarse directamente para construir consultas persistentes, ya que trabaja de la misma forma que los métodos de aprendizaje de perfiles explícitos, comentados en la Sección 3.1.

5.2.- Estructura del Algoritmo

En esta sección, presentamos un AG multiobjetivo que aprende automáticamente consultas persistentes representadas como consultas Booleanas ponderadas con pesos lingüísticos (consultas lingüísticas), consultas legítimas del SRI Lingüístico basado en información lingüística multigranular definido en el Capítulo 4.

Las principales características de las consultas con las que trabajaremos son:

- Están en forma normal conjuntiva (CNF) o disyuntiva (DNF).
- Están formadas por dos o más subexpresiones.
- Las subexpresiones que las forman están compuestas por dos o más términos.
- Los términos presentes en la consulta se pueden ponderar simultáneamente con tres pesos lingüísticos diferentes, asociados con tres semánticas diferentes: semántica de umbral simétrica, semántica de importancia relativa y semántica cuantitativa.

La figura 5.1 muestra un ejemplo de una consulta de este tipo.

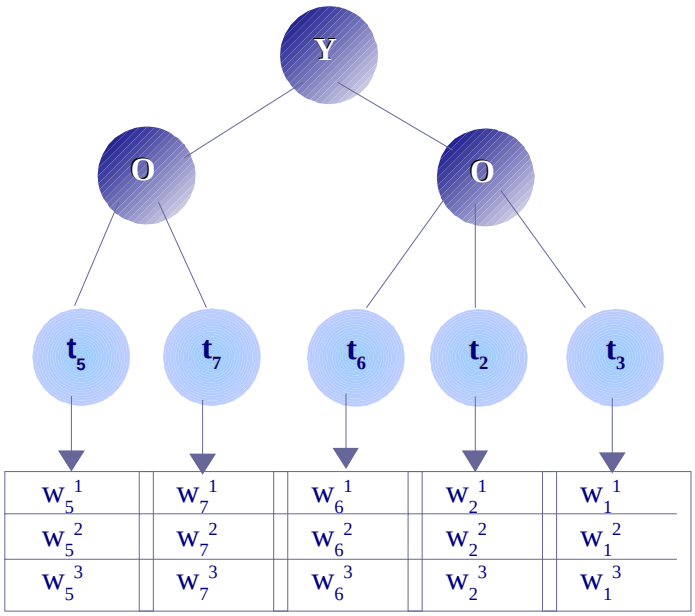


Figura 5.1: Consulta Booleana ponderada con pesos lingüísticos

5.2.1.- Representación

En [65], Fernández-Villacañas y Shackleton presentan un AG binario clásico que adapta árboles de consulta Booleanos, representando para ello los árboles de expresión como cadenas binarias. De manera similar, nosotros presentamos un esquema de codificación que permite representar los árboles de consulta lingüística en forma de vector de enteros, lo que simplifica la representación.

Un cromosoma *C* que codifica una consulta estará compuesto por dos partes (cromosoma con dos niveles de información), C_1 y C_2, las cuales codificarán la estructura de la consulta y los pesos asociados a los términos, respectivamente.

Para representar la estructura de la consulta, consideramos que todas las consultas están o en CNF, o en DNF, de manera que el cromosoma almacena únicamente las subexpresiones de las que está formada una consulta, sin tener que almacenar operadores (siempre son los mismos). Con esta representación, el árbol de consulta se codifica como un vector de números enteros, donde el 0 actúa como separador de subexpresiones mientras que el resto de números representan los términos.

Por otro lado, los pesos (etiquetas lingüísticas) asociados a los términos se representan como otro vector de enteros, donde cada peso se codifica como su posición en el conjunto de etiquetas al que pertenece (el enfoque ordinal se caracteriza por utilizar conjuntos de etiquetas totalmente ordenados, véase la Sección 4.2.1). Las tres etiquetas que ponderan a un mismo término se codificarán consecutivamente (terna de pesos).

La consulta de la Figura 5.1 se codificaría en un cromosoma *C* con la siguiente forma:

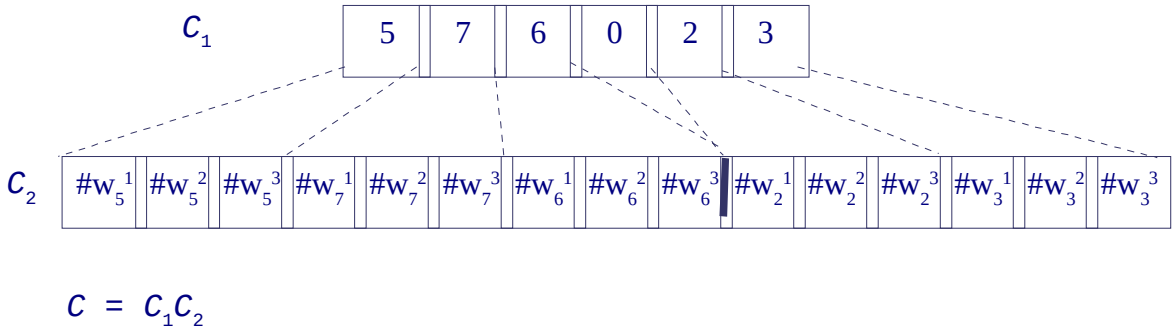


Figura 5.2: Estructura de un cromosoma

donde "*# peso*" es el orden que la etiqueta ocupa en el conjunto de etiquetas al que pertenece.

5.2.2.- Generación de la Población Inicial

Todos los cromosomas de la población inicial se obtienen de manera aleatoria, generando por separado las dos partes que los componen.

La parte que codifica el árbol de consulta (C_1) se genera teniendo en cuenta las siguientes tres propiedades:

- La consulta debe estar en forma normal.
- La consulta debe estar formada por dos o más subexpresiones.
- Las subexpresiones han de estar compuestas por dos o más términos.

Los términos que compondrán la consulta se seleccionan de aquellos incluidos en el conjunto de documentos proporcionados por el usuario, teniendo mayor probabilidad aquellos que aparezcan en los documentos relevantes frente a los que aparezcan en los irrelevantes.

Los pesos (C_2) se calculan aleatoriamente, haciendo variar cada gen en su respectivo intervalo.

5.2.3.- Evaluación de los Individuos

El valor de adaptación de una consulta con respecto al conjunto de documentos proporcionados por el usuario consiste en la optimización simultánea de los dos criterios clásicos para medir la efectividad de la RI, la precisión y la exhaustividad [176].

La **exhaustividad** es la proporción de documentos relevantes recuperados en una búsqueda determinada sobre el número de documentos relevantes para esa búsqueda en la base de datos, siendo su fórmula:

$$E = \frac{n^o \, de \, documentos \, relevantes \, recuperados}{n^o \, de \, documentos \, relevantes}$$

La **precisión** es la proporción de documentos relevantes recuperados sobre el número total de documentos recuperados, siendo su fórmula:

$$P = \frac{n^o \, de \, documentos \, relevantes \, recuperados}{n^o \, de \, documentos \, recuperados}$$

Como ya se comentado anteriormente, ambos criterios están inversamente relacionados [27], esto es, cuando la precisión sube, la exhaustividad normalmente baja y viceversa (véase la Sección 1.4). Por tanto, el algoritmo aborda la optimización de estos criterios utilizando un enfoque multiobjetivo en el que se intentan maximizar ambos.

¿CUÁL ES EL CONJUNTO DE DOCUMENTOS RECUPERADOS?

Una cuestión importante es cómo se determina el conjunto de documentos recuperados por una consulta. El algoritmo evalúa cada consulta en el SRI Lingüístico basado en información multigranular definido en el Capítulo 4, siguiendo los cincos pasos descritos en él:

1. Preprocesamiento de la consulta.
2. Evaluación de los átomos con respecto a la semántica de umbral.
3. Evaluación de los átomos con respecto a la semántica cuantitativa.
4. Evaluación de las subexpresiones y modelado de la semántica de importancia.
5. Evaluación de la consulta completa.

asociando a cada documento su correspondiente valor de recuperación RSV.

Finalmente, el conjunto de documentos recuperados lo formarán aquellos con un valor RSV igual o superior a un umbral establecido a priori.

5.2.4.- Enfoque Multiobjetivo Considerado

El AE multiobjetivo basado en el Pareto considerado para ser incorporado ha sido el SPEA [193][194], el mismo que se utilizó en el Capítulo 3 (Sección 3.4.2) para extender la propuesta de aprendizaje de consultas Booleanas de Smith y Smith, al mantener el concepto de elitismo.

El algoritmo utiliza dos poblaciones, una población externa (P_e) que mantiene las soluciones no dominadas encontradas desde el comienzo de la simulación; y una población actual (P).

En cada generación, se crea una nueva población seleccionando, mediante torneo binario, las mejores soluciones existentes tanto en la población externa como en la población actual ($P \cup P\_e$), y aplicando a los individuos seleccionados los operadores genéticos de cruce y mutación. Además, las nuevas soluciones no dominadas encontradas se comparan con la población externa existente con objeto de preservar las soluciones no dominadas resultantes.

El tamaño de la población externa se mantiene fijo utilizando un algoritmo de clustering.

5.2.5.- Operadores Genéticos

Debido a la naturaleza especial de los cromosomas que empleamos (doble nivel de información), el diseño de los operadores genéticos capaces de tratar con ellos se convierte en una tarea de suma importancia. Además, la estrecha relación entre las dos partes de un cromosoma, requiere operadores que trabajen de forma cooperativa en C_1 y C_2, con el fin de sacar el máximo rendimiento a la representación elegida.

Se puede observar claramente que la relación existente presentará ciertos problemas si no se actúa de manera adecuada. Por ejemplo, modificaciones en la primera parte del cromosoma deben reflejarse de manera automática en la segunda. Sería incorrecto modificar la estructura de la consulta, ya sea añadiendo, eliminando o cambiando términos y subexpresiones, y continuar trabajando con los mismos pesos. Por otro lado, es conveniente desarrollar la alteración de los cromosomas en un orden adecuado de manera que se obtengan descendientes con significado pleno.

Teniendo en cuenta estos aspectos, hemos considerado los operadores descritos en las secciones siguientes.

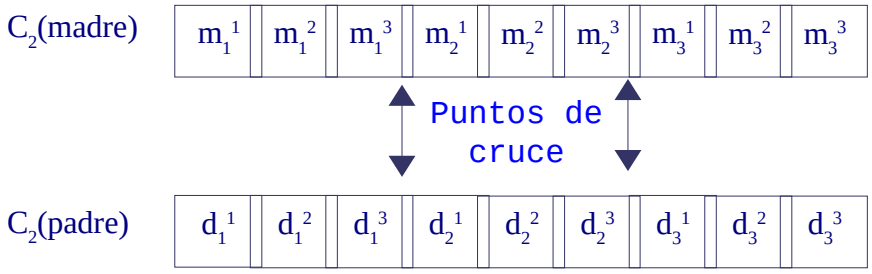
5.2.5.1.- Operadores de cruce

El cruce entre cromosomas dependerá de que la estructura de las consultas que codifican los padres sea la misma o no.

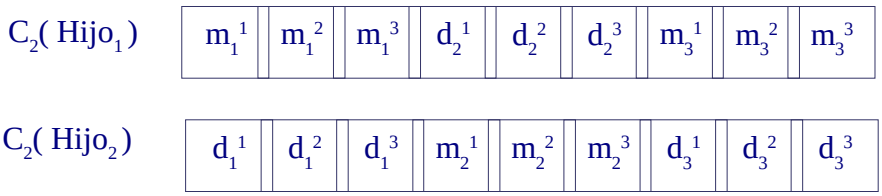
Cruce cuando ambos padres codifican la mima consulta ($C_1^1 = C_1^2$)

Si éste es el caso, la búsqueda genética ha localizado una zona prometedora en el espacio de búsqueda y hay que explotarla. Esta tarea la llevamos a cabo aplicando un cruce en dos puntos sobre C_2 (pesos) y, obviamente, manteniendo los valores de C_1 en los descendientes.

En la Figura 5.3, podemos ver un ejemplo del funcionamiento de este operador.



a) Cromosomas padres que se van a cruzar



b) Descendientes

Figura 5.3: Cruce en dos puntos

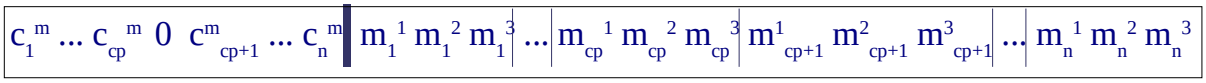
Los posibles puntos de cruce serán los límites entre ternas de pesos.

Cruce cuando los padres codifican diferentes consultas ($C_2^1 \neq C_2^2$)

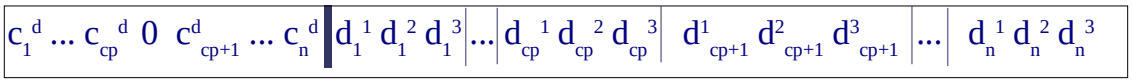
En este segundo caso, es recomendable el uso de la información codificada por los padres para explorar el espacio de búsqueda y descubrir nuevas zonas prometedoras. Para ello, se aplica un operador de cruce simple en ambas partes de los cromosomas. El operador actúa de la siguiente manera: se genera un punto de cruce aleatorio (pc) en C_1 para cada uno de los padres y se intercambian los genes existentes entre dicho punto de corte y el final de C_1. En C_2, el cruce se hace de la misma forma, utilizando los puntos de corte equivalentes.

La Figura 5.4 muestra un ejemplo para clarificar la aplicación de este operador.

C(madre)

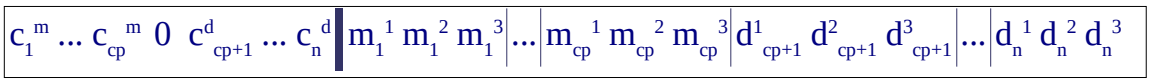


C(padre)

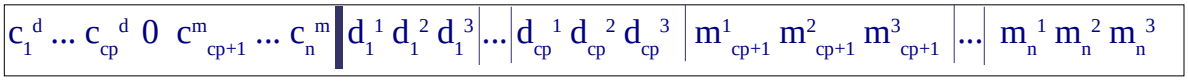


a) Cromosomas padres que se van a cruzar

C(Hijo$_1$)



C(Hijo$_2$)



b) Descendientes

Figura 5.4: Cruce explorativo

Los posibles puntos de cruce en C_1 serán los separadores entre subexpresiones.

El proceso completo permitirá al AG conseguir un equilibrio adecuado entre exploración y explotación en el espacio de búsqueda. Inicialmente, se ejecutarán un gran número de cruces simples en el cromosoma completo y muy pocos cruces en dos puntos en C_2. Este comportamiento produce que la búsqueda genética lleve a cabo a una extensa exploración, localizando zonas prometedoras hacia las que derivará la población en la sucesivas iteraciones. En este momento se incrementa la explotación de las nuevas zonas y se reduce la exploración del espacio. Por lo tanto, el número de cruces en dos puntos aumenta, mientras que la aplicación del cruce simple se decrementa, como se puede ver en la Figura 5.8.

5.2.5.2.- Operadores de mutación

Se han utilizado siete operadores diferentes, seis que actúan sobre C_1, y uno sobre C_2.

Mutación sobre C_2

El operador de mutación seleccionado para C_2 es similar al propuesto por Thrift en [170]. Cuando se lleva a cabo una mutación sobre un gen perteneciente a la segunda parte del cromosoma, la etiqueta se cambia por la anterior o la posterior en el conjunto (la decisión se toma de manera aleatoria). Sin embargo, si la etiqueta que se va a modificar es la primera o la última del conjunto, se realiza el único cambio posible.

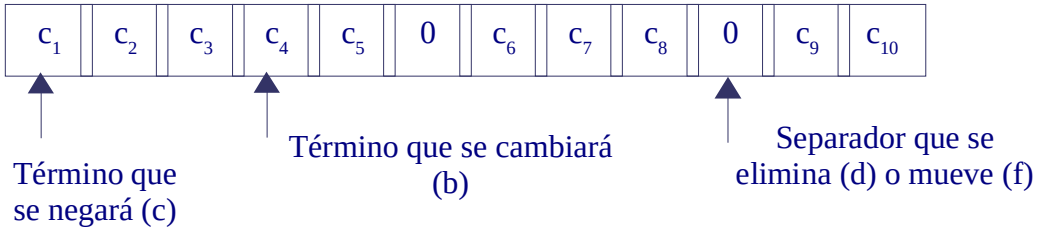
Mutación sobre C_1

Los operadores de mutación sobre C_1 son los siguientes:

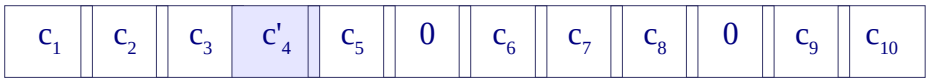
1. Cambio de un término por otro generado aleatoriamente.
2. Negación de un término.
3. Eliminación de un separador elegido aleatoriamente.
4. Adición de un separador en una posición aleatoria.
5. Desplazamiento un separador de una posición a otra.
6. Cambio de una subexpresión por otra nueva generada de manera aleatoria que tenga más o menos términos que la anterior.

Mientras que los cinco primeros operadores no afectan en nada a C_2, el cambiar una subexpresión por otra afecta en gran medida a C_2 (al aumentarse o disminuirse el número de términos). Por lo tanto, en el momento en que se modifica una subexpresión de C_1, la parte C_2 se actualiza automáticamente, moviendo a la derecha o la izquierda, según corresponda, los pesos válidos y generando aleatoriamente los pesos nuevos.

Las Figuras 5.5 y 5.6 muestran gráficamente como actúan los operadores de mutación.

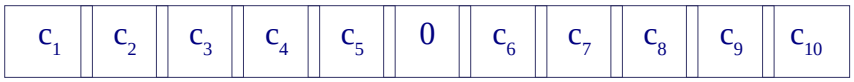


a) Parte C_1 de un cromosoma y posiciones aleatorias donde se aplicarán las diferentes mutaciones

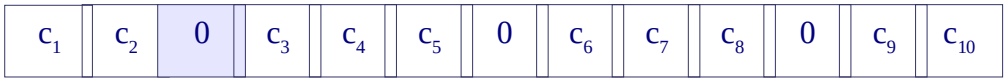


b) Cambio del término c_4 por c'_4

c) Negación del término c_1



d) Eliminación de un separador



e) Adición de un separador

f) Desplazamiento de un separador

Figura 5.5: Operadores de mutación que no afectan a C_2

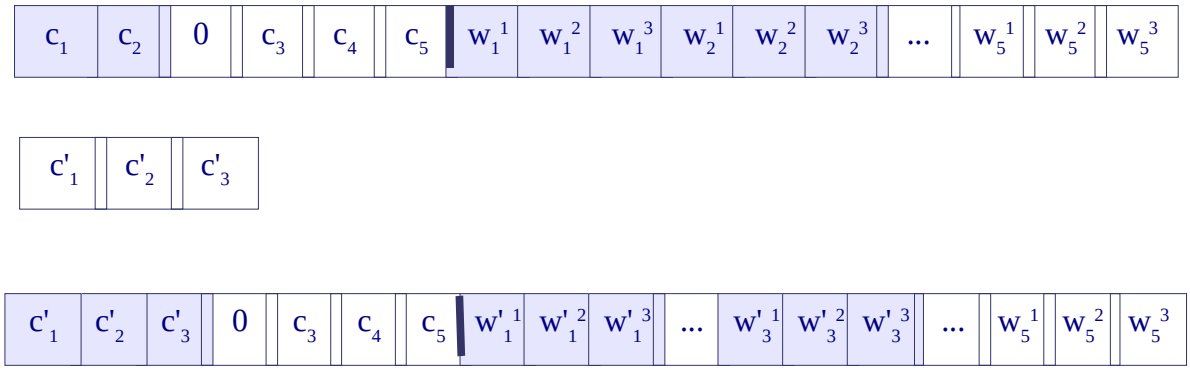


Figura 5.6: Cambio de una subexpresión por otra

El siguiente esquema muestra la aplicación de los operadores genéticos definidos anteriormente.

```
(1) Seleccionar los dos padres que se cruzarán
(2) Si aleatorio < Probabilidad de cruzar las partes C1
    (a)Si C1(madre) = C1(padre) entonces
          Cruce en dos puntos (explotativo)
       en otro caso
          Cruce estándar (explorativo)
       fin_si
    en otro caso
       Si aleatorio < Probabilidad de cruzar las partes C2
          Cruce en dos puntos
       fin_si
    fin_si
(3) Para cada hijo
   (a) Si aleatorio < Probabilidad de mutación en la parte C1
          Elegir una mutación de acuerdo a las probabilidades que
          se establezcan y mutar
   fin_si
   (b) Si la mutación no ha afectado a C2
          Calcular el numero de genes que se mutarán en C2 de
          acuerdo a la probabilidad de mutación en la parte C2 y
          utilizar la mutación de Thrift
       fin_si
fin_para
```

Finalmente, la Figura 5.7 muestra el ámbito de aplicación de los operadores genéticos propuestos, y la Figura 5.8 la progresión en el uso de los cruces explorativo y explotativo en la segunda ejecución de la consulta 157 de Cranfield.

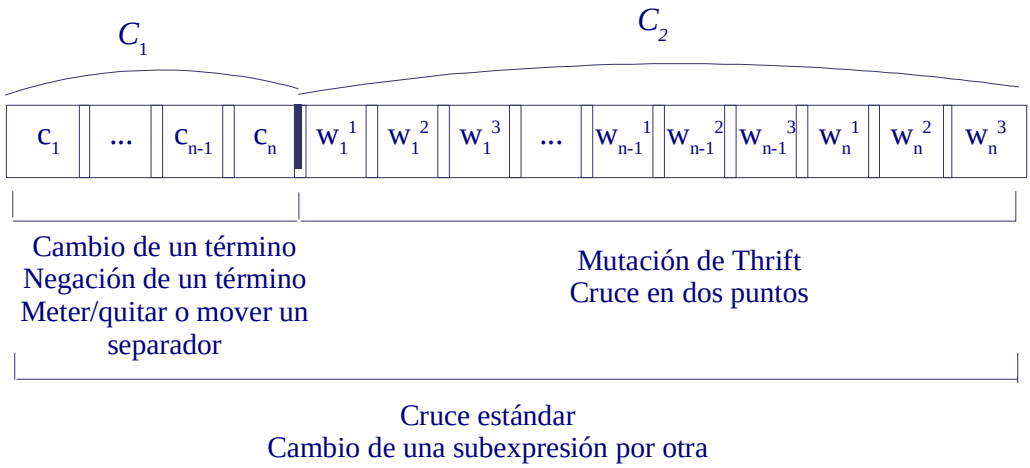


Figura 5.7: Ámbito de aplicación de los operadores genéticos

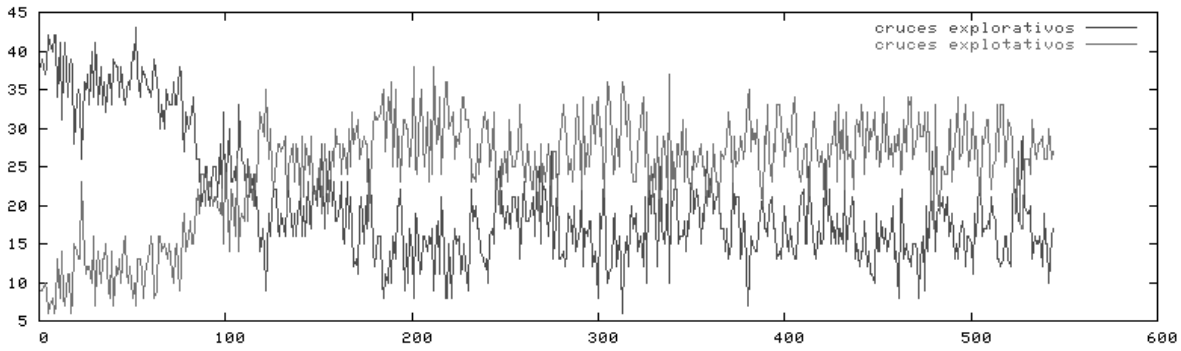


Figura 5.8: Número de cruces por generación para la 2ª ejecución de la consulta 157 de Cranfield

5.3.- Experimentación y Análisis de Resultados

Esta sección se encarga de describir los experimentos realizados y los resultados obtenidos por el algoritmo propuesto. Los experimentos se realizarán sobre las bases documentales Cranfield y CACM, utilizando el entorno experimental considerado en el Apéndice A.1.

5.3.1.- AG Multiobjetivo Lingüístico

Hemos ejecutado nuestra propuesta (SPEA-AG) 5 veces, cada una de ellas con una semilla diferente, para cada una de las 35 consultas consideradas. Los valores empleados para los distintos parámetros del algoritmo se recogen en la Tabla 5.1.

Parámetros	Valores
Tamaño de la población	100
Tamaño de la población elitista	25
Número de evaluaciones	50000
Tamaño del torneo	2
Probabilidad de Cruce en C_1 y C_2	0,8
Probabilidad de Mutación en C_1 y C_2	0,2
Número máximo de términos en la consulta	10 términos
Probabilidad de escoger un término relevante	0.8
Probabilidad de negar un término	0.3
Umbral de recuperación	3ª etiqueta

Tabla 126: Valores de parámetros considerados para el algoritmo SPEA-AG

La sección incluye seis subsecciones, la mitad de ellas dedicadas a la experimentación realizada con las diecisiete consultas seleccionadas de Cranfield y la otra mitad a las dieciocho de CACM. Las tres subsecciones correspondientes a cada colección están dedicadas, respectivamente, al análisis de los Paretos obtenidos y de la eficacia de las consultas presentes en los Pareto de acuerdo a las medidas clásicas de precisión y exhaustividad y a la precisión media.

MÉTRICAS UTILIZADAS PARA MEDIR LA CALIDAD DE LOS PARETOS

De las cinco métricas mencionadas en la Sección 3.2.3.1, haremos uso de cuatro, abandonando únicamente la métrica de similitud con el Pareto óptimo por la imposibilidad de aplicación en nuestro caso al no conocerse éste. En concreto, trabajaremos con el número de

soluciones contenidas en los conjuntos Pareto derivados y el número de éstas que son distintas *con respecto a los valores de los objetivos*[3]. En el caso de las métricas M_2 y M_3, trabajaremos también en el espacio objetivo (es decir, emplearemos M_2* y M_3*), que es donde nos interesa que estén bien distribuidas las consultas persistentes aprendidas para poder obtener la mayor variedad posible de balances exhaustividad-precisión.

CONJUNTO DE SOLUCIONES REPRESENTATIVAS DEL PARETO

Ante la imposibilidad de analizar todas las consultas existentes en todos los Paretos, se elige un conjunto de soluciones representativas de la fusión de los Paretos correspondientes a cada una de las ejecuciones, de igual forma que se hizo en el Capítulo 3, Sección 3.5.2.

CONJUNTOS DE ETIQUETAS UTILIZADOS

Como se indicó en la Sección 5.2, el algoritmo que proponemos aprende automáticamente consultas persistentes representadas como consultas Booleanas ponderadas con pesos lingüísticos (consultas lingüísticas), consultas legítimas del SRI Lingüístico basado en información lingüística multigranular definido en el Capítulo 4. En concreto, cada término estará ponderado con tres pesos valorados sobre tres semánticas diferentes. Por lo tanto, son necesarios tres conjuntos de etiquetas, cada uno para modelar una semántica, y además, un cuarto para expresar el RSV de los documentos.

De acuerdo con lo anterior, se han elegido cuatro conjuntos de etiquetas con diferente cardinalidad y semántica para valorar los pesos asociados con cada una de las semánticas y los RSVs. Estos conjuntos son los mismos que se utilizaron en el Sección 4.3.4, conjuntos con siete, cinco, nueve y once etiquetas. Sus representaciones se pueden ver en las Figuras 4.8, 4.9, 4.10 y 4.11, respectivamente.

3 En realidad, mostraremos el número de soluciones no dominadas que se obtengan, valor que será igual al número de soluciones diferentes respecto a los valores de los objetivos. Esta generalización es correcta, puesto que los Paretos proporcionados por el algoritmo SPEA están formados por los individuos de la población elitista que, como se comentó en la Sección 3.4, sólo contiene soluciones Pareto-optimales diferentes.

ESTRUCTURA DE LAS CONSULTAS

El SRI Lingüístico utilizado para evaluar las consultas tiene como condición que éstas estén en forma normal, bien DNF o CNF. En la Sección 5.2.1, indicamos que consideraríamos que todas las consultas estaban en una de las dos formas, pero solo en una, de forma que sólo necesitábamos codificar las subexpresiones.

De acuerdo con esto, los experimentos se han hecho considerando que todas las consultas están en CNF.

5.3.1.1.- Análisis de los Paretos obtenidos en la experimentación realizada con Cranfield

Las Tablas 5.2 a 5.18 recogen las estadísticas de los Paretos generados en las 5 ejecuciones del algoritmo SPEA-AG efectuadas sobre las 17 consultas de Cranfield. De izquierda a derecha, las columnas se corresponden con las cuatro métricas comentadas anteriormente: *N. Sol.* (Número de soluciones contenidas en el Pareto, igual al número de soluciones distintas existentes en dicho conjunto, en el espacio objetivo), M_2^* (valor medio obtenido para la métrica de distribución del Pareto en el espacio objetivo) y M_3^* (valor medio obtenido para la métrica de extensión del Pareto en el espacio objetivo). Finalmente, las dos últimas filas de cada tabla contienen las medias y las desviaciones típicas.

Posteriormente, la Tabla 5.19 resume los resultados de las diecisiete tablas anteriores, mostrando los valores medios y las desviaciones típicas en cada caso.

C. 1	*N_sol*	M_2^*	M_3^*
Ej. 1	5	2.500	1.130
Ej. 2	9	4.250	1.247
Ej. 3	9	4.375	1.268
Ej. 4	7	3.167	1.198
Ej. 5	8	3.571	1.199
Media	7.600	3.573	1.209
Desv.	0.669	0.311	0.021

Tabla 127: Estadísticas de los Paretos generados por el algoritmo SPEA-AG en la consulta 1 de Cranfield

C. 2	*N_sol*	M_2^*	M_3^*
Ej. 1	7	3.167	1.233
Ej. 2	5	2.250	1.073
Ej. 3	5	2.250	1.119
Ej. 4	8	3.714	1.256
Ej. 5	6	2.800	1.196
Media	6.200	2.836	1.175
Desv.	0.522	0.251	0.031

Tabla 128: Estadísticas de los Paretos generados por el algoritmo SPEA-AG en la consulta 2 de Cranfield

C. 3	*N_sol*	M_2^*	M_3^*
Ej. 1	1	0.000	0.000
Ej. 2	1	0.000	0.000
Ej. 3	2	1.000	0.671
Ej. 4	1	0.000	0.000
Ej. 5	1	0.000	0.000
Media	1.200	0.200	0.134
Desv.	0.179	0.179	0.120

Tabla 129: Estadísticas de los Paretos generados por el algoritmo SPEA-AG en la consulta 3 de Cranfield

C. 7	*N_sol*	M_2^*	M_3^*
Ej. 1	1	0.000	0.000
Ej. 2	2	1.000	0.764
Ej. 3	2	1.000	0.764
Ej. 4	2	1.000	1.070
Ej. 5	2	1.000	1.076
Media	1.800	0.800	0.735
Desv.	0.179	0.179	0.176

Tabla 130: Estadísticas de los Paretos generados por el algoritmo SPEA-AG en la consulta 7 de Cranfield

C. 8	*N_sol*	M_2^*	M_3^*
Ej. 1	3	1.500	1.041
Ej. 2	3	1.500	0.913
Ej. 3	3	1.500	0.934
Ej. 4	5	2.500	1.186
Ej. 5	2	1.000	0.949
Media	3.200	1.600	1.004
Desv.	0.438	0.219	0.045

Tabla 131: Estadísticas de los Paretos generados por el algoritmo SPEA-AG en la consulta 8 de Cranfield

C. 11	*N_sol*	M_2^*	M_3^*
Ej. 1	2	1.000	0.764
Ej. 2	1	0.000	0.000
Ej. 3	1	0.000	0.000
Ej. 4	2	1.000	0.671
Ej. 5	1	0.000	0.000
Media	1.400	0.400	0.287
Desv.	0.219	0.219	0.158

Tabla 132: Estadísticas de los Paretos generados por el algoritmo SPEA-AG en la consulta 11 de Cranfield

C. 19	*N_sol*	M_2^*	M_3^*
Ej. 1	2	1.000	0.606
Ej. 2	2	1.000	0.606
Ej. 3	2	1.000	0.803
Ej. 4	2	1.000	0.606
Ej. 5	1	0.000	0.000
Media	1.800	0.800	0.524
Desv.	0.179	0.179	0.122

Tabla 133: Estadísticas de los Paretos generados por el algoritmo SPEA-AG en la consulta 19 de Cranfield

C. 23	*N_sol*	M_2^*	M_3^*
Ej. 1	12	5.273	1.283
Ej. 2	9	4.125	1.272
Ej. 3	10	4.778	1.285
Ej. 4	12	5.455	1.297
Ej. 5	9	3.875	1.243
Media	10.400	4.701	1.276
Desv.	0.607	0.277	0.008

Tabla 134: Estadísticas de los Paretos generados por el algoritmo SPEA-AG en la consulta 23 de Cranfield

C. 26	*N_sol*	M_2^*	M_3^*
Ej. 1	2	1.000	0.979
Ej. 2	2	1.000	0.764
Ej. 3	2	1.000	0.764
Ej. 4	1	0.000	0.000
Ej. 5	1	0.000	0.000
Media	1.600	0.600	0.501
Desv.	0.219	0.219	0.186

Tabla 135: Estadísticas de los Paretos generados por el algoritmo SPEA-AG en la consulta 26 de Cranfield

C. 38	*N_sol*	M_2^*	M_3^*
Ej. 1	3	1.500	0.828
Ej. 2	2	1.000	0.606
Ej. 3	2	1.000	0.903
Ej. 4	4	2.000	1.212
Ej. 5	2	1.000	0.986
Media	2.600	1.300	0.907
Desv.	0.358	0.179	0.089

Tabla 136: Estadísticas de los Paretos generados por el algoritmo SPEA-AG en la consulta 38 de Cranfield

C. 39	*N_sol*	M_2^*	M_3^*
Ej. 1	4	2.000	1.108
Ej. 2	4	2.000	1.120
Ej. 3	4	2.000	1.174
Ej. 4	4	2.000	1.108
Ej. 5	2	1.000	0.802
Media	3.600	1.800	1.062
Desv.	0.358	0.179	0.059

Tabla 137: Estadísticas de los Paretos generados por el algoritmo SPEA-AG en la consulta 39 de Cranfield

C. 40	*N_sol*	M_2^*	M_3^*
Ej. 1	2	1.000	0.645
Ej. 2	3	1.500	1.098
Ej. 3	4	2.000	1.127
Ej. 4	3	1.500	1.078
Ej. 5	4	2.000	1.163
Media	3.200	1.600	1.022
Desv.	0.335	0.167	0.085

Tabla 138: Estadísticas de los Paretos generados por el algoritmo SPEA-AG en la consulta 40 de Cranfield

C. 47	*N_sol*	M_2^*	M_3^*
Ej. 1	5	2.500	1.016
Ej. 2	5	2.500	1.246
Ej. 3	3	1.500	0.934
Ej. 4	4	2.000	1.030
Ej. 5	4	2.000	1.097
Media	4.200	2.100	1.064
Desv.	0.335	0.167	0.047

Tabla 139: Estadísticas de los Paretos generados por el algoritmo SPEA-AG en la consulta 47 de Cranfield

C. 73	*N_sol*	M_2^*	M_3^*
Ej. 1	7	3.500	1.233
Ej. 2	6	3.000	1.187
Ej. 3	4	2.000	1.069
Ej. 4	5	2.500	0.982
Ej. 5	7	3.500	1.211
Media	5.800	2.900	1.137
Desv.	0.522	0.261	0.043

Tabla 140: Estadísticas de los Paretos generados por el algoritmo SPEA-AG en la consulta 73 de Cranfield

C. 157	*N_sol*	M_2^*	M_3^*
Ej. 1	11	5.000	1.171
Ej. 2	14	6.385	1.245
Ej. 3	14	6.308	1.180
Ej. 4	12	5.545	1.276
Ej. 5	10	4.667	1.275
Media	12.200	5.581	1.229
Desv.	0.716	0.307	0.020

Tabla 141: Estadísticas de los Paretos generados por el algoritmo SPEA-AG en la consulta 157 de Cranfield

C. 220	*N_sol*	M_2^*	M_3^*
Ej. 1	5	2.500	1.191
Ej. 2	5	2.500	1.206
Ej. 3	4	2.000	1.047
Ej. 4	6	3.000	1.022
Ej. 5	5	2.500	1.166
Media	5.000	2.500	1.127
Desv.	0.283	0.141	0.034

Tabla 142: Estadísticas de los Paretos generados por el algoritmo SPEA-AG en la consulta 220 de Cranfield

C. 225	*N_sol*	M_2^*	M_3^*
Ej. 1	8	3.571	1.205
Ej. 2	8	3.571	1.227
Ej. 3	6	2.600	1.170
Ej. 4	5	2.500	1.153
Ej. 5	8	3.571	1.252
Media	7.000	3.163	1.201
Desv.	0.566	0.224	0.016

Tabla 143: Estadísticas de los Paretos generados por el algoritmo SPEA-AG en la consulta 225 de Cranfield

	N_sol		***M_2^****		***M_3^****	
	Media	***Desviación***	***Media***	***Desviación***	***Media***	***Desviación***
C. 1	7.600	0.669	3.573	0.311	1.209	0.021
C. 2	6.200	0.522	2.836	0.251	1.175	0.031
C. 3	1.200	0.179	0.200	0.179	0.134	0.120
C. 7	1.800	0.179	0.800	0.179	0.735	0.176
C. 8	3.200	0.438	1.600	0.219	1.004	0.045
C. 11	1.400	0.219	0.400	0.219	0.287	0.158
C. 19	1.800	0.179	0.800	0.179	0.524	0.122
C. 23	10.400	0.607	4.701	0.277	**1.276**	0.008
C. 26	1.600	0.219	0.600	0.219	0.501	0.186
C. 38	2.600	0.358	1.300	0.179	0.907	0.089
C. 39	3.600	0.358	1.800	0.179	1.062	0.059
C. 40	3.200	0.335	1.600	0.167	1.022	0.085
C. 47	4.200	0.335	2.100	0.167	1.064	0.047
C. 73	5.800	0.522	2.900	0.261	1.137	0.043
C. 157	**12.200**	0.716	**5.581**	0.307	1.229	0.020
C. 220	5.000	0.283	2.500	0.141	1.127	0.034
C. 225	7.000	0.566	3.163	0.224	1.201	0.016

Tabla 144: Estadísticas de los Paretos generados por el algoritmo SPEA-AG en las consultas de Cranfield

Análisis de resultados

Podemos comenzar destacando el hecho de que, al igual que en Capítulo 3, se cumple el principal objetivo de nuestra propuesta, obtener varias consultas persistentes con diferente balance precisión-exhaustividad en una única ejecución, ya que los frentes de los Paretos obtenidos están bastante bien distribuidos como demuestran los altos valores de las métricas M_2^* y M_3^*.

La mayoría de las consultas generan un número de consultas persistentes con diferente balance de precisión-exhaustividad proporcional al número de documentos relevantes que tienen asociadas (aquellos casos en que se proporciona un mayor número de documentos relevantes, un mayor número de consultas persistentes forman el frente del Pareto). Entre las consultas con menor número de documentos relevantes, hay algunas que sólo consiguen

derivar una consulta persistente (p.e. las consultas 3 y 11), lo que se traduce en un Pareto que cubre un único punto del espacio en vez de una zona del mismo. Lógicamente, esta única solución se localiza en la esquina superior derecha del espacio de búsqueda, es decir, presenta una precisión y una exhaustividad de 1 (Figura 5.9), con lo que no se pueden encontrar más soluciones no dominadas al satisfacerse plenamente los dos objetivos.

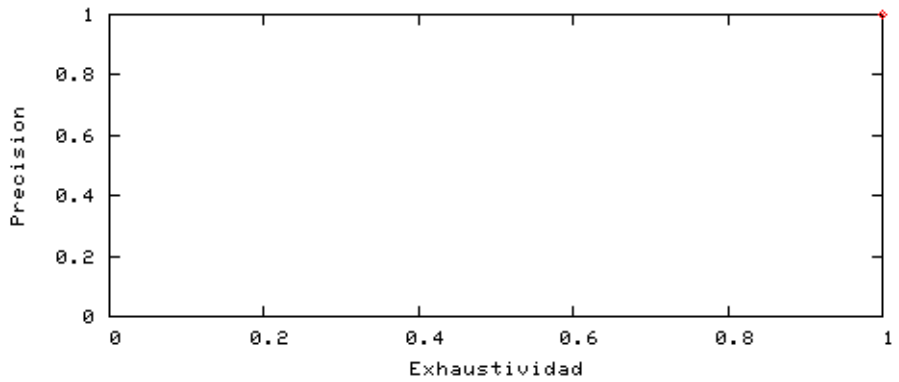


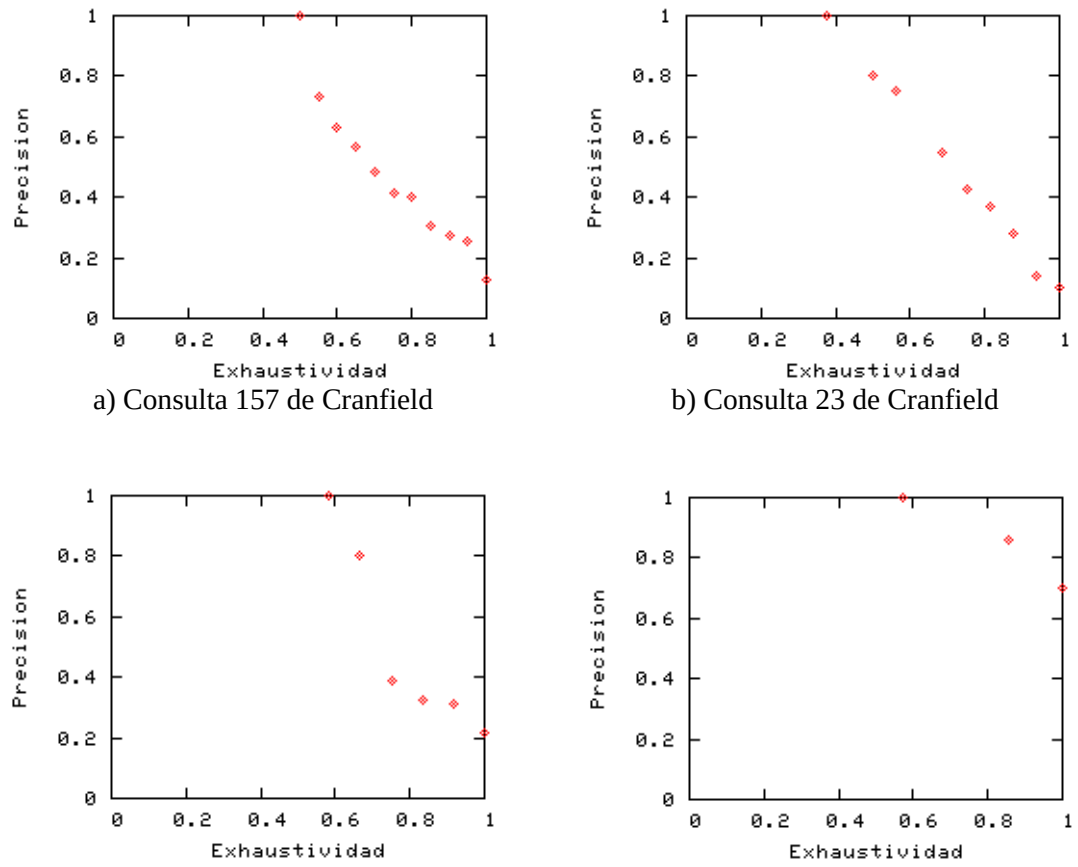
Figura 5.9: Frente de Pareto con una solución que satisface plenamente los dos objetivos

Los valores de las métricas M_2^* y M_3^* son también muy apropiados, destacando especialmente los de esta última, bastante cercanos a 1.4142, el máximo valor posible. Esto nos muestra como los frentes de los Paretos generados cubren una zona amplia del espacio. Obviamente, estas métricas no tienen sentido cuando el frente del Pareto está formado por una única solución, puesto que ésta no se puede comparar con ninguna otra, obteniéndose valores de 0 en ambas métricas para este tipo de consultas.

Por otro lado, las desviaciones típicas se encuentran alrededor de 0.5, 0.2 y 0.05 para el número de soluciones diferentes, M_2^* y M_3^*, respectivamente, lo que indica un comportamiento homogéneo del algoritmo.

Haciendo un análisis algo más pormenorizado, la consulta 157 (Tabla 141) es la que obtiene la media más alta en cuanto a soluciones distintas presentes en el frente del Pareto, alrededor de 12. Además, es esta misma consulta la que obtiene una mejor media en cuanto a la distribución de las soluciones en el frente del Pareto con un 5.581, junto con un 1.229 para M_3^*, muy cercano al máximo de la distancia euclídea (1.4142). Por su parte, la consulta 23 (Tabla 134) es la que obtiene la mejor media respecto a la extensión del frente no dominado con un total de 1.276 junto con la desviación típica más pequeña, tan sólo 0.008.

Como ejemplo, la Figura 5.10 muestra los frentes de los Paretos obtenidos para las consultas 157, 23, 225 y 47, compuestos, respectivamente, por 11, 9, 6 y 3 consultas persistentes. Recordemos que el eje X representa los valores de exhaustividad y el eje Y los de precisión. Como se hizo en [192], los Paretos obtenidos por las cinco ejecuciones realizadas para cada consulta se han fusionado y las soluciones dominadas se han eliminado del conjunto unificado antes de imprimir las curvas.



a) Consulta 157 de Cranfield

b) Consulta 23 de Cranfield

c) Consulta 225 de Cranfield

d) Consulta 47 de Cranfield

Figura 5.10: Frentes de los Paretos derivados por el algoritmo SPEA-AG sobre las consultas 157, 23, 225 y 47 de Cranfield

5.3.1.2.- Análisis de los Paretos obtenidos en la experimentación realizada con CACM

Las Tablas 5.20 a 5.37 recogen las estadísticas de los Paretos generados en las 5 ejecuciones del algoritmo SPEA-AG efectuadas sobre las 18 consultas de CACM. Por otro lado, la Tabla 5.38 recoge las estadísticas en media de los Paretos generados para CACM.

C. 4	***N_sol***	***M_2^****	***M_3^****
Ej. 1	4	2.000	1.155
Ej. 2	3	1.500	0.888
Ej. 3	2	1.000	0.645
Ej. 4	1	0.000	0.000
Ej. 5	2	1.000	0.902
Media	2.400	1.100	0.718
Desv.	0.456	0.297	0.176

Tabla 145: Estadísticas de los Paretos generados por el algoritmo SPEA-AG en la consulta 4 de CACM

C. 7	***N_sol***	***M_2^****	***M_3^****
Ej. 1	6	2.600	1.123
Ej. 2	6	3.000	1.141
Ej. 3	6	2.800	1.139
Ej. 4	6	2.800	1.171
Ej. 5	6	3.000	1.133
Media	6	2.840	1.141
Desv.	0.000	0.067	0.007

Tabla 146: Estadísticas de los Paretos generados por el algoritmo SPEA-AG en la consulta 7 de CACM

C. 9	***N_sol***	***M_2^****	***M_3^****
Ej. 1	2	1.000	0.671
Ej. 2	1	0.000	0.000
Ej. 3	2	1.000	0.764
Ej. 4	1	0.000	0.000
Ej. 5	1	0.000	0.000
Media	1.400	0.400	0.287
Desv.	0.219	0.219	0.158

Tabla 147: Estadísticas de los Paretos generados por el algoritmo SPEA-AG en la consulta 9 de CACM

C. 10	***N_sol***	***M_2^****	***M_3^****
Ej. 1	6	3.000	1.178
Ej. 2	5	2.250	1.088
Ej. 3	8	3.571	1.065
Ej. 4	7	3.333	1.256
Ej. 5	9	4.125	1.210
Media	7.000	3.256	1.160
Desv.	0.632	0.278	0.032

Tabla 148: Estadísticas de los Paretos generados por el algoritmo SPEA-AG en la consulta 10 de CACM

C. 14	*N_sol*	M_2^*	M_3^*
Ej. 1	6	2.800	0.991
Ej. 2	7	3.333	1.142
Ej. 3	8	3.429	1.054
Ej. 4	6	2.600	0.840
Ej. 5	8	3.714	1.079
Media	7.000	3.175	1.021
Desv.	0.400	0.185	0.046

Tabla 149: Estadísticas de los Paretos generados por el algoritmo SPEA-AG en la consulta 14 de CACM

C. 19	*N_sol*	M_2^*	M_3^*
Ej. 1	2	1.000	0.753
Ej. 2	2	1.000	0.942
Ej. 3	1	0.000	0.000
Ej. 4	2	1.000	0.606
Ej. 5	4	2.000	1.212
Media	2.200	1.000	0.702
Desv.	0.438	0.283	0.181

Tabla 150: Estadísticas de los Paretos generados por el algoritmo SPEA-AG en la consulta 19 de CACM

C. 24	*N_sol*	M_2^*	M_3^*
Ej. 1	2	1.000	0.690
Ej. 2	2	1.000	0.707
Ej. 3	2	1.000	0.788
Ej. 4	2	1.000	0.900
Ej. 5	3	1.500	1.009
Media	2.200	1.100	0.819
Desv.	0.179	0.089	0.054

Tabla 151: Estadísticas de los Paretos generados por el algoritmo SPEA-AG en la consulta 24 de CACM

C. 25	*N_sol*	M_2^*	M_3^*
Ej. 1	16	7.067	1.293
Ej. 2	16	7.067	1.267
Ej. 3	11	4.900	1.209
Ej. 4	17	7.500	1.285
Ej. 5	16	7.133	1.190
Media	15.200	6.733	1.249
Desv.	0.955	0.416	0.019

Tabla 152: Estadísticas de los Paretos generados por el algoritmo SPEA-AG en la consulta 25 de CACM

C. 26	*N_sol*	M_2^*	M_3^*
Ej. 1	4	1.667	0.886
Ej. 2	7	3.333	1.068
Ej. 3	4	2.500	1.030
Ej. 4	6	2.800	1.129
Ej. 5	6	2.800	1.153
Media	5.400	2.520	1.053
Desv.	0.537	0.270	0.042

Tabla 153: Estadísticas de los Paretos generados por el algoritmo SPEA-AG en la consulta 26 de CACM

C. 27	*N_sol*	M_2^*	M_3^*
Ej. 1	8	3.857	1.138
Ej. 2	7	3.333	1.161
Ej. 3	5	2.500	1.076
Ej. 4	8	3.571	1.212
Ej. 5	8	3.714	1.183
Media	7.200	3.395	1.154
Desv.	0.522	0.215	0.021

Tabla 154: Estadísticas de los Paretos generados por el algoritmo SPEA-AG en la consulta 27 de CACM

C. 40	*N_sol*	M_2^*	M_3^*
Ej. 1	3	1.500	0.880
Ej. 2	3	1.500	0.992
Ej. 3	2	1.000	0.903
Ej. 4	2	1.000	1.007
Ej. 5	2	1.000	1.014
Media	2.400	1.200	0.959
Desv.	0.219	0.110	0.025

Tabla 155: Estadísticas de los Paretos generados por el algoritmo SPEA-AG en la consulta 40 de CACM

C. 42	*N_sol*	M_2^*	M_3^*
Ej. 1	5	2.500	1.173
Ej. 2	6	3.000	1.152
Ej. 3	4	2.000	1.007
Ej. 4	5	2.500	1.173
Ej. 5	5	2.500	1.208
Media	5.000	2.500	1.143
Desv.	0.283	0.141	0.031

Tabla 156: Estadísticas de los Paretos generados por el algoritmo SPEA-AG en la consulta 42 de CACM

C. 43	*N_sol*	M_2^*	M_3^*
Ej. 1	10	4.444	1.249
Ej. 2	12	5.364	1.188
Ej. 3	10	4.444	1.207
Ej. 4	9	4.000	1.230
Ej. 5	12	5.545	1.265
Media	10.600	4.760	1.228
Desv.	0.537	0.265	0.012

Tabla 157: Estadísticas de los Paretos generados por el algoritmo SPEA-AG en la consulta 43 de CACM

C. 45	*N_sol*	M_2^*	M_3^*
Ej. 1	6	2.800	1.087
Ej. 2	5	2.500	1.149
Ej. 3	7	3.000	1.172
Ej. 4	4	1.667	1.004
Ej. 5	9	4.125	1.251
Media	6.200	2.818	1.133
Desv.	0.769	0.356	0.037

Tabla 158: Estadísticas de los Paretos generados por el algoritmo SPEA-AG en la consulta 45 de CACM

C. 58	*N_sol*	M_2^*	M_3^*
Ej. 1	5	2.500	1.107
Ej. 2	4	2.000	1.125
Ej. 3	6	2.800	1.173
Ej. 4	4	2.000	1.117
Ej. 5	5	2.500	1.111
Media	4.800	2.360	1.127
Desv.	0.335	0.140	0.011

Tabla 159: Estadísticas de los Paretos generados por el algoritmo SPEA-AG en la consulta 58 de CACM

C. 59	*N_sol*	M_2^*	M_3^*
Ej. 1	8	3.714	1.145
Ej. 2	8	3.571	1.105
Ej. 3	10	4.778	1.193
Ej. 4	10	4.444	1.171
Ej. 5	7	3.333	1.066
Media	8.600	3.968	1.136
Desv.	0.537	0.246	0.020

Tabla 160: Estadísticas de los Paretos generados por el algoritmo SPEA-AG en la consulta 59 de CACM

C. 60	*N_sol*	M_2^*	M_3^*
Ej. 1	7	3.333	1.127
Ej. 2	6	3.000	1.156
Ej. 3	5	2.500	1.209
Ej. 4	9	4.250	1.251
Ej. 5	6	3.000	1.151
Media	6.600	3.217	1.179
Desv.	0.607	0.260	0.020

Tabla 161: Estadísticas de los Paretos generados por el algoritmo SPEA-AG en la consulta 60 de CACM

C. 61	*N_sol*	M_2^*	M_3^*
Ej. 1	9	4.000	1.179
Ej. 2	6	2.600	1.120
Ej. 3	6	2.800	1.275
Ej. 4	9	3.857	1.115
Ej. 5	6	2.600	1.032
Media	7.200	3.175	1.144
Desv.	0.657	0.281	0.036

Tabla 162: Estadísticas de los Paretos generados por el algoritmo SPEA-AG en la consulta 61 de CACM

	N_sol		M_2^*		M_3^*	
	Media	***Desviación***	***Media***	***Desviación***	***Media***	***Desviación***
C. 4	2.400	0.456	1.100	0.297	0.718	0.176
C. 7	6.000	0.000	2.840	0.067	1.141	0.007
C. 9	1.400	0.219	0.400	0.219	0.287	0.158
C. 10	7.000	0.632	3.256	0.278	1.160	0.032
C. 14	7.000	0.400	3.175	0.185	1.021	0.046
C. 19	2.200	0.438	1.000	0.283	0.702	0.181
C. 24	2.200	0.179	1.100	0.089	0.819	0.054
C. 25	**15.200**	0.955	**6.733**	0.416	**1.249**	0.019
C. 26	5.400	0.537	2.520	0.270	1.053	0.042
C. 27	7.200	0.522	3.395	0.215	1.154	0.021
C. 40	2.400	0.219	1.200	0.110	0.959	0.025
C. 42	5.000	0.283	2.500	0.141	1.143	0.031
C. 43	10.600	0.537	4.760	0.265	1.228	0.012
C. 45	6.200	0.769	2.818	0.356	1.133	0.037
C. 58	4.800	0.335	2.360	0.140	1.127	0.011
C. 59	8.600	0.537	3.968	0.246	1.136	0.020
C. 60	6.600	0.607	3.217	0.260	1.179	0.020
C. 61	7.200	0.657	3.175	0.281	1.144	0.036

Tabla 163: Estadísticas de los Paretos generados por el algoritmo SPEA-AG en las consultas de CACM

Análisis de resultados

Las conclusiones son similares al caso anterior en lo que respecta a la calidad de los Paretos. De hecho, los números obtenidos son bastante parecidos a los de la experimentación de Cranfield, lo que muestra la robustez del algoritmo.

Las desviaciones típicas se encuentran alrededor de 0.5, 0.25 y 0.05 para el número de soluciones diferentes, M_2^* y M_3^*, respectivamente, lo que indica un comportamiento homogéneo del algoritmo.

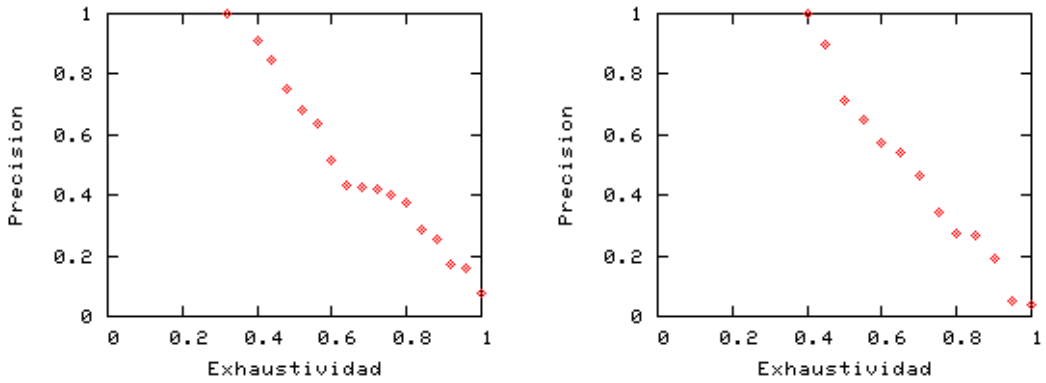
Como podemos observar en la Tabla 152, la consulta 25 acapara los mejores valores de las métricas. Así, presenta la media de Paretos con mayor número de soluciones no dominadas con un total de 15.200, el valor más alto en la distribución de soluciones en el frente del Pareto (6.733) y la mejor media respecto a la extensión del frente no dominado con un total 1.249, cercano al resultado óptimo deseado (1.4142).

La Figura 5.11 muestra los frentes del Pareto generados para las consultas 25, 43, 60 y 45, con 17, 13, 7, y 5 consultas persistentes, respectivamente. El eje X representa los valores de exhaustividad y el eje Y los de precisión. Al igual que en la experimentación de Cranfield, los Paretos obtenidos por las cinco ejecuciones realizadas para cada consulta se han fusionado y, antes de imprimir las curvas, las soluciones dominadas se han eliminado de dicho conjunto.

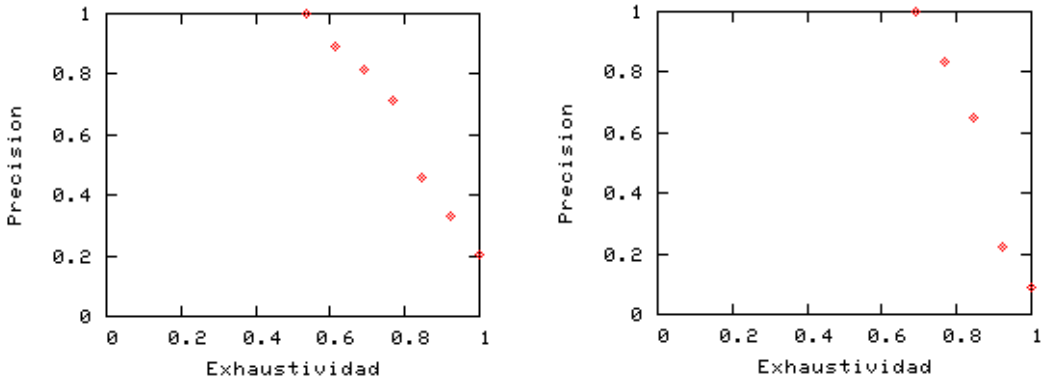
Número de soluciones en los frentes de los Paretos derivados

Igual que ocurría con los Paretos generados por el algoritmo SPEA-PG del Capítulo 3, el número de soluciones con diferente balance entre precisión y exhaustividad presentes en los frentes de los Paretos generados por el algoritmo SPEA-AG es pequeño en comparación con el tamaño de la población elitista. Este comportamiento se debe, como se comentó en la Sección 3.5.2.2, a que consideramos que dos soluciones son iguales si lo son en el espacio de objetivos, independientemente de la estructura que tengan.

Aunque se probó a realizar la comparación en el espacio de decisiones utilizando la distancia de edición [123], como medida de comparación entre árboles de expresión, el tiempo de ejecución se vio incrementado considerablemente, por lo que decidimos mantener la opción original, igual que en el Capítulo 3, dejando para trabajos futuros la búsqueda de nuevas funciones de similitud entre árboles de expresión.



a) Consulta 25 de CACM b) Consulta 43 de CACM



c) Consulta 60 de CACM d) Consulta 45 de CACM

Figura 5.11: Frentes de los Paretos derivados por el algoritmo SPEA-AG sobre las consultas 25, 43, 60 y 45 de CACM

5.3.1.3.- Análisis de la eficacia de las consultas representativas del Pareto de acuerdo a los criterios de precisión y exhaustividad en la experimentación realizada con Cranfield

Las Tablas 5.39 a 5.55 muestran la eficacia de la recuperación de las consultas seleccionadas del Pareto fusionado obtenido tras combinar los Paretos asociados a cada una de las 5 ejecuciones realizadas, para cada una de las consultas de Cranfield. La eficacia se mide de acuerdo a los dos criterios clásicos de recuperación, precisión y exhaustividad, utilizando un umbral para determinar qué conjunto de documentos es el recuperado.

Dentro de cada tabla, y de izquierda a derecha, las columnas recogen los datos siguientes:

i) Número de consulta seleccionada.

ii) Valores correspondientes a la evaluación de las consultas persistentes sobre el conjunto de entrenamiento: Número de nodos en el árbol que representa la consulta, Precisión, Exhaustividad, Número de documentos relevantes recuperados y Número total de documentos recuperados.

iii) Valores correspondientes a la evaluación de las consultas persistentes sobre el conjunto de prueba: Número de nodos en el árbol que representa la consulta, Precisión, Exhaustividad, Número de documentos relevantes recuperados y Número total de documentos recuperados.

C. 1	***ENTRENAMIENTO***					***PRUEBA***				
	Nodos	***P***	***E***	***#rel***	***#rec***	***Nodos***	***P***	***E***	***#rel***	***#rec***
1	14	0.087	1.000	14	161	11	0.040	0.467	7	174
2	15	0.355	0.786	11	31	15	0.071	0.133	2	28
3	15	0.562	0.643	9	16	7	0.000	0.000	0	7
4	15	0.889	0.571	8	9	7	0.000	0.000	0	8
5	15	1.000	0.500	7	8	7	0.000	0.000	0	7

Tabla 164: Eficacia de la recuperación en base a los criterios de precisión y exhaustividad para la consulta 1 de Cranfield

C. 2	***ENTRENAMIENTO***					***PRUEBA***				
	Nodos	***P***	***E***	***#rel***	***#rec***	***Nodos***	***P***	***E***	***#rel***	***#rec***
1	14	0.524	0.917	11	21	13	0.333	0.308	4	12
2	14	0.789	0.833	10	13	11	0.375	0.231	3	8
3	16	0.182	1.000	12	66	13	0.117	0.538	7	60
4	15	0.900	0.750	9	10	11	0.353	0.462	6	17
5	15	1.000	0.667	8	8	11	0.417	0.385	5	12

Tabla 165: Eficacia de la recuperación en base a los criterios de precisión y exhaustividad para la consulta 2 de Cranfield

C. 3	*ENTRENAMIENTO*					*PRUEBA*				
	Nodos	*P*	*E*	*#rel*	*#rec*	*Nodos*	*P*	*E*	*#rel*	*#rec*
1	13	1.000	1.000	4	4	0	0.000	0.000	0	0
2	15	1.000	1.000	4	4	0	0.000	0.000	0	0
3	14	1.000	1.000	4	4	10	0.714	1.000	5	7
4	15	1.000	1.000	4	4	8	0.000	0.000	0	4

Tabla 166: Eficacia de la recuperación en base a los criterios de precisión y exhaustividad para la consulta 3 de Cranfield

C. 7	*ENTRENAMIENTO*					*PRUEBA*				
	Nodos	*P*	*E*	*#rel*	*#rec*	*Nodos*	*P*	*E*	*#rel*	*#rec*
1	14	1.000	1.000	3	3	0	0.000	0.000	0	0

Tabla 167: Eficacia de la recuperación en base a los criterios de precisión y exhaustividad para la consulta 7 de Cranfield

C. 8	*ENTRENAMIENTO*					*PRUEBA*				
	Nodos	*P*	*E*	*#rel*	*#rec*	*Nodos*	*P*	*E*	*#rel*	*#rec*
1	12	0.600	1.000	6	10	0	0.000	0.000	0	0
2	15	0.833	0.833	5	6	10	0.000	0.000	0	0
3	14	1.000	0.667	4	4	12	0.000	0.000	0	0
4	12	1.000	0.667	4	4	12	0.000	0.000	0	0
5	14	1.000	0.667	4	4	13	0.077	0.333	2	26

Tabla 168: Eficacia de la recuperación en base a los criterios de precisión y exhaustividad para la consulta 8 de Cranfield

C. 11	*ENTRENAMIENTO*					*PRUEBA*				
	Nodos	*P*	*E*	*#rel*	*#rec*	*Nodos*	*P*	*E*	*#rel*	*#rec*
1	13	1.000	1.000	4	4	12	0.000	0.000	0	0
2	15	1.000	1.000	4	4	0	0.000	0.000	0	0
3	12	1.000	1.000	4	4	7	0.000	0.000	0	2

Tabla 169: Eficacia de la recuperación en base a los criterios de precisión y exhaustividad para la consulta 11 de Cranfield

C. 19	*ENTRENAMIENTO*					*PRUEBA*				
	Nodos	*P*	*E*	*#rel*	*#rec*	*Nodos*	*P*	*E*	*#rel*	*#rec*
1	16	1.000	1.000	5	5	10	0.250	0.200	1	4

Tabla 170: Eficacia de la recuperación en base a los criterios de precisión y exhaustividad para la consulta 19 de Cranfield

C.23	*ENTRENAMIENTO*					*PRUEBA*				
	Nodos	*P*	*E*	*#rel*	*#rec*	*Nodos*	*P*	*E*	*#rel*	*#rec*
1	12	0.142	0.938	15	106	10	0.085	0.941	16	188
2	15	0.371	0.812	13	35	12	0.053	0.059	1	19
3	14	0.550	0.688	11	20	14	0.059	0.059	1	17
4	14	0.750	0.562	9	12	13	0.429	0.176	3	7
5	14	1.000	0.375	6	6	13	0.333	0.118	2	6

Tabla 171: Eficacia de la recuperación en base a los criterios de precisión y exhaustividad para la consulta 23 de Cranfield

C. 26	*ENTRENAMIENTO*					*PRUEBA*				
	Nodos	***P***	***E***	***#rel***	***#rec***	***Nodos***	***P***	***E***	***#rel***	***#rec***
1	14	1.000	1.000	3	3	11	0.000	0.000	0	8
2	15	1.000	1.000	3	3	11	0.000	0.000	0	6

Tabla 172: Eficacia de la recuperación en base a los criterios de precisión y exhaustividad para la consulta 26 de Cranfield

C. 38	*ENTRENAMIENTO*					*PRUEBA*				
	Nodos	***P***	***E***	***#rel***	***#rec***	***Nodos***	***P***	***E***	***#rel***	***#rec***
1	14	1.000	0.800	4	4	0	0.000	0.000	0	0
2	14	1.000	0.800	4	4	13	0.083	0.167	1	12
3	13	0.833	1.000	5	6	11	0.000	0.000	0	3
4	14	1.000	0.800	4	4	10	0.000	0.000	0	2

Tabla 173: Eficacia de la recuperación en base a los criterios de precisión y exhaustividad para la consulta 38 de Cranfield

C. 39	*ENTRENAMIENTO*					*PRUEBA*				
	Nodos	***P***	***E***	***#rel***	***#rec***	***Nodos***	***P***	***E***	***#rel***	***#rec***
1	15	0.500	1.000	7	14	10	0.125	0.143	1	8
2	16	1.000	0.857	6	6	13	0.100	0.143	1	10

Tabla 174: Eficacia de la recuperación en base a los criterios de precisión y exhaustividad para la consulta 39 de Cranfield

C. 40	*ENTRENAMIENTO*					*PRUEBA*				
	Nodos	***P***	***E***	***#rel***	***#rec***	***Nodos***	***P***	***E***	***#rel***	***#rec***
1	15	0.750	1.000	6	8	9	0.000	0.000	0	5
2	16	1.000	0.833	5	5	7	0.000	0.000	0	2

Tabla 175: Eficacia de la recuperación en base a los criterios de precisión y exhaustividad para la consulta 40 de Cranfield

C. 47	ENTRENAMIENTO					PRUEBA				
	Nodos	P	E	#rel	#rec	Nodos	P	E	#rel	#rec
1	16	1.000	0.571	4	4	7	0.000	0.000	0	1
2	14	1.000	0.571	4	4	9	0.000	0.000	0	2
3	13	0.700	1.000	7	10	13	0.143	0.250	2	14
4	12	0.857	0.857	6	7	12	0.143	0.250	2	14

Tabla 176: Eficacia de la recuperación en base a los criterios de precisión y exhaustividad para la consulta 47 de Cranfield

C. 73	ENTRENAMIENTO					PRUEBA				
	Nodos	P	E	#rel	#rec	Nodos	P	E	#rel	#rec
1	15	0.435	1.000	10	23	15	0.042	0.091	1	24
2	15	0.529	0.900	9	17	15	0.067	0.091	1	15
3	14	0.889	0.800	8	9	10	0.100	0.091	1	10
4	13	1.000	0.700	7	7	12	0.111	0.091	1	9

Tabla 177: Eficacia de la recuperación en base a los criterios de precisión y exhaustividad para la consulta 73 de Cranfield

C. 157	ENTRENAMIENTO					PRUEBA				
	Nodos	P	E	#rel	#rec	Nodos	P	E	#rel	#rec
1	15	0.130	1.000	20	154	14	0.075	0.650	13	173
2	14	0.304	0.850	17	56	11	0.113	0.350	7	62
3	15	0.565	0.650	13	23	10	0.429	0.600	12	28
4	14	0.733	0.550	10	15	11	0.333	0.150	3	9
5	14	1.000	0.500	10	10	11	0.000	0.000	0	4

Tabla 178: Eficacia de la recuperación en base a los criterios de precisión y exhaustividad para la consulta 157 de Cranfield

C. 220	*ENTRENAMIENTO*					*PRUEBA*				
	Nodos	***P***	***E***	***#rel***	***#rec***	***Nodos***	***P***	***E***	***#rel***	***#rec***
1	16	0.455	1.000	10	22	16	0.174	0.400	4	23
2	13	1.000	0.600	6	6	12	0.000	0.000	0	3
3	14	0.750	0.900	9	12	14	0.000	0.000	0	12
4	14	0.800	0.800	8	10	13	0.000	0.000	0	13
5	14	0.875	0.700	7	8	13	0.000	0.000	0	11

Tabla 179: Eficacia de la recuperación en base a los criterios de precisión y exhaustividad para la consulta 220 de Cranfield

C. 225	*ENTRENAMIENTO*					*PRUEBA*				
	Nodos	***P***	***E***	***#rel***	***#rec***	***Nodos***	***P***	***E***	***#rel***	***#rec***
1	14	0.214	1.000	12	56	14	0.054	0.231	3	56
2	14	0.391	0.750	9	23	14	0.154	0.308	4	26
3	14	0.323	0.833	10	31	14	0.138	0.308	4	29
4	16	0.800	0.667	8	10	0	0.000	0.000	0	0
5	14	1.000	0.583	7	7	8	1.000	0.077	1	1

Tabla 180: Eficacia de la recuperación en base a los criterios de precisión y exhaustividad para la consulta 225 de Cranfield

Análisis de resultados

A continuación, analizaremos los resultados presentados en las tablas anteriores. En primer lugar, comentaremos los resultados de cada grupo de consultas (consultas con más de 20 documentos relevantes y consultas que presentan menos de 15 documentos relevantes, Tabla 5.56), de acuerdo a la efectividad de la recuperación. Seguidamente, analizaremos el tamaño de las mejores consultas persistentes aprendidas independientemente del número de documentos relevantes que tenga asociado la consulta.

Nº de documentos relevantes	*Consultas*
más de 20	1, 2, 23, 73, 157, 220, 225
menos de 15	3, 7, 8, 11, 19, 26, 38, 39, 40, 47

Tabla 181: Grupos de consultas de Cranfield

Eficacia de las consultas con más de 20 documentos relevantes

Al unificar los Paretos obtenidos en cada una de las ejecuciones correspondientes a una consulta y seleccionar un conjunto representativo, de acuerdo al procedimiento descrito en la Sección 3.5.2, observamos como dicho conjunto se compone de diversas consultas persistentes con diferente equilibrio precisión-exhaustividad, objetivo de esta propuesta.

La diversidad de soluciones se mueve entre consultas persistentes que son capaces de recuperar únicamente documentos relevantes, alrededor de la mitad; y consultas persistentes que representan perfectamente las necesidades del usuario y recuperan todos documentos relevantes, aunque junto con basura (documentos irrelevantes). Los valores de exhaustividad se encuentran bastante por encima de 0.5 en la mayoría de los casos, mientras que los de precisión se mueven en el rango [0,1].

Cuantos más documentos relevantes se recuperan, también se recuperan más irrelevantes, siendo este comportamiento más acusado cuando se recupera el total de relevantes. Por tanto, es preferible recuperar menos documentos relevantes, ya que irán acompañados de menor ruido, consiguiéndose un mayor equilibrio entre ambos criterios. Por ejemplo, en la consulta 2 (véase la Tabla 165), la tercera consulta persistente recupera todos los documentos relevantes (12) pero junto con 54 irrelevantes, lo que se traduce en valores de precisión y exhaustividad de 0.182 y 1.00, respectivamente. Por el contrario, la segunda consulta persistente elegida recupera un total de 13 documentos, 10 relevantes y 3 irrelevantes, con lo que pierde exhaustividad (0.83) pero gana precisión (0.789).

La eficacia de estas consultas a la hora de recuperar nuevos documentos que se adapten a las necesidades de un usuario (evaluación sobre el conjunto de prueba) es dispar. Por ejemplo, en las consultas 1 y 220, solo dos y una, respectivamente, de las cinco soluciones seleccionadas son capaces de recuperar algún documento relevante. Además, en el caso de la consulta 1esta recuperación viene acompañada de ruido, lo que dificulta la asimilación de nueva información por parte del usuario. El resto de las consultas (2, 23, 73, 157, 225), sí

recuperan documentos relevantes, con una cantidad de ruido más o menos aceptable en la mayoría de los casos.

Eficacia de las consultas con menos de 15 documentos relevantes

Como ya vimos en la Sección 5.3.1.1 al analizar los Paretos obtenidos por consultas con pocos documentos relevantes asociados, los frentes de dichos Paretos estaban formados por un número muy reducido de soluciones (una o dos) diferentes en el espacio objetivo, en su mayoría soluciones con una precisión y exhaustividad iguales a 1. En consecuencia, al fusionar los Paretos, parece normal que todas las soluciones incluidas en él presenten dichos valores, independientemente de que tengan o no la misma estructura. De ahí, que casi la totalidad de las consultas persistentes con menos de 15 documentos relevantes mostradas en las tablas anteriores tengan valores máximos de precisión y exhaustividad, y las que no, no bajen de 0.5 en ninguno de los dos criterios. Además, en algunas consultas, el número de soluciones diferentes, no sólo en el espacio de objetivos sino también en el espacio de decisiones, es inferior a cinco, lo que no permite seleccionar un conjunto completo de cinco soluciones.

En contraste con lo anterior, el comportamiento de estas consultas deja bastante que desear al ser evaluadas sobre el conjunto de prueba. La mayoría de las consultas persistentes escogidas no consiguen recuperar ningún documento relevante[4] y las pocas que lo hacen, no consiguen más de 1 o 2, lo que no proporciona mucha información nueva. Cabe destacar la tercera solución de la consulta 3 (Tabla 166), que recupera todos los nuevos documentos relevantes con tan solo un añadido de 2 irrelevantes.

Mejor consulta

La consulta que presenta mejor variedad de soluciones con diferente balance precisión-exhaustividad es la 2 (Tabla 165), al recuperar una media de 5 documentos relevantes con una desviación típica de 1.41, mientras que los valores de precisión están por encima de 0.3.

Sin embargo, la mejor consulta persistente en el proceso de obtener nuevos documentos relevantes es la tercera de la consulta 3, con una precisión de 0.714 y una exhaustividad de 1.00. La Figura 5.12 muestra la estructura de esta consulta.

4 No se tienen en cuenta las consultas que no se han podido traducir y que aparecen con valores de 0.

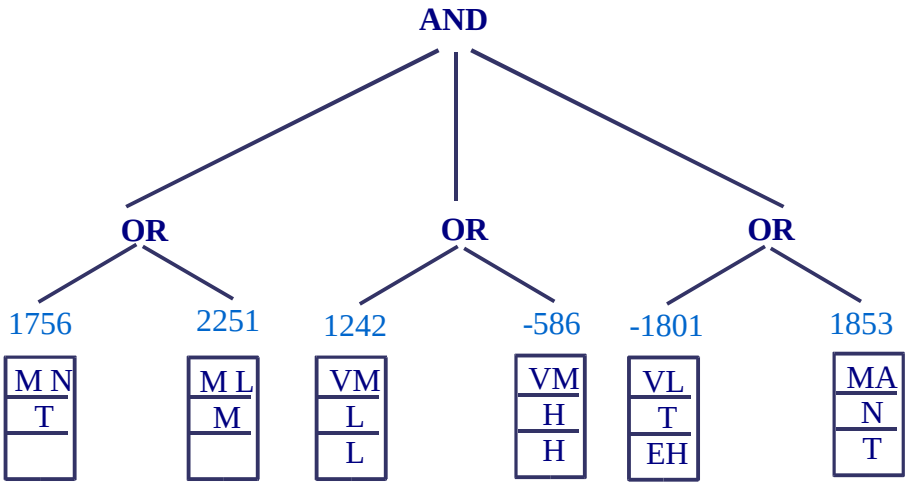


Figura 5.12: Mejor consulta persistente en el proceso de obtener nuevos documentos relevantes generada por el algoritmo SPEA-AG sobre la colección Cranfield (Solución 3 de la consulta 3)

Tamaño de los árboles

En todas las ejecuciones, la población ha convergido hasta presentar consultas de gran tamaño, lo que permite obtener mejor eficacia en la recuperación. Así, en la mayoría de los casos, el número de nodos (términos + operadores) de las consultas persistentes aprendidas se mueve entre el máximo (16) y 14, aunque algunas presentan un tamaño menor con 12 y 13 nodos.

En las consultas traducidas, el tamaño suele ser más pequeño en la mayoría de los casos, al no existir correspondencia para algunos términos en el conjunto de prueba. Existen, incluso, varios casos en los que las traducción no consigue ninguna consulta, lo que impide evaluarla.

5.3.1.4.- Análisis de la eficacia de las consultas representativas del Pareto de acuerdo a los criterios de precisión y exhaustividad en la experimentación realizada con CACM

Las Tablas 5.57 a 5.74 muestran la eficacia de la recuperación de las consultas seleccionadas del Pareto fusionado, obtenido tras combinar los Paretos asociados a cada una de las 5 ejecuciones realizadas, para cada una de las consultas de Cranfield. La eficacia se mide de acuerdo a los dos criterios básicos, la precisión y la exhaustividad.

C. 4	***ENTRENAMIENTO***					***PRUEBA***				
	Nodos	***P***	***E***	***#rel***	***#rec***	***Nodos***	***P***	***E***	***#rel***	***#rec***
1	14	1.000	1.000	6	6	11	0.000	0.000	0	2

Tabla 182: Eficacia de la recuperación en base a los criterios de precisión y exhaustividad para la consulta 4 de CACM

C. 7	***ENTRENAMIENTO***					***PRUEBA***				
	Nodos	***P***	***E***	***#rel***	***#rec***	***Nodos***	***P***	***E***	***#rel***	***#rec***
1	16	0.073	1.000	14	192	7	0.013	0.143	2	155
2	15	0.333	0.929	13	39	8	0.065	0.214	3	46
3	16	0.632	0.857	12	19	7	0.250	0.143	2	8
4	16	0.786	0.786	11	14	7	0.000	0.000	0	4
5	14	1.000	0.643	9	9	0	0.000	0.000	0	0

Tabla 183: Eficacia de la recuperación en base a los criterios de precisión y exhaustividad para la consulta 7 de CACM

C. 9	***ENTRENAMIENTO***					***PRUEBA***				
	Nodos	***P***	***E***	***#rel***	***#rec***	***Nodos***	***P***	***E***	***#rel***	***#rec***
1	14	1.000	1.000	4	4	12	0.286	0.400	2	7
2	15	1.000	1.000	4	4	15	0.250	0.400	2	8
3	14	1.000	1.000	4	4	10	0.250	0.200	1	4

Tabla 184: Eficacia de la recuperación en base a los criterios de precisión y exhaustividad para la consulta 9 de CACM

C. 10	*ENTRENAMIENTO*					*PRUEBA*				
	Nodos	*P*	*E*	*#rel*	*#rec*	*Nodos*	*P*	*E*	*#rel*	*#rec*
1	15	0.917	0.647	11	12	11	0.556	0.556	10	18
2	13	0.405	1.000	17	42	10	0.378	0.944	17	45
3	15	0.444	0.941	16	36	14	0.386	0.944	17	44
4	14	0.684	0.765	13	19	8	0.500	0.778	14	28
5	15	1.000	0.529	9	9	11	0.556	0.556	10	18

Tabla 185: Eficacia de la recuperación en base a los criterios de precisión y exhaustividad para la consulta 10 de CACM

C. 14	*ENTRENAMIENTO*					*PRUEBA*				
	Nodos	*P*	*E*	*#rel*	*#rec*	*Nodos*	*P*	*E*	*#rel*	*#rec*
1	15	0.818	0.818	18	22	11	0.640	0.727	16	25
2	16	0.677	0.955	21	31	10	0.571	0.182	4	7
3	16	0.564	1.000	22	39	10	0.529	0.409	9	17
4	15	0.760	0.864	19	25	10	0.157	0.727	16	102
5	16	1.000	0.682	15	15	0	0.000	0.000	0	0

Tabla 186: Eficacia de la recuperación en base a los criterios de precisión y exhaustividad para la consulta 14 de CACM

C. 19	*ENTRENAMIENTO*					*PRUEBA*				
	Nodos	*P*	*E*	*#rel*	*#rec*	*Nodos*	*P*	*E*	*#rel*	*#rec*
1	15	1.000	1.000	5	5	12	0.000	0.000	0	5

Tabla 187: Eficacia de la recuperación en base a los criterios de precisión y exhaustividad para la consulta 19 de CACM

C. 24	*ENTRENAMIENTO*					*PRUEBA*				
	Nodos	***P***	***E***	***#rel***	***#rec***	***Nodos***	***P***	***E***	***#rel***	***#rec***
1	15	1.000	0.833	5	5	14	0.000	0.000	0	2
2	15	1.000	0.833	5	5	0	0.000	0.000	0	0
3	14	0.857	1.000	6	7	0	0.000	0.000	0	0

Tabla 188: Eficacia de la recuperación en base a los criterios de precisión y exhaustividad para la consulta 24 de CACM

C. 25	*ENTRENAMIENTO*					*PRUEBA*				
	Nodos	***P***	***E***	***#rel***	***#rec***	***Nodos***	***P***	***E***	***#rel***	***#rec***
1	16	0.170	0.920	23	135	16	0.137	0.654	17	124
2	15	0.337	0.800	23	53	7	0.256	0.423	11	43
3	14	0.517	0.600	15	29	10	0.207	0.231	6	29
4	15	0.750	0.480	12	16	7	0.333	0.077	2	6
5	13	1.000	0.320	8	8	10	0.200	0.038	1	5

Tabla 189: Eficacia de la recuperación en base a los criterios de precisión y exhaustividad para la consulta 25 de CACM

C. 26	*ENTRENAMIENTO*					*PRUEBA*				
	Nodos	***P***	***E***	***#rel***	***#rec***	***Nodos***	***P***	***E***	***#rel***	***#rec***
1	16	1.000	0.800	12	12	13	0.600	0.400	6	10
2	16	0.682	1.000	15	22	13	0.571	0.800	12	21
3	16	0.875	0.933	14	16	10	0.667	0.667	10	15
4	15	0.929	0.867	13	14	12	0.562	0.600	9	16

Tabla 190: Eficacia de la recuperación en base a los criterios de precisión y exhaustividad para la consulta 26 de CACM

C. 27	ENTRENAMIENTO					PRUEBA				
	Nodos	P	E	#rel	#rec	Nodos	P	E	#rel	#rec
1	14	0.206	1.000	14	68	12	0.088	0.333	5	57
2	14	0.343	0.857	12	35	10	0.061	0.200	3	49
3	14	0.458	0.786	11	24	10	0.105	0.133	2	19
4	14	0.909	0.714	10	11	11	0.200	0.067	1	5
5	14	1.000	0.643	9	9	11	0.250	0.067	1	4

Tabla 191: Eficacia de la recuperación en base a los criterios de precisión y exhaustividad para la consulta 27 de CACM

C. 40	ENTRENAMIENTO					PRUEBA				
	Nodos	P	E	#rel	#rec	Nodos	P	E	#rel	#rec
1	14	1.000	0.800	4	4	14	1.000	0.200	1	1
2	15	1.000	0.800	4	4	11	0.250	0.200	1	4
3	14	1.000	0.800	4	4	12	0.000	0.000	0	5
4	15	0.625	1.000	5	8	0	0.000	0.000	0	0

Tabla 192: Eficacia de la recuperación en base a los criterios de precisión y exhaustividad para la consulta 40 de CACM

C. 42	ENTRENAMIENTO					PRUEBA				
	Nodos	P	E	#rel	#rec	Nodos	P	E	#rel	#rec
1	15	0.286	1.000	10	35	8	0.080	0.182	2	25
2	15	0.360	0.900	9	25	8	0.095	0.182	2	21
3	16	0.800	0.800	8	10	0	0.000	0.000	0	0
4	16	1.000	0.700	7	7	0	0.000	0.000	0	0

Tabla 193: Eficacia de la recuperación en base a los criterios de precisión y exhaustividad para la consulta 42 de CACM

C. 43	*ENTRENAMIENTO*					*PRUEBA*				
	Nodos	***P***	***E***	***#rel***	***#rec***	***Nodos***	***P***	***E***	***#rel***	***#rec***
1	15	0.188	0.900	18	96	8	0.145	0.571	12	83
2	16	0.341	0.750	15	44	10	0.207	0.571	12	58
3	14	0.571	0.600	12	21	10	0.370	0.476	10	27
4	14	0.714	0.500	10	14	0	0.000	0.000	0	0
5	14	1.000	0.400	8	8	10	0.364	0.190	4	11

Tabla 194: Eficacia de la recuperación en base a los criterios de precisión y exhaustividad para la consulta 43 de CACM

C. 45	*ENTRENAMIENTO*					*PRUEBA*				
	Nodos	***P***	***E***	***#rel***	***#rec***	***Nodos***	***P***	***E***	***#rel***	***#rec***
1	13	0.222	0.923	12	54	9	0.131	0.615	8	61
2	15	0.647	0.846	11	17	11	0.192	0.385	5	26
3	14	1.000	0.692	9	9	9	0.000	0.000	0	3
4	15	0.089	1.000	13	146	9	0.060	0.615	8	134
5	14	0.833	0.769	10	12	9	0.300	0.231	3	10

Tabla 195: Eficacia de la recuperación en base a los criterios de precisión y exhaustividad para la consulta 45 de CACM

C. 58	*ENTRENAMIENTO*					*PRUEBA*				
	Nodos	***P***	***E***	***#rel***	***#rec***	***Nodos***	***P***	***E***	***#rel***	***#rec***
1	14	0.068	1.000	15	219	7	0.040	0.533	8	199
2	16	0.310	0.867	13	42	0	0.000	0.000	0	0
3	16	0.500	0.800	12	24	0	0.000	0.000	0	0
4	15	0.688	0.733	11	16	0	0.000	0.000	0	0
5	15	1.000	0.667	10	10	0	0.000	0.000	0	0

Tabla 196: Eficacia de la recuperación en base a los criterios de precisión y exhaustividad para la consulta 58 de CACM

C. 59	ENTRENAMIENTO					PRUEBA				
	Nodos	P	E	#rel	#rec	Nodos	P	E	#rel	#rec
1	15	0.106	1.000	21	198	15	0.090	0.864	19	212
2	16	0.380	0.950	19	50	10	0.366	0.682	15	41
3	14	0.514	0.857	18	35	10	0.158	0.136	3	19
4	14	0.750	0.714	15	20	12	0.571	0.545	12	21
5	14	1.000	0.571	12	12	12	0.455	0.227	5	11

Tabla 197: Eficacia de la recuperación en base a los criterios de precisión y exhaustividad para la consulta 59 de CACM

C. 60	ENTRENAMIENTO					PRUEBA				
	Nodos	P	E	#rel	#rec	Nodos	P	E	#rel	#rec
1	16	0.203	1.000	13	64	10	0.119	0.571	8	67
2	14	0.333	0.923	12	36	10	0.067	0.143	2	30
3	13	0.458	0.846	11	24	10	0.091	0.143	2	22
4	16	0.714	0.769	10	14	10	0.471	0.571	8	17
5	13	1.000	0.538	7	7	10	0.250	0.071	1	4

Tabla 198: Eficacia de la recuperación en base a los criterios de precisión y exhaustividad para la consulta 60 de CACM

C. 61	ENTRENAMIENTO					PRUEBA				
	Nodos	P	E	#rel	#rec	Nodos	P	E	#rel	#rec
1	15	0.424	0.933	14	33	11	0.262	0.688	11	42
2	15	0.357	1.000	15	42	14	0.261	0.750	12	46
3	14	0.481	0.867	13	27	13	0.176	0.188	3	17
4	14	0.706	0.800	12	17	9	0.222	0.125	2	9
5	15	1.000	0.667	10	10	7	0.500	0.125	2	4

Tabla 199: Eficacia de la recuperación en base a los criterios de precisión y exhaustividad para la consulta 61 de CACM

Análisis de resultados

A continuación, analizaremos los resultados presentados en las tablas anteriores. Al igual que con la colección Cranfield, comentaremos en primer lugar los resultados de cada grupo de consultas (consultas con más de 30 documentos relevantes, consultas que presentan entre 21 y 30 documentos relevantes, y consultas con menos de 15 documentos relevantes, Tabla 5,75), de acuerdo a la efectividad de la recuperación. Seguidamente, analizaremos el tamaño de las mejores consultas persistentes aprendidas independientemente del número de documentos relevantes que tenga asociado la consulta.

Nº de documentos relevantes	*Consultas*
más de 30	10, 14, 25, 43, 59 y 61
entre 21 y 30	7, 26, 27, 42, 45, 58 y 60
menos de 15	4, 9, 19, 24 y 40

Tabla 200: Grupos de consultas de CACM

Eficacia de las consultas con más de 30 documentos relevantes

Igual que en la colección Cranfield, el conjunto de soluciones seleccionado de acuerdo al procedimiento descrito en la Sección 3.5.2 se compone de diversas consultas persistentes con diferente equilibrio precisión-exhaustividad, objetivo de esta propuesta.

La diversidad de soluciones se mueve entre consultas persistentes que son capaces de recuperar únicamente documentos relevantes, alrededor de la mitad, y consultas persistentes que representan perfectamente las necesidades del usuario y recuperan todos documentos relevantes, aunque junto con basura (documentos irrelevantes). Los valores de exhaustividad se encuentran bastante por encima de 0.5 en la mayoría de los casos, mientras que los de precisión también son aceptables. Salvo un par de casos en los que se recuperan más de 60 documentos, la recuperación no conlleva un ruido excesivo.

Las soluciones intermedias (precisión y exhaustividad entre 0.7 y 0.8) son las que mejor balance precisión-exhaustividad presentan, recuperando casi todos los documentos relevantes y disminuyendo el ruido.

Si nos fijamos en la Tabla 186, correspondiente a la consulta 14, podemos ver como la mejor consulta es la que ocupa la cuarta posición. Ésta recupera 25 documentos, de los cuales

19 son relevantes, obteniéndose una precisión de 0.760 y una exhaustividad de 0.864.

La eficacia de estas consultas a la hora de recuperar nuevos documentos que se adapten a la necesidades de un usuario (evaluación sobre el conjunto de prueba) es prometedora. Todas las consultas recuperan documentos relevantes y en muchos casos este número es bastante alto, por encima de la media. A su vez, los valores de la precisión se sitúan alrededor de 0.3 o por encima, lo que indica que el acceso a los nuevos documentos por parte de los usuarios, no se verá excesivamente entorpecido por una maraña de documentos irrelevantes.

Eficacia de las consultas que presentan entre 21 y 30 documentos relevantes

Estas consultas tienen un comportamiento muy similar al descrito en la sección anterior, aunque empiezan a aparecer consultas cuyos Paretos unificados no tienen ni siquiera cinco soluciones diferentes en el espacio de objetivos y decisiones (Tabla 193).

En lo que respecta a la evaluación sobre el conjunto de prueba, aparece alguna consulta que no es capaz de recupera ningún documento relevante[5] (Tabla 183), lo que nos anticipa el comportamiento generalizado en las consultas con menos de 15 documentos relevantes. El resto de consultas persistentes siguen recuperando documentos relevantes, aunque disminuyen los valores de precisión (aproximadamente de 0.1) y exhaustividad (por debajo de la media) respecto a los obtenidos por las consultas del grupo anterior.

Eficacia de las consultas con menos de 15 documentos relevantes

Como ya vimos en la Sección 5.3.1.2 al analizar los Paretos obtenidos por consultas con pocos documentos relevantes asociados, los frentes de dichos Paretos estaban formados por un número muy reducido de soluciones (una o dos) diferentes en el espacio objetivo, en su mayoría soluciones con una precisión y exhaustividad iguales a 1. En consecuencia, al fusionar los Paretos, parece normal que todas las soluciones incluidas en ellos presenten dichos valores, independientemente de que tengan o no la misma estructura. De ahí que la totalidad de las consultas persistentes mostradas en las tablas anteriores con menos de 15 documentos relevantes presenten valores máximos de precisión y exhaustividad. Además, en

5 Nos referimos a las consultas que han sido traducidas y que al ser evaluadas sobre nuevos documentos no han sido capaces de recuperar ninguno que se adapte a las necesidades de información del usuario; no a las consultas que no tienen traducción y aparecen como que no hubiesen recuperado nada, puesto que no han podido ni siquiera evaluarse sobre los nuevos documentos (Tabla 193, últimas dos filas).

algunas consultas, el número de soluciones diferentes, no sólo en el espacio de objetivos, sino también en el espacio de decisiones, es inferior a cinco, lo que no permite seleccionar un conjunto completo de cinco soluciones.

La eficacia de recuperación sobre el conjunto de prueba es bastante mala en general. De hecho, la mayoría de las consultas no consigue recuperar ningún documento relevante (no se tienen en cuenta consultas que no se hayan podido traducir). Las que sí lo hacen, presentan valores de exhaustividad y precisión alrededor de 0.2 y 0.25, respectivamente, es decir, recuperan muy pocos documentos, comportamiento que, aunque facilita el acceso de los usuarios a los nuevos documentos, no proporciona mucha información debido al bajo número de documentos relevantes que se obtienen.

En nuestra opinión, el mal comportamiento puede deberse, como ya comentamos anteriormente, a la diversidad de términos índice en los documentos relevantes para estas consultas, lo que impide a los términos existentes en los documentos del conjunto de entrenamiento describir apropiadamente los documentos relevantes del conjunto de prueba.

Mejor consulta

Una de las consultas que presentan una mayor diversidad en las soluciones que forman el frente del Pareto es la consulta 25. En la Tabla 189 podemos ver como las cinco soluciones presentan diferente equilibrio entre los criterios de precisión y exhaustividad. Comienzan con valores bajos de precisión (0.170) y altos de exhaustividad (0.920) y van aumentando y disminuyendo, respectivamente, hasta la solución con valores de precisión y exhaustividad iguales a 1.00 y 0.320.

Tamaño de los árboles

En lo que respecta a este aspecto, encontramos de nuevo el mismo comportamiento que en Cranfield: el algoritmo converge hasta presentar consultas de gran tamaño para mejorar la efectividad de la recuperación. La mayoría de consultas persistentes, están formadas por un número de nodos que se mueve entre el 16 (máximo) y 14, aunque algunas presentan un tamaño menor con 13 nodos.

5.3.1.5.- Análisis de la eficacia de las consultas representativas del Pareto de acuerdo a la precisión media en la experimentación realizada con Cranfield

Como ya comentamos al principio del capítulo, las consultas persistentes derivadas por el algoritmo multiobjetivo propuesto son consultas legítimas del SRI Lingüístico basado en información lingüística multigranular definido en el Capítulo 4. Este SRI presenta la potencialidad de devolver los documentos ordenados según su relevancia. Sería, por tanto, interesante que el mecanismo de evaluación fuese capaz de medir también esta capacidad.

Para ello, en vez de considerar un umbral de corte, cuya mala elección nos puede llevar a pensar en un mal comportamiento del SRI, estudiaremos también el comportamiento del sistema ante una consulta considerando que el conjunto de documentos recuperados es el conjunto total de documentos en la base y calculando la precisión media a once niveles de exhaustividad, de la forma que se indicó en la Sección 1.4. El sistema devolverá el conjunto de documentos ordenados de acuerdo a su RSV, interesándonos que los documentos relevantes estén en las primeras posiciones de la lista, de manera que para acceder a ellos no tengamos que examinar muchos irrelevantes.

Las Tablas 5.76 a 5.92 muestran, por lo tanto, la eficacia de la recuperación de las consultas seleccionadas del Pareto fusionado, de acuerdo a la precisión media a once niveles de exhaustividad.

Dentro de cada tabla, y de izquierda a derecha, las columnas recogen los datos siguientes:

i. Número de consulta seleccionada.

ii. Valores correspondientes a la evaluación sobre el conjunto de entrenamiento de las consultas persistentes: Número de nodos en el árbol que representa la consulta, Precisión media, Número de documentos relevantes en la base, Número de documentos relevantes recuperados en las *n* primeras posiciones, siendo *n* igual al número de documentos relevantes en la base.

iii. Valores correspondientes a la evaluación sobre el conjunto de prueba de las consultas persistente: Número de nodos en el árbol que representa la consulta, Precisión media, Número de documentos relevantes en la base, Número de documentos relevantes recuperados en las *n* primeras posiciones, siendo *n* igual al número de documentos relevantes en la base.

C. 1	*ENTRENAMIENTO*				*PRUEBA*			
	Nodos	*Pm*	*#D_rel*	*#rel_n*	*Nodos*	*Pm*	*#D_rel*	*#rel_n*
1	14	0.074	14	0	11	0.033	15	0
2	15	0.366	14	6	15	0.097	15	2
3	15	0.343	14	7	7	0.013	15	0
4	15	0.583	14	9	7	0.013	15	0
5	15	0.568	14	7	7	0.013	15	0

Tabla 201: Eficacia de la recuperación en base a la precisión media para la consulta 1 de Cranfield

C. 2	*ENTRENAMIENTO*				*PRUEBA*			
	Nodos	*Pm*	*#D_rel*	*#rel_n*	*Nodos*	*Pm*	*#D_rel*	*#rel_n*
1	14	0.648	12	7	13	0.146	13	4
2	14	0.657	12	9	11	0.113	13	3
3	16	0.493	12	5	13	0.202	13	3
4	15	0.713	12	9	11	0.288	13	4
5	15	0.643	12	8	11	0.272	13	5

Tabla 202: Eficacia de la recuperación en base a la precisión media para la consulta 2 de Cranfield

C. 3	*ENTRENAMIENTO*				*PRUEBA*			
	Nodos	*Pm*	*#D_rel*	*#rel_n*	*Nodos*	*Pm*	*#D_rel*	*#rel_n*
1	13	1.000	4	4	0	0.000	0	0
2	15	1.000	4	4	0	0.000	0	0
3	14	1.000	4	4	10	0.542	5	3
4	15	1.000	4	4	8	0.005	5	0

Tabla 203: Eficacia de la recuperación en base a la precisión media para la consulta 3 de Cranfield

C. 7	ENTRENAMIENTO				PRUEBA			
	Nodos	Pm	#D_rel	#rel_n	Nodos	Pm	#D_rel	#rel_n
1	14	1.000	3	3	0	0.000	0	0

Tabla 204: Eficacia de la recuperación en base a la precisión media para la consulta 7 de Cranfield

C. 8	ENTRENAMIENTO				PRUEBA			
	Nodos	Pm	#D_rel	#rel_n	Nodos	Pm	#D_rel	#rel_n
1	12	0.717	6	3	0	0.000	0	0
2	15	0.575	6	5	10	0.017	6	0
3	14	0.647	6	4	12	0.024	6	0
4	12	0.640	6	4	12	0.021	6	0
5	14	0.652	6	4	13	0.045	6	0

Tabla 205: Eficacia de la recuperación en base a la precisión media para la consulta 8 de Cranfield

C. 11	ENTRENAMIENTO				PRUEBA			
	Nodos	Pm	#D_rel	#rel_n	Nodos	Pm	#D_rel	#rel_n
1	13	1.000	4	4	12	0.004	4	0
2	15	1.000	4	4	0	0.000	0	0
3	12	1.000	4	4	7	0.004	4	0

Tabla 206: Eficacia de la recuperación en base a la precisión media para la consulta 11 de Cranfield

C. 19	ENTRENAMIENTO				PRUEBA			
	Nodos	Pm	#D_rel	#rel_n	Nodos	Pm	#D_rel	#rel_n
1	16	1.000	5	5	10	0.144	5	1

Tabla 207: Eficacia de la recuperación en base a la precisión media para la consulta 19 de Cranfield

C. 23	*ENTRENAMIENTO*				*PRUEBA*			
	Nodos	*Pm*	*#D_rel*	*#rel_n*	*Nodos*	*Pm*	*#D_rel*	*#rel_n*
1	12	0.162	16	3	10	0.226	17	3
2	15	0.510	16	9	12	0.047	17	1
3	14	0.538	16	10	14	0.045	17	1
4	14	0.508	16	9	13	0.114	17	3
5	14	0.380	16	6	13	0.079	17	2

Tabla 208: Eficacia de la recuperación en base a la precisión media para la consulta 23 de Cranfield

C. 26	*ENTRENAMIENTO*				*PRUEBA*			
	Nodos	*Pm*	*#D_rel*	*#rel_n*	*Nodos*	*Pm*	*#D_rel*	*#rel_n*
1	14	1.000	3	3	11	0.005	4	0
2	15	1.000	3	3	11	0.005	4	0

Tabla 209: Eficacia de la recuperación en base a la precisión media para la consulta 26 de Cranfield

C. 38	*ENTRENAMIENTO*				*PRUEBA*			
	Nodos	*Pm*	*#D_rel*	*#rel_n*	*Nodos*	*Pm*	*#D_rel*	*#rel_n*
1	14	0.820	5	4	0	0.000	0	0
2	14	0.820	5	4	13	0.043	6	1
3	13	0.970	5	4	11	0.014	6	0
4	14	0.819	5	4	10	0.014	6	0

Tabla 210: Eficacia de la recuperación en base a la precisión media para la consulta 38 de Cranfield

C. 39	*ENTRENAMIENTO*				*PRUEBA*			
	Nodos	***Pm***	***#D_rel***	***#rel_n***	***Nodos***	***Pm***	***#D_rel***	***#rel_n***
1	15	0.381	7	1	10	0.055	7	1
2	16	0.821	7	6	13	0.134	7	1

Tabla 211: Eficacia de la recuperación en base a la precisión media para la consulta 39 de Cranfield

C. 40	*ENTRENAMIENTO*				*PRUEBA*			
	Nodos	***Pm***	***#D_rel***	***#rel_n***	***Nodos***	***Pm***	***#D_rel***	***#rel_n***
1	15	0.955	6	5	9	0.014	7	0
2	16	0.821	6	5	7	0.010	7	0

Tabla 212: Eficacia de la recuperación en base a la precisión media para la consulta 40 de Cranfield

C. 47	*ENTRENAMIENTO*				*PRUEBA*			
	Nodos	***Pm***	***#D_rel***	***#rel_n***	***Nodos***	***Pm***	***#D_rel***	***#rel_n***
1	16	0.565	7	4	7	0.014	8	0
2	14	0.550	7	4	9	0.017	8	0
3	13	0.870	7	6	13	0.076	8	1
4	12	0.713	7	6	12	0.088	8	1

Tabla 213: Eficacia de la recuperación en base a la precisión media para la consulta 47 de Cranfield

C. 73	*ENTRENAMIENTO*				*PRUEBA*			
	Nodos	*Pm*	*#D_rel*	*#rel_n*	*Nodos*	*Pm*	*#D_rel*	*#rel_n*
1	15	0.657	10	6	15	0.027	11	1
2	15	0.617	10	5	15	0.029	11	1
3	14	0.712	10	8	10	0.092	11	1
4	13	0.748	10	7	12	0.080	11	1

Tabla 214: Eficacia de la recuperación en base a la precisión media para la consulta 73 de Cranfield

C. 157	*ENTRENAMIENTO*				*PRUEBA*			
	Nodos	*Pm*	*#D_rel*	*#rel_n*	*Nodos*	*Pm*	*#D_rel*	*#rel_n*
1	15	0.200	20	5	14	0.079	20	2
2	14	0.311	20	5	11	0.103	20	2
3	15	0.341	20	11	10	0.289	20	7
4	14	0.462	20	11	11	0.151	20	3
5	14	0.559	20	11	11	0.044	20	2

Tabla 215: Eficacia de la recuperación en base a la precisión media para la consulta 157 de Cranfield

C. 220	*ENTRENAMIENTO*				*PRUEBA*			
	Nodos	*Pm*	*#D_rel*	*#rel_n*	*Nodos*	*Pm*	*#D_rel*	*#rel_n*
1	16	0.603	10	6	16	0.099	10	1
2	13	0.649	10	6	12	0.017	10	0
3	14	0.805	10	7	14	0.056	10	0
4	14	0.736	10	8	13	0.028	10	0
5	14	0.707	10	7	13	0.029	10	0

Tabla 216: Eficacia de la recuperación en base a la precisión media para la consulta 220 de Cranfield

C. 225	ENTRENAMIENTO				PRUEBA			
	Nodos	Pm	#D_rel	#rel_n	Nodos	Pm	#D_rel	#rel_n
1	14	0.248	12	3	14	0.076	13	2
2	14	0.261	12	4	14	0.135	13	3
3	14	0.237	12	3	14	0.126	13	2
4	16	0.607	12	8	0	0.000	0	0
5	14	0.575	12	7	8	0.124	13	2

Tabla 217: Eficacia de la recuperación en base a la precisión media para la consulta 225 de Cranfield

Análisis de resultados

A continuación, analizaremos los resultados presentados en las tablas anteriores, agrupándolos por conjuntos de consultas (consultas con más de 20 documentos relevantes y consultas que presentan menos de 15 documentos relevantes, Tabla 181) de acuerdo a la capacidad de las consultas para situar a los documentos relevantes en las primeras posiciones. Las consultas persistentes seleccionadas son las mismas de la Sección 5.3.1.3, por lo que no se volverá a analizar el tamaño de las consultas, ni el número de consultas seleccionas, puesto que esto ya se hizo en dicha sección.

Eficacia de las consultas con más de 20 documentos relevantes

Los valores de la precisión media son bastante altos, indicando que bastantes documentos que satisfacen las necesidades del usuario se encuentran en las primeras posiciones de la lista, sin que aparezcan mezclados con muchos documentos irrelevantes. Este comportamiento es muy deseable puesto que el usuario no tendrá que examinar muchos documentos antes de encontrar aquellos que realmente le interesan, evitándole perder el tiempo y que se desanime.

Por lo general, más de la mitad de los documentos relevantes se encuentran situados en las *n* primeras posiciones de la lista (con *n* igual al número de documentos relevantes). El comportamiento ideal sería que todos los relevantes se encontrasen en las primeras posiciones.

Un mayor número de documentos relevantes en las primeras posiciones no se traduce en

mayor valor de precisión media, sino que también se tienen en cuenta el resto de documentos. Si nos fijamos en la Tabla 215, correspondiente a la consultas 157, podemos ver como las consultas persistentes que ocupan la segunda y tercera líneas presentan prácticamente el mismo valor de precisión media (0.311 y 0.341), aún cuando una sitúa el doble de documentos que la otra en las primeras posiciones (5 y 11 para las consultas 2 y 3, respectivamente). La diferencia estriba en las posiciones que ocupan el resto de documentos. Aunque la tercera sitúa 11 documentos en las primeras posiciones, el resto de documentos relevantes se encuentran a partir de la posición 300, mientras que la segunda localiza 17 entre las 50 primeras posiciones.

Al evaluar las consultas persistentes sobre bases con documentos desconocidos en el proceso de aprendizaje, los valores de la precisión media disminuyen, localizándose muy pocos documentos relevantes en las primeras posiciones, lo que perjudica al usuario que necesita analizar muchos documentos innecesarios antes de encontrar alguno que realmente le interese.

Eficacia de las consultas con menos de 15 documentos relevantes

Como ya se comentó en la Sección 5.3.1.3, los Paretos fusionados de este grupo de consultas están formados por pocas soluciones, las mayoría con valores máximos de precisión y exhaustividad. Esto se refleja en valores de precisión media muy altos, cercanos al máximo e incluso iguales a él en la mayoría de los casos. Este comportamiento es predecible si pensamos que si sólo se recuperaban documentos relevantes, lo lógico es que estos tengan un valor RSV mayor que los no recuperados (que no superaron el corte establecido por el umbral) y que, por tanto, al ordenarlos estarán por delante.

El único caso extraño es la primera consulta persistente seleccionada para la consulta 39 que, con un valor de 0.381 para la precisión media, sólo sitúa un documento relevante en las primeras posiciones.

En contraste con lo anterior, cuando evaluamos estas consultas persistentes sobre el conjunto de prueba nos encontramos con que casi ninguna posiciona documentos en la parte alta de la lista, como demuestran los bajos valores de precisión media, por debajo de 0.1. De hecho, analizando el número de documentos, vemos como la tónica general son 0 ó 1 documentos relevantes en las *n* primeras posiciones. Esto sería medianamente admisible si el

resto de documentos aparecieran en las 20 posiciones siguientes (hay que tener en cuenta que, al contar con pocos documentos relevantes, el número de irrelevantes en esas primeras posiciones sería igualmente pequeño), pues no supondría al usuario tener que examinar demasiados documentos inútiles. Sin embargo, analizando en más profundidad las posiciones que ocupan los documentos relevantes, resulta que, salvo un par de ocasiones en las que algún documento relevante aparece en ese rango propuesto, en el resto estos documentos empiezan a aparecer a partir de la posición 100, y en algunos casos incluso a partir de la 400.

Como viene siendo habitual, la consulta 3 destaca positivamente entre los pésimos comportamientos de sus homólogas (Tabla 203). Con una precisión media de 0.56, tres de los cinco documentos relevantes se sitúan en las cinco primeras posiciones, y los otros dos a continuación. Concretamente, las posiciones que ocupan son la 3, 4, 5, 6 y 7.

Mejor consulta

De ambos grupos de consultas, la consulta persistente con mejor comportamiento es la seleccionada en tercer lugar para la consulta 3 (véase la Tabla 203). Cuando se evalúa sobre el conjunto de entrenamiento, consigue el comportamiento ideal, al recuperar todos los documentos en primer lugar. Cuando se lanza sobre documentos no utilizados en el proceso de aprendizaje, su comportamiento es excepcionalmente bueno, al recuperar los documentos interesantes en las posiciones de 3 a la 7, consiguiendo una precisión media de 0.56. Esto facilitará al usuario la tarea de obtener nueva información. La Figura 5.12 muestra la estructura de esta consulta persistente.

5.3.1.6.- Análisis de la eficacia de las consultas representativas del Pareto de acuerdo a la precisión media en la experimentación realizada con CACM

Las Tablas 5.93 a 5.110 muestran la eficacia de la recuperación de las consultas seleccionadas del Pareto fusionado, en base a la precisión media a once niveles de exhaustividad, calculada como se indica en la Sección 1.4.

C. 4	***ENTRENAMIENTO***				***PRUEBA***			
	Nodos	***Pm***	***#D_rel***	***#rel_n***	***Nodos***	***Pm***	***#D_rel***	***#rel_n***
1	15	1.000	6	6	11	0.008	6	0

Tabla 218: Eficacia de la recuperación en base a la precisión media para la consulta 4 de CACM

C. 7	***ENTRENAMIENTO***				***PRUEBA***			
	Nodos	***Pm***	***#D_rel***	***#rel_n***	***Nodos***	***Pm***	***#D_rel***	***#rel_n***
1	16	0.247	14	4	7	0.021	14	0
2	15	0.252	14	3	8	0.031	14	0
3	16	0.671	14	11	7	0.060	14	2
4	16	0.640	14	11	7	0.022	14	0
5	14	0.714	14	10	0	0.000	0	0

Tabla 219: Eficacia de la recuperación en base a la precisión media para la consulta 7de CACM

C. 9	***ENTRENAMIENTO***				***PRUEBA***			
	Nodos	***Pm***	***#D_rel***	***#rel_n***	***Nodos***	***Pm***	***#D_rel***	***#rel_n***
1	14	1.000	4	4	12	0.237	5	2
2	15	1.000	4	4	15	0.130	5	1
3	14	1.000	4	4	10	0.124	5	1

Tabla 220: Eficacia de la recuperación en base a la precisión media para la consulta 9 de CACM

C. 10	ENTRENAMIENTO				PRUEBA			
	Nodos	Pm	#D_rel	#rel_n	Nodos	Pm	#D_rel	#rel_n
1	15	0.638	17	11	11	0.322	18	10
2	13	0.382	17	6	10	0.320	18	4
3	15	0.371	17	5	14	0.385	18	6
4	14	0.592	17	12	8	0.534	18	11
5	15	0.582	17	9	11	0.352	18	10

Tabla 221: Eficacia de la recuperación en base a la precisión media para la consulta 10 de CACM

C. 14	ENTRENAMIENTO				PRUEBA			
	Nodos	Pm	#D_rel	#rel_n	Nodos	Pm	#D_rel	#rel_n
1	15	0.687	22	18	11	0.557	22	13
2	16	0.637	22	14	10	0.178	22	4
3	16	0.708	22	12	10	0.304	22	9
4	15	0.619	22	17	10	0.369	22	10
5	16	0.648	22	15	0	0.000	0	0

Tabla 222: Eficacia de la recuperación en base a la precisión media para la consulta 14 de CACM

C. 19	ENTRENAMIENTO				PRUEBA			
	Nodos	Pm	#D_rel	#rel_n	Nodos	Pm	#D_rel	#rel_n
1	15	1.000	5	5	12	0.005	6	0

Tabla 223: Eficacia de la recuperación en base a la precisión media para la consulta 19 de CACM

C. 24	*ENTRENAMIENTO*				*PRUEBA*			
	Nodos	*Pm*	*#D_rel*	*#rel_n*	*Nodos*	*Pm*	*#D_rel*	*#rel_n*
1	15	0.820	6	5	12	0.014	7	0
2	15	0.821	6	5	0	0.000	0	0
3	14	0.729	6	5	0	0.000	0	0

Tabla 224: Eficacia de la recuperación en base a la precisión media para la consulta 24 de CACM

C. 25	*ENTRENAMIENTO*				*PRUEBA*			
	Nodos	*Pm*	*#D_rel*	*#rel_n*	*Nodos*	*Pm*	*#D_rel*	*#rel_n*
1	16	0.207	25	2	16	0.081	26	1
2	15	0.512	25	13	7	0.176	26	6
3	14	0.473	25	12	10	0.083	26	5
4	15	0.419	25	12	7	0.042	26	2
5	13	0.398	25	8	10	0.066	26	1

Tabla 225: Eficacia de la recuperación en base a la precisión media para la consulta 25 de CACM

C. 26	*ENTRENAMIENTO*				*PRUEBA*			
	Nodos	*Pm*	*#D_rel*	*#rel_n*	*Nodos*	*Pm*	*#D_rel*	*#rel_n*
1	16	0.827	15	12	13	0.387	15	6
2	16	0.803	15	12	13	0.540	15	10
3	16	0.857	15	13	10	0.443	15	10
4	15	0.803	15	13	12	0.422	15	8

Tabla 226: Eficacia de la recuperación en base a la precisión media para la consulta 26 de CACM

C. 27	ENTRENAMIENTO				PRUEBA			
	Nodos	Pm	#D_rel	#rel_n	Nodos	Pm	#D_rel	#rel_n
1	14	0.271	14	4	12	0.057	15	1
2	14	0.283	14	5	10	0.031	15	0
3	14	0.315	14	5	10	0.035	15	1
4	14	0.676	14	10	11	0.059	15	1
5	14	0.661	14	9	11	0.114	15	1

Tabla 227: Eficacia de la recuperación en base a la precisión media para la consulta 27 de CACM

C. 40	ENTRENAMIENTO				PRUEBA			
	Nodos	Pm	#D_rel	#rel_n	Nodos	Pm	#D_rel	#rel_n
1	14	0.820	5	4	14	0.291	5	1
2	15	0.819	5	4	11	0.076	5	1
3	14	0.825	5	4	12	0.009	5	0
4	15	0.815	5	3	0	0.000	0	0

Tabla 228: Eficacia de la recuperación en base a la precisión media para la consulta 40 de CACM

C. 42	ENTRENAMIENTO				PRUEBA			
	Nodos	Pm	#D_rel	#rel_n	Nodos	Pm	#D_rel	#rel_n
1	15	0.557	10	4	8	0.053	11	1
2	15	0.572	10	4	8	0.072	11	1
3	16	0.651	10	8	0	0.000	0	0
4	16	0.742	10	7	0	0.000	0	0

Tabla 229: Eficacia de la recuperación en base a la precisión media para la consulta 42 de CACM

C. 43	*ENTRENAMIENTO*				*PRUEBA*			
	Nodos	*Pm*	*#D_rel*	*#rel_n*	*Nodos*	*Pm*	*#D_rel*	*#rel_n*
1	15	0.132	20	1	8	0.299	21	4
2	16	0.292	20	9	10	0.236	21	7
3	14	0.433	20	11	10	0.524	21	10
4	14	0.538	20	11	0	0	0	0
5	14	0.543	20	10	10	0.320	21	7

Tabla 230: Eficacia de la recuperación en base a la precisión media para la consulta 43 de CACM

C. 45	*ENTRENAMIENTO*				*PRUEBA*			
	Nodos	*Pm*	*#D_rel*	*#rel_n*	*Nodos*	*Pm*	*#D_rel*	*#rel_n*
1	13	0.472	13	6	9	0.158	13	4
2	15	0.518	13	7	11	0.152	13	4
3	14	0.644	13	9	9	0.034	13	0
4	15	0.269	13	2	9	0.053	13	0
5	14	0.715	13	10	9	0.211	13	3

Tabla 231: Eficacia de la recuperación en base a la precisión media para la consulta 45 de CACM

C. 58	*ENTRENAMIENTO*				*PRUEBA*			
	Nodos	*Pm*	*#D_rel*	*#rel_n*	*Nodos*	*Pm*	*#D_rel*	*#rel_n*
1	14	0.038	15	0	7	0.117	15	1
2	16	0.272	15	3	0	0.000	0	0
3	16	0.407	15	8	0	0.000	0	0
4	15	0.513	15	10	0	0.000	0	0
5	15	0.644	15	10	0	0.000	0	0

Tabla 232: Eficacia de la recuperación en base a la precisión media para la consulta 58 de CACM

C. 59	ENTRENAMIENTO				PRUEBA			
	Nodos	Pm	#D_rel	#rel_n	Nodos	Pm	#D_rel	#rel_n
1	15	0.080	21	0	15	0.083	22	1
2	16	0.241	21	1	10	0.220	22	2
3	14	0.659	21	15	10	0.141	22	4
4	14	0.472	21	15	12	0.344	22	12
5	14	0.554	21	12	12	0.135	22	5

Tabla 233: Eficacia de la recuperación en base a la precisión media para la consulta 59 de CACM

C. 60	ENTRENAMIENTO				PRUEBA			
	Nodos	Pm	#D_rel	#rel_n	Nodos	Pm	#D_rel	#rel_n
1	16	0.373	13	5	10	0.048	14	0
2	14	0.291	13	3	10	0.044	14	0
3	13	0.676	13	8	10	0.068	14	2
4	16	0.505	13	9	10	0.186	14	5
5	13	0.554	13	7	10	0.045	14	1

Tabla 234: Eficacia de la recuperación en base a la precisión media para la consulta 60 de CACM

C. 61	ENTRENAMIENTO				PRUEBA			
	Nodos	Pm	#D_rel	#rel_n	Nodos	Pm	#D_rel	#rel_n
1	15	0.624	15	8	11	0.284	16	7
2	15	0.360	15	6	14	0.341	16	7
3	14	0.443	15	7	13	0.161	16	3
4	14	0.428	15	10	9	0.052	16	2
5	15	0.671	15	10	7	0.095	16	2

Tabla 235: Eficacia de la recuperación en base a la precisión media para la consulta 61 de CACM

Análisis de resultados

A continuación, analizaremos los resultados presentados en las tablas anteriores. Al igual que en Cranfield, los agruparemos por conjuntos de consultas (consultas con más de 30 documentos relevantes, consultas que presentan entre 21 y 30 documentos relevantes, y consultas con menos de 15 documentos relevantes, Tabla 200) de acuerdo a la capacidad de las consultas para situar a los documentos relevantes en las primeras posiciones. Las consultas persistentes seleccionadas son las mismas de la Sección 5.3.1.4, por lo que no se volverá a analizar el tamaño de las consultas, ni el número de consultas seleccionas, puesto que esto ya se hizo en dicha sección.

Eficacia de las consultas con más de 30 documentos relevantes

Las tablas correspondientes muestran como sobre el conjunto de entrenamiento, los documentos relevantes consiguen RSVs altos, lo que les permite situarse en lo alto de la lista de documentos recuperados, como demuestran los valores de precisión media, situados alrededor de 0.5, y el número de documentos relevantes en las primeras posiciones, aproximadamente la mitad del total. Además, en muchos casos, el resto de documentos están situados en las siguientes posiciones, lo que está muy bien. Por ejemplo, los resultados de la tercera consulta persistente de la consulta 14 (Tabla 222), nos indican que de los 22 documentos relevantes, 12 están en las primeras posiciones. Además, el alto valor de precisión media (0.708) nos asegura que el resto de documentos no se encuentran muy lejos de los primeros, como realmente ocurre, al estar localizados dentro de las primeras 40 posiciones.

Si nos fijamos en los resultados obtenidos al evaluar las consultas sobre nuevos documentos, observamos que, aunque las precisiones medias bajan respecto al conjunto de entrenamiento, se mueven alrededor de 0.25, salvo en el caso de la consulta 25. Además, el número de documentos relevantes en las primeras posiciones también es alto, lo que simplifica el acceso a la nueva información por parte del usuario.

Eficacia de las consultas que presentan entre 21 y 30 documentos relevantes

Estas consultas tienen un comportamiento muy similar al descrito en la sección anterior. Aunque disminuyen un poco los valores de precisión media, todavía se localizan alrededor de

la mitad de los documentos relevantes en las primeras posiciones. La reducción en los valores de precisión implica la aparición de documentos no relevantes en posiciones cercanas al cenit.

Se reduce el número de documentos relevantes que aparecen en la primeras posiciones del conjunto de documentos de prueba y, en consecuencia, los valores de precisión media. Al haber menos documentos relevantes, resulta más difícil conseguir que estén todos en las primeras posiciones, y éstos se encuentran más desperdigados. Esto se puede deber a que la consulta esté formada por términos que aparecen sólo en ciertos documentos relevantes y no en todos.

Por otro lado, aparecen consultas que no consiguen situar ningún documento que satisfaga las necesidades de información del usuario en la cabeza de la lista y que además presentan valores bajos de precisión media (en torno a 0.04), lo que indica que los documentos que son interesantes están muy mezclados con los que no lo son.

Eficacia de las consultas con menos de 15 documentos relevantes

Como ya se comentó en la Sección 5.3.1.4, los Paretos fusionados de este grupo de consultas están formados por pocas soluciones, la mayoría con valores máximos de precisión y exhaustividad. Esto se refleja en valores de precisión media muy altos, cercanos al máximo e incluso iguales a él en la mayoría de los casos. Este es un comportamiento predecible si pensamos que si sólo se recuperaban documentos relevantes, lo lógico es que estos tengan un valor RSV mayor que los no recuperados (no superaron el corte establecido por el umbral) y, por tanto, estén por delante al ordenarlos.

En contraste con lo anterior, cuando evaluamos estas consultas persistentes sobre el conjunto de prueba, nos encontramos con que cuatro de las nueve consultas persistentes derivadas[6] no posicionan ningún documento relevante en lo alto de la lista, como demuestran los valores bajos de precisión media (de aproximadamente 0.008). Las otras cinco consultas sí consiguen localizar algún que otro documento relevante en la cabeza de la lista y los valores de precisión media (por encima de 0.12) indican que algún otro no debe estar muy mal situado (Tabla 220).

6 Sólo consideramos las consultas persistentes que tienen traducción y que, por lo tanto, han podido ser evaluadas.

Mejor consulta

De las dieciocho consultas seleccionadas de CACM, la que mejor comportamiento en media presenta, tanto en lo que se refiere sobre el conjunto de entrenamiento como sobre el conjunto de prueba, es la consulta 26 (véase la Tabla 226).

Sobre el conjunto de entrenamiento, las cuatro consultas seleccionadas sitúan 12 ó 13 de los 15 documentos relevantes posibles entre los primeros documentos recuperados. Además, la precisión media es muy homogénea, todas tienen más de 0.8 y menos de 0.86. Al evaluarlas sobre documentos no utilizados en el proceso de aprendizaje, su comportamiento es bastante bueno. Los valores de precisión media se mueven entre 0.387 y 0.540, lo que unido al número de documentos relevantes en las primeras posiciones (10, 10, 8, 6), aseguran que al usuario le será fácil acceder a la nueva información, sin malgastar excesivo tiempo en examinar documentos que sólo consigan desanimarlo.

La Figura 5.13 muestra la estructura de la segunda consulta persistente seleccionada.

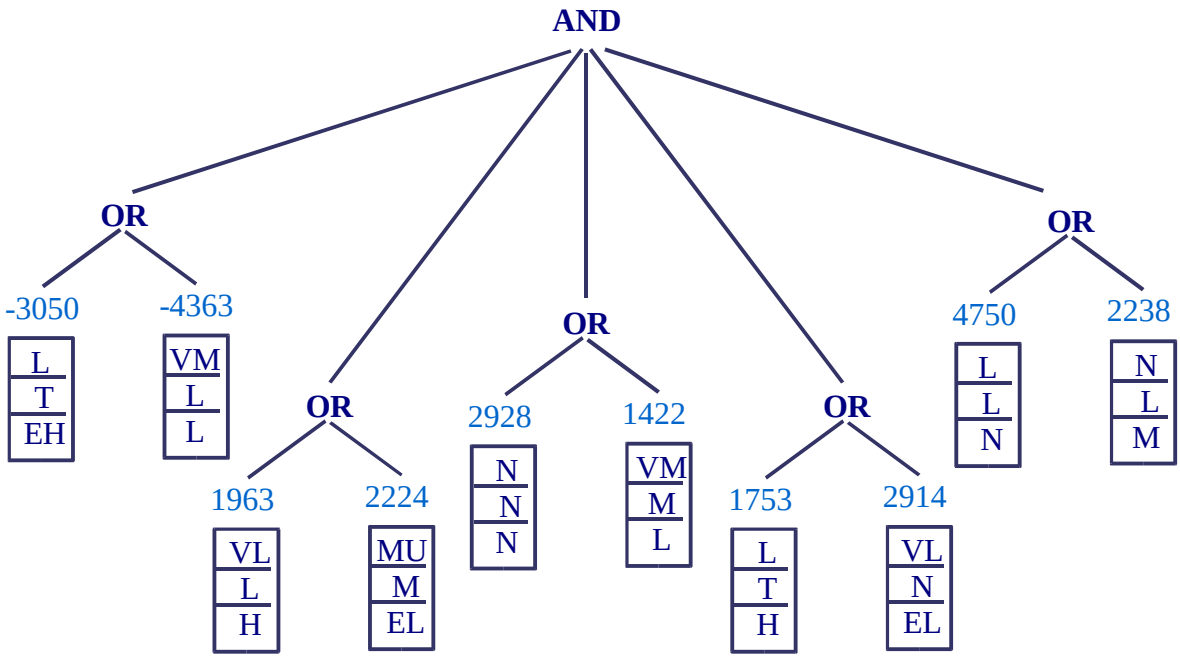


Figura 5.13: Mejor consulta persistente generada por el algoritmo de SPEA-AG sobre la colección CACM (Solución 2 de la consulta 26)

5.3.1.7.- Relación entre precisión media y medidas clásicas usando un umbral

Una mala elección del umbral para determinar qué documentos se recuperan y cuáles no, puede llevar a pensar que el SRI se comporta mal ante las consultas cuando en realidad lo está haciendo correctamente. Como hemos comentado anteriormente, utilizar la potencialidad del sistema para devolver los documentos ordenados en función de su RSV, nos puede ayudar a determinar cómo de bien se comporta el sistema en función de las posiciones que ocupen los documentos que nos aportan nueva información. Veamos a continuación la relación que existe entre los resultados obtenidos imponiendo un umbral y los obtenidos utilizando la capacidad de ordenación.

Las soluciones que no son capaces de recuperar ningún documento relevante al imponer un umbral, tampoco lo son de situar documentos relevantes en posiciones altas del conjunto total de documentos recuperados, como demuestran no sólo el 0 en el número de documentos relevantes en las *n* primeras posiciones, sino los bajos valores de precisión media. Es necesario pasar por muchos documentos no relevantes antes de llegar a alguno que aporte nueva información.

Sin embargo, cuando ocurre al contrario y la recuperación ha sido efectiva al 100% con los criterios clásicos imponiendo un umbral, lógicamente la precisión media alcanza su valor máximo.

Cuando la utilización del umbral deriva en una recuperación con excesivo ruido, los valores de precisión media son bajos y sólo unos pocos documentos se localizan en las primeras posiciones, a pesar de que con el umbral se recuperaban muchos relevantes. Lo único positivo que podemos sacar de esto es que sabemos que en las *n* primeras posiciones vamos a encontrar con seguridad un cierto número de relevantes. Así, en la consulta 23 (Tabla 171), al imponer un umbral se recuperan 188 documentos de los cuales 16 son relevantes, lo que supondría tener que analizar 188 documentos, para poder encontrar relevantes. Por su parte, al utilizar la precisión media (Tabla 208), sabemos que 3 documentos que interesan se van a encontrar al analizar los 17 primeros. Además, por el valor de precisión media podemos asegurar que en posiciones no muy lejanas también encontraremos documentos útiles a nuestras necesidades. De hecho, 8 documentos relevantes se encuentran entre los 30 primeros (2, 3, 4, 19, 21, 22, 26 y 30), lo que es un comportamiento muy deseable; y si queremos

equipararlo con la evaluación usando umbral, el decimosexto documento relevante aparece en la posición 188.

En general, al imponer un umbral para determinar los documentos recuperados, valores altos de precisión se traducen en que todos los documentos relevantes que se recuperan se sitúan en al inicio de la lista ordenada. Conforme desciende la precisión, también desciende la precisión media y el número de documentos interesantes en las primeras posiciones (véase la consulta 7 de CACM, Tablas 183 y 219).

5.3.1.8.- Resumen

Globalmente, podemos destacar los siguientes aspectos en los experimentos realizados con el algoritmo SPEA-AG sobre las bases Cranfield y CACM:

- El número de soluciones con diferente balance entre precisión y exhaustividad presentes en los frentes de los Paretos es pequeño en comparación con el tamaño de la población elitista. Este comportamiento se debe a que consideramos que dos soluciones son iguales si lo son en el espacio de objetivos, independientemente de la estructura que tengan.

- Consultas con un mayor número de documentos relevantes asociados consiguen Paretos con un mayor número de soluciones y, además, cubren una zona más amplia del espacio. Muestra de esto, aparte de los valores de las métricas, son los conjuntos de soluciones obtenidos. En ellos podemos observar como las consultas con más documentos relevantes derivan en una sola ejecución varias consultas con diferente balance precisión-exhaustividad mientras que, conforme disminuye el número de documentos relevantes, los conjuntos presentan menor diversidad de consultas persistentes, la mayoría de ellas con valores muy altos de precisión y exhaustividad.

- La eficacia de las consultas para obtener nuevos documentos relevantes, no conocidos durante el proceso de aprendizaje, es mayor para aquellas consultas que, en vez de presentar valores extremos de los criterios, consiguen equilibrarlos.

- Los experimentos realizados sobre Cranfield indican un alto grado de sobreaprendizaje. Se obtienen buenos resultados sobre el conjunto de entrenamiento, pero al evaluar las

consultas sobre datos no utilizados en el proceso de aprendizaje, con el fin de obtener nueva información, los resultados dejan bastante que desear.

- El uso de la potencialidad del SRI para devolver los documentos ordenados según su relevancia nos muestra como, en la mayoría de los casos, podemos encontrar un número aceptable de documentos que se adaptan a nuestras necesidades entre las primeras posiciones, lo que facilita el acceso a la nueva información.

5.4.- Enfoque Clásico para la Generación de Perfiles

En la Sección 5.2, hemos descrito nuestra propuesta para aprender perfiles representados como consultas Booleanas ponderadas con pesos lingüísticos y, en la 5.3, hemos presentado los resultados obtenidos en la experimentación con las bases Cranfield y CACM. Sin embargo, la estructura más común de perfil es la denominada "*bag of words*", la cual consiste en un conjunto de palabras clave que representan los intereses del usuario. Por tanto, parece necesario comparar nuestra propuesta con métodos clásicos de creación de perfiles, para ver como de competitiva es.

Como método de comparación, utilizaremos uno de los que mejor comportamiento presentan en la actualidad, basado en el modelo de espacio vectorial y en la teoría de probabilidades. En concreto, consideramos el valor de selección de Robertson (VSR) [148] como fórmula para la creación de los perfiles y la función Okapi BM25 [150] como función de similitud para el emparejamiento del perfil y los documentos.

5.4.1.- Valor de Selección de Robertson

Robertson [147][148] ideó un método de selección de términos basado en la teoría de la probabilidad. El valor de selección de Robertson (VSR) se define como:

$$VSR = (p_i - q_i) RW_i \quad (1)$$

donde $p_i = P(w_i = 1|R)$ es la probabilidad de la presencia del término i supuesto que un documento es relevante, y $q_i = P(w_i = 1|\ \overline{R})$, es la probabilidad de la presencia del término i cuando un documento no es relevante. Los pesos de relevancia para cada término i, RW_i, se calculan como:

$$RW_i = \log \frac{p_i(1-q_i)}{q_i(1-p_i)}$$

RW_i se puede obtener en base al modelo probabilístico binario independiente y las reglas Bayesianas de decisión [176][148][149]. De acuerdo a [148], q_i se puede despreciar con toda seguridad en la ecuación (1), ya que o q_i es mucho más pequeña que p_i, o RW_i es mucho más pequeña por grande que sea q_i. De esta forma, la fórmula final del VSR se puede simplificar a:

$$VSR = p_i * RW_i = p_i * \log \frac{p_i(1-q_i)}{q_i(1-p_i)}$$

TABLAS DE CONTINGENCIA

Las tablas de contingencia son una de las técnicas estadísticas más usadas en el análisis de datos y que pueden ser usadas para modelar fórmulas basadas en probabilidades, como es el caso del valor de selección de Robertson.

Para nuestro problema, es necesario calcular una tabla de contingencia para cada término j en el conjunto de entrenamiento. Cada una de estas tablas tiene la forma mostrada en la Figura 5.14.

	Relevantes	***No-relevantes***	
término *j* = 1	A	B	A+B
término *j* = 0	C	D	C+D
	A+C	B+D	N

Figura 5.14: Tabla de contingencia del término j

Aquí, *A* es el número de documentos relevantes en los que aparece el término *j*, *B* es el número de documentos no relevantes en los que aparece el término *j*, *C* es el número de documentos relevantes en los que no aparece el término *j* y *D* es el número de documentos no relevantes en los que término *j* no aparece. *N* se define como la suma de *A*, *B*, *C* y *D*.

De acuerdo a la información contenida en la tabla de contingencia mostrada en la Figura 5.14, $p_i = A / (A+C)$ y $q_i = B / (B+D)$, e ignorando A+C que es una constante para todos los términos, el VSR se puede simplificar a:

$$VSR = A * \log \frac{A*D}{B*C}$$

Todos los términos de los documentos relevantes se ordenan de manera descendente de acuerdo a su VSR. Para un tamaño de perfil (p.e., 200), los 200 términos con mayor valor de VSR y sus pesos serán incluidos en el perfil. El tamaño del perfil hace referencia al número de términos (junto con sus pesos) que es necesario incluir para describir las preferencias del usuario de la mejor manera posible.

5.4.2.- Función de Similitud

El valor de selección de Robertson, junto con la simplificación basada en la tabla de contingencia, nos permite determinar que términos son elegidos para formar parte del perfil. Sin embargo, para poder medir la eficacia de estos perfiles necesitamos una función que nos permita medir el grado de similitud o concordancia entre ellos y los nuevos documentos.

La función de similitud elegida para este fin es Okapi BM25 [150], que se define de la siguiente manera:

$$\mathrm{Sim}_O(Q,D) = \sum_{T \in P} \frac{3 \times tf}{0.5 + 1.5 \times \frac{length}{length_{avg}}} \times \log \frac{N - df + 0.5}{df + 0.5} \times QTW$$

donde tenemos las siguientes equivalencias:

tf	frecuencia del término en el documento D
QTW	estrategia de ponderación del término en la consulta Q. Se suele utilizar tf como valor de la estrategia de ponderación
df	número de documentos en la colección en los que él término está presente
length	longitud del documento D (número de término que lo representan)
$length_{avg}$	longitud media de los documentos en la colección

5.4.3.- Evaluación de los perfiles

Puesto que el método utilizado está basado en el modelo de espacio vectorial, presenta la potencialidad de poder devolver los documentos ordenados de acuerdo a su relevancia, por lo que sería interesante que el mecanismo de evaluación fuese capaz de medir también esta capacidad.

Para ello utilizamos la precisión media a once niveles de exhaustividad (véase Sección 1.4), lo que nos permitirá, además, una comparación equitativa con nuestra propuesta, ya que ésta fue una de las medidas utilizada para su evaluación.

5.4.4.- Experimentación y Análisis de Resultados

Esta sección se encarga de describir los experimentos realizados y los resultados obtenidos por el algoritmo VSR-OKAPI. Los experimentos se realizarán sobre las bases documentales Cranfield y CACM, utilizando el entorno experimental considerado en el Apéndice A.1.

Hemos ejecutado el algoritmo clásico de generación de perfiles (VSR-OKAPI) una única vez para cada una de las 35 consultas consideradas, ya que no existe componente aleatoria alguna en el mismo. Los valores empleados para los distintos parámetros del algoritmo se recogen en la Tabla 5.1.

Parámetros	Valores
Tamaño del perfil	10
Valor de QTW	VSR

Tabla 236: Valores de parámetros considerados para el algoritmo VSR-OKAPI

Hemos considerado un tamaño de perfil igual a 10 (número de términos que lo componen) para que la comparación con nuestra propuesta sea equiparable (las consultas generadas por ella pueden contener un máximo de 10 términos, véase la Tabla 126).

5.4.4.1.- Resultados obtenidos con la colección Cranfield

La Tabla 5.112 presentan los resultados obtenidos para cada una de las diecisiete consultas de Cranfield consideradas (véase el Apéndice A.1), sobre el conjunto de entrenamiento (parte izquierda de la tabla) y sobre el conjunto de prueba (parte derecha).

Dentro de cada tabla, y de izquierda a derecha, las columnas recogen los datos siguientes:

i. Número de consulta

ii. Valores correspondientes a la evaluación de las consultas persistentes sobre el conjunto

de entrenamiento: Precisión media, Número de documentos relevantes en la base, Número de documentos relevantes recuperados en las *n* primeras posiciones, siendo *n* igual al número de documentos relevantes en la base.

iii. Valores correspondientes a la evaluación de las consultas persistentes sobre el conjunto de prueba: Precisión media, Número de documentos relevantes en la base, Número de documentos relevantes recuperados en las *n* primeras posiciones, siendo *n* igual al número de documentos relevantes en la base.

	ENTRENAMIENTO			***PRUEBA***		
	Pm	***#D_rel***	***#rel_n***	***Pm***	***#D_rel***	***#rel_n***
C. 1	0.510	14	6	0.484	15	6
C. 2	0.522	12	5	0.789	13	9
C. 3	0.945	4	3	0.888	5	4
C. 7	0.909	3	2	0.162	3	1
C. 8	0.499	6	4	0.141	6	0
C. 11	0.864	4	3	0.615	4	3
C. 19	0.970	5	4	0.604	5	4
C. 23	0.475	16	7	0.357	17	7
C. 26	0.773	3	2	0.010	4	0
C. 38	0.442	5	1	0.056	6	1
C. 39	0.569	7	3	0.021	7	0
C. 40	0.569	6	3	0.052	7	0
C. 47	0.768	7	5	0.438	8	3
C. 73	0.616	10	5	0.268	11	2
C. 157	0.532	20	10	0.298	20	7
C. 220	0.725	10	6	0.544	10	5
C. 225	0.370	12	5	0.099	13	2

Tabla 237: Resultados del algoritmo de VSR-OKAPI para las consultas de Cranfield

Análisis de resultados

A continuación, analizaremos los resultados presentados en la tabla anterior. Como hemos venido haciendo, comentaremos los resultados de cada grupo de consultas (consultas con más de 20 documentos relevantes y consultas que presentan menos de 15 documentos relevantes, Tabla 181) de acuerdo a la capacidad de las consultas para situar a los documentos relevantes en las primeras posiciones.

Eficacia de las consultas con más de 20 documentos relevantes

La evaluación sobre el conjunto de entrenamiento de los perfiles generados pone de manifiesto un comportamiento adecuado, como demuestran los valores de precisión media de alrededor de 0.5, así como el alto número de documentos que aportan información situados en la cabeza de la lista. La proporción de éstos se encuentra, así mismo, sobre el 50%.

Cuando los perfiles se comparan contra documentos no utilizados en el proceso de creación, los resultados siguen siendo muy prometedores. De hecho, cinco de las siete consultas consiguen una precisión media alrededor de 0.4. Por su parte, las consultas 2 y 225, destacan positiva y negativamente, respectivamente. La primera, con una precisión media de 0.789 y 10 de los 13 documentos relevantes en la cabeza del conjunto de recuperados; y la segunda, con una precisión de 0.099 y solamente 2 de los 13 documentos que aportan información situados en la primeras posiciones.

Eficacia de las consultas con menos de 15 documentos relevantes

El comportamiento de los perfiles que representan las necesidades de información indicadas por consultas con un número pequeño de documentos relevantes, al ser evaluados sobre los mismos conjuntos que se utilizaron para su creación, es muy bueno. Podemos ver como las precisiones medias se localizan bien cercanas al máximo (1), pero sin llegar a él, o bien, alrededor de 0.5; siendo ambos comportamientos aceptables. La proporción de documentos relevantes es, además, muy elevada.

Sin embargo, cuando estos mismos perfiles se evalúan sobre documentos de los que no se tiene ningún conocimiento anterior, nos encontramos con que seis de las diez consultas presentan valores de precisión media muy bajos (no llegando a 0.05 o superando ligeramente el 0.1). Además, la proporción de documentos relevantes al comienzo del conjunto de

recuperados es prácticamente igual a 0. Este comportamiento dificulta seriamente el acceso a nueva información por parte del usuario.

Por otra parte, el comportamiento de las cuatro consultas restantes (3, 11, 19 y 47) es muy alentador, al presentar valores de precisión media por encima de 0.5 y una proporción mayor de número de documentos que satisfacen las necesidades de información.

Mejor consulta

Como ha venido siendo habitual en todos las experimentaciones realizadas a lo largo de esta memoria con la colección Cranfield, la consulta con mejor comportamiento ha sido la consulta 3. Tanto en entrenamiento como en prueba, consigue que todos los documentos relevantes que tiene asociados menos uno, estén situados entre los primeros. Además, los altos valores de precisión media nos indican que el documento que falta debe estar muy cerca de las posiciones de cabeza.

5.4.4.2.- Resultados obtenidos con la colección CACM

La Tabla 5.113 presentan los resultados obtenidos para cada una de las dieciocho consultas de CACM consideradas (véase el Apéndice A.1), sobre el conjunto de entrenamiento (parte izquierda de la tabla) y sobre el conjunto de prueba (parte derecha).

	ENTRENAMIENTO			***PRUEBA***		
	Pm	***#D_rel***	***#rel_n***	***Pm***	***#D_rel***	***#rel_n***
C. 4	0.763	6	4	0.138	6	0
C. 7	0.673	14	9	0.496	14	7
C. 9	0.723	4	2	0.182	5	1
C. 10	0.747	17	12	0.676	18	13
C. 14	0.847	22	16	0.746	22	14
C. 19	0.625	5	2	0.024	6	0
C. 24	0.751	6	4	0.293	7	2
C. 25	0.540	25	12	0.525	26	13
C. 26	0.880	15	12	0.761	15	10
C. 27	0.631	14	8	0.443	15	5
C. 40	0.688	5	3	0.393	5	2
C. 42	0.595	10	5	0.310	11	4
C. 43	0.374	20	10	0.498	21	10
C. 45	0.579	13	7	0.292	13	6
C. 58	0.710	15	10	0.265	15	7
C. 59	0.456	21	10	0.548	22	15
C. 60	0.575	13	7	0.339	14	4
C. 61	0.676	15	10	0.348	16	6

Tabla 238: Resultados del algoritmo de VSR-OKAPI para las consultas de CACM

Análisis de resultados

A continuación, analizaremos los resultados presentados en la tabla anterior. Como hemos venido haciendo, comentaremos los resultados de cada grupo de consultas (consultas con más de 30 documentos relevantes, consultas que presentan entre 21 y 30 documentos relevantes, y consultas con menos de 15 documentos relevantes, Tabla 200) de acuerdo a la capacidad de las consultas para situar los documentos relevantes en las primeras posiciones

Eficacia de las consultas con más de 30 documentos relevantes

Las precisiones medias alcanzadas por las consultas incluidas en este grupo sobre el conjunto de entrenamiento son variadas, moviéndose entre el 0.374 de la 43 y el 0.847 de la consulta 14. La proporción de documentos relevantes entre las primeras posiciones es superior al 0.5.

De igual manera, los altos valores de precisión media (en torno a 0.5) alcanzados al emparejar perfiles y documentos de prueba nos indican que no aparecen muchos documentos irrelevantes en las primeras posiciones del conjunto de recuperados, lo que facilitará el acceso a la nueva información por parte de los usuarios. Prueba de esto es, también, la proporción de documentos que aportan información en las primeras posiciones del conjunto de documentos final.

Eficacia de las consultas que presentan entre 21 y 30 documentos relevantes

Estas consultas tienen un comportamiento muy similar al descrito en la sección anterior, tanto sobre el conjunto de entrenamiento como sobre el de prueba, aunque sobre este último tanto los valores de precisión media como la proporción de documentos relevantes en las primeras posiciones se ven algo reducidos. Sobre el conjunto entrenamiento se puede considerar algo mejor, puesto que ninguna consulta baja de 0.55 en lo que respecta a precisión media.

Eficacia de las consultas con menos de 15 documentos relevantes

La evaluación sobre el conjunto de entrenamiento de este grupo de consultas se mantiene en línea con el resto, caracterizada por valores de precisión media entre 0.65 y 0.75 y una proporción de documentos relevantes situados entre los primeros de más del 50%.

Sobre el conjunto de prueba, la precisión media disminuye respecto a los apartados anteriores, localizándose alrededor de 0.2. Además, dos de las cinco consultas no consiguen situar ningún documento que aporte nueva información entre los primeros.

Mejor consulta

Con una precisión media de 0.880 y 12 documentos relevantes entre los 15 primeros, la consulta 26 es la que mejor comportamiento presenta sobre el conjunto de entrenamiento. Además, también consigue los mejores resultados sobre el conjunto de prueba, con una precisión media de 0.761 y 10 documentos relevantes entre los 15 primeros.

5.4.4.3.- Resumen

Globalmente, podemos destacar que este método clásico para la generación de perfiles tiene un comportamiento muy apropiado.

Los valores de precisión media obtenidos al evaluar los perfiles sobre el conjunto de documentos utilizados para su generación son bastante buenos. Se mantienen por encima de 0.4 en todos los casos menos dos (consulta 43 de CACM y 225 de Cranfield), que aún así se sitúan por encima de 0.3.

Cuando la evaluación tiene lugar sobre documentos totalmente desconocidos, podemos encontrar dos tipos de comportamientos:

a) Consultas con valores de precisión media aceptables. Este tipo de consultas presentan valores por encima de 0.2 y, en la mayoría de los casos alrededor de 0.5, lo que facilita al usuario el acceso a la nueva información.

b) Consultas con valores de precisión muy bajos. En estos casos, los valores se encuentran por debajo de 0.2 e incluso por debajo de 0.1, lo que indica que muchos documentos no relevantes ocupan posiciones de importancia en el conjunto final de recuperados. Además, en algunas ocasiones, ningún documento de los que aportaría nueva información es situado entre los primeros. Esto dificulta en gran medida el acceso a los nuevos documentos, puesto que será necesario examinar muchos documentos que no interesan antes de encontrar alguno que sea útil.

5.5.- Caracterización de Perfiles como Consultas *versus* Representación Vectorial

Representar el perfil como una consulta persistente, en lugar de como la clásica estructura de "bag of words", nos proporciona:

- ☞ Más flexibilidad, al poder usar cualquier modelo de RI para formularla: Booleano, Booleano extendido, Lingüístico, ...
- ☞ Más expresividad, al usar términos y operadores Booleanos (Y, O, NO) para representar las necesidades de información, lo que es más interpretable para el ser humano.

Además, si representamos las consultas persistentes como consultas lingüísticas, dotamos de mayor interpretabilidad a los perfiles.

De acuerdo con todo esto, en este capítulo hemos presentado un algoritmo multiobjetivo capaz de aprender de manera automática consultas Booleanas con pesos lingüísticos, que caracterizan los perfiles; obteniéndose resultados prometedores.

Sin embargo, los métodos basados en el modelo de espacio vectorial, como el descrito en la Sección 5.4 (VSR-OKAPI) dan los mejores resultados a la hora de generar perfiles.

Por tanto debemos preguntarnos, **"¿Es nuestra propuesta competitiva con los métodos clásicos de generación de perfiles?"**

Antes de seguir, es necesario establecer un mecanismo de comparación. Con nuestro algoritmo, cada consulta tiene asociada entre una y cinco soluciones diferentes. Como se comentó en la Sección 5.3.1, nuestra propuesta se ha ejecutado cinco veces, y ante la imposibilidad de analizar todas las consultas existentes (en cada ejecución se generan varias soluciones con diferente balance precisión-exhaustividad), se ha seleccionado un conjunto de soluciones representativas de la fusión de los Paretos correspondientes a cada ejecución. Sin embargo, VSR-OKAPI presenta una solución por consulta, al ejecutarse una sola vez (al no presentar componente aleatoria alguna) y generar únicamente una solución en cada ejecución.

Con el fin de que la comparación sea más sencilla y, buscando comparar un único resultado por algoritmo, hemos derivado dos únicos resultados para cada una de las consultas,

consiguiendo, de esta forma, dos grupos de resultados para nuestro algoritmo. Uno formado por resultados medios y otro por los mejores resultados sobre el conjunto de prueba. En las subsecciones siguientes se muestran estos resultados, así como el procedimiento seguido para obtenerlos.

5.5.1.- Resultados Medios del Algoritmo SPEA-AG

Las Tablas 5.114 y 5.115 muestran los resultados medios correspondiente a las mejores consultas persistentes en precisión media, derivadas por el algoritmo SPEA-AG para Cranfield y CACM, respectivamente.

Para conseguir estos resultados, se ha seleccionado del frente del Pareto generado en cada una de las ejecuciones, la consulta persistente con mayor valor de precisión media, en total cinco consultas. Finalmente, hemos calculado la media (para cada parámetro) de estas cinco consultas. Este procedimiento se ha repetido para cada una de las 35 consultas. En la Figura 5.15 se puede ver el proceso de manera gráfica.

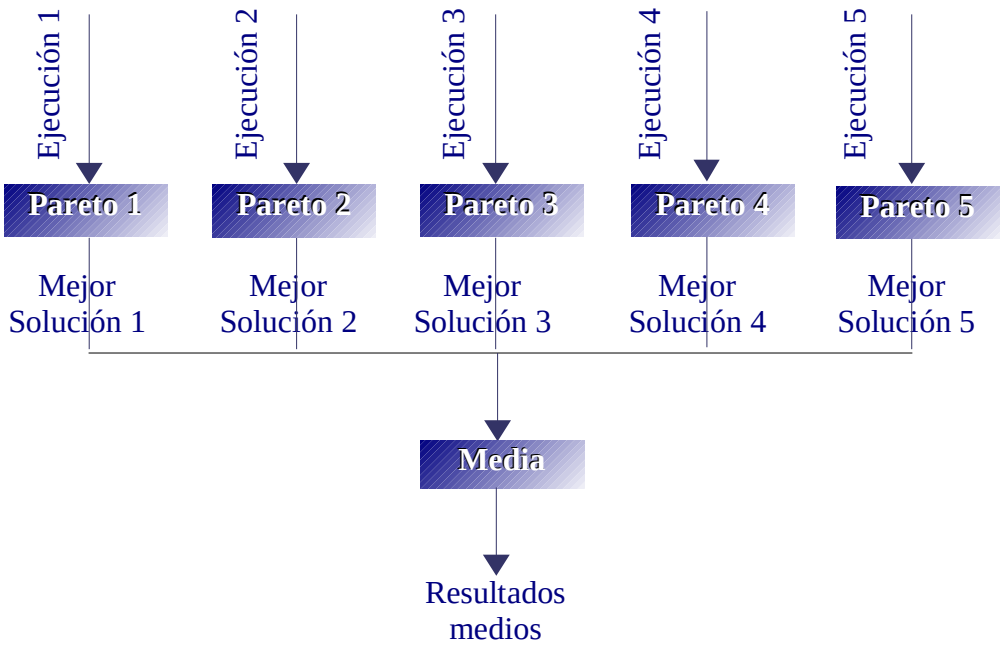


Figura 5.15: Proceso para obtener los resultados medios del algoritmo SPEA-AG

	ENTRENAMIENTO			*PRUEBA*		
	Pm	***#D_rel***	***#rel_n***	***Pm***	***#D_rel***	***#rel_n***
C. 1	0.528	14	7.8	0.070	15	0.75
C. 2	0.615	12	7.2	0.143	13	2.75
C. 3	0.980	4	3.8	0.330	5	2
C. 7	0.765	3	2.2	0.156	3	0.25
C. 8	0.657	6	3.8	0.024	6	0
C. 11	0.913	4	3.4	0.021	4	0.25
C. 19	0.931	5	4.2	0.034	5	0.2
C. 23	0.461	16	8	0.113	17	2.4
C. 26	0.820	3	2.4	0.006	4	0
C. 38	0.847	5	3.8	0.029	6	0.2
C. 39	0.704	7	5	0.051	7	0.25
C. 40	0.706	6	4	0.015	7	0.2
C. 47	0.684	7	4.4	0.089	8	0.5
C. 73	0.627	10	5.6	0.057	11	0.75
C. 157	0.464	20	9.4	0.112	20	2.2
C. 220	0.677	10	6.2	0.173	10	1.2
C. 225	0.515	12	6.2	0.078	13	0.75

Tabla 239: Resultados medios del algoritmo de SPEA-AG para las consultas de Cranfield

Podemos observar como los resultados son bastante buenos sobre el conjunto de entrenamiento. La precisión media se sitúa por encima de 0.5 salvo en las consultas 23 y 157 de Cranfield y la 25 de CACM.

La proporción de documentos útiles en las primeras posiciones del conjunto de recuperados es de la mitad en consultas con más de 20 documentos, en el caso de Cranfield, y de algo más en CACM. En consultas con un menor número de documentos relevantes, la proporción aumenta.

	ENTRENAMIENTO			*PRUEBA*		
	Pm	***#D_rel***	***#rel_n***	***Pm***	***#D_rel***	***#rel_n***
C. 4	0.823	6	4.4	0.052	6	0.2
C. 7	0.649	14	9	0.055	14	1.6
C. 9	0.946	4	3.8	0.145	5	1.2
C. 10	0.631	17	10	0.450	18	10.33
C. 14	0.740	22	15.8	0.381	22	11.25
C. 19	0.946	5	4.4	0.026	6	0
C. 24	0.774	6	4.6	0.046	7	0.25
C. 25	0.477	25	12.8	0.157	26	3.8
C. 26	0.751	15	10.8	0.410	15	7.5
C. 27	0.596	14	8	0.140	15	2
C. 40	0.828	5	3.8	0.211	5	0.66
C. 42	0.674	10	6.2	0.052	11	1.25
C. 43	0.523	20	10.6	0.225	21	5
C. 45	0.617	13	8.2	0.105	13	1
C. 58	0.603	15	8.8	0.072	15	0.5
C. 59	0.595	21	13	0.170	22	5.4
C. 60	0.555	13	7	0.118	14	3
C. 61	0.642	15	9.4	0.121	16	3

Tabla 240: Resultados medios del algoritmo SPEA-AG para las consultas de CACM

La parte derecha de las tablas nos muestra como los resultados sobre los conjuntos de prueba son bastante bajos en comparación con el conjunto de entrenamiento. Vemos como se produce un claro efecto de sobreaprendizaje. En Cranfield, salvo la consulta 3, el resto no suben de 0.15. Por otro lado, con CACM se consiguen resultados algo más aceptables, sobre todo en consultas con más de 30 documentos relevantes (todas por encima de 0.15). Destacan la consulta 10 y la 26 con más de 0.4.

5.5.2.- Mejores Resultados para SPEA-AG sobre el Conjunto de Prueba

Con cualquier algoritmo de aprendizaje de consultas, lo que buscamos es que se consiga aportar nueva información que satisfaga las necesidades de un usuario al evaluar dichas consultas sobre conjuntos diferentes de los usados en el proceso de aprendizaje. En vista de esto, las Tablas 5.116 y 5.117 presentan los resultados obtenidos por la consulta persistente con mayor valor de precisión media sobre el conjunto de prueba, para cada consulta de Cranfield y CACM, respectivamente. La mejor consulta se elige de entre las cinco consultas previamente seleccionadas para obtener los resultados medios. La Figura 5.16 muestra gráficamente la forma de obtener esta consulta.

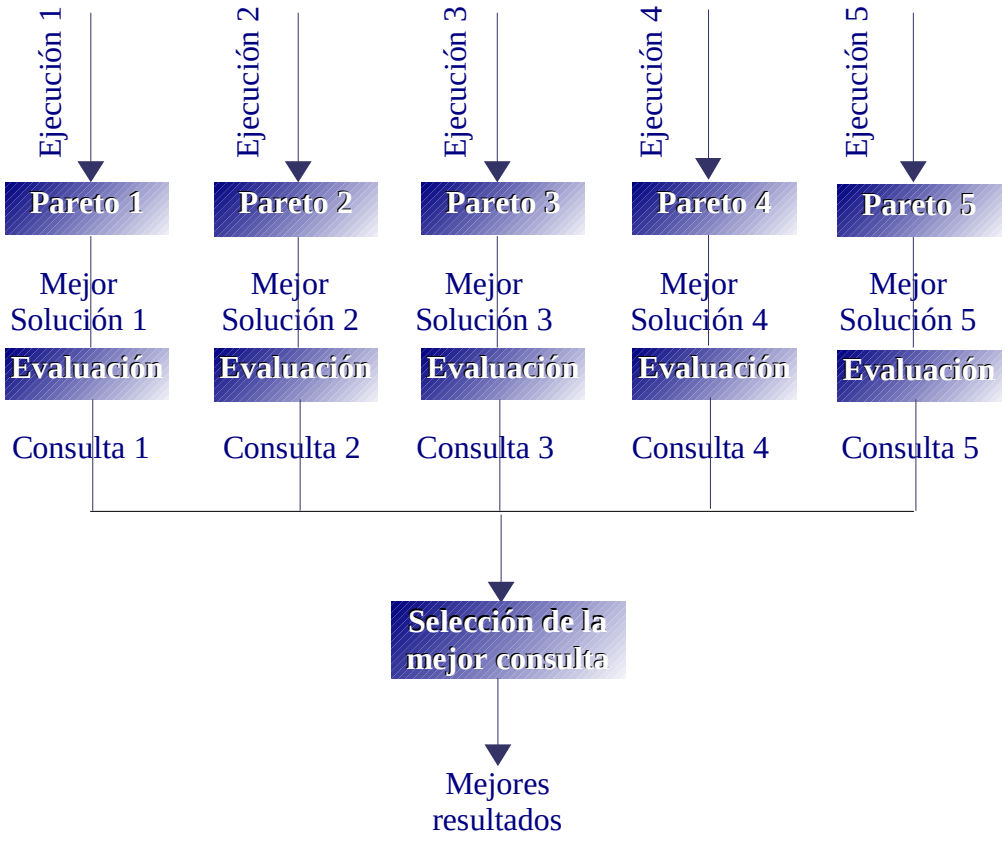


Figura 5.16: Proceso para obtener los mejores resultados del algoritmo SPEA-AG

	ENTRENAMIENTO			***PRUEBA***		
	Pm	***#D_rel***	***#rel_n***	***Pm***	***#D_rel***	***#rel_n***
C. 1	0.467	14	8	0.201	15	2
C. 2	0.725	12	7	0.289	13	4
C. 3	1.000	4	4	0.542	5	3
C. 7	0.637	3	2	0.245	3	1
C. 8	0.652	6	4	0.045	6	0
C. 11	0.729	4	3	0.072	4	1
C. 19	1.000	5	5	0.144	5	1
C. 23	0.426	16	8	0.213	17	5
C. 26	0.818	3	2	0.007	4	0
C. 38	0.912	5	4	0.050	6	0
C. 39	0.821	7	6	0.134	7	1
C. 40	0.549	6	3	0.033	7	1
C. 47	0.740	7	5	0.237	8	1
C. 73	0.748	10	7	0.080	11	1
C. 157	0.454	20	11	0.422	20	7
C. 220	0.647	10	5	0.349	10	3
C. 225	0.484	12	5	0.126	13	1

Tabla 241: Mejores resultados del algoritmo de SPEA-AG sobre el conjunto de prueba para las consultas de Cranfield

Las consultas seleccionadas son las mejores en precisión media al ser evaluadas sobre los conjuntos de prueba, lo que no supone que tengan que serlo, necesariamente, también sobre el conjunto de entrenamiento. De hecho, en muchas ocasiones, la mejor consulta sobre el conjunto de prueba es una de las que presentan un comportamiento medio en el conjunto de entrenamiento o incluso el peor. Esto nos indica que no siempre lo interesante son los mejores resultados en el conjunto de entrenamiento (se puede estar produciendo sobreaprendizaje), sino que son preferibles valores intermedios.

En general, los mejores resultados sobre el conjunto de prueba superan con creces a los medios, alrededor del doble.

	ENTRENAMIENTO			***PRUEBA***		
	Pm	***#D_rel***	***#rel_n***	***Pm***	***#D_rel***	***#rel_n***
C. 4	0.646	6	3	0.148	6	1
C. 7	0.658	14	8	0.150	14	5
C. 9	1.000	4	4	0.237	5	2
C. 10	0.670	17	12	0.534	18	11
C. 14	0.818	22	16	0.581	22	14
C. 19	1.000	5	5	0.087	6	0
C. 24	0.639	6	4	0.096	7	1
C. 25	0.471	25	11	0.185	26	2
C. 26	0.632	15	9	0.466	15	7
C. 27	0.583	14	7	0.190	15	3
C. 40	0.894	5	4	0.333	5	1
C. 42	0.676	10	6	0.112	11	2
C. 43	0.565	20	11	0.320	21	7
C. 45	0.715	13	10	0.211	13	3
C. 58	0.643	15	10	0.116	15	1
C. 59	0.595	21	13	0.218	22	9
C. 60	0.612	13	8	0.185	14	5
C. 61	0.568	15	9	0.198	16	3

Tabla 242: Mejores resultados del algoritmo de SPEA-AG sobre el conjunto de prueba para las consultas de CACM

En la colección Cranfield, encontramos dos comportamientos diferentes: consultas cuyos valores de precisión media se sitúan por encima de 0.2 y alcanzan un máximo de 0.542 (p.e., consulta 3), y consultas con precisiones medias por debajo de 0.15, siendo la peor la 26 (0.007).

En el caso de CACM, salvo tres consultas (la 19, 24 y 40), el resto presentan valores por encima de 0.1. Las consultas con mayor número de documentos relevantes consiguen valores más altos de precisión media. El valor máximo lo consigue la consulta 14 con una precisión media de 0.581, seguida de cerca por la consulta 10 (0.534) y la 26 (0.466).

5.5.3.- VSR-OKAPI *versus* SPEA-AG sobre Cranfield

Evaluación sobre el conjunto de entrenamiento

Si comparamos los resultados obtenidos por nuestra propuesta sobre el conjunto de entrenamiento con los conseguidos por el método clásico (Tabla 5.114), observamos que nuestro método es competitivo. En once de las diecisiete consultas, la propuesta multiobjetivo consigue mejores resultados que VSR-OKAPI. En las que no, las diferencias son muy pequeñas.

Una de las consultas donde la diferencia entre un algoritmo y otro es más acusada a nuestro favor, es la 38. Mientras que con VSR-OKAPI se consigue un valor de 0.442, situando un único documento relevante en las primeras posiciones, la propuesta multiobjetivo sube hasta 0.847 con una media de 3.8 documentos sobre 5 en la cabeza del conjunto final de recuperados.

Las consultas en las que nuestro algoritmo presenta peor comportamiento son la 23, 157, 220, 7, 19 y 47. La mayor diferencia, en este caso en nuestra contra, es de 0.145, correspondiente a la consulta 7. El método basado en VSR y OKAPI consigue una precisión media de 0.91, con una proporción de 2 documentos sobre el total de 3, mientras que SPEA-AG se queda en 0.765 aunque la proporción de documentos es prácticamente la misma.

Como hemos comentado antes, los mejores resultados sobre el conjunto de prueba no tienen por qué coincidir con los mejores en entrenamiento. De hecho, al hacer la comparación con las consultas con mejor resultado sobre el conjunto de nuevos documentos (parte izquierda de las Tablas 5.112 (VSR-OKAPI) y 5.116 (SPEA-AG)), vemos como OKAPI es mejor en entrenamiento en ocho de las diecisiete consultas.

Las mayores diferencias a nuestro favor, las sigue presentando la consulta 38, y en nuestra contra la 7.

Las Figuras 5.17 y 5.18, muestran de manera gráfica el comportamiento de los dos algoritmos sobre el conjunto de entrenamiento. La primera presenta la comparación de los valores de precisión media correspondientes a VSR-OKAPI (Tabla 5.110) y a los resultados medios de SPEA-AG (Tabla 5.114). Por su parte, la Figura 5.18 muestra la misma

comparativa pero utilizando como valores de nuestro algoritmo los correspondientes a las consultas con mejor comportamiento sobre el conjunto de prueba (Tabla 5.116).

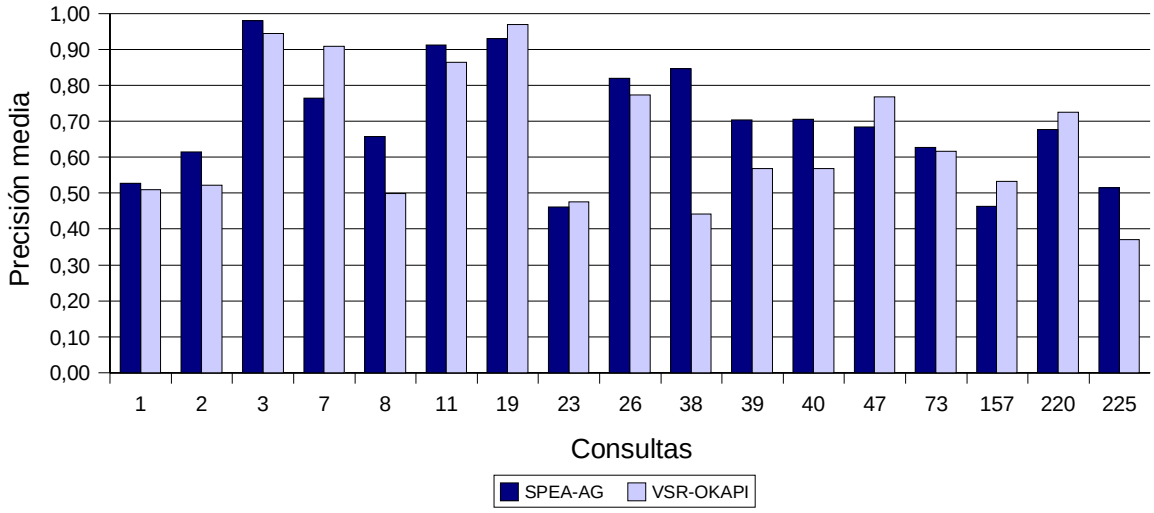


Figura 5.17: Comparativa de SPEA-AG (considerando resultados medios) y VSR-OKAPI sobre el conjunto de entrenamiento para las consultas de Cranfield

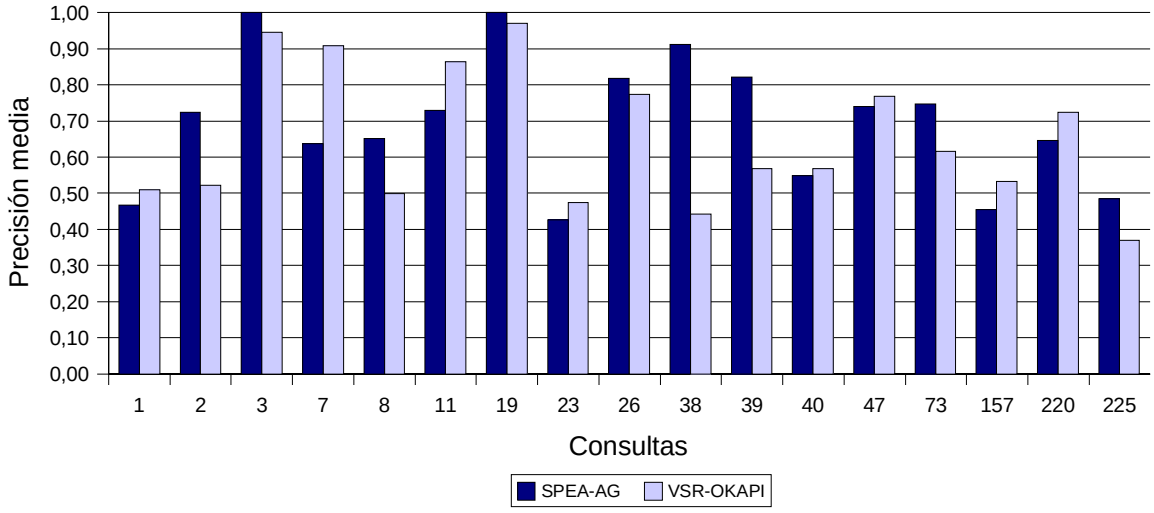


Figura 5.18: Comparativa de SPEA-AG (considerando mejores resultados) y VSR-OKAPI sobre el conjunto de entrenamiento para las consultas de Cranfield

Evaluación sobre el conjunto de prueba

Centrándonos ahora en la evaluación sobre el conjunto de prueba, vemos que comparando los resultados medios (parte derecha de la Tabla 5.114) con VSR-OKAPI, éste presenta mucho mejor comportamiento. Podemos destacar, sin embargo, tres consultas, una en la que obtenemos mejores resultados (0.05 frente a 0.02) y otras dos donde las diferencias son mínimas (0.078 frente a 0.098 y 0.156 frente a 0.161), en comparación con el resto de consultas.

En el resto de consultas, VSR-OKAPI obtiene mucho mejores resultados que nuestro método. Así, por ejemplo, en la consulta 2, mientras que con nuestra propuesta sólo conseguimos una media de 0.143 para la precisión media, OKAPI alcanza un 0.788, situando, además, 9 documentos relevantes entre los 13 primeros, frente a los 2.75 de SPEA-AG.

Si la comparación la hacemos sobre los mejores resultados de precisión media (parte derecha de la Tabla 5.116), vemos que SPEA-AG consigue resultados ligeramente mejores que VSR-OKAPI en cinco consultas (157, 225, 7, 38 y 39). Por ejemplo, la consulta 157 tiene asociada una precisión media de 0.422 en el caso del algoritmo SPEA-AG y de 0.298 en el caso de VSR-OKAPI. La proporción de documentos en las primeras posiciones es la misma para los dos (11 de 20), lo que sugiere que la diferencia estriba en las posiciones que ocupan los restantes 13 documentos relevantes.

En el resto de consultas, las diferencias se reducen respecto al caso medio, aunque siguen siendo significativas. Por ejemplo, la consulta 3 tiene un valor máximo de 0.542 en el caso multiobjetivo frente al 0.888 del método clásico.

Los comportamientos que acabamos de describir pueden verse de manera gráfica en las Figuras 5.19 y 5.20. Ambas figuras representan la misma comparativa que la descrita para las Figuras 5.17 y 5.18, respectivamente, sólo que sobre el conjunto de prueba.

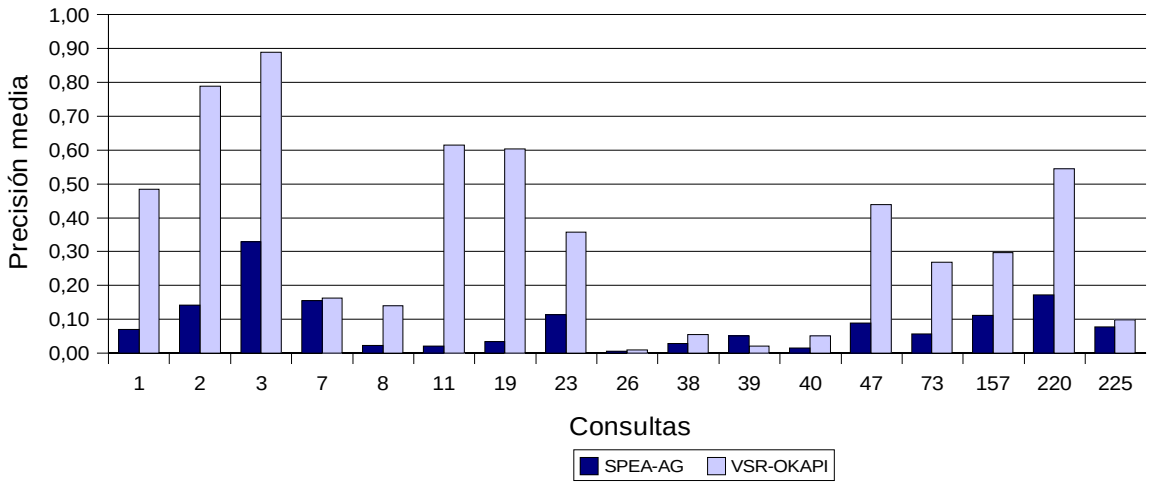


Figura 5.19: Comparativa de SPEA-AG (considerando resultados medios) y VSR-OKAPI sobre el conjunto de prueba para la consultas de Cranfield

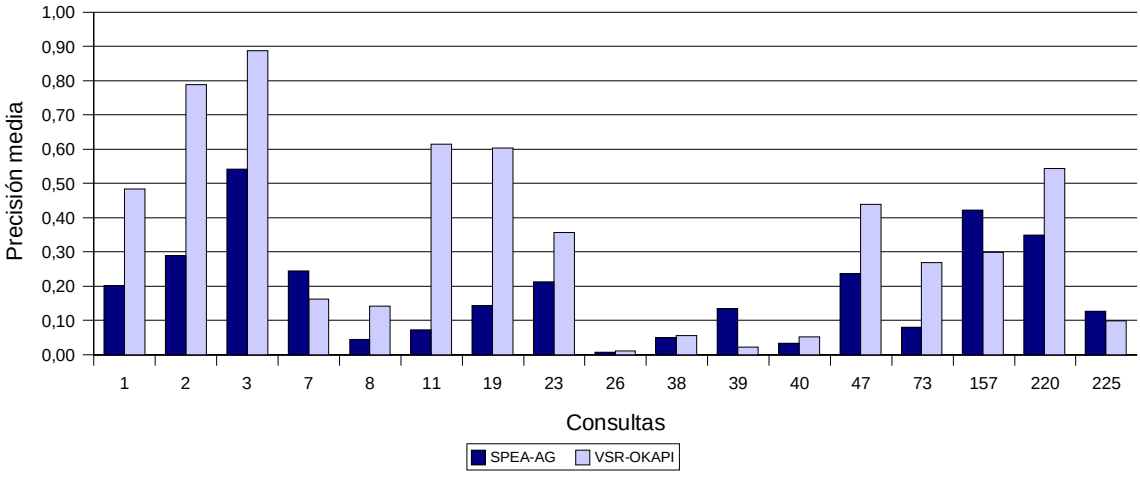
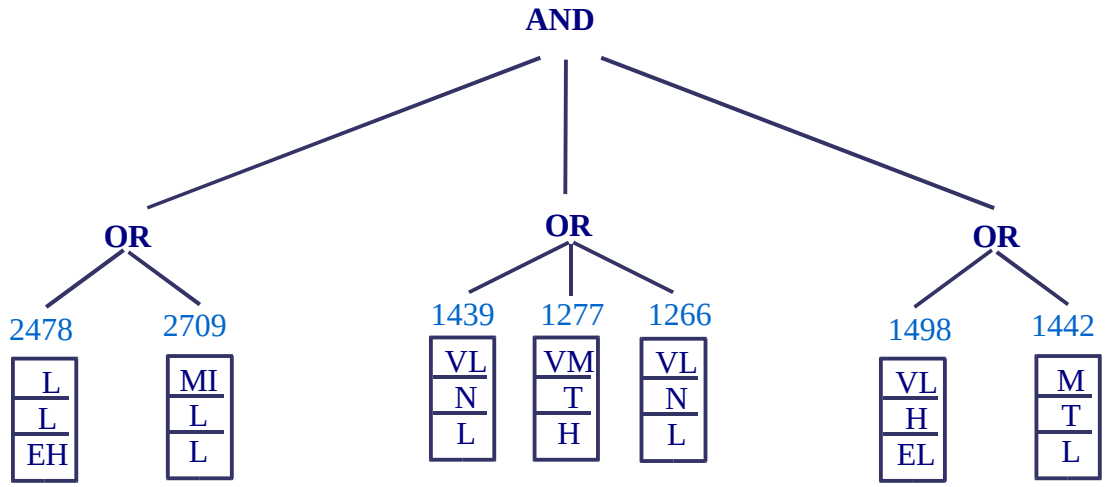


Figura 5.20: Comparativa de SPEA-AG (considerando mejores resultados) y VSR-OKAPI sobre el conjunto de prueba para las consultas de Cranfield

Composición de los perfiles

Una factor importante que tenemos que tener en cuenta en la comparación entre SPEA-AG y VSR-OKAPI es la composición de los perfiles generados. En el caso del algoritmo clásico los perfiles están formados, únicamente, por los *n* términos con mayor valor VSR, sin tenerse ningún conocimiento sobre la relación que existe o puede existir entre ellos. Por el contrario, los perfiles generados por nuestra propuesta están formados por una serie de términos y un conjunto de operadores Booleanos que determinan la relación existente entre los términos seleccionados.

Las Figuras 5.21 y 5.22 muestran la composición de los perfiles obtenidos por los dos métodos sobre el conjunto de prueba para las consultas 2 y 157 de Cranfield, respectivamente.



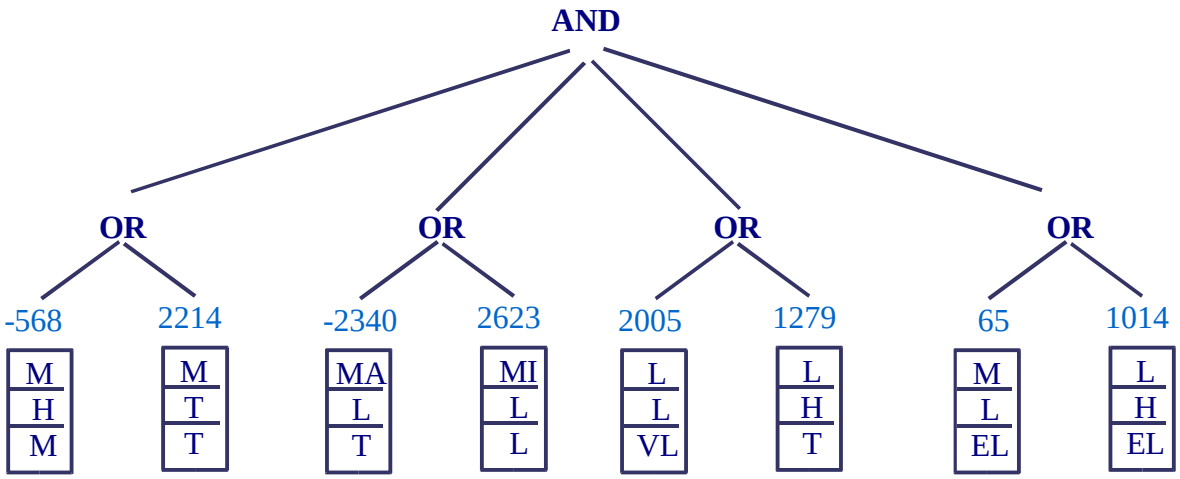
a) Perfil generado por el algoritmo SPEA-AG

1794 (11.56)	2437 (9.66)	2526 (8.67)
693 (8.35)	847 (7.69)	1887 (7.52)

b) Perfil generado por el algoritmo VSR-OKAPI

Figura 5.21: Perfiles generados por SPEA-AG y VSR-OKAPI para la consulta 157 de Cranfield sobre el conjunto de prueba

En el caso de la consulta 2, VSR-OKAPI consigue valores de precisión media bastante superiores a los de nuestra propuesta, 0.789 frente a 0.289 (Figura 5.20). Sin embargo, como demuestra la Figura 5.21, la composición del perfil generado por SPEA-AG es mucho más clara que la del perfil generado por el método clásico, ya que se han aprendido, a la vez, los términos que los forman y las relaciones existente entre ellos.



a) Perfil generado por el algoritmo SPEA-AG

1116 (45.33)	2623 (45.06)	146 (26.39)	2929 (24.17)	1674 (23.02)
557 (20.79)	3 (17.39)	1952 (16.63)	1973 (16.09)	1279 (12.23)

b) Perfil generado por el algoritmo VSR-OKAPI

Figura 5.22: Perfiles generados por SPEA-AG y VSR-OKAPI para la consulta 2 de Cranfield sobre el conjunto de prueba

A pesar de que nuestra propuesta obtiene sobre el conjunto de prueba peores resultados que VSR-OKAPI en la mayoría de los casos, la representación del perfil como una consulta (términos + operadores) nos aporta mayor expresividad, al aprender no sólo los términos que compondrán el perfil, sino también los operadores Booleanos que los relacionan. Además, la representación de los pesos como etiquetas lingüísticas (SPEA-AG) es más interpretable que el uso de pesos numéricos (VSR-OKAPI).

5.5.4.- VSR-OKAPI *versus* SPEA-AG sobre CACM

Evaluación sobre el conjunto de entrenamiento

Los resultados obtenidos sobre el conjunto de entrenamiento son bastante parecidos para ambos algoritmos cuando consideramos los resultados medios de SPEA-AG. Cada uno de ellos supera al otro en la mitad de las consultas (véanse las Tablas 5.113 y 5.115), pero sin que las diferencias sean demasiado elevadas en ninguno de los dos casos, permitiéndonos competir con VSR-OKAPI.

Las consultas persistentes generadas por nuestra propuesta multiobjetivo obtienen, en todos los casos, valores de precisión media superiores a 0.5, con un máximo de 0.946 para las consultas 9 y 19, y un mínimo de 0.477 correspondiente a la consulta 25. En el caso de VSR-OKAPI, los valores máximo y mínimo son 0.880 y 0.374, correspondientes a las consultas 26 y 25, respectivamente.

De igual forma, la proporción de documentos que aportan información, situados en las primeras posiciones del conjunto de recuperados, es también parecida, como demuestran los resultados recogidos en las Tablas 5.113 y 5.115.

Entre las consultas en las que SPEA-AG supera a VSR-OKAPI, las soluciones más distanciadas (aproximadamente en 0.3) corresponden a la consulta 19, con unos valores de precisión media de 0.946 y 0.625, respectivamente. En el lado contrario, la diferencia más significativa (0.129) corresponde a la consulta 26 con una precisión media de 0.751 en el caso del algoritmo multiobjetivo y de 0.880 para el método clásico.

La Figura 5.23 muestra un diagrama de barras comparativo de los valores de precisión media conseguidos por cada algoritmo para cada una de las 18 consultas. En ella se puede observar gráficamente las consultas en las que SPEA-AG supera a VSR-OKAPI y viceversa. Por otro lado, la igualdad de alturas entre las barras de ambos algoritmos nos indica que las diferencias entre las soluciones proporcionadas por ellos no son excesivamente altas.

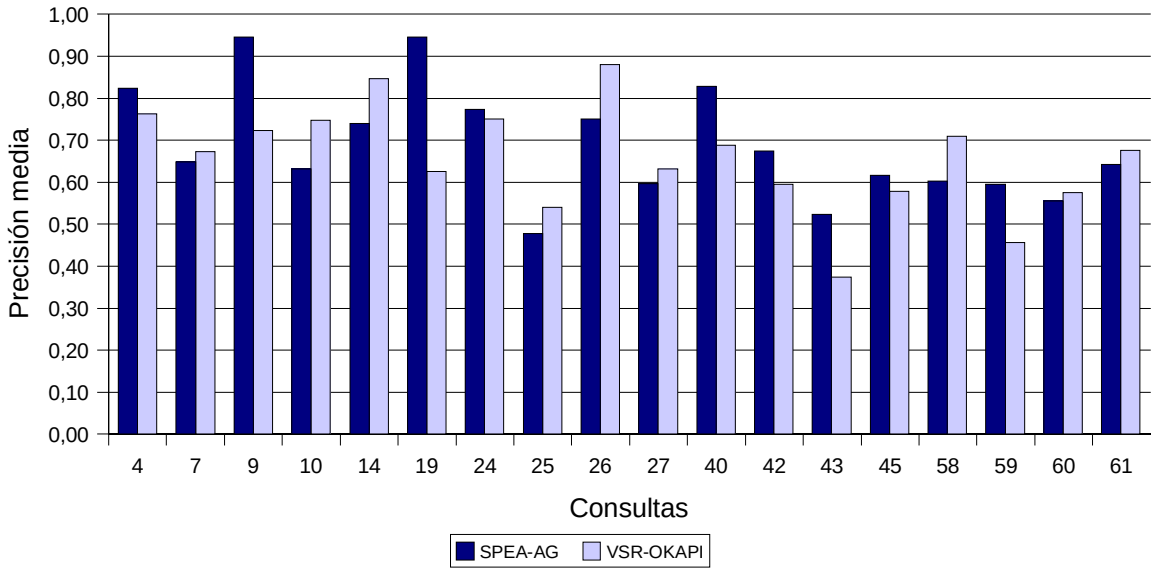


Figura 5.23: Comparativa de SPEA-AG (considerando resultados medios) y VSR-OKAPI sobre el conjunto de entrenamiento para las consultas de CACM

Si, por el contrario, consideramos para la comparación los mejores resultados de SPEA-AG sobre el conjunto de prueba (Tabla 5.117), VSR-OKAPI presenta mejor comportamiento en diez de las dieciocho consultas (Figura 5.24). El aumento del número de consulta en las que VSR-OKAPI nos supera es normal, ya que los resultados de SPEA-AG que estamos utilizando en la comparación son los mejores obtenidos sobre el conjunto de prueba, y como comentamos anteriormente, estos no tienen por qué corresponderse, necesariamente, con los mejores resultados en entrenamiento (resultados que estamos analizando actualmente). Caben destacar positivamente las consultas 9 y 19 que consiguen una precisión media igual al máximo.

En la Figura 5.24, podemos ver gráficamente las diferencias existentes entre los valores de precisión media obtenidos por SPEA-AG (considerando los mejores resultados en el conjunto prueba) y VSR-OKAPI. Las alturas de las barras nos indica que la mayoría de las consultas consiguen valores de precisión medias alrededor de 0.6; destacando las consultas 9, 19 y 42.

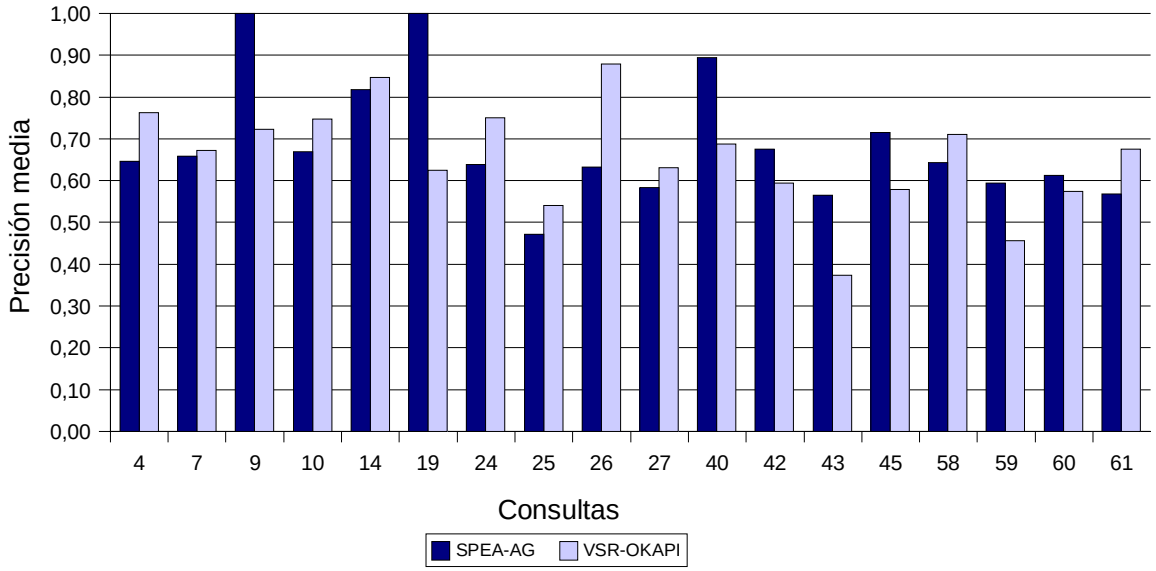


Figura 5.24: Comparativa de SPEA-AG (considerando los mejores resultados) y VSR-OKAPI sobre el conjunto de entrenamiento para las consultas de CACM

Evaluación sobre el conjunto de prueba

Por otro lado, en la evaluación sobre el conjunto de prueba, ocurre igual que en el caso de Cranfield. Cuando comparamos con los resultados medios (parte derecha de la Tabla 5.115), VSR-OKAPI consigue mejores resultados que SPEA-AG en todas las consultas excepto una (consulta 19). Los resultados medios de precisión media, valga la redundancia, de este algoritmo se mueven alrededor de 0.15, destacando las consultas 10 (0.45), 14 (0.38) y 26 (0.41), mientras que en el caso de VSR-OKAPI, los valores se sitúan en torno a 0.5 y, salvo uno, el resto no bajan de 0.1 (véase la Figura 5.25).

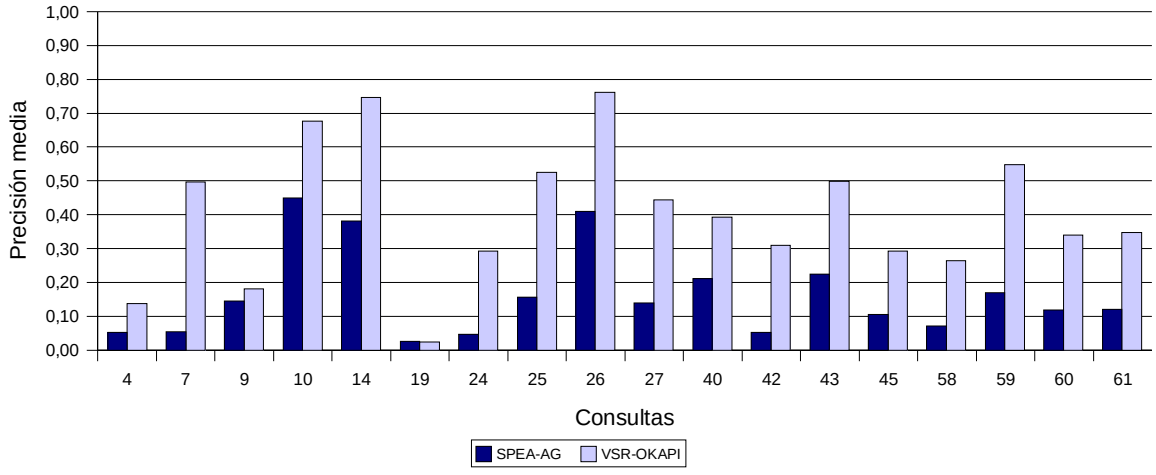


Figura 5.25: Comparativa de SPEA-AG (considerando resultados medios) y VSR-OKAPI sobre el conjunto de prueba para las consultas de CACM

Los mejores resultados de precisión media sobre el conjunto de prueba (parte derecha de la Tabla 5.117) nos permiten acercarnos algo más a los resultados de VSR-OKAPI. Ahora, únicamente dos consultas tienen valores por debajo de 0.1 (consultas 19 y 24), mientras que el resto se mueve en torno a 0.25. Las consultas que alcanzan mejores valores con SPEA-AG que con VSR-OKAPI se corresponden con las de menor número de documentos relevantes, en concreto, la número 4, 9 y 19, aunque las diferencias son muy pequeñas, al igual que los valores de precisión media.

En la Figura 5.26 podemos ver como las consultas 10, 14 y 26, a pesar de presentar los valores más altos de precisión media (0.534, 0.581 y 0.466, respectivamente), siguen diferenciándose de los resultados obtenidos por VSR-OKAPI. No obstante, estas diferencias no son muy elevadas. Por ejemplo, en el caso de la consulta 10, la diferencia es sólo de 0.142.

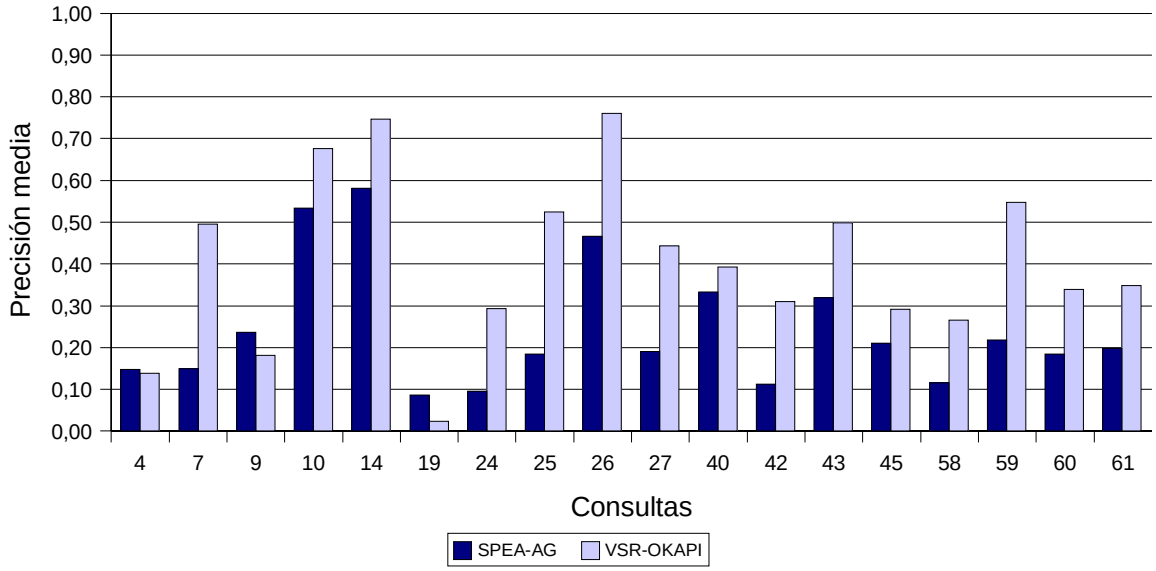
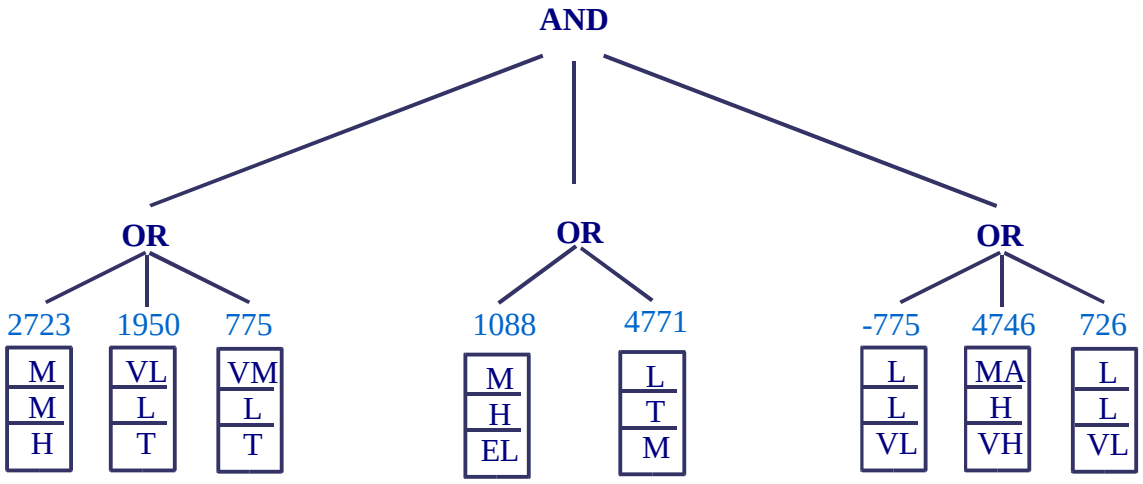


Figura 5.26: Comparativa de SPEA-AG (considerando los mejores resultados) y VSR-OKAPI sobre el conjunto de prueba para las consultas de CACM

En lo que respecta a la situación de los documentos que aportan información, vemos como no hay excesiva diferencia entre un algoritmo y otro, siendo en algunos casos exactamente los mismos números. Por tanto, las diferencias en los valores de precisión estribarán, sobre todo, en las posiciones que ocupen el resto de documentos relevantes, aunque también se verán influenciadas por las posiciones que ocupan los documentos dentro de esas *n* primeras.

Composición de los perfiles

Al igual que en el caso de Cranfield, las Figuras 5.27 y 5.28 muestran, a modo de ejemplo, los perfiles generados por SPEA-AG y VSR-OKAPI y, lanzados sobre el conjunto de prueba, para dos consultas, la 9 y la 10 respectivamente. El comportamiento de ambos algoritmo sobre estas consultas es diferentes. Para la consulta 9, nuestra propuesta consigue mejores resultados que VSR-OKAPI, mientras que para la consulta 10 es el método clásico el que obtiene un valor mayor de precisión media.

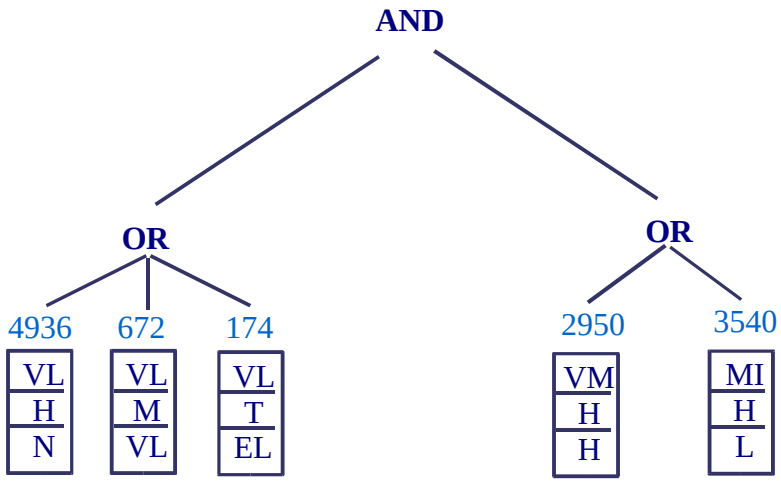


a) Perfil generado por el algoritmo SPEA-AG

2493 (17.27)	4778 (13.36)	1126 (11.55)	3272 (10.85)	770 (9.76)
2957 (9.53)	1919 (9.45)	2280 (8.99)	864 (8.35)	4790 (7.90)

b) Perfil generado por el algoritmo VSR-OKAPI

Figura 5.27: Perfiles generados por SPEA-AG y VSR-OKAPI para la consulta 9 de CACM sobre el conjunto de prueba



a) Perfil generado por el algoritmo SPEA-AG

3540 (59.13)	777 (36.83)	3100 (24.75)	1292 (21.50)	3076 (20.87)
5152 (20.35)	445 (19.83)	2570 (17.92)	2393 (15.14)	5087 (12.87)

b) Perfil generado por el algoritmo VSR-OKAPI

Figura 5.28: Perfiles generados por SPEA-AG y VSR-OKAPI para la consulta 10 de CACM sobre el conjunto de prueba

En ambas figuras podemos observar como los perfiles generados por nuestra propuesta aportan mayor expresividad, al almacenar tanto los términos que representan las necesidades de información del usuario como los operadores Booleanos que los relacionan. Esta propiedad unida a la posibilidad de obtener diferentes perfiles en una misma ejecución compensa los malos resultados sobre el conjunto de prueba, colocándonos en una posición medianamente competitiva.

5.5.5.- Resumen

- En vista de los resultados, podemos afirmar que nuestro algoritmo es competitivo con el método clásico sobre el conjunto de entrenamiento, puesto que ambos se comportan de manera muy similar. Incluso, en algunos casos, nuestra propuesta mejora a VSR-OKAPI.

- Como era de esperar, el método clásico de generación de perfiles consigue mejores resultados que nosotros sobre los conjuntos de prueba. Sin embargo, en algunos casos las diferencias no son demasiado grandes, lo que unido al hecho de que con nuestra propuesta conseguimos derivar varias consultas persistentes en una única ejecución, nos coloca en una situación favorable para refinar la metodología y obtener resultados competitivos.

- La forma de representar los perfiles utilizada por SPEA-AG, como consultas clásicas de RI, hace que los perfiles de usuario generados por este algoritmo sean más expresivos que los obtenidos por el algoritmo clásico, al aprender no sólo los términos, sino también la estructura del perfil.

- Las consultas con menor número de documentos relevantes alcanzan los valores más bajos de precisión media en ambos algoritmos.

- La consulta 3 de Cranfield y las consultas 10, 14 y 26 de CACM son las que presentan un mejor comportamiento.

- Las diferencias en los valores de precisión media entre los dos algoritmos no se reflejan en la proporción de documentos relevantes que se localizan en la cabeza del conjunto de documentos recuperados.

- La obtención de los mejores resultados en entrenamiento no implica la de los mejores resultados en prueba.

COMENTARIOS FINALES

A modo de resumen, en este apartado de la memoria recogemos las principales conclusiones derivadas de la labor investigadora descrita a lo largo de la misma. El trabajo realizado no sólo nos ha permitido obtener resultados muy interesantes, sino que, a raíz de la investigación realizada, han quedado abiertas diferentes líneas de trabajo para un desarrollo futuro.

Conclusiones Generales del Trabajo

A lo largo de esta memoria nuestro objetivo principal ha sido el de presentar técnicas que mejoren la representación de las necesidades de información, puntuales y permanentes, de los usuarios, en los SRI documentales. La caracterización de los perfiles de usuario como consultas clásicas de RI, su aprendizaje utilizando AEs y el uso de Información Lingüística Difusa han sido nuestro aporte en esta investigación. Atendiendo a estos aspectos, los resultados obtenidos en esta memoria pueden resumirse en los siguientes apartados:

Uso de Técnicas de Soft Computing en el Campo de la RI

La aplicación de técnicas de Soft Computing a la RI aporta mayor flexibilidad a los SRI. Así, la Lógica Difusa y los AEs son dos potentes herramientas que mejoran el rendimiento de estos sistemas. La primera permite modelar la subjetividad y la incertidumbre existentes en la actividad de la RI (p.e, en la estimación de la relevancia de un documento respecto a una consulta o en la formulación de una consulta que representa las necesidades de información del usuario); mientras que los segundos son muy útiles en lo que se refiere a resolución de problemas de RI mediante búsquedas en espacios complejos [34].

Uso de Información Lingüística Difusa en la Mejora de la Interacción Directa Usuario-Sistema

1. El uso de Información Lingüística en los SRI permite a los usuarios expresar de forma cualitativa sus necesidades de información [17][18][19][95][96][114]. Sin embargo,

estas propuestas presentan dos limitaciones:

- ☞ No permiten que los elementos de una consulta se ponderen simultáneamente de acuerdo a varias semánticas.
- ☞ Las entradas y la salida de los SRI se valoran sobre el mismo conjunto de etiquetas S, reduciendo las posibilidades de comunicación entre el usuario y el sistema, a la vez que disminuyendo la expresividad, al utilizar un mismo conjunto de etiquetas para expresar conceptos diferentes.

2. La utilización del MLDM permite resolver estas deficiencias, a la par que dota de más flexibilidad y facilidad de representación a las consultas instantáneas en el proceso de RI. En concreto, hemos propuesto un SRI Lingüístico Multigranular que utiliza diferentes conjuntos de etiquetas con distinta granularidad y/o semántica para representar los diferentes tipos de información que pueden aparecer en el proceso de RI, permitiendo la ponderación de un elemento por varias semánticas de manera simultánea [97][98][99][100].

Caracterización de los Perfiles de Usuario como Consultas Booleanas y su Aprendizaje utilizando un Algoritmo de PG Multiobjetivo

1. Caracterizar las necesidades de información persistentes (perfiles de usuario) como consultas clásicas de RI (consultas persistentes) proporciona:

   - ☞ Más flexibilidad, al poder usar cualquier modelo de RI para formularlas: Booleano, Booleano extendido, Lingüístico, ...
   - ☞ Más expresividad, al usar términos y operadores Booleanos (Y, O, NO) para representar las necesidades de información, lo que es más interpretable para el ser humano.

2. El uso de AEs multiobjetivo nos ha permitido resolver el problema del aprendizaje automático de consultas persistentes de una forma muy satisfactoria, generando varias consultas con distinto balance entre precisión y exhaustividad en una sola ejecución del algoritmo[35][36][37].

De hecho, haciendo un recorrido por los resultados obtenidos, podemos comprobar como en 29 de las 35 consultas consideradas se generan poblaciones de más de una solución, consiguiéndose mayor diversidad que con el algoritmo de Smith y Smith [165]. En las otras 6 consultas (4 de Cranfield y 2 de CACM), el número de soluciones se reduce a una única consulta que satisface plenamente ambos objetivos, localizándose en la esquina superior derecha del espacio de búsqueda.

Además, al evaluar los perfiles de usuarios generados sobre nuevos documentos, no sólo conseguimos buenos resultados, sino que recuperamos menos documentos basura que el algoritmo original, lo que facilita el acceso de los usuarios a la nueva información.

Uso de Información Lingüística en la Representación de Consultas Persistentes

1. Los buenos resultados en un proceso de búsqueda de información dependen de la habilidad del usuario para expresar sus necesidades de información mediante una consulta. Combinando la idea de representar los perfiles de usuario como consultas clásicas de RI y el uso del MLDM, hemos conseguido perfiles que, caracterizados como consultas lingüísticas multigranulares, son más interpretables, flexibles y expresivos, logrando una representación de las necesidades de información más acorde con el usuario.

2. La potencialidad del SRI Multigranular para devolver los documentos ordenados de acuerdo a su relevancia ha permitido que un número aceptable de documentos que se adaptan a la necesidades de información del usuario se sitúen entre los primeros del conjunto de documentos recuperados, facilitando el acceso a la nueva información.

3. La representación de los perfiles como consultas clásicas de RI y su aprendizaje mediante un AE multiobjetivo es competitiva sobre los conjuntos de entrenamiento, presentando un comportamiento muy similar o incluso mejor que el método clásico de aprendizaje de perfiles (VSR-OKAPI), como demuestran los resultados obtenidos en la experimentación realizada sobre Cranfield y CACM en el Capítulo 5.

4. En los conjuntos de prueba, nuestro desarrollo sale perdiendo en la mayor parte de los ejemplos. No obstante, la posibilidad de obtener diferentes consultas en una ejecución y poder aprender no sólo los términos, sino también la estructura del perfil, junto con el hecho de que las diferencias en los resultados no sean demasiado altas en algunos casos, nos coloca en una situación favorable para refinar el método a corto plazo y obtener así resultados competitivos en este aspecto.

5. El número de documentos relevantes que tiene asociados una consulta es un factor determinante de los resultados de recuperación obtenidos con ella. Cuanto menos documentos relevantes tenga, mejores resultados se obtendrán sobre el conjunto de entrenamiento y peores sobre el de prueba.

6. En el término medio se encuentra la virtud. Encontramos un fenómeno de sobreaprendizaje. En general, las consultas con un comportamiento extremadamente bueno sobre el conjunto de entrenamiento, presentan el comportamiento inverso sobre el conjunto de prueba. Por el contrario, las consultas con resultados medianamente buenos en el conjunto de entrenamiento, presentan resultados igualmente buenos sobre los conjuntos de prueba.

Trabajos Futuros

A raíz de la investigación realizada, han quedado abiertas diferentes líneas de trabajo para un desarrollo futuro que pueden clasificarse en dos vertientes: una primera que agrupa las futuras líneas de investigación asociadas con el uso de Información Lingüística y otra que incluye las relacionadas con el uso de AEs.

A) Vertiente Lingüística

1. Diseño de un SRI Lingüístico multigranular multinivel

Los SRI Lingüísticos existentes no permiten que los usuarios puedan usar las distintas partes de una consulta para expresar sus necesidades de información y ello limita su

rendimiento y la expresividad de los usuarios. Nos planteamos, por tanto, desarrollar un nuevo SRI Lingüístico que introduzca más medios para que los usuarios expresen sus necesidades de información. Estará formado por un subsistema de consultas ponderadas multinivel basado en información lingüística difusa y su correspondiente subsistema de evaluación, e incorporará las siguientes características:

a) *Un esquema de ponderación multinivel*, de modo que, en cada consulta, el usuario pueda asignar pesos lingüísticos difusos a cada uno de sus componentes (términos, subexpresiones, operadores Booleanos y consulta completa). Así conseguimos que los usuarios puedan participar más activamente en el proceso de RI.

b) *Un esquema semántico multinivel consistente*, de manera que la semántica asociada a los pesos de cada nivel sea compatible con las del resto de niveles.

c) *La posibilidad de usar información lingüística multigranular y balanceada para valorar los pesos*, de modo que se puedan usar distintos conjuntos de etiquetas para ponderar los diferentes niveles de una consulta: i) con distinta semántica y cardinalidad, como en esta memoria, y/o ii) no simétricamente distribuidos con respecto de la etiqueta central y con diferentes cardinalidades a ambos lados de la misma (balanceada).

2. Mejora del rendimiento de los SRI basados en un enfoque lingüístico ordinal

El SRI Lingüístico propuesto en esta memoria, así como los presentados en [19][95][96], se basan en un enfoque lingüístico ordinal [88] y, por tanto, se ven afectados por dos problemas característicos del modelado lingüístico ordinal [87]:

☞*Pérdida de precisión*. El enfoque lingüístico ordinal trabaja con dominios lingüísticos discretos, lo que implica cierta limitación a la hora de representar la información, por ejemplo, al representar los grados de relevancia.

☞*Pérdida de información*. Los operadores de agregación para información lingüística ordinal utilizan operaciones de aproximación en su definición (p.e. la operación de redondeo), lo que causa pérdida de información.

Nos planteamos utilizar un enfoque lingüístico distinto, que solucione los problemas

anteriores y, además, mejore el rendimiento de los SRI como, por ejemplo, un enfoque lingüístico basado en el modelo de 2-tuplas [87].

3. Desarrollo de una interfaz gráfica para el uso del SRI Lingüístico multigranular

Somos conscientes de que el SRI Lingüístico propuesto es complejo para ser empleado directamente por un usuario no experto. De hecho, para que un usuario pueda expresar de manera correcta sus necesidades de información, es necesario que tenga un conocimiento perfecto de las semánticas utilizadas para la ponderación, así como de los conjuntos de etiquetas en los que son valoradas.

Nos proponemos desarrollar una interfaz amigable que facilite la utilización del SRI Lingüístico por parte de los usuarios, evitando que tengan que conocer los entresijos del mismo.

B) VERTIENTE EVOLUTIVA

1. Mejora en la selección y traducción de las consultas de los Paretos obtenidos

Como comentamos en las conclusiones, nuestra propuesta del Capítulo 5 sale perdiendo en la mayor parte de los ejemplos sobre los conjuntos de prueba, al compararla con el algoritmo de VSR-OKAPI. No obstante, la posibilidad de obtener diferentes consultas en una ejecución y poder aprender no sólo los términos, sino también la estructura del perfil, nos coloca en una situación favorable para refinar la metodología y obtener así mejores resultados. En vista de que las diferencias no son excesivamente elevadas y de que incluso en algunos casos conseguimos superar al algoritmo clásico, pensamos que incluyendo algunas mejoras podemos conseguir ser más competitivos.

En concreto, nos propondremos incluir dos mejoras:

a) *Modificar el procedimiento de validación de los algoritmos*. En vez de utilizar la clásica división de los conjuntos de datos en entrenamiento y prueba, consideraremos la división de los datos en entrenamiento, validación y prueba, al igual que hacen Fan y otros en sus trabajos más recientes [61][60]. De esta manera, una vez aprendidas las consultas persistentes (utilizando el conjunto de entrenamiento), las lanzaremos sobre

el conjunto de validación, empleado para seleccionar las mejores, que serán las que, a su vez, se evaluarán sobre el conjunto de prueba. Con la incorporación de este conjunto de datos intermedios, intentaremos eliminar del conjunto final las soluciones con un alto grado de sobreaprendizaje.

b) *Mejorar el algoritmo de traducción de las consultas.* En los resultados mostrados en el Capítulo 5, varias consultas no llegaban ni siquiera a ser evaluadas al no poder se traducidas (como quedaba demostrado por la aparición de consultas con tamaño 0). Esto es consecuencia del algoritmo utilizado para traducir las consultas persistentes de entrenamiento a prueba (véase el Apéndice A.1), el cual elimina una expresión en el momento en que se queda con un único término, desapareciendo dicho término de la consulta global. Nos plantearemos modificar este algoritmo de traducción para que, en vez de eliminar la subexpresión y, por tanto el término, reestructure la consulta.

2. Mejora de las técnicas multiobjetivo propuestas

El reducido número de soluciones en los Paretos, en proporción al tamaño de la población elitista, se debe, como se comentó en los Capítulos 3 y 5, a que consideramos que dos soluciones son iguales si lo son en el espacio de objetivos, independientemente de la estructura que tengan. Nos interesaría, por tanto, encontrar medidas de similitud entre expresiones de consultas que nos permitan quedarnos con el mayor número de soluciones iguales en ambos objetivos, pero diferentes en la estructura (es decir, consultas con el mismo valor de precisión y exhaustividad pero con distintas expresiones), con el propósito de trabajar en el espacio de decisiones.

3. Aplicación de las técnicas evolutivas multiobjetivo a otros problemas de RI

En nuestra opinión, el empleo de AEs multiobjetivo en RI presenta una gran potencialidad puesto que el problema fundamental de la RI es bi-objetivo en sí al tener que optimizar dos criterios contradictorios, la precisión y la exhaustividad. Por tanto, creemos que el empleo de este tipo de AEs en otros problemas del campo de la RI podría proporcionar resultados muy prometedores.

APÉNDICES

A.1.-DISEÑO EXPERIMENTAL

Este apéndice está dedicado a describir el entorno experimental de aprendizaje automático de PQ que hemos diseñado para evaluar el rendimiento de las propuestas que hemos introducido en los Capítulos 3 y 5. Para ello, comenzaremos repasando brevemente la metodología de evaluación propuesta por Cleverdon y otros [28], que constituye la base del modelo tradicional de evaluación de SRIs, y que será la considerada en los experimentos realizados en esta memoria. A continuación, pasaremos a estudiar la composición de las bases documentales de test que emplearemos en nuestra experimentación, Cranfield y CACM, y describiremos el proceso realizado para obtener los vectores documentales. Posteriormente, analizaremos el procesamiento efectuado sobre algunas de las consultas asociadas a las dos bases anteriores para diseñar nuestro entorno de aprendizaje automático de PQ, así como otros factores de la experimentación.

A.1.1.-Modelo Algorítmico de Evaluación

El *modelo tradicional o algorítmico de evaluación* de SRIs está basado en el uso de medidas de relevancia para evaluar la eficacia del sistema en el proceso de RI. Dicho modelo se denomina también *modelo Cranfield*, en honor del lugar de procedencia de uno de los investigadores que más activamente trabajó en su desarrollo, C.W. Cleverdon.

A mediados de la década de los 60, tras una serie de pruebas preliminares que constituyeron el proyecto denominado *Cranfield I*, un grupo de investigadores del College of Aeronautics de Cranfield en el Reino Unido dirigidos por Cleverdon realizó una nueva batería de pruebas para evaluar la calidad de treinta y tres sistemas de indización distintos [26] en el proceso de RI, considerando directamente la relevancia como medida de rendimiento (eficacia) del SRI.

Para ello, diseñaron una base compuesta por 1400 documentos sobre Aeronáutica y compusieron una batería de 211 consultas generadas por autores de documentos seleccionados de la base. Antes de realizar ninguna búsqueda, determinaron los documentos relevantes para cada consulta empleando un proceso multietápico en el que intervinieron en

primer lugar estudiantes de Aeronáutica de Cranfield y, posteriormente, los propios autores de las consultas.

De este modo, el experimento presentaba tres componentes principales:

1. La colección de documentos de prueba.
2. El conjunto de consultas.
3. El conjunto de juicios de relevancia (valoraciones binarias que indicaban si el documento era juzgado como relevante para cada consulta).

Cada vez que se ejecutaba una consulta en el sistema, se determinaban los siguientes conjuntos de documentos:

a: conjunto de documentos relevantes recuperados

b: conjunto de documentos no relevantes recuperados

c: conjunto de documentos relevantes no recuperados

d: conjunto de documentos no relevantes no recuperados

De este modo, los juicios de relevancia asociados a cada consulta se emplearon para medir el rendimiento de los distintos mecanismos de indización considerados, haciendo uso por primera vez de los conceptos (hoy ya clásicos) de precisión y exhaustividad (véase la Sección 1.4). Ambas medidas se obtenían a partir de los conjuntos de documentos mencionados de la siguiente forma:

$$P = \frac{a}{a+b} \quad ; \quad E = \frac{a}{a+c}$$

Este experimento, que pasó a ser conocido como *Cranfield II*, sentó las bases de la evaluación algorítmica de SRI desde aquel momento. Poco después, Salton desarrolló el SRI SMART [152] y lo evaluó haciendo uso de distintas bases documentales, con juicios de relevancia conocidos, siguiendo una filosofía muy similar a la de Cleverdon. De hecho, una de las primeras colecciones considerada en SMART fue la empleada en Cranfield II, que pasó a denominarse colección Cranfield desde ese momento. La principal novedad del modelo de

evaluación aplicado en SMART era que, puesto que SMART estaba basado en el modelo de espacio vectorial y aplicaba, por tanto, emparejamiento parcial en lugar de exacto, incorporaba un mecanismo de evaluación capaz de valorar no sólo la eficacia absoluta de la recuperación sino también el orden de aparición de los documentos recuperados. Para ello, Salton y su equipo hicieron uso de las curvas de exhaustividad-precisión que se analizaron en la Sección 1.4.

El modelo tradicional de evaluación ha sido muy criticado, principalmente por las razones siguientes [175]:

1. La dificultad de obtención de los juicios de relevancia asociados a los documentos de la colección (entre otros aspectos, debido a la subjetividad y la necesidad de un recorrido exhaustivo de la base documental).

2. El escalado de los resultados obtenidos en bases documentales de pequeño tamaño a las bases documentales reales, de gran tamaño.

3. El hecho de que las representaciones de los documentos se obtengan a partir de resúmenes de los mismos y provengan del texto completo.

Sin embargo, el método tradicional facilita el diseño del experimento de evaluación y permite a los investigadores comparar los resultados de los estudios realizados de una forma sencilla [127]. Además, como argumenta Olvera [139], a pesar de las dificultades a nivel teórico y práctico que genera el uso de medidas de exhaustividad y precisión, éstas ofrecen, al menos, medidas intersubjetivamente aceptables y homogéneas, y gozan ya de suficientes referentes como para permitir comparar el funcionamiento de los SRI.

Por último, el problema de la escalabilidad de los resultados a grandes bases documentales y la indización de documentos a texto completo va quedando progresivamente atenuado al aumentar el tamaño de las colecciones de test consideradas. Un ejemplo claro lo constituyen las colecciones TREC (*Text REtrieval Conference*), provenientes de una iniciativa liderada por Donna Harman a principios de los noventa en el *NIST* (*National Institute of Standards and Technology*), en Maryland. La iniciativa consistió en el desarrollo, desde 1992, de un congreso anual, denominado TREC[7], dedicado a la experimentación con grandes colecciones de test. Desde ese momento, en cada edición del congreso se han diseñado

7 http://trec.nist.gov/

nuevos experimentos de referencia para evaluar distintos aspectos (filtrado de información, RI interactiva, procesamiento del lenguaje natural, RI en la web, etc.) en el marco de la calidad de la recuperación en SRIs [5].

Sin embargo, como López-Pujalte menciona en [127], las colecciones TREC tienen el inconveniente de no incluir algunas características presentes en las colecciones pequeñas empleadas en SMART. Por un lado, el empleo de estas colecciones requieren una gran cantidad de recursos, lo que dificulta su manejo. Por otro, las consultas de TREC presentan una superposición muy pequeña entre ellas y no son muy útiles para investigar el impacto de técnicas que se benefician de información obtenida de dependencias entre la consulta actual y las consultas pasadas del usuario, cuestión que recibió una atención especial en la conferencia TREC-7 [5].

De este modo, existe una gran cantidad de investigadores en el área que trabajan con colecciones de tamaño medio. Así, el empleo de las bases documentales consideradas por Edward Fox en la realización de su tesis doctoral en 1983 [71] está muy extendido. Este investigador trabajó con las cinco colecciones siguientes [70]:

- ADI: documentos sobre Ciencias de la Información.
- CACM: artículos publicados en *Communications of the ACM*.
- CISI: documentos más citados del ISI.
- INSPEC: resúmenes de electrónica, informática y física.
- MEDLINE: artículos médicos de la base de datos MEDLINE de la Biblioteca Nacional de Medicina.

La Tabla A.1.1 muestra algunos datos sobre las cinco colecciones comentadas.

En este trabajo haremos uso de dos colecciones, la clásica de Cranfield y la CACM, disponibles en la distribución del SRI de Salton, SMART. Describiremos la composición de ambas en la sección siguiente.

Colección	Número de documentos	Número de consultas	Ámbito
ADI	82	35	Ciencias de la Información
CACM	3.204	64	Informática
CISI	1.460	112	Ciencias de la Información
INSPEC	12.684	77	Informática e Ingeniería Eléctrica
MEDLINE	1.033	30	Medicina

Tabla A.1.1: Características de las colecciones de test empleadas por Fox

El uso de las colecciones TREC en el campo de la RI ha incrementado considerablemente en los últimos tiempos. Sin embargo, nuestras propuestas se enmarcan dentro de en un entorno de IQBE donde el usuario proporciona un conjunto de documentos relevantes e irrelevantes como punto de partida para la representación de sus necesidades de información a largo plazo como una PQ. Por lo tanto, consideramos que no sería realista utilizar colecciones compuestas por un elevado número de documentos puesto que un usuario, en condiciones normales, no estaría capacitado para suministrar muchos documentos como entrada a un proceso de aprendizaje automático.

Por esta razón, hemos preferido hacer uso de algunas de las colecciones clásicas de RI que presentan un menor tamaño, pero que nos permiten una representación mucho más apropiada de un marco de aplicación que semeja lo que ocurre en la realidad.

En lo que respecta a las medidas de efectividad consideradas, trabajaremos con la exhaustividad y precisión clásica como medidas para la evaluación de nuestros sistemas a lo largo del proceso evolutivo, y utilizaremos la precisión media a 11 niveles de exhaustividad (ver sección 1.4) como medida final de comparación, en aquellos sistemas que permitan ordenar los documentos finales de acuerdo a su relevancia.

Esta elección se ha hecho con el fin de mantener un entorno experimental uniforme y poder realizar comparaciones realistas, ya que el algoritmo presentado en el capítulo 3 está basado en el modelo Booleano clásico y todos los algoritmos propuestos se basan en un enfoque multiobjetivo lo que permitirá derivar varias consultas que optimicen simultáneamente ambos criterios.

Así, para el caso de los algoritmos multiobjetivo, bien propuestos en esta memoria o que se utilicen como comparación, se ha escogido como función objetivo la optimización de

ambos criterios a la vez, mientras que en los algoritmos de comparación mono-objetivos se ha adoptado como función objetivo una combinación ponderada de exhaustividad y precisión de la siguiente forma:

$$F(C) = \alpha \cdot E + \beta \cdot P$$

A.1.2.-Colecciones de Test y consultas consideradas

Para el estudio experimental realizado en esta tesis doctoral consideraremos las colecciones Cranfield y CACM, que pasamos a describir brevemente en las dos subsecciones siguientes. A continuación, explicaremos el proceso documental que hemos aplicado para vectorizar ambas colecciones.

A.1.2.1.-La Colección de Cranfield

Como hemos comentado en la sección anterior, Cranfield fue una de las primeras bases documentales empleadas como colecciones de test en el área, por lo que goza de gran prestigio entre los investigadores y ha sido empleada en una gran cantidad de estudios en el campo de la RI.

En su estado actual, está compuesta por 1.398 documentos sobre Ingeniería Aeronáutica y 225 consultas, para todas las cuales se conocen los juicios de relevancia. La media de documentos relevantes para cada consulta es de aproximadamente 7, un número bastante bajo.

Para realizar nuestra experimentación, hemos seleccionado dos bloque de consultas, por un lado, aquellas que presentan 20 o más documentos relevantes; y por otro, 10 consultas con 15 o menos documentos relevantes. Con esta selección de consultas, hemos querido presentar un “setup” que refleje la mayoría de las situaciones que pueden ocurrir cuando el usuario proporciona los documentos que sirven como punto de partida al proceso de creación de consultas persistentes que representan sus intereses a largo plazo. Recogemos situaciones que van desde que el usuario proporcione un número sustancial de documentos relevantes (p.e., 40 para la consulta 157) a qué sólo sea capaz de identificar unos pocos, como en el caso de la consulta 7 que tiene asociados 6 documentos relevantes.

De este modo, las 17 consultas seleccionadas se recogen en las Tablas A.1.2 y A.1.3.

Número de consulta	Número de documentos relevantes
1	29
2	25
23	33
73	21
157	40
220	20
225	25

Tabla A.1.2: Consultas seleccionadas en Cranfield con 20 o más documentos relevantes

Número de consulta	Número de documentos relevantes
3	9
7	6
8	12
11	8
19	10
26	7
38	11
39	14
40	13
47	15

Tabla A.1.3: Consultas seleccionadas en Cranfield con menos de 15 documentos relevantes

A.1.2.2.-La Colección CACM

La base documental CACM está formada por 3.204 artículos publicados en la revista *Communications of the ACM*, una de las revistas clásicas de Informática, entre 1958 y 1979, comprendiendo desde el primer número de la revista hasta el último publicado en el año 1979.

Tal y como se comenta en [5], además del texto de los documentos en sí, la colección también incluye información de campos estructurados, a los que Fox denomina conceptos. Dichos campos son: nombre de los autores, información sobre la fecha, raíces de las palabras del título y el resumen, categorías derivadas de un esquema de clasificación jerárquico, referencias directas entre artículos de la colección, conexiones de emparejamiento bibliográfico (referencias comunes entre dos documentos) y número de co-citaciones para cada par de artículos de la colección.

La base documental CACM incluye un total de 64 consultas, de las que en 52 se conocen los juicios de relevancia. El número de documentos relevantes para cada una de ellas ronda los 15 en media, el cual, aunque es superior al de Cranfield, sigue sin ser un número demasiado alto[8], por lo que los valores de precisión y exhaustividad tienden a ser bajos.

Al igual que en la colección Cranfield, para nuestra experimentación hemos seleccionado dos bloques diferentes de consultas: aquellas que presentan más de 20 documentos relevantes y 5 con menos de 15 documentos relevantes. En total, se han escogido las 18 consultas que se muestran en las Tablas A.1.4 y A.1.5.

Número de consulta	**Número de documentos relevantes**
7	28
10	35
14	44
25	51
26	30
27	29
42	21
43	41
45	26
58	30
59	43
60	27
61	31

Tabla A.1.4: Consultas seleccionadas en en CACM con mas de 20 documentos relevantes

Número de consulta	**Número de documentos relevantes**
4	12
9	9
19	11
24	13
40	10

Tabla A.1.5: Consultas seleccionadas en CACM con menos de 15 documentos relevantes

8 Por ejemplo, la colección CISI, que está compuesta por un número mucho menor de documentos (1.460), presenta una media de casi 50 documentos relevantes por consulta.

A.1.2.3.-Procesamiento de las Colecciones para Obtener los Vectores Documentales

Una vez analizadas las colecciones de test consideradas, vamos a describir brevemente el proceso seguido para su vectorización. Para obtener los vectores documentales, hemos hecho uso del SRI SMART de Salton [152].

Así, hemos seguido el proceso documental habitual recogido en la Figura 1.4 del Capítulo 1. Una vez procesados los documentos y obtenidos los términos contenidos en ellos, se han eliminado las palabras vacías haciendo uso de la "stoplist" que incorpora SMART y se ha aplicado igualmente el algoritmo de reducción a la raíz ("stemming") de este SRI.

Llegados a este momento, ya tenemos determinados los términos índice que considerarían nuestros vectores documentales (3.857 en Cranfield y 7562 en CACM) y podemos construir los ficheros inversos que recogen la frecuencia de cada uno de esos términos en cada documento. A la hora de proceder a la ponderación de los vectores documentales, SMART proporciona distintas posibilidades para la función de ponderación, realizando este proceso en tres pasos [154]:

- *Cálculo del peso básico*: Existen tres opciones: indización binaria, uso de la frecuencia del término (*tf*) y uso de la frecuencia del término aumentada[9]. Las tres opciones se notan respectivamente por las letras *b*, *n* y *a*.

- *Modificación del peso calculado en la etapa anterior*: Generalmente, esta modificación se efectúa haciendo uso de información procedente de la colección completa, para aumentar el peso de los términos menos comunes y disminuir el de los más usuales. De nuevo, se pueden considerar tres posibilidades: no efectuar modificación (letra *n*), realizar una indización probabilística (letra *p*) y multiplicar por la frecuencia inversa del término en los documentos de la colección (*idf*) (letra *t*).

- *Normalización de los vectores obtenidos*: En este caso, sólo existen dos opciones: no normalizar los pesos (letra *n*) y normalizarlos dividiendo sus valores entre la medida del coseno, es decir, entre la raíz cuadrada de la suma de los pesos al cuadrado (letra *c*).

9 El término aumentado es una traducción directa de "*augmented*" en la documentación de SMART. En realidad, no tenemos claro qué tipo de procesamiento efectúa SMART sobre la frecuencia del término cuando se aplica esta opción.

De este modo, se pueden considerar distintas ponderaciones tales como las siguientes:

- *Esquema nnn*:

    1. *n*: El peso básico es la frecuencia del término (*tf*).
    2. *n*: No se combina la *tf* con ninguna otra información de la colección.
    3. *n*: No se normaliza el vector.

    El peso final será directamente la frecuencia del término en el documento (*tf*).

- *Esquema ntn*:

    1. *n*: El peso básico es la frecuencia del término (*tf*).
    2. *t*: Se multiplica por el *idf*.
    3. *n*: No se normaliza el vector.

    El peso final sería el *tf·idf* de cada término, esquema de indización muy habitual, propuesto por Salton (véase la Sección 1.3.2.1).

- *Esquema ntc*:

    1. *n*: El peso básico es la frecuencia del término (*tf*).
    2. *t*: Se multiplica por el *idf*.
    3. c: Se normaliza el vector.

    El peso final sería el *tf·idf* de cada término normalizado por la medida del coseno.

Del mismo modo, el esquema *bnn* correspondería a una indización binaria clásica y el *npn* a una indización probabilística. En este trabajo hemos optado por el esquema *ntc*, es decir, por la función IDF de Salton normalizada.

Finalmente, los ficheros inversos obtenidos a partir de SMART, presentan la siguiente estructura:

Número documento	??	Número término	Peso	Término
1	0	t_{11}	w_{11}	nombre t_{11}
...	...	...	...	...
M	0	t_{mn}	w_{m1n}	nombre t_{mn}

en la que las columnas tienen la siguiente interpretación:

- *Número documento*: Número del documento de la colección en cuestión (de 1 a 1.398 en Cranfield y de 1 a 3.204 en CACM).
- *??*: Columna con todos los valores a 0, de la que no sabemos el significado. Seguramente es una columna libre que SMART mantiene para emplearla con algún propósito.
- *Número término*: número del término asociado al documento en cuestión. A pesar de haber realizado la eliminación de palabras vacías y el stemming, SMART no reenumera los términos obtenidos después de estos procesos, por lo que se mantiene la numeración original. De este modo, aparecen “huecos” en la numeración de los términos en el fichero inverso.
- *Peso*: peso del término en el documento en cuestión, dependiente de la función de ponderación considerada, como hemos visto anteriormente.
- *Término*: Forma del término reducida a la raíz.

A continuación mostramos, a modo de ejemplo, las primeras líneas del fichero inverso considerado para indicar la colección Cranfield (*cran_docs.ntc*):

```
1       0       350     0.10638         due
1       0       408     0.07052         basi
1       0       487     0.04358         angl
1       0       620     0.04453         effect
1       0       730     0.08600         propel
1       0       1443    0.03623         ratio
1       0       1513    0.45814         destal
1       0       2693    0.63449         slipstream
1       0       2842    0.07305         control
1       0       3184    0.02497         bound
1       0       3213    0.04435         increas
1       0       3801    0.04549         load
1       0       4068    0.26271         incr
```

Una vez obtenidos estos ficheros inversos con la ponderación deseada, realizamos un sencillo procesamiento para reenumerar los términos índice para eliminar los “huecos” que comentábamos y para, finalmente, construir los vectores documentales. Este último proceso se lleva a cabo mediante dos sencillos programas que hemos realizado en lenguaje C:

- *Reajuste*: Programa que toma como entrada el fichero inverso generado por SMART y devuelve como salida un nuevo fichero inverso con una nueva numeración continua para los términos índice.

- *Vector*: Programa que toma como entrada el fichero inverso reenumerado y devuelve como salida un fichero binario que contiene los vectores documentales. Este último fichero será la base documental definitiva que emplearán los algoritmos desarrollados en los Capítulos 3 y 5.

A.1.2.4.-División de las Bases de Datos en Entrenamiento y Prueba.

El objetivo de un sistema de aprendizaje automático es obtener las consultas que mejor definan las necesidades de información de acuerdo con un conjunto de documentos, pero que luego se comporten lo mejor posible cuando sean ejecutadas sobre conjuntos de documentos diferentes a los utilizados para el aprendizaje.

Por esta razón, con objetivo de definir un entorno experimental realista, hemos dividido la base de datos documental en dos subconjuntos, uno de entrenamiento, sobre el que se realizará el aprendizaje de consultas, y otro de prueba, para validar el comportamiento de las consultas generadas con el primer conjunto. La Figura A.1.1 muestra el uso de los conjuntos de entrenamiento y prueba en el proceso de aprendizaje de consultas.

La división de las bases de datos se ha hecho de acuerdo a la siguiente proporción: 50% de documentos relevantes e irrelevantes en el conjunto de entrenamiento y, por lo tanto, el 50% restante de documentos relevantes e irrelevante en el conjunto de prueba.

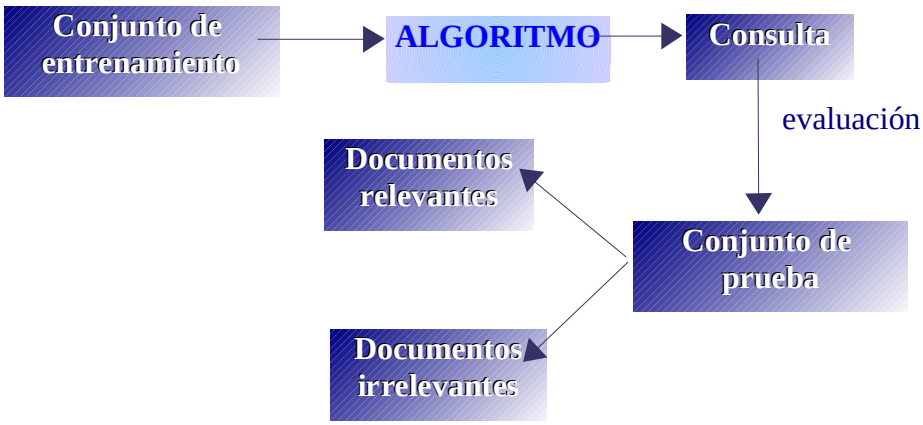


Figura A.1.1: Proceso de aprendizaje

La separación de los datos para entrenamiento y prueba en la proporción establecida (50% de relevantes, 50% irrelevantes) no se hace de forma exacta, no se calcula el 50% del número total de documentos relevantes y el 50% de los irrelevantes y se añaden exactamente ese número de documentos. El procedimiento seguido para la obtención de las particiones es el siguiente:

```
 Umbral_rel = 0.5
Umbral_irr = 0.5

Para cada di   conjunto de documentos de la base completa
     u  = aleatorio (0,1)
     si di es relevante
       si (u < Umbral_rel)
         añadir di al conjunto de entrenamiento
       else
         añadir di al conjunto de prueba
       fin_si
     si_ no
       si (u < Umbral_irr)
         añadir di al conjunto de entrenamiento
       else
         añadir di al conjunto de prueba
       fin_si
     fin_si
fin_para
```

Al seguir este procedimiento, en función de la semilla que se utilice, se obtendrán conjuntos de diferentes tamaños.

En las dos secciones siguientes, se muestran el número de documentos y el número de términos, que se obtienen para cada una de las particiones realizadas y consultas consideradas.

A.1.2.4.1.-Particiones obtenidas sobre la base documental Cranfield

La Tabla A.1.6 muestra el número de documentos relevantes e irrelevantes y el número de términos, respectivamente, asociados a cada consulta, tras realizar la división correspondiente, a los conjuntos de entrenamiento y prueba.

ENTRENAMIENTO				**PRUEBA**			
Q	**Nº Documentos**		**Nº Términos**	**Q**	**Nº Documentos**		**Nº Términos**
	Rel.	**Irr.**			**Rel.**	**Irr.**	
1	14	692	3004	**1**	15	677	2873
2	12	661	2867	**2**	13	712	3023
3	4	699	3003	**3**	5	690	2873
7	3	701	3004	**7**	3	691	2873
8	6	699	3004	**8**	6	687	2873
11	4	702	3007	**11**	4	688	2869
19	5	701	3007	**19**	5	687	2870
23	16	694	2920	**23**	17	671	2946
26	3	702	3004	**26**	4	689	2873
38	5	700	3004	**38**	6	687	2873
39	7	697	3002	**39**	7	687	2877
40	6	700	3010	**40**	7	685	2863
47	7	698	3004	**47**	8	685	2873
73	10	654	2887	**73**	11	723	2993
157	20	667	2871	**157**	20	681	3009
220	10	697	2934	**220**	10	681	2971
225	12	693	2887	**225**	13	680	3014

Tabla A.1.6: Particiones de Cranfield (50%, 50%)

A.1.2.4.2.-Particiones obtenidas sobre la base documental CACM

La Tabla A.1.7 muestra el número de documentos relevantes e irrelevantes y el número de términos, respectivamente, asociados a cada consulta, tras realizar la división correspondiente, a los conjuntos de entrenamiento y prueba.

ENTRENAMIENTO				PRUEBA			
Q	**Nº Documentos**		**Nº Términos**	**Q**	**Nº Documentos**		**Nº Términos**
	Rel.	**Irr.**			**Rel.**	**Irr.**	
4	6	1596	5201	**4**	6	1596	5175
7	14	1588	5202	**7**	14	1588	5175
9	4	1597	5198	**9**	5	1598	5178
10	17	1584	5198	**10**	18	1585	5178
14	22	1580	5278	**14**	22	1580	5124
19	5	1596	5198	**19**	6	1597	5178
24	6	1595	5198	**24**	7	1596	5184
25	25	1576	5179	**25**	26	1577	5206
26	15	1587	5198	**26**	15	1587	5173
27	14	1587	5197	**27**	15	1588	5175
40	5	1597	5201	**40**	5	1597	5175
42	10	1591	5199	**42**	11	1592	5179
43	20	1581	5251	**43**	21	1582	5117
45	13	1589	5203	**45**	13	1589	5169
58	15	1587	5199	**58**	15	1587	5173
59	21	1580	5062	**59**	22	1581	5295
60	13	1588	5198	**60**	14	1589	5178
61	15	1586	5107	**61**	16	1587	5264

Tabla A.1.7: Particiones de CACM (50%, 50%)

A.1.2.4.3.-Adaptación de las consultas de entrenamiento a prueba.

Al dividir las bases de datos documentales en los correspondientes conjuntos de entrenamiento y prueba, los términos que aparecen en cada uno de los conjuntos cambia su numeración respecto a la que tenía en la base original. Además, existirán términos que no aparezcan en el conjunto de prueba, aún cuando si están presentes en el de entrenamiento.

Lo primero es realizar una correspondencia entre los términos de cada conjunto, es decir, asociar a la numeración de entrenamiento la correspondiente en prueba; para ello, se genera un fichero con tres columnas: término, numeración asociada a dicho término en el conjunto de entrenamiento y numeración correspondiente en el conjunto de prueba. El fichero tendrá tantas líneas como términos comunes haya en ambas colecciones. A continuación se muestra un extracto del fichero de correspondencia de la consulta 73 de Cranfield:

```
skin                    1       28
account                 3       35
plat                    4       52
encompass               6       61
film                    7       64
tsien                   8       75
bow                     9       97
total                   10      103
predict                 11      105
row                     12      106
```

Podemos ver como los términos con numeración 2 y 5 en la base de entrenamiento no tienen correspondencia en la base de prueba.

Por otra parte, las consultas derivadas en el proceso de aprendizaje pueden estar formadas por términos presentes o no en el fichero de correspondencia. Si todos los término que forman la consulta derivada en el proceso de aprendizaje tienen su correspondencia en el conjunto de prueba, la traducción consistirá, únicamente, en cambiar la numeración.

Las Figuras A.1.2 y A.1.3 muestran una consulta derivada en el proceso de aprendizaje y su traducción, respectivamente.

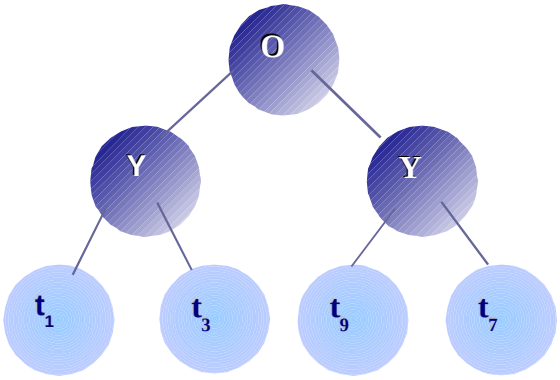


Figura A.1.2: Consulta derivada en el proceso de aprendizaje

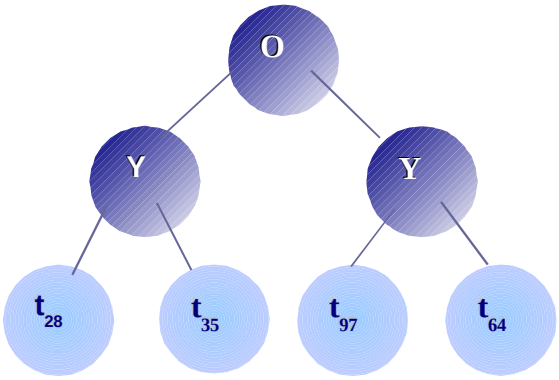


Figura A.1.3: Traducción de la consulta

Sin embargo, si la consulta está formada por términos que no se encuentran en el fichero de correspondencia, será necesario eliminarlos, adaptando la consulta para que siga teniendo una estructura correcta.

La manera de adaptar la consulta dependerá de si ésta se encuentra en forma normal (disyuntiva o conjuntiva) o si por el contrario sólo está representada como un árbol binario.

Si la consulta se encuentra en forma normal, se eliminarán la hoja del árbol que tiene asociado el término en cuestión, desechando cualquier subexpresión con menos de 2 términos. De igual forma, si como resultado del proceso anterior, la consulta resultado sólo constará de una subexpresión, consideraremos que la consulta no tiene traducción posible. Finalmente, a los términos que sí tienen su pareja en el conjunto de prueba, bastará con cambiarles la numeración.

La Figura A.1.4 muestra un ejemplo de traducción cuando existen nodos sin correspondencia en el conjunto de prueba.

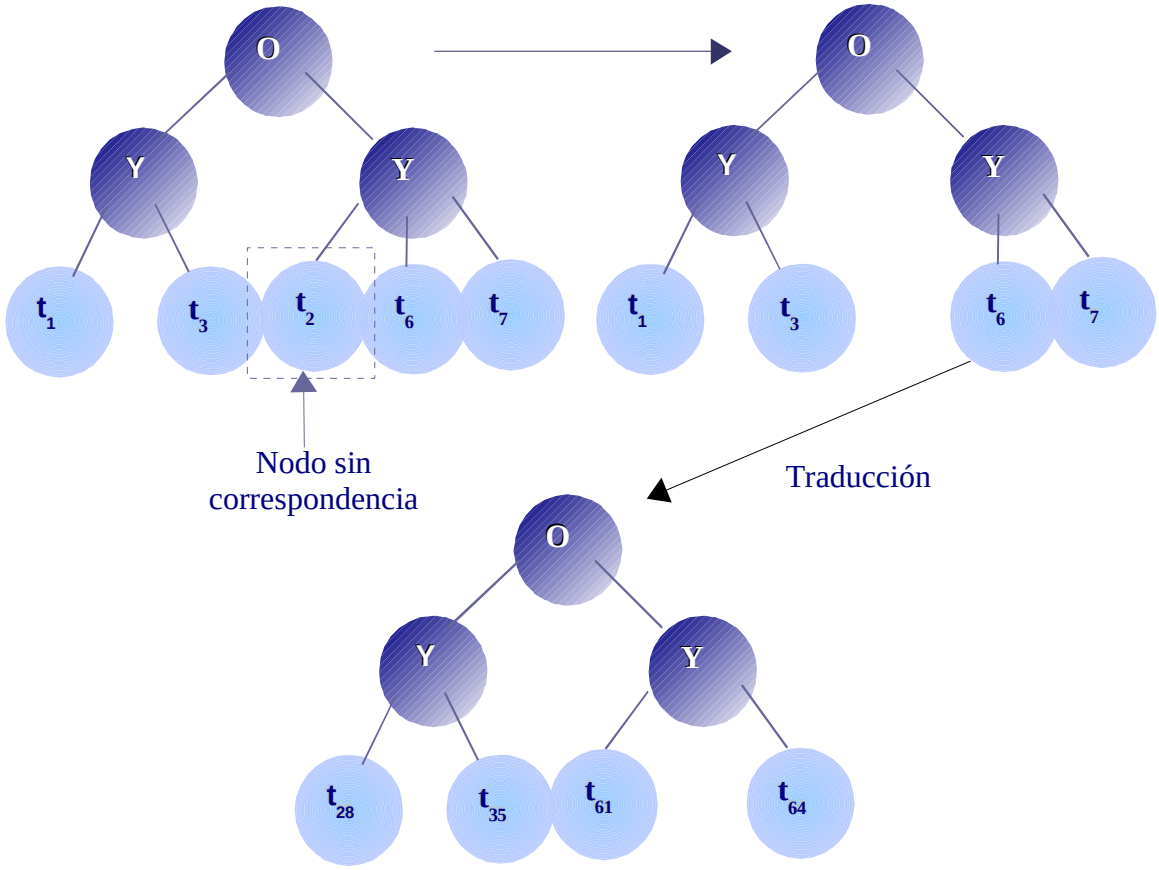


Figura A.1.4: Proceso de traducción cuando hay nodos sin correspondencia.

En el segundo caso el procedimiento consistirá en eliminar, por un lado, la hoja del árbol que tiene asociado el término en cuestión, y por otro, el nodo interno (operador) del que cuelga dicha hoja, colocando en lugar del nodo eliminado, el otro subárbol asociado con él. Este proceso se repetirá tantas veces como sea necesario hasta que se consiga una consulta con una estructura correcta. Finalmente, a los términos que sí tienen su pareja en el conjunto de prueba, bastará con cambiarles la numeración.

La Figura A.1.5 muestra un ejemplo de traducción cuando existen nodos sin correspondencia en el conjunto de prueba.

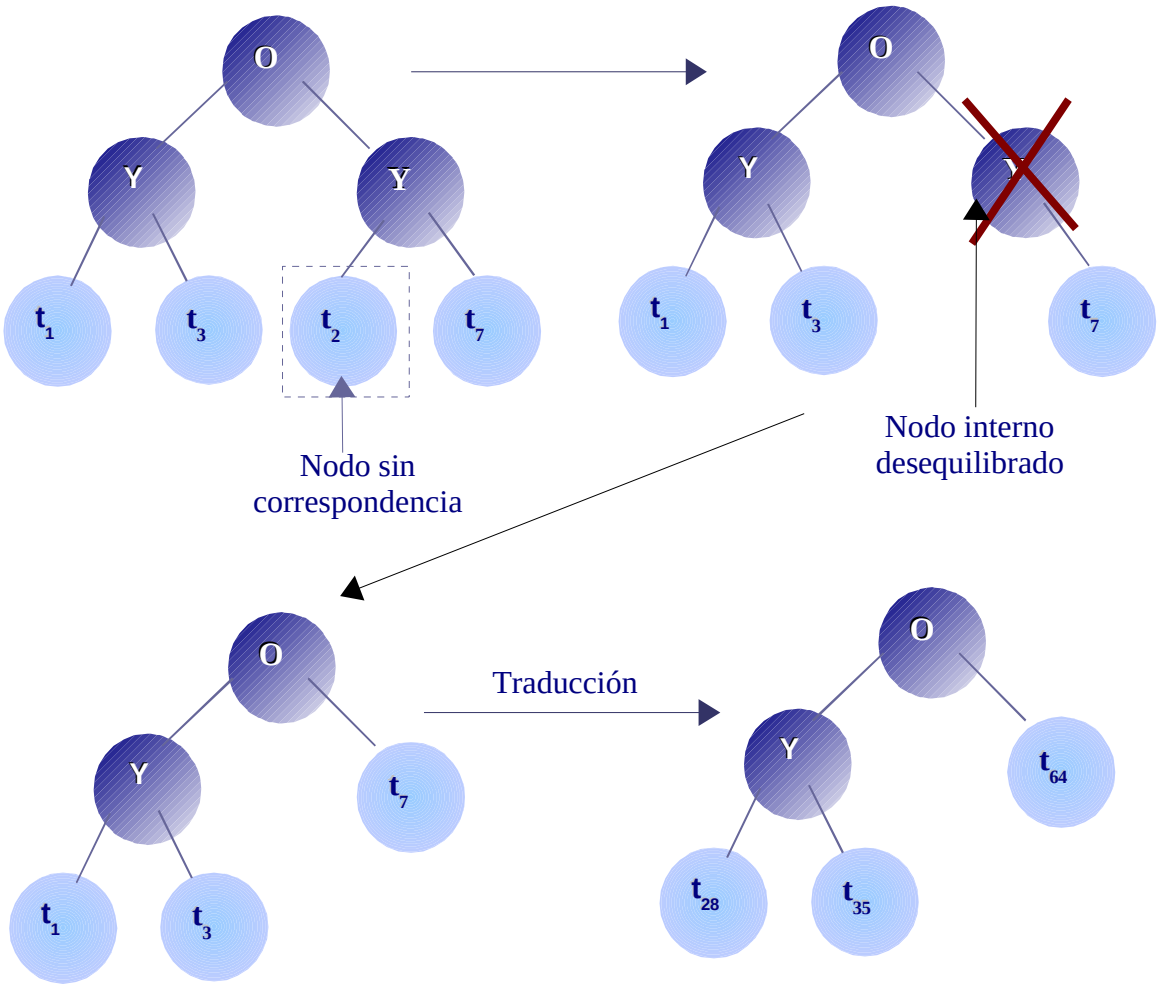


Figura A.1.5: Proceso de traducción cuando hay nodos sin correspondencia.

A.1.3.-Diseño experimental

Una vez analizadas las bases documentales y las consultas que vamos a emplear en nuestra experimentación, el siguiente paso consiste en describir nuestro entorno para generación de PQ, lo que realizaremos en esta sección.

En teoría, en un entorno de este tipo debe existir un usuario que proporcione documentos relacionados con su necesidad de información (es decir, documentos que para él son relevantes), que definen un patrón del tipo de documento que desearía obtener en sus búsquedas posteriores. Opcionalmente, el usuario puede también aportar documentos que no son relevantes para él (que no representan su necesidad de información), es decir, patrones que especifican el tipo de documento que no desearía recuperar. De este modo, se acotaría mejor la búsqueda y el algoritmo podría generar perfiles más específicos.

En nuestro caso, vamos a trabajar haciendo uso de una metodología habitual en el área, la considerada por Kraft y otros en [116], y por Smith y Smith en [165]. El papel del usuario que aporta documentos lo jugarán las consultas de las colecciones consideradas y, más concretamente, los juicios de relevancia asociados a las mismas. De este modo, por ejemplo, si en la partición de entrenamiento de la colección Cranfield existen 13 documentos que son relevantes para la consulta 1, dicha consulta representaría una situación en la que el usuario aporta 13 documentos que están relacionados con su necesidad de información y 693 (es decir, el total de documentos en la partición, 706, menos los 13 relevantes), que no lo son.

Por tanto, en todos los casos, el algoritmo de aprendizaje de perfiles toma como conjunto documental de entrada la colección completa (o en nuestro caso la partición de entrenamiento), clasificada en dos grupos de documentos, los relevantes y los no relevantes para la necesidad de información del usuario. En principio, la situación podría parecer poco realista, ya que se podría argumentar que el número de documentos irrelevantes es excesivo para que un usuario lo proporcione en un entorno real de RI. Sin embargo, este entorno experimental es muy apropiado desde un punto de vista algorítmico, ya que permite discriminar de mejor modo la calidad de este tipo de técnicas al ser más complicado que las consultas generadas sean capaces de no recuperar el alto número de documentos irrelevantes existentes, consiguiendo a su vez un valor de precisión alto en la recuperación.

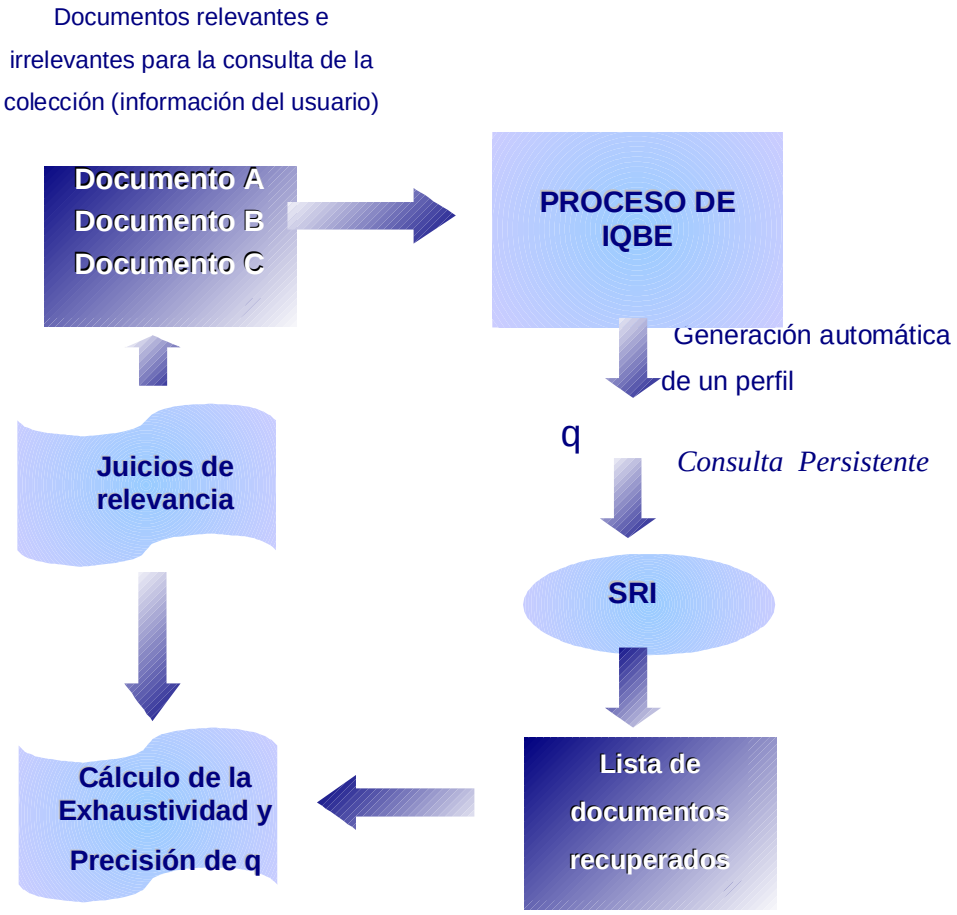


Figura A.1.6: Representación gráfica del entorno experimental considerado

El entorno experimental considerado queda reflejado gráficamente en la Figura A.1.6. Es importante destacar que, al contrario que en las aplicaciones de retroalimentación por relevancia en las que se consideran las estructuras de las consultas asociadas a las colecciones y se procesan del mismo modo que se procesan los documentos, en nuestro caso sólo haremos uso de los juicios de relevancia de las mismas, ya que el propio proceso aprenderá los perfiles partiendo de cero.

Una vez especificado el entorno experimental, vamos a analizar otros aspectos relacionados con la metodología experimental. En primer lugar, es importante destacar que realizaremos varias ejecuciones para cada algoritmo y consulta. Este es un procedimiento habitual cuando se trabaja con algoritmos probabilísticos como los AE, es decir, algoritmos que incluyen componentes aleatorias en sus decisiones y que, por tanto, pueden proporcionar

soluciones diferentes ante las mismas entradas en distintas ejecuciones.

De este modo, no es conveniente evaluar la calidad de este tipo de algoritmos realizando una única ejecución ya que, al depender de la aleatoriedad, podríamos encontrarnos con que la solución obtenida es mucho peor o mucho mejor de las que habitualmente obtendría ese algoritmo. Por tanto, el proceso experimental habitual consiste en ejecutar varias veces el algoritmo, fijando varias semillas distintas para el generador de números aleatorios, y en obtener resultados promedio de las soluciones devueltas por el algoritmo en las ejecuciones realizadas. Así, se trabaja con dos parámetros para medir la calidad del algoritmo: la mejor solución obtenida y la robustez, es decir, la calidad promedio del algoritmo.

En nuestro caso, ejecutaremos 5 veces cada algoritmo desarrollado para cada una de las consultas seleccionadas en Cranfield y CACM. Todas las ejecuciones se efectuarán con las mismas semillas, para que la comparación sea realista. Es decir, la primera ejecución realizada con cualquiera de los algoritmos considerará la semilla A, la segunda, la B, y así sucesivamente. En concreto, las 5 semillas escogidas son las siguientes: 123456, 56781, 522, 1114 y 23452.

Otro de los factores a destacar es la elección de los valores de los parámetros para la ejecución de los distintos algoritmos. El problema radica en el alto número de parámetros existentes y en la gran cantidad de valores que pueden tomar. Hemos optado por escoger los más habituales en el área de la Computación Evolutiva y por realizar experimentaciones preliminares para determinar aquellos de los que no se tiene información sobre una buena elección.

Así, el tamaño de la población se ha fijado a 800 individuos en los algoritmos basados en PG, siguiendo la pauta habitual de que los algoritmos de PG deben trabajar con tamaños de población grandes para no perder diversidad durante la búsqueda y evitar la convergencia prematura; y a 100 individuos en el algoritmo basado en AG.

Las probabilidades de cruce y mutación (P_c y P_m) se han fijado a 0,8 y 0,2, respectivamente, valores usuales en el área. Para el tamaño del torneo en el mecanismo de selección, se ha optado por el valor 2, ya que el torneo binario ha demostrado muy buen comportamiento en una gran cantidad de aplicaciones de los AE. Por último, el número de evaluaciones de los algoritmos se ha concretado en 50.000, es decir, los algoritmos finalizarán

su ejecución cuando hayan llamado 50.000 veces a la función de evaluación[10].

Todas las ejecuciones realizadas en esta memoria han sido efectuadas en un ordenador con procesador Pentium IV a 2.4 Hz y con 1 GB de memoria RAM.

10 Es importante destacar el hecho de que se detendrá la ejecución del algoritmo en la generación en la que se alcance la evaluación número 50.000, es decir, que no finalizaremos automática la ejecución en el momento en que se evalúe la solución número 50.000, sino que esperaremos a terminar la generación actual. Por esta razón, el algoritmo evaluará habitualmente algunas soluciones más, aunque este número no será significativo.

A.2.-OPTIMIZACIÓN MULTIOBJETIVO CLÁSICA

Esta apéndice está dedicado a introducir la optimización evolutiva multiobjetivo, que ha sido considerada en nuestras propuestas para la derivación de consultas persistentes (perfiles) con diferentes balances entre exhaustividad y precisión en una sola ejecución. Repasaremos los conceptos básicos de la optimización multiobjetivo y las técnicas clásicas de resolución de problemas.

A.2.1.-Optimización Multiobjetivo

Muchos problemas reales se caracterizan por la existencia de múltiples medidas de actuación, las cuales deberían ser optimizadas, o al menos ser satisfechas, simultáneamente. Como el propio nombre sugiere, el problema de la optimización multiobjetivo consiste en el proceso de optimización simultánea de más de una función objetivo [30].

La falta de metodologías para resolver este tipo de problemas llevó a que, en un principio, se resolvieran como problemas monobjetivo. Sin embargo, no es correcto tomar esta determinación ya que existen diferencias entre los principios en que se basan los algoritmos que tratan un solo objetivo y los que trabajan con varios. De esta forma, al trabajar con problemas monobjetivo nos enfrentamos con la búsqueda de una solución que optimice esa única función objetivo, tarea distinta a la que se nos plantea al trabajar con problemas multiobjetivo. En este último caso, no pretendemos encontrar una solución óptima que se corresponda a cada una de las funciones objetivo, sino varias soluciones que satisfagan todos los objetivos a la vez de la mejor manera posible.

En este primer acercamiento intentaremos ver cuál es la principal tarea que se plantea al resolver un problema multiobjetivo: un problema de optimización multiobjetivo presenta un cierto número de funciones objetivo, las cuales hay que maximizar o minimizar. Como en un problema de optimización con un solo objetivo, también suele existir un número de restricciones que debe satisfacer cualquier solución factible.

La forma general de un problema de optimización multiobjetivo es la siguiente:

$$\textbf{Minimizar/ Maximizar : } f_m(x),\ m=1,...,NF;$$

$$\textbf{Sujeto a : } g_j(x) \geq 0;\ j=0,...,J;$$

$$h_k(x)=0;\ k=0,\ldots,K;$$

$$x_i^{(L)} \leq x_i \leq x_i^{(U)},\ i=1,\ \ldots,\ n$$

Una solución *x* es un vector de n variables de decisión $x = (x_1, x_2, ..., x_n)$. Como vemos, asociadas con el problema hay *J* restricciones de desigualdad y *K* de igualdad. El último conjunto de restricciones, denominado límite de las variables, restringe cada variable de decisión x para que tome un valor entre $x_i^{(L)}$ y $x_i^{(U)}$. Estos límites constituyen el espacio de decisión de las variables.

Si alguna solución x satisface todas las restricciones y los límites de las variables se la conoce como solución *factible*.

La mayoría de los algoritmos de optimización multiobjetivo usan el concepto de dominancia es su búsqueda del óptimo. A continuación describimos con detalle este concepto.

Concepto de dominancia:

En los algoritmos de optimización multiobjetivo, la preferencia entre dos soluciones se especifica en función de que una domine a la otra.

Sean NF funciones objetivo, definimos el operador ⊗, que trabaja a nivel de valores de función objetivo, y decimos que i ⊗ j si la solución i es mejor que la solución j en un objetivo en particular. Por otro lado, i ¬⊗ j si la solución i es peor que la solución j en un objetivo en particular. Por ejemplo, si la función objetivo trata de minimizar las soluciones, el operador ⊗ significará <, mientras que si la función objetivo busca maximizar, el operador será equivalente a >.

Una solución *x* se dice dominada por otra *y*, si y sólo si se cumplen las siguientes condiciones:

1. La solución x no es peor que *y* en todos los objetivos; $\neg (f_j(y) \otimes f_j(x))$, $\forall j = 1, 2, ..., NF$.

2. La solución *x* es estrictamente mejor que *y* en al menos un objetivo; $\exists j = 1, 2, ..., NF \mid f_j(x) \otimes f_j(y)$.

Si alguna de estas dos condiciones no se cumpliera, la solución *x* no dominaría a la solución *y*. Si la solución *x* domina a la solución *y* podremos decir que:

1. *y* es dominada por *x*.

2. *x* es no dominada por *y*.

3. *x* es no inferior a *y*.

La Figura A.2.1 muestra un ejemplo para aclarar el concepto.

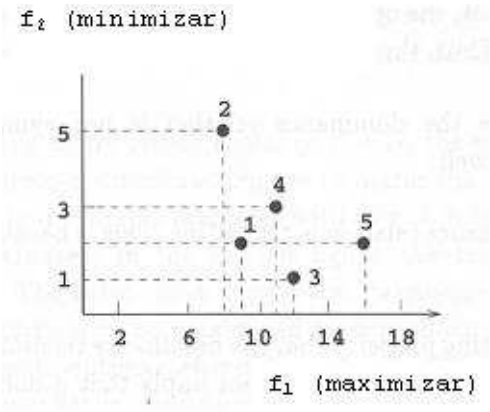


Figura A.2.1: Dominancia entre varias soluciones

Supongamos un problema con dos funciones objetivo: la primera de las funciones f_1 será una función a maximizar, mientras que f_2 será una función a minimizar.

Considerando la optimización conjunta de las dos funciones objetivo, es difícil encontrar una solución que sea la mejor respecto a ambas. Utilizando la definición de dominancia podremos decidir qué solución es la mejor de dos soluciones dadas en términos de ambos objetivos.

Por ejemplo: si comparamos las soluciones 1 y 2, observamos que la solución 1 es mejor que la 2 en las dos funciones objetivo f_1 y f_2. Esto supone que se cumplen las dos condiciones de dominancia, luego podremos afirmar que la solución 1 domina a la solución 2.

De forma intuitiva podemos decir que si una solución x domina a otra y, entonces x es mejor que y para la optimización multiobjetivo.

El concepto de dominancia proporciona una forma de comparar soluciones con múltiples objetivos. Como ya hemos comentado, la mayoría de los métodos de optimización multiobjetivo usan este concepto para buscar soluciones no dominadas.

Conjunto de Soluciones no Dominadas:

Si en el ejemplo que mostramos antes comparamos la solución 3 con la 5, observamos que la solución 5 es mejor que la solución 3 en el objetivo f_1 mientras que la solución 5 es peor que la solución 3 en el segundo objetivo, f_2. Vemos que la primera condición no se cumple para ambas soluciones, lo que simplemente nos dice que no podemos concluir que la solución 5 domine a la solución 3 ni tampoco que la 3 domine a la 5. Cuando esto ocurre, es costumbre decir que las soluciones 3 y 5 son no dominadas una respecto de la otra. Teniendo en cuenta ambos objetivos, no podemos decidir cuál de las dos soluciones es la mejor.

Para un conjunto de soluciones dado, podemos realizar todas las posibles comparaciones de pares de soluciones y encontrar cuáles dominan a cuáles y qué soluciones son no dominadas con respecto a las otras. Al final esperamos conseguir un conjunto de soluciones tales que cualesquiera dos no dominen una a la otra, es decir, un conjunto en el que todas las soluciones son no dominadas entre sí. Este conjunto se conoce como *conjunto de soluciones no-dominadas*.

Este conjunto también tiene otra propiedad: dada cualquiera solución que no pertenezca a él, siempre podremos encontrar una solución del conjunto que la domine. Así, este conjunto particular tiene la propiedad de dominar a todas las soluciones que no pertenecen al mismo. En términos simples, esto significa que las soluciones de este conjunto son mejores comparadas con el resto de soluciones.

Pareto-optimalidad:

Dado un conjunto de soluciones P, el conjunto de soluciones no dominadas P' está formado por aquellas que no son dominadas por ningún miembro del conjunto P.

Cuando el conjunto P es el espacio de búsqueda completo, P=S, el conjunto no dominado resultante se denomina conjunto Pareto-optimal.

A.2.2.-Técnicas Clásicas para Resolver Problemas Multiobjetivo

Una dificultad común en la optimización multiobjetivo es la aparición de un conflicto entre objetivos [85], es decir, ninguna de las soluciones factibles sea óptima simultáneamente para todos los objetivos. En este caso, la solución matemática más adecuada es quedarse con aquellas soluciones que ofrezcan el menor conflicto posible entre objetivos. Estas soluciones pueden verse como puntos en el espacio de búsqueda que están colocados de forma óptima a partir de los óptimos individuales de cada objetivo, aunque puede que dichas soluciones no satisfagan las preferencias del experto que quiera establecer algunas prioridades asociadas a los objetivos.

Para encontrar tales puntos, todas las técnicas clásicas reducen el vector objetivo a un escalar, es decir, a un único objetivo. En estos casos, en realidad, se trabaja con un problema sustituto buscando una solución sujeta a las restricciones especificadas.

A continuación vamos a ver tres de las técnicas clásicas más comunes para afrontar problemas con múltiples objetivos. Posteriormente, dedicaremos una sección a analizar los inconvenientes que presentan.

Optimización Mediante Ponderación de los Objetivos.

Ésta es probablemente la más simple de todas las técnicas clásicas. En este caso, las funciones objetivo se combinan en una función objetivo global, F, de la siguiente manera [77]:

$$F = \sum_{i=1}^{NF} w_i f_i(x)$$

donde los pesos w_i se definen como:

$$\Sigma_{i-1}^{NF} w_i = 1 \qquad \forall w_i \in R, \ \ 0 \leq w_i \leq 1$$

En este método, la solución óptima se controla mediante un vector de pesos w de forma que la preferencia de un objetivo puede cambiarse modificando dichos pesos. En muchos casos, primero se optimiza cada objetivo individualmente y después se calcula el valor de la función objetivo completa para cada uno de ellos. Así podemos conseguir evaluar la importancia que ejerce cada objetivo y encontrar un vector de pesos adecuado. Después, la solución final aceptada se calcula optimizando F según los pesos establecidos.

Las únicas ventajas de usar esta técnica es que se puede potenciar a un objetivo frente a otro y que la solución obtenida es normalmente Pareto-optimal.

Optimización Mediante Funciones de Distancia

Con esta técnica, la reducción a un escalar se lleva a cabo usando un vector de nivel de demanda, $\bar{y}$, que debe especificar el experto. Por esta razón, suele denominarse "programación por metas" (goal programming) [46]. En este caso, la función F se obtiene por medio de la siguiente fórmula:

$$F = \left[\sum_{i=1}^{NF} f_i(x) - \overline{y_i^r} \right]^{1/r} \qquad ,1 \leq r \leq \infty$$

Normalmente, se elige una métrica Euclídea r = 2, considerando $\bar{y}$ como óptimos de los objetivos individuales. Es importante recalcar que la solución obtenida depende enormemente del vector de nivel de demanda establecido, de modo que si éste es malo, no se llegará a una solución Pareto-optimal. Como la solución no está garantizada, el experto debe tener un conocimiento profundo de los óptimos individuales de cada objetivo para establecer adecuadamente $\bar{y}$. De esta forma, el método busca la meta indicada (representada por $\bar{y}$) para cada objetivo.

Esta técnica es similar a la anterior. La única diferencia es que ahora se requiere saber la meta de cada objetivo mientras que en el enfoque de ponderación era necesario conocer su importancia relativa.

Optimización Mediante Formulación Min-Max

Esta técnica intenta minimizar las desviaciones relativas de cada función objetivo a partir de óptimos individuales, esto es, intenta minimizar el conflicto entre objetivos [140]. El problema Min-Max se define como:

```
Minimizar f(x) = max [Zj (x)] ,   j = 1, 2, ..., NF ,
```

donde Z_j (x) se define para el valor óptimo positivo $\bar{f}_j > 0$ como:

$$Z_j(x) = \frac{|f_i - \overline{f_j}|}{\overline{f_j}}, \qquad j = 1,2,\ldots,NF$$

Esta técnica puede obtener la mejor solución cuando los objetivos a optimizar tienen igual prioridad. Sin embargo, la prioridad de cada objetivo puede alterarse utilizando pesos en la fórmula. También es posible introducir un vector de nivel de demanda.

Inconvenientes de las Técnicas Clásicas

Todas las técnicas clásicas que se han utilizado para resolver problemas multiobjetivo tienen graves inconvenientes que han dado lugar a que no sean adecuadas en muchas ocasiones. A continuación, mencionamos los más significativos:

- Dado que los distintos objetivos se combinan para formar uno único, sólo podremos obtener una solución Pareto-optimal simultáneamente. En situaciones reales, los expertos necesitan con frecuencia varias alternativas para decidir, pero estas técnicas no pueden ofrecerlas.

- Además, para realizar esta combinación, muchas veces es necesario tener un conocimiento sobre el problema que generalmente es difícil de obtener.

- No funcionan bien cuando los objetivos no son fiables o tienen un espacio de variables discontinuas.

- Si las funciones objetivo no son determinísticas, la elección de un vector de pesos o de niveles de demanda entraña gran dificultad.

- Son muy sensibles y dependientes de los pesos o niveles de demanda usados.

Para solucionar estos problemas han surgido técnicas avanzadas de optimización multiobjetivo basadas en Enfriamiento Simulado [8] [122], AGs [29], [53], EEs [117], etc.

BIBLIOGRAFÍA

BIBLIOGRAFÍA

[1] Bäck, T., *Evolutionary Algorithms in Theory and Practice*, Oxford University Press, 1996.

[2] Bäck, T., Fogel, D. B., Michalewicz, Z., *Handbook of Evolutionary Computation*, Institute of Physics Publishing and Oxford University Press, 1997.

[3] Bäck, T., Schwefel, H.P., Evolution Strategies I: Variants and their Computational Implementation. Periux, J., Winter, G., Galán, M., Cuesta, P., Eds. *Genetic Algorithms in Engineering and Computer Science* (Wiley, Chichester), pp. 111-126, 1995.

[4] Baeza-Yates, R., Information Retrieval in the Web: Beyond Current Search Engines, *International Journal of Approximate Reasoning*, 34(2-3), pp. 97-104, 2003.

[5] Baeza-Yates, R., Ribeiro-Neto, B., *Modern Information Retrieval*, Addison-Wesley, 1999.

[6] Banzhaf, W., Nordin, P., Keller, R., Francone, F., *Genetic Programming - An Introduction: On the Automatic Evolution of Computer Programs and Its Applications*, Academic Press / Morgan Kaufmann, 1998.

[7] Belkin, N.J. and Croft, W.B, Information Filtering and Information Retrieval: Two Sides of the Same Coin?, *Communications of the ACM*, 35(12), pp. 29-38, 1992.

[8] Bennage, W. A. and Dhingra, A. K., Single and Multiobjective Structural Optimization in Discrete-continuos Variables using Simulated Annealing, *International Journal for Numerical Methods in Engineering*, 38, pp. 2753-2773, 1995.

[9] Bergström, A., Jaksetic, P., and Nordin, P., Enhancing Information Retrieval by Automatic Acquisition of Textual Relations using Genetic Programming, Proceedings of 2000 International Conference on Intelligent User Interfaces, pp. 29-32, 2000.

[10] Bolc, L., Kowalski, A. and Kozlowska, M., A Natural Language Information Retrieval System with Extensions Towards Fuzzy Reasoning, *International Journal of Man-Machine Studies*, 23(4), pp. 335-367, 1996.

[11] Bonissone, P.P., Soft Computing: The Convergence of Emerging Reasoning Technologies, *Soft Computing*, 1(1), pp. 6-18, 1997.

[12] Bonissone, P.P., Decker, K.S., Selecting Uncertainty Calculi and Granularity: An Experiment in Trading-off Precision and Complexity. Kanal, L.H., Lemmer, J.F., Eds. *Uncertainty in Artificial Intelligence* (North-Holland), pp. 217-247, 1986.

[13] Bookstein, A., Fuzzy Request: An approach to Weighted Boolean Searches, *Journal of the American Society for Information Science and Technology*, 31, pp. 240-247, 1980.

[14] Bookstein, A., Outline of a General Probabilistic Retrieval Model, *Journal of Documentation*, 39(2), pp. 63-72, 1983.

[15] Bordogna, G., Carrara, P., Pasi, G., Query Terms Weights as Constraints in Fuzzy Information Retrieval, *Information Processing & Management*, 27(1), pp. 15-26, 1991.

[16] Bordogna, G., Carrara, P., Pasi, G., Fuzzy Approaches to Extend Boolean Information Retrieval. Bosc, P., Kacpryk, J., Eds. *Fuzzy Sets and Possibility Theory in Database Management Systems* (Springer Verlag), pp. 231-274, 1995.

[17] Bordogna, G., Pasi, G., A Fuzzy Linguistic Approach Generalizing Boolean Information Retrieval: A Model and its Evaluation, *Journal of the American Society for Information Science*, 44(2), pp. 70-82, 1993.

[18] Bordogna, G., Pasi, G., Linguistic Aggregation Operators of Selection Criteria in Fuzzy Information Retrieval, *International Journal of Intelligent Systems*, 10, pp. 233-248, 1995.

[19] Bordogna, G., Pasi, G., An Ordinal Information Retrieval Model, *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 9(1), pp. 63-75, 2001.

[20] Boughanem, M, Chrisment, C., Tamine, L., Genetic Approach to Query Space Exploration, *Information Retrieval*, 1(3), pp. 175-192, 1999.

[21] Boughanem, M, Chrisment, C., Tamine, L., On Using Genetic Algorithms for Multimodal Relevance Optimization in Information Retrieval, *Journal of the American Society for Information Science and Technology*, 53(11), pp. 934-942, 2002.

[22] Boughanem, M, Chrisment, C., Tamine, L., Multiple Query Evaluation Based on an Enhaced Genetic Algorithm, *Information Processing & Management*, 39(2), pp. 215-231, 2003.

[23] Buell, D., Kraft, D.H., A Model for a Weighted Retrieval System, *Journal of the American Society for Information Science*, 32, pp. 211-216, 1981.

[24] Buell, D. A., Kraft, D.H., Threshold Values and Boolean Retrieval System, *Information Processing & Management*, 17(3), pp. 127-136, 1981.

[25] Cater, S. C. and Kraft, D. H., A Generalization and Clarification on the Waller-Kraft Wish-List, *Information Processing & Management*, 25(1), pp. 15-25, 1989.

[26] Cleverdon, C.W., Design and Evaluation of Information System, *Annual Review of Information Science and Technology*, 6, pp. 42-73, 1971.

[27] Cleverdon, C.W., On the Inverse Relationship of Recall and Precision, *Journal of Documentation*, 28, pp. 195-201, 1972.

[28] Cleverdon, C.W., Keen, E.M., "Factors Determining the Performance of Indexing Systems", College of Aeronautics, Cranfield, UK, Informe Técnico, 1966.

[29] Coello, C. A., Van Veldhuizen, D. A., and Lamont, G. B., *Evolutionary Algorithms for Solving Multi-objective Problems*, Klumer Academia Publishers, 2002.

[30] Cohon, J.L., *Multiobjective Programming and Planning*, Academic Press, 1978.

[31] Cooper, W., Getting Beyond Boole, *Information Processing & Management*, 24, pp. 243-248, 1988.

[32] Cordón, O., Herrera, F., Hoffmann, F., Magdalena, L., *Genetic Fuzzy Systems. Evolutionary Tuning and Learning of Fuzzy Knowledge Bases*, World Scientific, 2001.

[33] Cordón, O., Herrera-Viedma, E., Preface to the Special Issue on Soft Computing Applications to Intelligent Information Retrieval on the Internet, *International Journal of Approximate Reasoning*, 34(2-3), pp. 89-95, 2003.

[34] Cordón, O., Herrera-Viedma, E., López-Pujalte, C., Luque, M., Zarco, C., A Review on the Application of Evolutionary Computation to Information Retrieval, *International Journal of Approximate Reasoning*, 34(2-3), pp. 241-264, 2003.

[35] Cordón, O., Herrera-Viedma, E., Luque, M., Evolutionary Learning of Boolean Queries by Multiobjective Genetic Programming, Lecture Notes in Computer Science, 2439, pp. 710-719, 2002.

[36] Cordón, O., Herrera-Viedma, E., Luque, M., Validación de un Algoritmo de PG Multiobjetivo para el Aprendizaje de Consultas Booleanas en un Entorno Realista de Recuperación de Información, Actas del Segundo Congreso Español sobre Metaheurísticas, Algoritmos Evolutivos y Bioinspirados (MAEB'03), pp. 437-444, 2003.

[37] Cordón, O., Herrera-Viedma, E., Luque, M., Improving the Learning of Boolean Queries by means of a Multiobjective IQBE Evolutionary Algorithm, Information Processing & Management, 2005, por aparecer.

[38] Cordón, O., Moya, F., Zarco, C., Breve Estudio sobre la Aplicación de los Algoritmos Genéticos a la Recuperación de la Información, Actas del Sexto Congreso ISKO-España, pp. 179-186, 1999.

[39] Cordón, O., Moya, F., Zarco, C., A GA-P Algorithm to Automatically Formulate Extended Boolean Queries for a Fuzzy Information Retrieval System., *Mathware & Soft Computing*, 7(2-3), pp. 309-322, 2000.

[40] Cordón, O., Moya, F., Zarco, C., A New Evolutionary Algorithm Combining Simulated Annealing and Genetic Programming for Relevance Feedback in Fuzzy Inforamation Retrieval Systems, *Soft Computing*, 6(5), pp. 308-319, 2002.

[41] Cordón, O. Moya, F., Zarco, C., Automatic Learning of Multiple Extended Boolean Queries by Multiobjective GA-P Algorithms. Loia, V., Nikravesh, M., Zadeh, L. A., Eds. *Fuzzy Logic and the Internet* (Springer), pp. 47-70, 2004.

[42] Crestani, F., Pasi, G., *Soft Computing in Information Retrieval: Techniques and Applications*, Physica-Verlag, New York, 2000.

[43] Crestani, F., Pasi, G., Handling Vagueness, Subjectivity, and Imprecision in Information Access: An Introducction to the Special Issue, *Information Processing & Management*, 39(2), pp. 161-165, 2003.

[44] Croft, W.B., Experiments with Representation in a Document Retrieval System, *Information Technology: Research and Development*, 2(1), pp. 1-21, 1983.

[45] Croft, W.B., Harper, D.J., Using Probabilistic Models of Document Retrieval without Relevance Information, *Journal of Documentation*, 35(4), pp. 285-295, 1979.

[46] Charnes, A., Cooper, W. W., *Management Models and Industrial Applications of Linear Programming*, John Wiley, 1961.

[47] Chen, H., Machine Learning for Information Retrieval: Neural Networks, Symbolic Learning, and Genetic Algorithms, *Journal of the American Society for Information Science*, 46(3), pp. 194-216, 1995.

[48] Chen, H., Preface to the Special Issue: "Web Retrieval and Mining", *Decision Support Systems*, 35, pp. 1-5, 2003.

[49] Chen, H., Shankaranarayanan, G., She, L. Iyer, A., A Machine Learning Approach to Inductive Query by Example: An Experiment Using Relevance Feedback, ID3, Genetic Algorithms, and Simulated Annealing, *Journal of the American Society for Information Science*, 49(8), pp. 693-705, 1998.

[50] Chen, Y., Shahabi, C., Automatically Improving the Accuracy of User Profiles with Genetic Algorithm, Proceedings of IASTED International Conference on Artificial Intelligence and Soft Computing, pp. 283-288, 2001.

[51] Cho, S., Lee, J., A Human-Oriented Image Retrieval System Using Interactive Genetic Algorithm, *IEEE Transactions on Systems, Man and Cybernetics. Part A: Systems and Humans*, 32(3), pp. 452-458, 2002.

[52] Deb, K., Agrawal, S., Pratap, A., Meyarivan, T., A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II, Proceedings of the Parallel Problem Solving from Nature VI Conference, pp. 849-858, 2000.

[53] Deb, K., Goldberg, D.E, An Investigation of Niche and Species Formation In Genetic Function Optimisation, Third International Conference on Genetic Algorithms, pp. 42-50, 1989.

[54] Dervin, B., Nilan, M., Information Needs and Uses, *Annual Review of Information Science and Technology*, 21, pp. 3-33, 1986.

[55] Dillon, M., Desper, J., Automatic Relevance Feedback in Boolean Retrieval Systems, *Journal of Documentation*, 36(3), pp. 197-208, 1980.

[56] Dillon, M., Ulmschneider, J., Desper, J., A Prevalence Formula for Automatic Relevance Feedback in Boolean Systems, *Information Processing & Management*, 19(1), pp. 27-36, 1983.

[57] Ellis, D., The Physical and Cognitve Paradigms in Information Retrieval Research, *Journal of Documentation*, 48(1), pp. 45-64, 1992.

[58] Eshelman, L.J., Schafer, D.J., Real Coded Genetic Algorithms and Interval-schemata. L.D. Whitley, Eds. *Foundations of Genetic Algorithms 2* (Morgan Kaufmann), pp. 187-202, 1993.

[59] Fan, W., Gordon, M.D., and Pathak, P., Automatic Generation of Matching Functions by Genetic Programming for Effective Information Retrieval, Proceedings of the 1999 Americas Conference on Information Systems, pp. 49-51, 1999.

[60] Fan, W., Gordon, M.D., Pathak, P., Effective Profiling of Consumer Information Retrieval Needs: A Unified Framework and Empirical Comparision, Decision Support Systems, 2005, por aparecer.

[61] Fan, W., Gordon, M.D., Pathak, P., An Integrated Two-Stage Model for Intelligent Information Routing, Sometido a Decision Support Systems, 2004.

[62] Fan, W., Gordon, M.D and Patkak, P., Discovery of Context-Specific Ranking Functions for Effective Information Retrieval Using Genetic Programming, *IEEE Transaction on Knowledge and Data Engineering*, 16(4), pp. 523-527, 2004.

[63] Fan, W. Gordon, M.D., Pathak, P., A Generic Ranking Function Discovery Framework by Genetic Programming for Information Retrieval, *Information Processing & Management*, 40(4), pp. 587-602, 2004.

[64] Fernández-Villacañas, J.L., BTGP and Information Retrieval, Proceedings of the Second International Conference of Adaptative Computing in Engineering Design and Control (ACEDC'96), pp. 297-309, 1996.

[65] Fernández-Villacañas, J.L., Shacketon, M., Investigation of the Importance of the Genotype-Phenotype Mapping in Information Retrieval, *Future Generation of Computer Systems*, 19(1), pp. 55-68, 2003.

[66] Fogel, D. B., *System Identification trough Simulated Evolution: A Machine Learning Approach*, Ginn Press, 1991.

[67] Fogel, L.J., Owens, A.J., Walsh, M.J., *Artificial Intelligence through Simulated Evolution*, John Willey and Sons, 1966.

[68] Fonseca, C.M., Fleming, P.J., Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization, Genetic Algorithms: Proceedings of the Fifth International Conference, pp. 416-423, 1993.

[69] Fonseca, C.M., Fleming, P.J., On the Performance Assesment and Comparision of Stochastic Multiobjective Optimiziers, Proceedings of the Fourth International Conference on Parallel Problem Solving from Nature (PPSN IV), pp. 584-593, 1996.

[70] Fox, E.A., Characterization of Two New Experimental Collections in Computer and Information Science Containing Textual and Bibliographic Concepts, Universidad de Cornell, Informe Técnico, 1983.

[71] Fox, E. A., Extending the Boolean and Vector Space Models of Information Retrieval with P-Norm Queries and Multiple Concept Types. Tesis Doctoral. 1983. .

[72] Frakes, W., Stemming Algorithms. Frakes, W., Baeza-Yates, R., Eds. *Information Retrieval. Data Structures & Algorithms* (Prentice-Hall), pp. 131-160, 1992.

[73] Frakes, W., Introduction to Information Storage and Retrieval Systems. Frakes, W., Baeza-Yates, R., Eds. *Information Retrieval. Data Structure & Algorithms* (Prentice-Hall), pp. 1-12, 1992.

[74] Freitas, A.A., *Data Mining and Knowledge Discovery with Evolutionary Algorithms*, Springer, 2002.

[75] Fuhr, N., Probabilistic Models in Information Retrieval, *Computer Journal*, 35(3), pp. 243-255, 1992.

[76] Furnas, G.W., Landauer, T.K., Gomez, L.M., Dumais, S.T., The Vocabulary Problem in Human-System Communication, *Communications of ACM*, 30(11), pp. 947-971, 1987.

[77] Gass, S., Saaty, T.L., The Computational Algorithm for the Parametric Objective Function, *Naval Research Logistics Quaterly*, 2, pp. 39-45, 1955.

[78] Geyer-Schulz, A., *Fuzzy Rule-Based Expert Systems and Genetic Machine Learning*, Physica-Verlag, 1995.

[79] Goldberg, D.E., *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, 1989.

[80] Gordon, M.D., Probabilistic and Genetic Algorithms for Document Retrieval, *Communications of the ACM*, 31(10), pp. 1208-1218, 1988.

[81] Gordon, M. D., User-Based Document Clustering by Redescribing Subject Descriptions with a Genetic Algorithm, *Journal of the American Society for Information Science*, 42(5), pp. 311-322, 1991.

[82] Grefenstette, J.J., *Genetic Algorithms for Machine Learning*, Kluwer Academic Publishers, 1995.

[83] Hajela, P., Lee, E., Lin, C.Y., Genetic Algorithms in Structural Topology Optimization, Proceedings of the NATO Advanced Research Workshop on Topology Design of Structures, pp. 117-133, 1993.

[84] Hanani, U., Shapira, B., Shoval, P., Information Filtering: Overview of Issues, Research and Systems, *User Modeling and User-Adapted Interaction*, 11(3), pp. 203-259, 2001.

[85] Hans, A.E., *Multicriteria Optimization in Engineering and Sciences, Mathematical concepts and Methods in Science and Engineering*, Plenum Press, 1988.

[86] Harman, D. W., An Experimental Study of Factors Important in Document Ranking, Proceedings of the 9th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 186-193, 1986.

[87] Herrera, F., and Martínez, L., A 2-tuple Fuzzy Linguistic Representation Model for Computing with Words, *IEEE Transaction on Fuzzy Systems*, 8(6), pp. 746-752, 2000.

[88] Herrera, F., Herrera-Viedma, E., Aggregation Operators for Linguistic Weighted Information, *IEEE Transactions on Systems, Man and Cybernetics*, 27, pp. 646-656, 1997.

[89] Herrera, F., Herrera-Viedma, E., Linguistic Decision Analysis: Steps for Solving Decision Problems Under Linguistic Information, *Fuzzy Sets and Systems*, 115, pp. 67-82, 2000.

[90] Herrera, F., Herrera-Viedma, E., Martínez, L., A Fusion Approach for Managing Multi-Granularity Linguistic Term Sets in Decision Making, *Fuzzy Sets and Systems*, 114(1), pp. 43-58, 2000.

[91] Herrera, F., Herrera-Viedma, E., Verdagay, J.L., Direct Approach Processes in Group Decision Making Using Linguistic OWA Operators, *Fuzzy Sets and Systems*, 79, pp. 175-190, 1996.

[92] Herrera, F., Verdegay, J.L., Linguistic Assessments in Group Decision, First European Congress as Fuzzy and Intelligent Technologies (EUFIT'93), pp. 941-948, 1993.

[93] Herrera-Viedma, E., An Information Retrieval System with Ordinal Linguistic Weighted Queries Based on Two Weighting Semantics, Proceedings of the 7th International Conference on Information Processing and Management of Uncertainty in Knowledge-Bases Systems (IPMU'00), Vol. I, pp. 454-461, 2000.

[94] Herrera-Viedma, E., Modeling the Query Subsystem of a Linguistic IRS for Expressing Qualitative and Quantitative Restrictions on Query Terms, Actas del X Congreso Español sobre Tecnologías y Lógica Fuzzy (ESTYLF'00), pp. 181-186, 2000.

[95] Herrera-Viedma, E., Modeling the Retrieval Process of an Information Retrieval System Using an Ordinal Fuzzy Linguistic Approach, *Journal of the American Society for Information Science and Technology*, 52(6), pp. 460-475, 2001.

[96] Herrera-Viedma, E., An Information Retrieval System with Ordinal Linguistic Weighted Queries based on Two Weighting Elements, *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 9(1), pp. 77-88, 2001.

[97] Herrera-Viedma, E., Cordón, O., Herrera, J.C., Luque, M., An IRS Based on Multi-granular Linguistic Information, Proceedings of the 7th International Conference of the International Society for Knowledge Organization (ISKO'02), pp. 372-378, 2002.

[98] Herrera-Viedma, E., Cordón, O., Luque, M., López, A.G., A Model of a Fuzzy Multi-Granular Linguistic IRS, Proceedings of the 10th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU 2004), pp. 1365-1372, 2004.

[99] Herrera-Viedma, E., Cordón, O., Luque, M., López, A.G., Muñoz, A.M., A Model of Fuzzy Linguistic IRS Based on Multigranular Linguistic Information, *International Journal of Approximate Reasoning*, 34(2-3), pp. 221-239, 2003.

[100] Herrera-Viedma, E., Herrera, F., Cordón, O., Luque, M., López, A.G, An Information Retrieval System with Weighted Querying Based on Multi-Granular Linguistic Information, Proceedings of EUROFUSE 2002, pp. 127-133, 2002.

[101] Holland, J. H., *Adaptation in Natural and Artificial Systems*, Ann Arbor: The University of Michigan Press, 1975.

[102] Horn, J., Nafploitis, N., Goldberg, D., A Niched Pareto Genetic Algorithm for Multi-Objective Optimization, Proceedings of the First IEEE Conference on Evolutionary Computation, pp. 82-87, 1994.

[103] Horn, J-T., Yeh, C.-C., Applying Genetic Algorithms to Query Optimization in Document Retrieval, *Information processing & Management*, 36(5), pp. 737-759, 2000.

[104] Howard, L.M., D'Angelo, D.J., The GA-P: A Genetic Algorithm and Genetic Programming Hybrid, *IEEE Expert*, 10(3), pp. 11-15, 1995.

[105] Hsinchun, C., Yi-Ming, C., Ramsey, M., Yang, C., An Intelligent Personal Spider (Agent) for Dynamic Internet/Intranet Searching, *Decision Support Systems*, 23(1), pp. 41-58, 1998.

[106] Ide, E., New Experiments in Relevance Feedback. Salton, G., Eds. *The SMART Retrieval System* (Prentice Hall), pp. 337-354, 1971.

[107] Ingwersen, P., Cognitive Perspective of Information Retrieval Interaction: Elements of a Cognitive IR Theory, *Journal of Documentation*, 52(1), pp. 3-50, 1996.

[108] Ingwersen, P., Willet, P., An Introduction to Algorithmic and Cognitive Approaches for Information Retrieval, *Libri*, 45(3-4), pp. 169-177, 1995.

[109] Kato, S., Iisaku, S., An Image Retrieval Method Based on a Genetic Algorithm, Proceedings of the Twelfth International Conference on Information Networking (ICOIN-12), pp. 333-336, 1998.

[110] Knowles, J.D., Corne, D.W., Approximating the Non-Dominated Front using the Pareto archived Evolution Strategy, *Evolutionary Computation*, 8(2), pp. 149-172, 2000.

[111] Kobayashi, M., Takeda, K., Information Retrieval on the Web, *ACM Computing Surveys*, 32(2), pp. 144-173, 2000.

[112] Korfhage, R., *Information Storage and Retrieval*, New York: Wiley, 1997.

[113] Koza, J., *Genetic Programming: On the Programming Computers by Means of Natural Selection*, Cambridge: MIT Press, 1992.

[114] Kraft, D.H., Bordogna, G., Pasi, G., An Extended Fuzzy Linguistic Approach to Generalize Boolean Information Retrieval, *Information Sciences*, 2(3), pp. 119-134, 1994.

[115] Kraft, D.H., Buell, D.A, Fuzzy Sets and Generalized Boolean Retrieval Systems, *International Journal of Man-Machine Studies*, 19, pp. 45-56, 1983.

[116] Kraft, D.H., Petry, F.E., Buckles, B.P., Sadasivan, T., Genetic Algorithms for Query Optimization in Information Retrieval: Relevance Feedback. Sanchez, E., Shibata, T., Zadeh, L.A., Eds. *Genetic Algorithms and Fuzzy Logic Systems* (World Scientific), pp. 155-173, 1997.

[117] Kursawe, F., A Variant of Evolution Strategies for Vector Optimization, Proceeding of the First Parallel Problem Solving from Nature, pp. 193-197, 1991.

[118] Lancaster, F.W., Criteria by Which Information Retrieval Sytstems May Be Evaluated. Lancaster, F.W., Eds. *Information Retrieval Systems - Characteristics, Testing and Evaluation* (Nueva York: Willey), 1979.

[119] Larsen, H., Marín, N., Martín, M.J., Vila, M.A., Using Genetic Feature Selection for Optimizing User Profile, *Mathware & Soft Computing*, 7(2-3), pp. 275-286, 2000.

[120] Lawrence, S., Giles, C.L., Searching the World Wide Web, *Science*, 280(5360), pp. 98-100, 1998.

[121] Lawrence, S., Giles, C.L., Searching the Web: General and Scientific Information Access, *IEEE Communications Magazine*, 37(1), pp. 116-122, 1999.

[122] Lee, S., Wang, H., Modified Simulated Annealing for Multiple Objective Engineering Design Optimization, *Journal of Intelligent Manufacturing*, 3, pp. 101-108, 1992.

[123] Levenshtein, V.I., Binary Codes Capable of Correcting Deletions, Insertions and Reversal, *Soviet Physics-Doklady*, 6, pp. 705-710, 1996.

[124] Lewis, D.D., Appliying Support Vector Machines to the TREC-2001 Batch Filtering and Routing Tasks, Proceedings of the Tenth Text Retrieval Conference, pp. 1-5, 2001.

[125] Lochbaum, K., Streeter, L., Comparing and Combining the Efectiveness of Latent Semantic Indexing and the Ordinary Vector Space Model for Information Retrieval, *Information Processing & Management*, 25(6), pp. 665-676, 1989.

[126] Loia, V., Luengo, P., An Evolutionary Approach to Automatic Web Page Categorization and Updating, First Asia-Pacific Conference, pp. 292-302, 2001.

[127] López-Pujalte, C., Algoritmos Genéticos Aplicados a la Retroalimentación por Relevancia. Tesis Doctoral. 2000. Universidad de Granada.

[128] López-Pujalte, C., Guerrero-Bote, V., Moya, F., A Test of Genetic Algorithms in Relevance Feedback, *Information Processing & Management*, 38(1), pp. 793-805, 2002.

[129] López-Pujalte, C., Guerrero-Bote, V., Moya, F., Order-Based Fitness Functions for Genetic Algorithms Applied to Relevance Feedback, *Journal of the American Society for Information Science and Technology*, 54(2), pp. 152-160, 2003.

[130] Marchioni, G., *Information Seeking in Electronic Environments*, Cambridge University Press, 1995.

[131] Martín-Bautista, M.J., Vila, M.A., Larsen, H.L., A Fuzzy Genetic Algorithm Approach to an Adaptative Information Retrieval Agent, *Journal of the American Society for Information Science*, 50(9), pp. 760-771, 1999.

[132] Maruyama, M., The Second Sybernectics: Desviation-Amplifying Casual Processes, *American Scientist*, 51(2), pp. 164-179, 1963.

[133] Michalewicz, Z., *Genetic Algorithms + Data Structures = Evolution Programs*, Springer-Verlag, 1996.

[134] Miller, G.A., The Magical Number Seven, Plus Minus Two: Some Limits on Our Capacity of Processing Information, *Psychological Review*, 63, pp. 81-97, 1956.

[135] Mitchell, T., *Machine Learning*, Mc-Graw Hill, 1997.

[136] Miyamoto, S., *Fuzzy Sets in Information Retrieval and Cluster Analysis*, Kluwer Academic Publishers, 1990.

[137] Nikravesh, M., Loia, V., Azvine, B., Fuzzy Logic and the Internet (FLINT): Internet, World Wide Web and Search Engines, *Soft Computing*, 6(5), pp. 287-299, 2002.

[138] Oard, D.W., Marchionini, G., A Conceptual Framework for Text Filtering, University of Maryland, College Park, CS-TR-3643, 1996.

[139] Olvera, M.D., Evaluación de la Recuperación de Información en Internet: Un Modelo Experimental. Tesis Doctoral. 1999. Universidad de Granada.

[140] Osyczka, A., An Approach to Multicriterion Optimization Problems for Engineering Design, *Computer Methods in Applied Mechanics and Engineering*, 15, pp. 309-333, 1978.

[141] Pasi, G., Intelligent Information Retrieval: Some Research Trends. Benítez, J.M. , Cordón, O., Hoffman, F., Roy, R., Eds. *Advances in Soft Computing. Engineering Design and Manufacturing* (Springer), pp. 157-171, 2003.

[142] Pathak, P., Gordon, M.D., Fan, W., Effective Information Retrieval using Genetic Algorithms Based Matching Function Adaptation, Proceedings of the 33rd Hawaii International Conference on System Science, (HICSS), pp. , 2000.

[143] Porter, M.F., An Algorithm for Suffix Stripping, *Program*, 14(3), pp. 130-137, 1980.

[144] Radecki, T., Fuzzy Set Theoretical Approach to Document Retrieval Information, *Information Processing & Management*, 15, pp. 247-260, 1979.

[145] Robertson, A.M., Willet, P., Generation of Equifrecuent Groups of Words Using a Genetic Algorithm, *Journal of Documentation*, 50(3), pp. 213-232, 1994.

[146] Robertson, A.M., Willet, P., An Upperbound to the Performance of Ranked-Output Searching: Optimal Weighting of Query Terms Using a Genetic Algorithm, *Journal of Documentation*, 52(4), pp. 405-420, 1996.

[147] Robertson, S., On Relevance Weight Estimation and Query Expasion, *Journal of Documentation*, 42(3), pp. 182-188, 1986.

[148] Robertson, S., On Term Selection for Query Expansion, *Journal of Documentation*, 46, pp. 359-364, 1990.

[149] Robertson, S.E., Sparck-Jones, K., Relevance Weighting of Search Terms, *Journal of the American Society for Information Science*, 27, pp. 129-145, 1976.

[150] Robertson, S.E., Walker, S., Jones, S., Hancock-Beaulieu, M.H., Gatford, M., Okapi at TREC-4. Harman, D.K, Eds. *Proceedings of the Fourth Text Retrieval Conference* (NIST Special Publications), pp. 73-97, 1996.

[151] Rocchio, J.J., Relevance Feedback in Information Retrieval. Salton, G., Eds. *The SMART Retrieval System: Experiments in Automatic Processing* (Prentice-Hall), pp. 313-323, 1971.

[152] Salton, G., *The Smart Retrieval System. Experiments in Automatic Document Processing*, Englewood Cliffs: Prentice-Hall, 1971.

[153] Salton, G., *Automatic Text Processing: The Transformation, Analysis and Retrieval of Information by Computer*, Addison-Wesley, 1989.

[154] Salton, G., Buckley, C., Term-weighting Approaches in Automatic Text Retrieval, *Information Processing & Management*, 24(5), pp. 513-523, 1988.

[155] Salton, G., McGill, M.J., *An Introduction to Modern Information Retrieval*, McGraw-Hill, 1983.

[156] Sanchez, E., Importance in Knowledge-based Systems, *Information Systems*, 14(6), pp. 455-464, 1989.

[157] Sanchez, E., Miyano, H., Brachet, J.P., Optimization of Fuzzy Queries with Genetic Algorithms. Application to a Data Base of Patents in Biomedical Engineering, Sixth IFSA World Congress, pp. 293-296, 1995.

[158] Sánchez, L., A Niching Scheme for Steady State GA-P and its Applications to Fuzzy Rule Based Classifiers Induction, *Mathware & Soft Computing*, 7(2-3), pp. 337-350, 2000.

[159] Sánchez, L., Couso, I., Corrales, J.A., Combining GP Operators with SA Search to Evolve Fuzzy Rule Based Classifiers, *Information Sciences*, 136(1-4), pp. 175-191, 2001.

[160] Schaffer, J.D., Multiple Objective Optimization with Vector Evaluated Genetic Algorithms, Proceedings of the First International Conference on Genetic Algorithms, pp. 93-100, 1985.

[161] Schaffer, J D., *Some Experiments in Machine Learning Using Vector Evaluated Genetic Algorithms*, Nashville, Vanderbilt University, 1984.

[162] Schwefel, H.P., *Evolution and Optimum Seeking*, New York: John Wiley and Sons, 1995.

[163] Shih, T.K., Agent Evolution Computing, *Information Sciences*, 137(1-4), pp. 53-73, 2001.

[164] Smith, L.C, Artificial Intelligence and Information Retrieval, *Annual Review of Information Science and Technology*, 22, pp. 41-77, 1987.

[165] Smith, M.P., Smith, M., The Use of Genetic Programming to Build Boolean Queries for Text Retrieval Through Relevance Feedback, *Journal of Information Science*, 23(6), pp. 423-431, 1997.

[166] Sparck Jones, K., A Statistical Interpretation of Term Specificity and Its Application in Retrieval, *Journal of Documentation*, 28(1), pp. 11-20, 1972.

[167] Sparck Jones, K., Experiments in Relevance Weighting of Search Terms, *Imformation Processing & Management*, 15, pp. 133-144, 1979.

[168] Srinivas, N., Deb, K., Multi-objective Function Optimization using Non-dominated sorting Genetic Algorithms, *Evolutionary Computation*, 2(3), pp. 221-248, 1994.

[169] Stejic, Z., Takam, Y., Hirota, K., Genetic Algorithm-based Relevance Feedback for Image Retrieval using Local Similarity Patterns, *Information Processing & Management*, 34(4), pp. 405-415, 2003.

[170] Thrift, P., Fuzzy Logic Synthesis with Genetic Algorithms, Proceedings of the Fourth International Conference on Genetic Algorithms, pp. 509-513, 1991.

[171] Tong, M., Bonissone, P.P., A Linguistic Approach to Decision Making with Fuzzy Sets, *IEEE Transactions on Systems, Man and Cybernetics1*, 10(11), pp. 716-723, 1980.

[172] Torra, V., Negation Functions Based Semantics for Ordered Linguistic Labels, *International Journal of Intelligent Systems*, 11, pp. 975-988, 1996.

[173] Torra, V., Aggregation of Linguistic Labels when Semantics is based on Antonyms, *International Journal of Intelligent Systems*, 16, pp. 513-524, 2001.

[174] Trotman, A., Choosing Document Structure Weights, *Information Processing and Management*, 41(2), pp. 243-264, 2005.

[175] Turtle, H.R., Inference Networks for Document Retrieval. Tesis Doctoral. 1990. Universidad de Massachusetts.

[176] van Rijsbergen, C.J., *Information Retrieval*, Butterworth, 1979.

[177] Vrajitouru, D., Crossover Improvement for the Genetic Algorithm in Information Retrieval, *Information Processing & Management*, 34(4), pp. 405-415, 1998.

[178] Walker, R.L., Assessment of the Web using Genetic Programming, Proceedings of the Genetic and Evolutionary Computation Conference, pp. 1750-1755, 1999.

[179] Walker, R.L., Search Engine Case Study: Searching the Web using Genetic Programming and MPI, *Parallel Computing*, 27(1-2), pp. 71-89, 2001.

[180] Waller, W.G, Kraft, D.H, A Mathematical Model of a Weighted Boolean Retrieval System, *Information Processing & Management*, 15(5), pp. 235-245, 1979.

[181] Wiener, N., *The Human use of Human Beings: Cybernetics and Society*, Boston: Hougton Mifflin, 1950.

[182] Yager, R.R., A Note on Weighted Queries in Information Retrieval Systems., *Journal of the American Society for Information Science*, 38(1), pp. 23-24, 1986.

[183] Yager, R.R., On Ordered Weighted Averaging Aggregation Operators in Multicriteria Decision Making, *IEEE Trans. on Systems and Cybernetics*, 18(1), pp. 183-190, 1988.

[184] Yager, R.R., An Approach to Ordinal Decision Making, *International Journal of Approximate Reasoning*, 12, pp. 237-261, 1995.

[185] Yang, J.J, Korfhage, R.R., Query Modification Using Genetic Algorithms in Vector Space Models, *International Journal of Expert Systems*, 7(2), pp. 165-191, 1994.

[186] Zadeh, L.A., Fuzzy Sets, *Information and Control*, 8(3), pp. 338-353, 1965.

[187] Zadeh, L.A., The Concept of a Linguistic Variable and its Applications to Approximate Reasoning. Parts I, II, and III, *Information Sciences*, 8, pp. 199-224, 8, pp. 301-357, 9, pp. 43-80, 1975.

[188] Zadeh, L.A., Fuzzy Logic, Neural Networks and Soft Computing, *Communications of the ACM*, 37(3), pp. 77-84, 1994.

[189] Zadeh, L.A., What is Soft Computing?, *Soft Computing*, 1(1), pp. 1, 1997.

[190] Zadeh, L.A., Kacprzyk, J., *Fuzzy Logic for the Management of Uncertainty*, John Wiley, New York, 1992.

[191] Zipf, G.K., Thiele, L., *Human Behavior and the Principle of Least Effort*, Addison Wesley, Cambridge, 1949.

[192] Zitler, E., Deb, K., Thiele, L., Comparison of Multiobjective Evolutionary Algorithms: Empirical Results, *Evolutionary Computation*, 8(2), pp. 173-195, 2000.

[193] Zitzler, E., Thiele, L., An Evolutionary Algorithm for Multiobjective Optimization: The Strength Pareto Approach, Zürich, Switzerland, Informe Técnico, 1998.

[194] Zitzler, E., Thiele, L., Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach, *IEEE Transactions on Evolutionary Computation*, 3(4), pp. 257-271, 1999.