VNiVERSiDAD Đ SALAMANCA

DEPARTAMENTO Đ iNFORMÁTiCA Y AVTOMÁTiCA

CLASiFiCACiÓN AVTOMÁTiCA Đ iNFORMACiÓN EN PORTALES WEB MEDiANTE TÉCNiCAS Đ *CLVSTERiNG*

TESiS DOCTORAL

JVAN CARLOS ÁLVAREZ GARCíA

VNiVERSiDAD Đ SALAMANCA

2010

VNiVERSiDAD Đ SALAMANCA

DEPARTAMENTO Đ iNFORMÁTiCA Y AVTOMÁTiCA

TESiS DOCTORAL

CLASiFiCACiÓN AVTOMÁTiCA Đ iNFORMACiÓN EN PORTALES WEB MEDiANTE TÉCNiCAS Đ *CLVSTERiNG*

JVAN CARLOS ÁLVAREZ GARCíA

DiRECTOR:

DR. CARLOS GARCíA-FiGVEROLA PANiAGVA

2010

Don Carlos García-Figuerola Paniagua, Profesor Catedrático de Escuela Universitaria del Departamento de Informática y Automática de la Universidad de Salamanca

HACE CONSTAR

que don Juan Carlos Álvarez García, Ingeniero Informático por la Universidad de Salamanca, ha realizado bajo mi dirección la Memoria que lleva por título

Clasificación automática de información en portales web mediante técnicas de clustering,

con el fin de obtener el grado de Doctor por la Universidad de Salamanca.

Y para que surta los efectos oportunos firmo en Salamanca, a 1 de febrero de 2010.

A los que buscan hacer sus sueños realidad.

RESUMEN

La expresión, “Recuperación de la Información” (*Information Retrieval*), hace referencia al tratamiento automatizado que se lleva a cabo para poder dar respuesta a una necesidad de información. Engloba por una parte aspectos relacionados con la representación, almacenamiento y organización de la información y por otra parte aspectos relacionados con la eficiencia en la presentación de resultados como consecuencia de consultas. Se trata de proporcionar al usuario información válida que le sea relevante, no solamente datos, en la medida de lo posible clasificada o ponderada en cuanto a su grado de utilidad.

Son diversos los algoritmos de clasificación que se han utilizado. Suelen operar de acuerdo a un conjunto de premisas, que en muchos casos van a ser medibles o ponderables, que darán lugar a los distintos modelos de recuperación de la información. Modelos clásicos como el booleano, el vectorial o el probabilístico. Modelos alternativos a los clásicos, como el de conjuntos finitos, el booleano extendido, el espacio vector generalizado, el de indexación por la semántica latente, el de redes neuronales, el de red de inferencias o el de red de creencias. Se ha pretendido dar una visión de conjunto de todos ellos y una clasificación.

Por otra parte las técnicas de *clustering* son técnicas de análisis de datos en las que se aplican las observaciones según su similitud. Sus campos de aplicación son de lo más variados: actividades empresariales, microeconomía, información geográfica, bioinformática, genómica, segmentación de imágenes, procesamiento del lenguaje natural y un largo etcétera que incluye aspectos que queremos abordar como es la clasificación de documentos en Recuperación de la Información.

Se ha pretendido aplicar las reglas de *clustering* para clasificar de forma automatizada grandes cantidades de información que manejan habitualmente los directorios de muchos portales web, buscando repartir los documentos de colecciones en grupos, de forma que puedan ser aplicados posteriormente a otros problemas prácticos como puede ser la agrupación de documentos obtenidos en las búsquedas web, o la visualización de directorios.

Para ello ha sido necesario analizar las distintas técnicas de *clustering* de documentos para analizar sus métodos y determinar cuál se adapta mejor a la clasificación de documentos provenientes de sitios web y modelar un proceso determinando y caracterizando sus distintas fases que permita combinar modelos de recuperación de la información con enfoques de técnicas de *clustering*.

De esta forma se aborda un tema de gran interés para el usuario de las tecnologías de la información y la comunicación como es la mejora en la localización de contenidos ante la creciente avalancha de datos y su temporalidad. Por otra parte se busca brindar a los portales web nuevas formas de poder presentar la información que complementen a las ya existentes.

Tabla de contenidos

Tabla de contenidos

Tabla de contenidos

Tabla de contenidos

Lista De Figuras

Lista De Figuras

Lista De Tablas

1 INTRODUCCIÓN A LA RECUPERACIÓN DE LA INFORMACIÓN

La expresión, “Recuperación de la Información” (*Information Retrieval*), hace referencia al tratamiento automatizado que se lleva a cabo para poder dar respuesta a una necesidad de información. Se trata de una terminología aceptada y extendida [Rigsbergen, 1979] [Baeza-Yates y Ribeiro-Neto, 1999], que engloba por una parte aspectos relacionados con la representación, almacenamiento y organización de la información y por otra parte aspectos relacionados con la eficiencia en la presentación de resultados como consecuencia de una consulta.

Nos estamos refiriendo a sistemas de recuperación de información informatizados, también conocidos como motores de búsqueda, sobre todo en el campo de la Web.

Un aspecto importante a tener en cuenta es que no se trata de proporcionarle datos al usuario sin más, sino de proporcionarle información que le sea relevante. No vale recuperar todo, sino recuperar información válida y en la medida de lo posible clasificada o ponderada en cuanto a su grado de utilidad.

A diferencia de los sistemas de recuperación de datos, típicos de las bases de datos, en los que se obtienen todos los documentos que contienen unas palabras clave especificadas, los sistemas de recuperación de información deben poder determinar cuáles de esos documentos son los relevantes para la consulta especificada por el usuario, cuáles de ellos son los más importantes e incluso descartar los que, aun conteniendo las palabras clave, no satisfacen sus necesidades.

Es necesaria una interpretación de los documentos, para ello no basta con extraer información sintáctica y semántica para poder aportar al usuario la respuesta que necesita, sino que se necesita establecer mecanismos que permitan determinar el grado de relevancia de cada uno de los documentos que se obtienen.

El mecanismo habitual que tiene un usuario para comunicar su necesidad de información a un sistema de recuperación de información es la consulta en el lenguaje que maneje el sistema, también se habla de expresiones de consulta. Tenemos que distinguir la forma en que el usuario busca la información, podría establecer una consulta claramente definida cuando conoce el tipo de resultados que tiene que obtener, o podría navegar en los documentos cambiando entre unos y otros según fuera

interpretando lo que encontrara. Se trata, pues, de dos tareas diferenciadas, el primer caso sería una recuperación de datos o de información y en el segundo caso se trataría de una navegación. Los sistemas clásicos de recuperación de la información serían un ejemplo del primer tipo de tarea, mientras que el hipertexto sería un claro ejemplo de navegación. Entre uno y otro podemos encontrar bastantes variantes de modelos. Las primeras clasificaciones aceptadas datan de finales de los años 80 [Belkin y Croft, 1987], y distinguen por una parte sistemas que buscan la coincidencia total entre consulta y documentos recuperados, por otra parte modelos de coincidencia parcial (lógico, vectorial, probabilístico, lógica difusa, etc.), y un tercer grupo de modelos en red (*cluster*, *browsing*).

1.1 Operaciones con los documentos

Una labor habitual que se ha venido haciendo con los documentos para poder localizarlos y acceder rápidamente a ellos ha sido la de utilizar índices. Se trata de determinar una serie de palabras claves o términos índice que son característicos del documento y que van a permitir recuperarlo. Esta tarea de seleccionar estos términos se puede hacer de forma automática, mediante herramientas informáticas, o bien de forma manual mediante un experto en el campo del documento. Este proceso puede llevar bastante tiempo e incluso derivar en inconsistencias [Hooper, 1965] [Stubbs et al., 2001].

El caso extremo sería utilizar todas las palabras del documento como términos índice, cosa posible si contamos con equipos informáticos que puedan soportarlo. No obstante, esto puede ser una carga de trabajo considerable, si las colecciones que se manejan son muy extensas. Se busca, por tanto, reducir el conjunto de índices, para ello se utiliza la eliminación de palabras vacías (como pueden ser artículos o conjunciones), la reducción de palabras a su raíz o la identificación de grupos de nombres. Otro problema añadido es la sinonimia y polisemia propia del lenguaje natural, un mismo concepto se puede representar por varias palabras y una misma palabra puede tener

distintos significados. Por todo ello, se suele buscar una solución intermedia entre indexar por un reducido número de palabras dadas por un especialista, o todas las palabras del documento.

1.2 El proceso de recuperación

El proceso de recuperación de la información podría incluirse como una parte de las operaciones que pueden realizarse sobre libros y documentos y que ya se detallaban en el *Tratado de Documentación* de Otlet [Otlet, 1934]. Entre estas operaciones unas eran estrictamente documentológicas, otras eran de uso y otras documentales, que son las que nos interesan.

Chaumier, habla de *cadena documental*, refiriéndose a las operaciones sucesivas y conectadas que intervienen en los sistemas documentales [Chaumier, 1979]. Considera tres fases: colecta o entrada en el sistema (adquisición, selección, registro), tratamiento (análisis y recuperación) y difusión (orientada a satisfacer las necesidades de búsqueda de los usuarios). No todos los autores se ponen de acuerdo en cuanto a las divisiones, y sobre todo a las subdivisiones, porque el límite entre unas fases y otras apenas existe. En lo que sí coinciden es en la utilización del término *proceso documental*, aunque mientras que Courrier lo entiende como un proceso de circulación de la información entre documentos y usuarios [Courrier, 1976], López Yepes lo considera como un procedimiento que posibilite la dinamización de la información [López Yepes y Sagredo, 1981].

Conceptualmente las fases anteriores se corresponden con las fases que se detallan a continuación, pero revisadas y adaptadas.

1.2.1 Definición de la base de datos de textos

El primer paso que hay que dar en el proceso de recuperación es definir la base de datos de textos. Suele hacerlo el administrador de la base de datos y para ello debe especificar: los documentos que se van a utilizar, las operaciones que se van a poder llevar a cabo sobre los documentos, y el modelo o estructura de los textos. Una vez hecho esto tendremos una vista lógica de los documentos, con lo que podremos construir el índice del texto.

1.2.2 Análisis documental

Son muchas las definiciones que se han dado sobre análisis documental. Algunas de ellas son:

- Como término genérico, "es la determinación exacta de los elementos o componentes de un complejo cualquiera", mientras que refiriéndose a análisis de contenido, "la investigación técnica con el fin de la descripción objetiva, sistemática y cuantitativa del contenido evidente de una comunicación [FID, 1958].
- "Toda operación o conjunto de operaciones enfocadas a representar un documento dado bajo una forma diferente de la original, bien se trate de traducirlo, resumirlo, indexarlo…, para facilitar la consulta o la recuperación por los especialistas interesados" [Gardin et al., 1964].
- "Todo reconocimiento y estudio que se hace de un documento" [García Gutiérrez, 1984].
- "Un conjunto de procedimientos efectuados con el fin de expresar el contenido de los documentos, bajo formas destinadas a facilitar la recuperación de la información" [Cunha, 1987].

Un estudio pormenorizado de las fuentes y de los aspectos teóricos del análisis documental puede consultarse en [Pinto Molina, 1993].

A nivel práctico, el análisis documental consiste en extraer de un documento los elementos que sirvan para una representación condensada del mismo. Su objetivo es identificar el documento mediante el tipo de índice utilizado, de forma que permite su rápida localización y recuperación.

Podemos hablar de análisis manual o tradicional y de análisis automatizado o no tradicional, dependiendo del tipo se pueden distinguir diferentes fases. En el sistema tradicional podemos distinguir catalogación (descripción bibliográfica o análisis formal), indización, clasificación y resumen, mientras que en el no tradicional tendríamos indización automática, extracto automático y evaluación de la concordancia (*matching*).

El índice nos permitirá manejar rápidamente grandes volúmenes de información. La estructura que puede tomar el índice puede ser muy variada. Algunos tipos de indización son los siguientes:

- Clasificación sistemática. Basada en una estructura en forma de árbol que se ha establecido en las materias. La primera clasificación de este tipo fue la DDC (*Dewey Decimal Clasificación*) que más tarde dio lugar a la más conocida UDC (o CDU, Clasificación Decimal Universal). Se establecen 10 grupos de materias, que se van subdividiendo en varios subniveles. Un documento se clasificará dándole una serie de dígitos que especificarán los distintos niveles en los que se encuadre. Presenta la ventaja de que es independiente del idioma y del alfabeto.
- Palabras clave o términos índice. Es un sistema ideado por Mortimer Taube. Se describe un documento seleccionando un conjunto de palabras significativas, en lenguaje natural del documento. Cada clave recibe el nombre de *unitérmino* [Taube, 1955]. Sus principales inconvenientes son el exceso de falsas combinaciones y la abundancia de palabras polisémicas, homonímicas, sinónimas, ambiguas y vacías.
- Descriptores. Fue iniciada por Calvin N. Mooers (1941). En este caso los términos que se utilizan pertenecen a un vocabulario documental establecido y sistematizado. Los términos utilizados pueden ser simples o compuestos de varias palabras.

Una descripción más amplia de estos tipos de indización y de otros sistemas puede consultarse en [Gimeno Perelló, 1995].

Una alternativa que se va a plantear será si utilizar un lenguaje controlado o bien indexar por todas las palabras del documento. En el caso de optar por todas las palabras del documento suele elaborarse una lista de palabras vacías, que no aportan información relevante sobre el documento, como pueden ser artículos, adverbios, pronombres, preposiciones, conjunciones o exclamaciones. Esta lista de palabras vacías servirá para indexar un documento por el resto de palabras que no figuran en la lista.

Si por el contrario se decide utilizar un lenguaje controlado, se utilizarán glosarios, que nos ayudarán a elegir un término entre los distintos sinónimos. Los términos del glosario no guardan relación semántica entre ellos por lo que pueden aparecer en orden alfabético para su rápida localización.

Otro elemento que supone un paso adelante con respecto al glosario, es el tesauro. Según la norma ISO 2788 / TC 46, los tesauros se pueden definir según su función y según su estructura:

- Por su función, son un instrumento de control terminológico utilizado para transponer a un lenguaje más estricto el idioma natural empleado en los documentos y por los indicadores.
- Por su estructura, se trata de un vocabulario controlado y dinámico de términos que tienen entre ellos relaciones semánticas y genéricas y que se aplican a un dominio particular del conocimiento.

Los términos preferentes del tesauro se denominan descriptores. El tesauro contendrá sólo una de las formas del concepto. Reflejará sus relaciones jerárquicas o asociativas con otros descriptores, mediante una notación estandarizada.

Como ventaja el tesauro permite moverse en la jerarquía establecida, accediendo a conceptos relacionados, pero ello conlleva un proceso informático más complejo.

1.2.3 Localización de documentos

Una vez construida la base de datos e indexada corresponderá al usuario plantear sus necesidades mediante sus consultas. Para establecer las consultas será necesario plantearlas de acuerdo a las operaciones establecidas para la base de datos.

Por último la consulta será procesada sobre la base de datos, dando lugar a unos resultados consistentes en unos documentos, que deberán ser ordenados según su relevancia antes de ser presentados al usuario.

Si los resultados obtenidos no son satisfactorios puede iniciarse un proceso de realimentación en el que el usuario reformula la consulta.

2 EVOLUCIÓN HISTÓRICA

Desde épocas remotas el hombre ha organizado la información para poder acceder más cómodamente a ella. Un caso cotidiano son las tablas de contenidos o los índices de los libros. Podemos considerar los índices como el elemento precursor de los sistemas de recuperación de información.

Subiendo de nivel, las bibliotecas y sus sistemas de catalogación son otro hito importante. Sus primeros sistemas basados en tarjetas dieron paso a catálogos automatizados pero con limitadas prestaciones como búsquedas por autor o por título. Con el tiempo fueron perfeccionándose y añadiendo prestaciones como búsquedas por materia, por editoriales o combinaciones de términos.

A finales de los años 50 se aglutinan varios factores que dan un fuerte impulso a los inicios de la recuperación de la información. Aumenta considerablemente la producción científica de documentos, procedentes de áreas muy distintas, unido al desarrollo de la informática y la utilización de ordenadores, hace que se desarrollen sistemas mucho más operativos y rápidos.

Surge la idea [Luhn, 1957] de diseñar los sistemas de recuperación de textos comparando los identificadores de contenido del texto con las consultas. También introduce la utilización de la frecuencia de aparición de los términos en un documento para determinar si son representativos del documento.

En los años posteriores, 60 y 70, se tiene en cuenta la relación entre términos que aparecen juntos, por autores como Kuhns, Maron, Robertson, Slites o Spark Jones, apareciendo recopilados en algunos trabajos posteriores [Rijsbergen, 1979] [Chen, 1995]. Más recientemente se ha retomado esta idea, desarrollando algoritmos que tienen en cuenta la frecuencia en la secuencia de las palabras [Li et al., 2008].

En los años 80 el principal objetivo de la recuperación de la información era indexar textos de forma que se pudieran realizar búsquedas en colecciones de documentos. En estos años se hace común organizar colecciones de documentos en bases de datos para su tratamiento mediante el ordenador. Entre 1980 y 1989 el número de bases de datos creció de unas 600 a 4000, según [Cuadra-Elsevier, 1990]. Las bases de datos son el instrumento por naturaleza de la búsqueda informativa operada por medio de las tecnologías *ad hoc* [López Yepes, 1998]. Pero, aunque se va popularizando la realización de búsquedas en línea en las bases de datos, la complejidad

de los lenguajes de consulta, junto con la poca frecuencia y continuidad en las consultas, hace que sea necesaria, aún, la asistencia de personal intermediario cualificado.

Surgen de forma paralela otras técnicas basadas en el conocimiento que conducen a los sistemas expertos.

Al principio de los 90 la aparición de la Web trae consigo la aparición de los elementos multimedia y del hipertexto [Canals, 1990], [Woodhead, 1991], [Caridad y Moscoso, 1991], [Nielsen, 1995], [Díaz, et al., 1996], lo que provoca que se desarrollen nuevas herramientas que permitan recuperar información de este tipo. Pero también trae consigo una difusión vertiginosa de la información, lo que provoca una multiplicación de los documentos con los que hay que trabajar, haciendo más difíciles las tareas de recuperación [Salton, 1989].

En nuestros días la investigación en el terreno de la recuperación de la información ha ampliado sus horizontes y ha incluido modelado y clasificación de documentos, interfaces de usuario, visualización de datos, filtrado, lenguajes, etc. [Frakes y Baeza-Yates, 1992], [Kowalski, 1997], [Grossman y Frieder, 1998], [Baeza-Yates y Ribeiro-Neto, 1999].

3 CLASIFICACIÓN DE LOS MODELOS DE RECUPERACIÓN DE LA INFORMACIÓN

Cuando recuperamos un conjunto de documentos como respuesta a una consulta deberíamos discernir cuáles son relevantes para el usuario y cuáles podríamos descartar. Para tomar esta decisión deberemos utilizar algún algoritmo que permita establecer un orden en los documentos recuperados, de forma que aparezcan primero los que van a ser más importantes para el usuario. Este proceso de ordenación va a ser el elemento principal de la recuperación de la información. Son muchos los estudios que se han llevado a cabo sobre los algoritmos, comparándolos y viendo las ventajas de uno u otro dependiendo de su aplicación a casos concretos [Liu y Yu, 2005] [vonLuxburg, 2007] [Zimmermann y De Raedt, 2009].

Los algoritmos de clasificación van a operar de acuerdo a un conjunto de premisas, que en muchos casos van a ser medibles o ponderables, que darán lugar a los distintos modelos de recuperación de la información.

A la hora de clasificar los modelos de recuperación de la información habría que distinguir dos tipos de tareas por parte del usuario: recuperación: *ad hoc*[1] (o bien por filtrado) y navegación. Dentro de los sistemas de recuperación *ad hoc* o filtrado podríamos hacer dos grupos: los modelos clásicos y los modelos estructurados. En los modelos clásicos estarían el *booleano*, el vectorial y el probabilístico. En los modelos de recuperación de texto estructurado tendríamos las listas no solapadas y el modelo de los nodos cercanos.

En los sistemas de navegación tendríamos los modelos de texto plano, los de estructura dirigida y el hipertexto.

Para cada uno de los modelos clásicos a lo largo de los años se han ido proponiendo alternativas. Para los modelos *booleanos* tendríamos el modelo *booleano* extendido y el difuso. Para los modelos algebraicos la alternativa serían el vector generalizado, el índice semántico latente y las redes neuronales. Por último para los modelos probabilísticos tendríamos las redes de inferencia y las redes de creencias. Todos ellos vienen descritos en [Baeza-Yates y Ribeiro-Neto, 1999].

La figura 1 muestra en modo de resumen esta clasificación.

[1] Literalmente *para esto*. Se refiere a soluciones elaboradas de forma específica para un problema difícilmente generalizables.

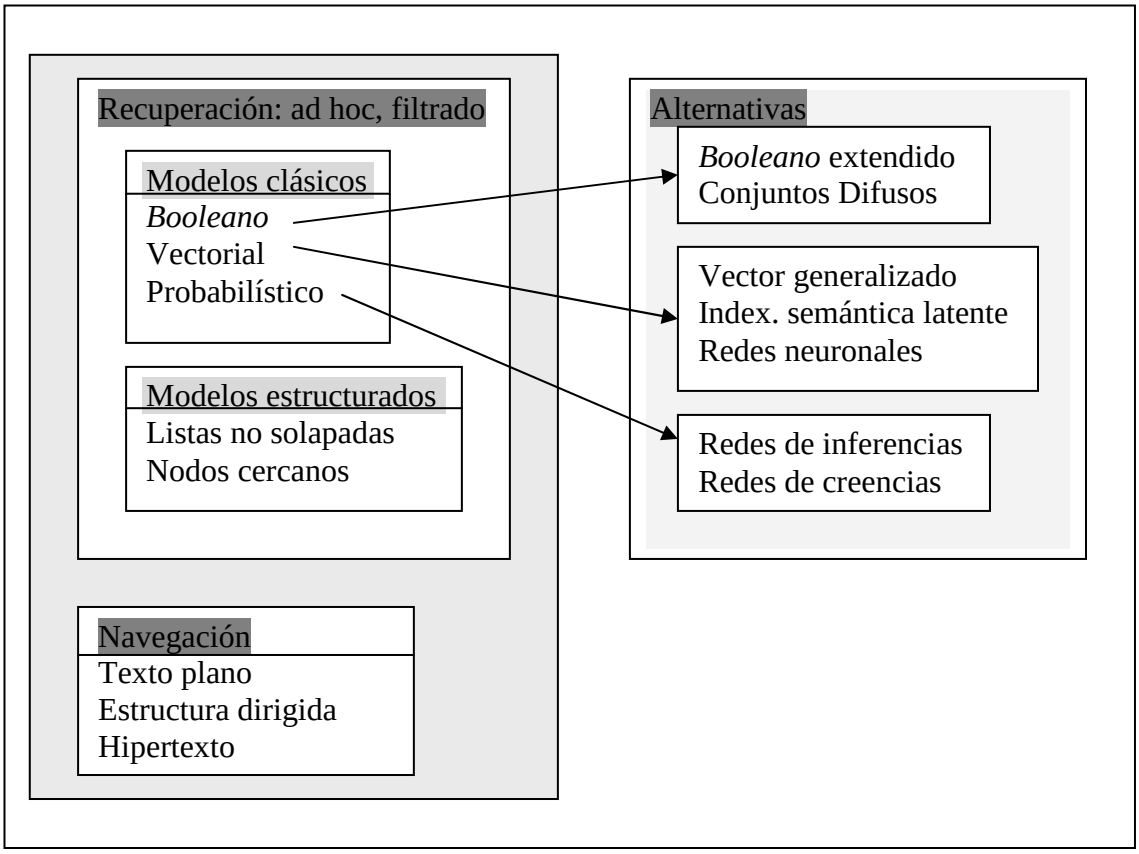

Figura 1. Clasificación de los modelos de recuperación de la información.

3.1 Modelos clásicos

Los modelos clásicos consideran que cada documento está descrito por un conjunto de palabras clave, denominadas términos índice. Estas palabras tienen la característica semántica de poder ayudar a recordar los temas principales de un documento. Suelen ser nombres porque tienen significado por sí mismos y su semántica es más sencilla de identificar y captar, pero no todos ellos servirán de igual forma para describir el contenido del documento. Un factor a tener en cuenta será la frecuencia con que una palabra aparece en el documento, a mayor frecuencia en un conjunto de documentos, menor información aporta para clasificarlos.

Los distintos términos utilizados para indexar un documento tendrán distinta importancia para describir el contenido del documento, por lo que se establecerán unos

pesos numéricos, que cuantifican la importancia del término para describir el contenido semántico del documento.

Sea *t* el número de términos en el sistema y k_i un término genérico. $K = \{k_1,\ldots, kt\}$ el conjunto de todos los términos. Un peso $\omega_{i,j} > 0$ está asociado con cada término k_i de un documento d_j. Para cada término que no aparece en un documento su peso es $\omega_{i,j} = 0$. Con cada documento d_j está asociado un vector de términos $\vec{d}_j = (\omega_{1,j}, \omega_{2,j}, ..., \omega_{t,j})$. Además sea g_i una función que devuelve el peso asociado a un término k_i, en un vector t-dimensional, $g_i(\vec{d}_j) = \omega_{i,j}$.

Se considera, por simplificación, que los pesos entre sí son independientes, aunque en la realidad la ocurrencia de una palabra no es independiente del resto. No obstante ninguna de las aproximaciones que se han ido proponiendo desde hace años ha conseguido demostrar que sea más ventajoso manejar la correlación entre términos, cosa por otra parte que dificulta enormemente su tratamiento.

3.1.1 Modelo booleano

Su planteamiento es sencillo, por lo que se ha utilizado hace algunos años en algunos sistemas de recuperación comerciales. Consiste en indexar el documento por uno o varios términos tomados de un glosario, de un tesauro, o simplemente de un conjunto de términos que no contiene la lista de palabras vacías. Su funcionamiento está basado en la teoría de conjuntos y en el álgebra de Boole.

Las consultas son planteadas como expresiones *booleanas*, con tres conectores: *not*, *and* y *or*, que pueden representarse como una disyunción de vectores conjuntivos (DNF *Disjunctive Normal Form*). Tienen una semántica muy precisa, pero esto hace que tengan la desventaja de no añadir ningún grado de ponderación en los términos. Se recuperan las referencias de todos los documentos que cumplan la expresión pero en el orden que el sistema los ha evaluado, sin aportar un posible grado de relevancia para el usuario.

Otra dificultad añadida es la elaboración de las expresiones de consultas, que, aunque pueden ser muy potentes y precisas, pueden presentar un alto grado de dificultad y ser complicadas para muchos de los usuarios finales.

Como el planteamiento del modelo es que un término índice esté o no en el modelo, el peso que se le va a asignar va a ser un valor binario, $\omega_{i,j}$ = {0, 1}. El resultado de aplicar una consulta va a ser dos subconjuntos: el de los documentos que contienen el término por el que se está indexando y su complementario, el de los documentos que no lo contienen.

Formalmente, si una consulta *q* es una expresión *booleana*, que expresamos en forma normal *booleana* $\vec{q}_{dnf}$, siendo $\vec{q}_{cc}$ alguno de sus componentes conjuntivos. Se define la similitud, afinidad o semejanza de un documento d_j con la consulta *q* como:

$$sim(d_j, q) = \left\{ \begin{array}{l} 1 \ si \ \exists \ \vec{q}_{cc} \ | (\vec{q}_{cc} \in \vec{q}_{dnf}) \wedge (\forall k_i, g_i(\vec{d}_j) = g_i(\vec{q}_{cc})) \\ 0 \ en \ otro \ caso \end{array} \right\}$$

Cuando valga 1 se considera que el documento d_j es relevante para la consulta *q*.

3.1.2 Modelo vectorial

El modelo vectorial [Salton y Lesk, 1968] [Salton, 1971a] propone asignar pesos que permitan ponderar la importancia que cada término de indexación tiene en el documento. Estos pesos, positivos y no binarios, permitirán determinar el grado de afinidad de los documentos almacenados, con las consultas formuladas por los usuarios. Las referencias recuperadas por una consulta podrán ordenarse en función de su similitud con la consulta.

Tanto los documentos como las consultas se representan mediante vectores. Si $\omega_{i,q}$ es el peso asociado con el par [k_i, *q*], entonces el vector que representa la consulta

es $\vec{q} = (\omega_{1,q}, \omega_{2,q}, ..., \omega_{t,q})$, siendo *t* el número total de términos índice en el sistema, mientras que el vector que representa un documento es $\vec{d}_j = (\omega_{1,j}, \omega_{2,j}, ..., \omega_{t,j})$.

Para evaluar el grado de similitud entre un documento y una consulta de un usuario, el modelo propone medir el grado de correlación entre sus respectivos vectores mediante el coseno del ángulo que forman ambos vectores:

$$sim(d_j, q) = \frac{\vec{d}_j \bullet \vec{q}}{\left|\vec{d}_j\right| \times \left|\vec{q}\right|} = \frac{\sum_{i=1}^{t} \omega_{i,j} \times \omega_{i,q}}{\sqrt{\sum_{i=1}^{t} \omega_{i,j}^2} \times \sqrt{\sum_{j=1}^{t} \omega_{i,q}^2}}$$

Como los valores de los pesos de ambos vectores son mayores que 0, el valor que se obtiene de similitud está comprendido entre 0 y 1 y nos va a permitir fijar valores umbrales, por encima de los cuales consideraremos que los documentos tienen relevancia para el usuario.

La forma de calcular el peso de cada término en el vector del documento [Salton y McGill, 1983] puede ser muy variada [Salton y Buckkey, 1988] [Harman, 1992], pero en cualquier caso hay que tener en cuenta que a mayor frecuencia de aparición de un término en un documento mayor peso debe tener. No obstante hay que tener en cuenta también la frecuencia de aparición en la colección de documentos, porque si aparece frecuentemente en todos sus documentos no servirá como discriminante. También hay que tener en cuenta en la ponderación el tamaño de los documentos, si un término aparece igual número de veces en dos documentos de distinto tamaño, deberá tener mayor ponderación en el documento más corto.

Para evaluar la capacidad de un término de representar a un documento, la primera medida es la frecuencia de aparición del término en el documento (*tf term frecuency*) que es el número de veces que aparece en el documento. Pero es importante conocer el grado de discriminación de un término con respecto a los documentos de la colección, cuanto más aparezca en todos ellos menos útil es, por lo que se utiliza la inversa de la frecuencia de documentos (*idf inverse document frequency*).

Formalmente, si N es el número total de documentos en el sistema, n_i el número de documentos en los que aparece el término k_i, y $freq_{i,j}$ la frecuencia del término k_i en

el documento d_j, entonces se define la frecuencia normalizada $f_{i,j}$ del término k_i en el documento d_j como:

$$f_{i,j} = \frac{freq_{i,j}}{\max_l freq_{l,j}}$$

La inversa de la frecuencia del documento k_i, se define como:

$$idf_i = \log \frac{N}{n_i}$$

El esquema de asignación de pesos a los términos más conocido es dado por el producto simple [Harman, 1992]:

$$\omega_{i,j} = f_{i,j} \times \log \frac{N}{n_i}$$

aunque también se han realizado variaciones sobre esta fórmula. A este conjunto de estrategias de asignación de pesos a los términos se las denomina esquemas *tf-idf*.

Algunas variaciones de la asignación de pesos vienen descritas en [Salton y Buckley, 1988]. Para los pesos asignados a los resultados de las consultas proponen:

$$\omega_{i,q} = (0.5 + \frac{0.5\, freq_{i,q}}{\max_l freq_{l,q}}) \times \log \frac{N}{n_i}$$

donde $freq_{i,q}$ es la frecuencia del término k_i en el texto de la consulta q.

Para finalizar, se pueden destacar como ventajas:

- La utilización de esquemas que asignan pesos a los términos mejora el rendimiento en la recuperación.
- La estrategia que permite utilizar coincidencias parciales permite recuperar documentos que se aproximan a los solicitados por el usuario y que podrían servirle.
- Permite ordenar los documentos de acuerdo al grado de similitud con la consulta.

3.1.3 Modelo probabilístico

Este modelo [Robertson y Sparck Jones, 1976] también se conoce como modelo de recuperación de independencia binaria (BIR *Binary Independence Retrieval*). La idea es que, dada una consulta de un usuario, debe existir un conjunto de documentos relevantes y no otros.

Si diéramos la descripción de la solución ideal no deberíamos tener dificultades para obtener los documentos deseados, el problema es que no conocemos cuáles son estas propiedades. Lo único que sabemos es que podemos utilizar unos términos para indexar el documento que semánticamente deberían caracterizar estas propiedades. Partiendo de esta conjetura inicial podemos generar una descripción probabilística preliminar del conjunto solución ideal, para obtener un primer conjunto de documentos. Mediante un proceso interactivo con el usuario, que irá indicando cuáles de los documentos recuperados son realmente relevantes y cuáles no, se irá refinando la descripción del conjunto solución ideal.

El modelo trata de estimar la probabilidad de que el usuario encuentre el documento d_j relevante. Esta probabilidad dependerá sólo de la consulta y de la representación del documento. Se asume que hay un subconjunto de documentos *R* que el usuario prefiere como solución para la consulta dada *q*, y por tanto otro subconjunto contendrá los no relevantes.

Los términos índice tanto de los documentos como de las consultas son valores binarios {0, 1}. Siendo *R* el conjunto de documentos inicialmente propuestos para ser relevantes, y $\overline{R}$ su complementario (los nos relevantes), se define la similitud de un documento d_j con una consulta *q* como:

$$sim(d_j, q) = \frac{P(R \mid \vec{d}_j)}{P(\overline{R} \mid \vec{d}_j)}$$

siendo $P(R \mid \vec{d}_j)$ la probabilidad de que el documento d_j sea relevante en la consulta *q*.

Utilizando las reglas de Bayes:

$$sim(d_j, q) = \frac{P(\vec{d}_j \mid R) \times P(R)}{P(\vec{d}_j \mid \overline{R}) \times P(\overline{R})}$$

$P(\vec{d}_j \mid R)$ representa la probabilidad de que aleatoriamente elijamos el documento d_j del conjunto R de los documentos relevantes, mientras que $P(R)$ es la probabilidad de que elijamos aleatoriamente un documento del conjunto R. Como $P(R)$ y $P(\overline{R})$ toman el mismo valor en todos los documentos de la colección, podemos simplificar la expresión anterior:

$$sim(d_j, q) \approx \frac{P(\vec{d}_j \mid R)}{P(\vec{d}_j \mid \overline{R})}$$

La ventaja de este modelo es que los documentos recuperados se obtienen en orden decreciente de su probabilidad de ser relevantes, pero sus desventajas son: la necesidad de partir de subconjuntos iniciales de supuestos documentos relevantes y no relevantes, el no considerar la frecuencia con que cada término aparece en los documentos y el considerar que los términos son independientes, aunque como ya se ha comentado para el modelo vectorial, no está claro que en la práctica se obtengan mejores resultados considerando esta dependencia.

3.2 Alternativas a los modelos clásicos

3.2.1 Alternativas a los modelos de teoría de conjuntos

3.2.1.1 Modelo de conjuntos difusos

Mientras que el álgebra de Boole trabaja con conjuntos deterministas, en los que los elementos pertenecen a un conjunto o están excluidos de él, la lógica difusa trata con clases que no son excluyentes, sus límites se solapan, por lo que los elementos pueden pertenecer a diferentes clases con distintos grados de pertenencia.

La teoría de conjuntos difusos [Zadeh, 1993] trata de la representación de clases cuyos límites no están bien definidos. El elemento fundamental es la función de pertenencia que permite asociar los elementos con las clases. La función tomará valores en el intervalo [0, 1], 0 indicará que un elemento no pertenece a una clase y 1 indicará que pertenece completamente a la clase, el resto de valores intermedios darán idea del grado de pertenencia a la clase.

Las operaciones más habituales con estos conjuntos son el complemento, la unión y la intersección. Si *A* y *B* son subconjuntos difusos de un universo del discurso *U* y $\mu_A : U \to [0,1]$ la función que asocia cada elemento *u* con un valor en el intervalo *[0, 1]*, las operaciones anteriores se pueden definir como:

$$\mu_{\bar{A}}(u) = 1 - \mu_A(u)$$

$$\mu_{A \cup B}(u) = \max(\mu_A(u), \mu_B(u))$$

$$\mu_{A \cap B}(u) = \min(\mu_A(u), \mu_B(u))$$

Existen varios modelos de recuperación basados en la teoría de conjuntos difusos que se han ido proponiendo a lo largo de los años. Uno de los más conocidos [Ogawa et al., 1991] se basa en la idea de expandir el conjunto de términos índice de la consulta con los términos relacionados obtenidos de un tesauro como si se tratara de documentos relevantes adicionales. El tesauro se construye mediante una matriz de correlación término-término en la que se puede incluir la relación entre cada dos términos. Con esta matriz podemos definir el conjunto difuso asociado con cada término.

El usuario realiza las consultas mediante expresiones *booleanas* en forma normal disyuntiva, como en el modelo *booleano* clásico, y también de igual forma se obtienen los elementos relevantes de la solución, pero teniendo en cuenta que tratamos con conjuntos difusos. Además el modelo utiliza sumas y productos, en lugar de máximos y mínimos, para calcular el grado global de pertenencia de un documento en el conjunto difuso definido por la consulta del usuario.

3.2.1.2 Modelo *booleano* extendido

Este modelo [Salton et al., 1983] utiliza pesos, en el rango [0,1], para cada término índice para indicar el grado en que el documento se caracteriza por ese término. También se utilizan pesos en los términos utilizados en las preguntas. La solución se obtiene mediante la semejanza de los términos de los documentos de la colección y los términos de la consulta.

Surge como crítica al modelo *booleano,* ya que considera que la utilización de consultas como expresiones en forma normal conjuntiva (o daría igual disyuntiva) hace que se descarten documentos que tienen relación con alguno de los términos de la consulta, pero no con todos ellos, lo que no es demasiado lógico.

3.2.2 Alternativas a los modelos algebraicos

3.2.2.1 Modelo espacio vector generalizado

A menudo la independencia de los términos índice se ha interpretado en el sentido más restrictivo como que los términos de los vectores son ortogonales, es decir que $\vec{k}_i \bullet \vec{k}_j = 0$. Este modelo [Wong et al., 1985] propone que los pesos son considerados independientes pero no ortogonales.

Dado el conjunto {k_1, k_2,…, k_t} de términos índice en una colección, y siendo $\omega_{i,j}$ el peso asociado con el par documento-término [k_i, dj]. Si los pesos son binarios todas las posibles co-ocurrencias de los términos pueden ser representadas por un conjunto de 2^t términos mínimos dados por m_1= (0,0,…,0), m_2= (1,0,…,0),…, m_2^t= (1,1,…,1). La función g_i (m_j) devuelve el peso {0,1} del término k_i en el término mínimo m_j.

La idea es introducir un conjunto de vectores ortogonales asociados con el conjunto de términos mínimos y adoptar este conjunto como base del subespacio de interés.

3.2.2.2 Modelo de indexación por la semántica latente (LSI)

Como se ha visto con modelos anteriores el utilizar términos índice para resumir los documentos y las consultas puede llevar a que los resultados obtenidos sean pobres, principalmente por dos motivos:

- se pueden obtener como solución muchos documentos inconexos.
- documentos relevantes para el usuario puede que no estén indexados por los términos utilizados en la búsqueda.

Al utilizar términos para indexar estamos perdiendo la esencia del documento, ya que las ideas están más relacionadas con los conceptos del documento que con los términos de indexación. Se trataría de emparejar documentos mediante conceptos, relacionaríamos un documento con una consulta cuando compartieran conceptos, es decir cuando tuviera una semántica relacionada, pudiendo por tanto prescindir de los términos índice.

Este modelo está considerado como una extensión del modelo vectorial. La idea original de este modelo [Furnas et al., 1988] es asociar cada documento con un vector consulta en el menor espacio dimensional con el que se puedan recoger los conceptos. Se intenta que la recuperación en este espacio reducido sea superior que la que se conseguía con los términos índice.

La forma de llevar a cabo la indexación por la semántica latente (LSI) [Deerwester et al., 1990] es utilizar la descomposición singular de matrices (SVD *Singular Value Decomposition*). Esta técnica consiste en poder descomponer cualquier matriz X (t x d) en el producto de: una matriz ortogonal de columnas T_0 (t x m), una matriz diagonal S_0 (m x m) con elementos positivos o cero, que son los valores singulares de X, y una matriz transpuesta de una matriz D_0 (m x d).

$$X = T_0 S_0 D_0'$$

El modelo LSI busca la reducción de ruido en los documentos encontrados. Pueden recuperarse, y por tanto descartar, documentos que tengan algún término de la consulta pero que sin embargo no sean relevantes por su significado. Esto se consigue, ya que los documentos de similar semántica se localizan en posiciones cercanas en un espacio multidimensional dado por el sistema de matrices.

Se parte de la matriz de asociación de términos con documentos *X*, que tiene *t* filas que representan los términos y *d* columnas que representan los documentos y cada posición $X_{i,\ j}$ tiene la frecuencia del término *i* en el documento *j*. Esta matriz se descompone mediante la técnica SVD, obteniéndose las tres matrices, a las que posteriormente se reducen sus dimensiones para eliminar el ruido.

3.2.2.3 Modelos de redes neuronales

Los modelos de redes neuronales se han utilizado en la recuperación de la información por sus buenos resultados en la comparación de patrones, ya que es necesario comparar los valores de las consultas con los de los documentos almacenados y ordenar los resultados.

Uno de los modelos de redes neuronales [Wilkinson y Kingston, 1991] propone una red de tres capas: una capa para los términos de los documentos, otra para los términos de las consultas y una tercera para los documentos en sí mismos.

El proceso de inferencia lo inician los nodos correspondientes a los términos de las consultas enviando señales a los nodos de los términos de los documentos (nivel 2), que propagan las señales para que lleguen a los nodos de los documentos. De esta forma se completa una primera fase, que no termina ya que los nodos finales de los documentos inician una propagación en sentido contrario, hacia los nodos de los términos de los documentos, pudiéndose repetir el proceso de propagación y realimentación entre las capas 2 y 3, hasta que las señales en el proceso de propagación sean débiles, momento en el que se detendrá. Se puede observar este proceso en la figura 2.

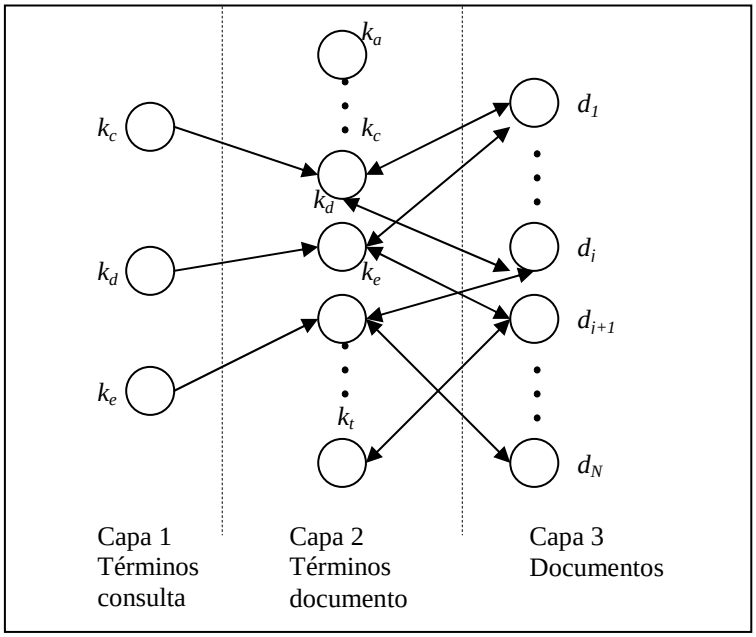

Figura 2. Red neuronal para Recuperación de la Información.

A los nodos iniciales de las consultas se les asigna un nivel de activación de 1, que será el valor máximo que se irá atenuando mediante los nodos del resto de capas.

Para mejorar los resultados de la recuperación se hace que la red continúe con el proceso de propagación después del primer ciclo, como si se tratara de un ciclo de realimentación, pudiendo fijar un umbral mínimo de activación por debajo del cual los nodos de los documentos finales no enviarían mas señales.

3.2.3 Alternativas a los modelos probabilísticos

Las principales alternativas a los modelos probabilísticos han sido las distintas variantes de las redes *Bayesianas* [Pearl, 1988]. Una red *bayesiana* es un grafo cuyos nodos representan variables y los arcos describen la relación causal entre las variables. Como se puede ver en la figura 3. El grado de las relaciones causales es expresado mediante probabilidades condicionales.

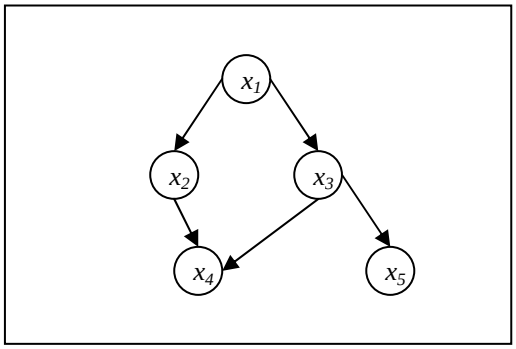

Figura 3. Red Bayesiana.

Si x_i es un nodo en una red *G*, y Γ_{x_i} es el conjunto de sus nodos padres, la influencia de Γ_{x_i} en x_i puede especificarse por un conjunto de funciones $Fi(x_i, \Gamma_{x_i})$, que satisfacen:

$$\sum_{\forall x_i} Fi(x_i, \Gamma_{x_i}) = 1$$

$$0 \leq Fi(x_i, \Gamma_{x_i}) \leq 1$$

La distribución de probabilidad conjunta de la red de la figura 3 sería:

$P(x_1,x_2,x_3,x_4,x_5) = P(x_1)P(x_2|x_1)P(x_3|x_1)P(x_4|x_2,x_3)P(x_5|x_3)$

3.2.3.1 Modelo de red de inferencia

El modelo de redes de inferencia [Turtle y Croft, 1990, 1991] adopta una visión epistemológica del problema de la recuperación de la información. Asocia variables aleatorias con los términos índice, con las consultas y con los documentos. Una variable aleatoria asociada con un documento, significa que ese documento está siendo observado en la búsqueda de documentos relevantes. La observación de ese documento asegura la confianza en las variables asociadas a sus términos índice.

Tanto los términos índice como los documentos son representados mediante nodos de la red. Los arcos están dirigidos desde los nodos de los documentos hasta sus respectivos nodos de términos índice para indicar que la observación del documento aumenta la confianza en sus términos índice asociados. Se puede ver en la figura 4.

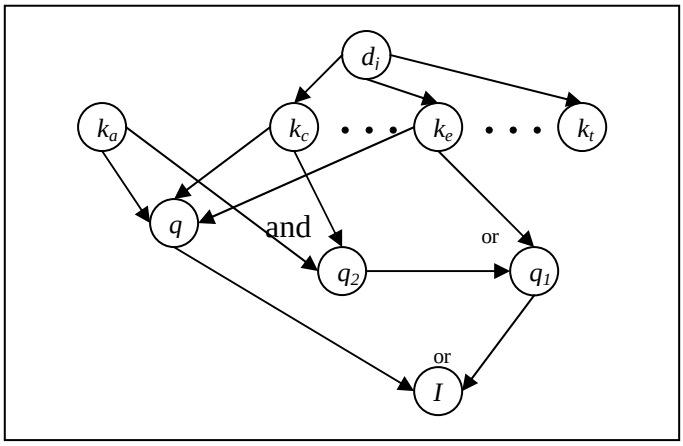

Figura 4. Modelo de red de inferencia.

Las variables aleatorias asociadas con las consultas de los usuarios también han sido representadas mediante nodos en la red. La confianza en estos nodos es una función de las confianzas en los nodos asociados con los términos de la consulta. Los arcos parten de los nodos de los términos índice de los documentos hacia los nodos de las consultas. En la figura 4, podemos observar que la consulta *q* está compuesta por los términos k_a, k_c, y k_e, ya que hay arcos que llegan hasta ella desde esos nodos. También hay otros nodos consulta *q*2 y *q*1, para representar alternativas a la consulta original, *q*1 representaría $q1 = (k_a \wedge k_c) \vee k_e$

3.2.3.2 Modelo de red de creencias (*Belief Network Model*)

El modelo de red de creencias [Ribeiro-Neto et al., 1996] también está basado en un interpretación epistemológica de las probabilidades, pero adoptando un espacio definido simple. Como resultado persigue una topología de red ligeramente diferente que proporciona una separación entre el documento y las partes de consulta de la red.

La consulta del usuario *q* se representa como un nodo de la red al que se le ha asociado una variable aleatoria binaria, a la que también haremos referencia como *q*. Esta variable será puesta a 1 cuando cubra completamente el espacio *K*. Cuando evaluamos *P* (*q*) calculamos el grado en que *q* completa el espacio *K*. Esto es

equivalente a evaluar el grado de creencia asociada con la siguiente proposición: ¿Es cierto que *q* cubra todos los posibles conceptos en *K*?

Un documento d_j se representa como un nodo de la red al que se le asocia una variable aleatoria binaria que también se denota por d_j, y vale 1 cuando cubre completamente el espacio *K*. Cuando evaluamos $P(d_j)$, calculamos el grado en que d_j cubre el espacio *K*. Equivalente a evaluar el grado de creencia de la proposición: ¿Es cierto que d_j cubre completamente todos los conceptos de *K*?

Las consultas de usuario y los documentos en la colección son modelados como subconjuntos del conjunto de términos índice. Cada uno de estos subconjuntos es interpretado como un concepto embebido en el concepto de espacio *K* que trabaja como un simple espacio común.

Una red de este tipo sería la de la figura 5. La diferencia con una red de inferencia, es que en la red de creencias los documentos y las consultas son tratados de igual forma y apuntados por sus respectivos términos índice, mientras que en las de inferencia los documentos no eran apuntados por nadie.

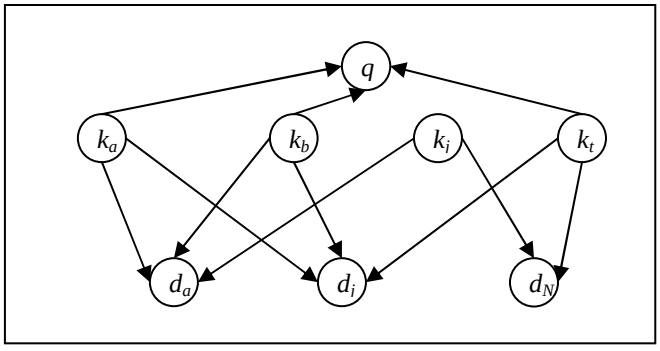

Figura 5. Modelo de red de creencias.

La clasificación de un documento recuperado con respecto a una consulta dada es interpretada como una relación de correspondencia que refleja el grado en que se cubre el documento por la consulta.

3.3 Modelos de texto estructurado

Se trata de modelos que combinan información contenida en el texto con información en la estructura del documento.

3.3.1 Listas no solapadas o no coincidentes

El modelo basado en listas no solapadas (*Non-Overlapping Lists*) [Burkowski, 1992] propone dividir el texto completo de cada documento en regiones o zonas de texto no solapadas y almacenarlas en una lista. Como hay muchas formas de obtener las regiones, se generarán muchas listas. Un ejemplo podría ser la lista de capítulos, la de secciones y la de subsecciones. Estas listas se guardarían por separado en distintas estructuras de datos. Si bien cada una de las listas no tiene zonas superpuestas, si comparáramos las distintas listas entre sí, veríamos que hay zonas que pueden solaparse.

Para permitir búsquedas por términos índice y por regiones de texto, se construye un fichero invertido en el que cada componente estructural soporta una entrada en el índice. Asociado con cada entrada hay una lista con las ocurrencias de esa entrada en las regiones de texto. Además, cada una de las listas podría fácilmente ser asociada con el tradicional fichero invertido con las palabras del texto.

3.3.2 Nodos cercanos o próximos

El modelo basada en nodos cercanos (*Proximal Nodes*) [Baeza-Yates y Navarro, 1996] [Navarro y Baeza-Yates, 1995, 1997] propone un modelo que permite la definición de estructuras índice jerárquicas independientes sobre el mismo texto de los documentos. Cada una de estas estructuras índice es una jerarquía compuesta de capítulos, secciones, párrafos, páginas y líneas, que son llamados nodos. A cada uno de

estos nodos se le asocia una región de texto. Además dos jerarquías distintas pueden hacer referencia a regiones de texto solapadas.

Para una consulta que hace referencia a distintas jerarquías, la solución que aporta está formada sólo por nodos de una de ellas, lo que hace que sea más rápido a costa de perder expresividad.

Además se mantienen listas invertidas cuyas entradas (términos) indican todas las posiciones en el texto del documento en las que aparece ese término. Estarían incluidas las posiciones en los capítulos, secciones, subsecciones, etc.

El lenguaje de consulta permite utilizar expresiones regulares, para realizar búsquedas por cadenas, pero también permite la búsqueda por nombre de componentes de la estructura, para buscar por ejemplo por capítulos, o la combinación de ambos tipos de búsqueda. Por tanto permite consultas más complejas que las que se pueden formular con el modelo de las listas no solapadas.

3.4 Modelos de navegación

Los modelos anteriores están planteados con la idea de que el usuario plantee una consulta concreta, pero hay ocasiones en las que el usuario lo que desea es investigar sobre algún tema, o simplemente buscar referencias. En este caso el usuario está navegando u hojeando (*browsing*), en lugar de realizar búsquedas. En ambos casos el usuario sabe qué pretende, cuáles son sus objetivos.

3.4.1 Texto plano

El usuario explora el espacio de un documento como si tuviera una organización plana. Los documentos podrían ser representados como puntos en un plano o bien como elementos en una lista. El usuario miraría los documentos y cuando se interesara por uno entraría dentro de él. Podría buscar palabras dentro del documento o visitar los documentos que están próximos al visitado. El proceso en conjunto se llama

realimentación con relevancia (*relevante feedback*), porque podríamos ir modificando las palabras empleadas en la consulta, según se fueran obteniendo resultados.

Una vez acceda a un documento podría recorrerlo secuencialmente o moverse mediante el teclado de arriba abajo, pero con el inconveniente de que se pierde el contexto en el que el documento se encuentra.

3.4.2 Estructura dirigida

Para organizar la navegación por los documentos se pueden agrupar en estructuras como pueden ser los directorios, que son estructuras de clase que permiten agrupar los documentos por tópicos relacionados. Las jerarquías de clase se han utilizado para clasificar documentos casi desde que éstos existen y seguirán siéndolo pero adaptadas a las nuevas tecnologías.

Podemos utilizar estructuras guiadas para movernos entre documentos y también para movernos dentro de un documento. Imaginemos por ejemplo un libro electrónico, podríamos movernos por él a través del índice pasando a los capítulos o a las secciones, o cualquier otra división que tuviera llegando en último término hasta el texto plano.

En estos casos es muy importante la herramienta de navegación, que nos permitirá movernos entre las distintas partes y volver a puntos anteriores o al índice. Estas herramientas pueden incluir mapas del sitio muy interesantes para tener una visión global, o ir marcando las zonas visitadas, o mostrar la propia estructura organizativa y el punto en el que nos encontramos.

3.4.3 Hipertexto

El hipertexto [Canals, 1990], [Woodhead, 1991], [Caridad y Moscoso, 1991], [Nielsen, 1995], [Díaz, et al., 1996], es una estructura de navegación altamente interactiva, que permite al usuario moverse por los documentos en pantalla sin necesidad de tener que recorrerlos secuencialmente. Consiste en una serie de nodos relacionados mediante enlaces dentro de una estructura gráfica. Accediendo a uno de estos enlaces nos movemos a otro punto del documento o a otro documento. A cada

nodo se le asocia una región de texto, que puede ser un capítulo en un libro, una sección en un artículo o una página Web.

Si tenemos dos nodos y un enlace entre ambos, en el primer nodo habrá un cadena de texto, que al ser seleccionada nos pasará al segundo nodo, por tanto podemos estar leyendo el primer nodo y pasar al segundo en el momento que queramos sin más que pulsar sobre el enlace. Normalmente estos enlaces aparecen en un color distinto al del texto habitual o bien subrayados y además cambian de color al ser utilizados.

La principal ventaja del hipertexto es que el usuario puede acceder a la información de acuerdo a sus necesidades. En contra tiene, el poderse perder entre los distintos enlaces, por eso es habitual el disponer de mapas de los documentos, que son gráficos que muestran los nodos que están siendo visitados y permiten centrarse al usuario. Además también le permite conocer las partes que ha recorrido.

Aunque el usuario tiene libertad para moverse, tiene que seguir unos caminos establecidos, aunque las posibilidades sean muchas. Por tanto el diseñador tendrá que tener en cuenta las posibles necesidades de los usuarios a la hora de realizar sus trabajos, es por tanto muy importante la labor de modelado del dominio, intentando que esté organizado de forma jerárquica de manera que se facilite la navegación.

El hipertexto ha sido la base de la concepción y el diseño de HTML (*HyperText Markup Language*) [Tittel et al., 1996], y del protocolo HTTP (*HyperText Transfer Protocol*) que dieron origen a la *World Wide Web*.

4 EVALUACIÓN DE LA RECUPERACIÓN

Los sistemas de recuperación de la información tienen que ser evaluados, como cualquier otro sistema informático, antes de ser finalizada su implementación [Baeza-Yates y Ribeiro-Neto, 1999]. Se debe comprobar que sus funcionalidades están conseguidas, se debe verificar que no se producen errores, probando cada una de sus funciones, y por último el rendimiento del sistema.

Los principales factores a medir en cuanto a rendimiento del sistema son el tiempo y el espacio. Tiempo que emplea en ofrecer una respuesta, y espacio que utiliza en el sistema. Otras métricas que también se utilizan en recuperación de la información son la precisión de los datos que se obtienen al procesar una consulta.

4.1 Medidas para la evaluación

El diccionario de la Real Academia Española define *relevancia* como “cualidad o condición de relevante, importancia, significación”. Aplicado a la recuperación de la información sería la medida en que uno o varios documentos recuperados se ajustan al objetivo de la búsqueda.

Se trata de un concepto muy importante, ya que depende totalmente del sujeto receptor de la información. Lo que para un individuo es relevante para otro puede carecer de todo interés, por tanto no aportar información. Se trata pues de un concepto subjetivo, luego difícilmente medible [Mizzaro, 98].

El término relevancia fue formulado en los años 30 y 40 por S. C. Bradford, que lo incluyó el término en algunos de sus artículos [Saracevic, 1975], si bien no empezó a utilizarse como medida de evaluación en recuperación de la información hasta los experimentos realizados por ASTIA (*Armed Services Technical Information Agency*), en los años 50, sobre la recuperación de documentos, a través de términos extraídos del

título y el resumen de conjuntos de documentos [Ellis, 1990]. Para una revisión del término se puede consultar [Gómez, 2003].

Los parámetros que se utilicen para evaluar cualquier sistema deberían ser cuantificables, porque ello conlleva que sean criterios más objetivos y además permiten la comparación entre distintos sistemas.

La evaluación de los sistemas en la mayoría de los casos está basada en la comparación entre los resultados que el sistema en evaluación produce al aplicarle una colección de documentos de referencia, cuando se le somete a unas consultas prefijadas, con unos resultados esperados facilitados por especialistas.

La comparación de los resultados dados por el sistema evaluado y los documentos relevantes dados por los expertos, su similitud, nos dará un grado de bondad del sistema que estamos evaluando.

Los dos parámetros más utilizados para la evaluación de la recuperación son la exhaustividad y la precisión, propuestos por [Cleverdon y Keen, 1966], [Cleverdon et al., 1966].

4.1.1 Exhaustividad (*recall*) y precisión

Exhaustividad hace referencia al grado en que se han encontrado todos los documentos de la colección que eran relevantes para la consulta dada, mientras precisión hace referencia al grado de aciertos, más fácilmente a los fallos que se han producido, dados por los documentos recuperados que no debían haberse obtenido.

Consideremos un ejemplo de consulta, recogido en la figura 6, sobre una colección de referencia, cuyo conjunto de documentos relevantes sería *R*. Sea por tanto |*R*| el número de documentos en el conjunto de los relevantes. Sea el conjunto de documentos *A*, la respuesta que ha dado el sistema que estamos evaluando, y |*A*| el número de documentos de este conjunto. Por último denotamos *Ra* al conjunto intersección de *R* y *A*, cuyo número de documentos es |*Ra*|.

Exhaustividad es el conjunto de documentos relevantes seleccionados con respecto al total de relevantes.

$$Exhaustividad = \frac{|Ra|}{|R|}$$

Precisión es el conjunto de documentos relevantes con respecto al total de los seleccionados.

$$\Pr ecisión = \frac{|Ra|}{|A|}$$

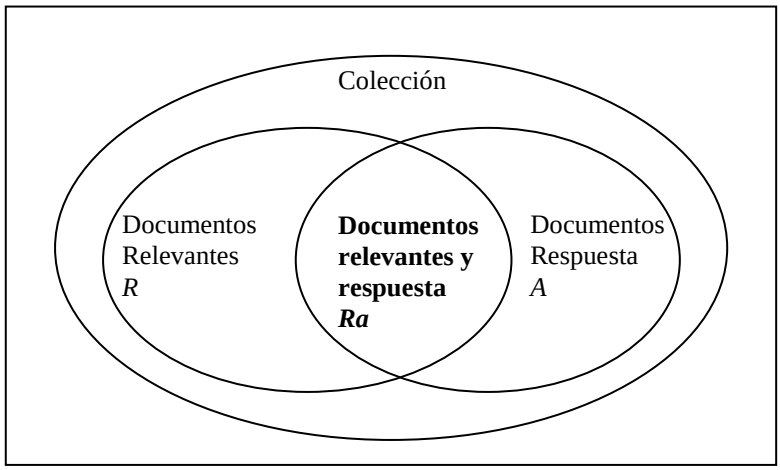

Figura 6. Exhaustividad y precisión en un ejemplo de solicitud de información.

Estos dos parámetros toman valores entre 0 y 1.

Estos parámetros definidos de esta forma serían valores teóricos porque en la práctica el usuario no obtendría todos los documentos recuperados a la vez, sino que iría mirándolos según se le presentaran en cuanto al grado de relevancia. A medida que los fuera viendo los valores de exhaustividad y precisión irían cambiando.

Supongamos que el número de documentos relevantes es 10 (|*R*|=10), y que la respuesta del sistema nos da los siguientes 16 documentos, marcados en negrita los que son relevantes:

(**d5**, **d76**, d24, **d9**, d88, d15, **d67**, **d12**, d19, **d27**, **d24**, d43, **d234**, **d127**, **d78**, d23)

Cuando obtenemos el primer documento la precisión es 1, porque es 1 relevante de un recuperado, la exhaustividad es de 0,10 porque es 1 documento relevante de un total de 10. Al obtener el segundo, como también es relevante, la precisión se mantiene, pero la exhaustividad aumenta a 0,20 ya que son dos relevantes de un total de 10. Al

recuperar el tercer documento, sin embargo la precisión disminuye por tratarse de un no relevante, con lo que pasa a ser de 0,66 y se mantiene la exhaustividad. Si continuamos analizando obtenemos los datos de la tabla 1.

Tabla 1. Precisión y exhaustividad para un ejemplo de recuperación.

A	Ra	Precisión	Exhaustividad
1	1	1,00	0,10
2	2	1,00	0,20
3	2	0,66	0,20
4	3	0,75	0,30
5	3	0,60	0,30
6	3	0,50	0,30
7	4	0,57	0,40
8	5	0,63	0,50
9	5	0,56	0,50
10	6	0,60	0,60
11	7	0,64	0,70
12	7	0,58	0,70
13	8	0,61	0,80
14	9	0,64	0,90
15	10	0,66	1,00
16	10	0	1,00

Analizando la tabla 1, se observa que la precisión va disminuyendo a medida que obtenemos más documentos, sin embargo la exhaustividad va aumentando, hasta llegar al valor máximo que es 1, y se produce cuando hemos recuperado todos los documentos relevantes, a partir de ese momento, por convenio, la precisión se hace 0.

Si representamos la precisión frente a la exhaustividad para los valores de la tabla 1 obtenemos la gráfica de la figura 7.

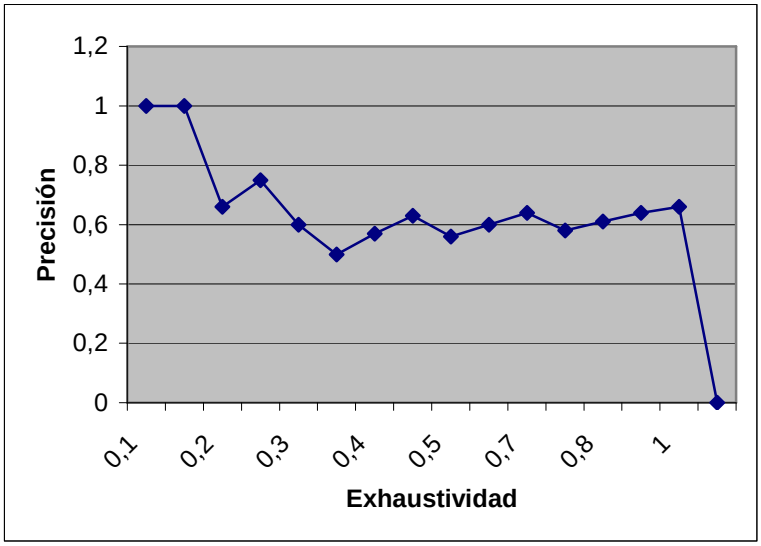

Figura 7. Diagrama precisión-exhaustividad.

Estos valores son para una consulta, si tuviéramos varias y quisiéramos evaluar un algoritmo sobre todas ellas, utilizaríamos la precisión media:

$$\overline{P}(r) = \sum_{i=1}^{Nq} \frac{Pi(r)}{Nq}$$

Siendo $\overline{P}(r)$ la precisión media en el nivel de exhaustividad *r* (primer valor en el que la exhaustividad alcanza el valor 1), *Nq* el número de consultas usadas y *Pi*(*r*) la precisión en el nivel de exhaustividad *r* para la consulta *i*.

Como los niveles de exhaustividad para cada consulta puede ser distinta para los 11 niveles de exhaustividad (0%, 10%, 20%,…,100%) es necesario utilizar una interpolación. Si r_j hace referencia al nivel de exhaustividad *j*, que puede ser uno de los 11 posibles, por ejemplo 60%, entonces:

$$P(r_j) = \max_{r_j \le r \le r_{j+1}} P(r)$$

Es decir, la precisión interpolada en el nivel de exhaustividad *j* es el máximo de las precisiones en el nivel *j* y el *j*+1. En la tabla 2 se recogen los valores de precisión de nuestro ejemplo para los 11 niveles de exhaustividad, y en la figura 8 se representan.

Tabla 2. Exhaustividad y precisión interpolada para el ejemplo.

Exhaustividad	Precisión
0	1,00
10	1,00
20	1,00
30	0,75
40	0,63
50	0,63
60	0,64
70	0,64
80	0,64
90	0,66
100	0,66

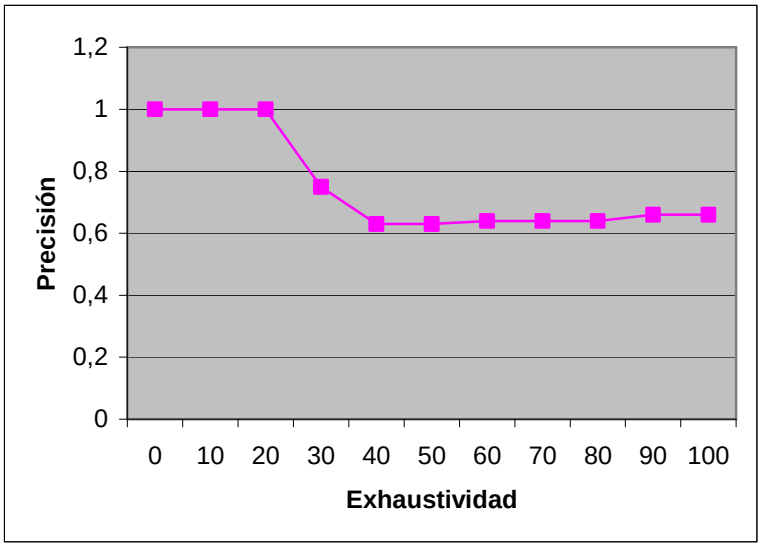

Figura 8. Diagrama precisión-exhaustividad interpolada.

La curva de precisión frente a exhaustividad que resulta de la media de varias consultas normalmente puede ser tomada como referente de un algoritmo y servir para comparar distintos métodos.

4.1.2 Resumen de valores individuales

Aunque la curva promedio de precisión contra exhaustividad es muy utilizada, puede haber situaciones en las que nos sea útil comparar los resultados de la

recuperación de un algoritmo para una consulta individual. Por dos motivos: porque la precisión media sobre varias consultas puede esconder la desviación en su comportamiento de un algoritmo, y porque cuando comparamos dos algoritmos podría interesarnos saber cuándo uno de ellos funciona mejor que el otro. En estos casos se hace necesario, tomar valores individuales en lugar de valores promedios. Estos valores individuales que se eligen, y que pueden ser varios, hay que considerarlos como un resumen de la correspondiente curva precisión-exhaustividad.

4.1.2.1 Precisión media en los documentos relevantes vistos

Se trata de obtener un valor resumen tomando la precisión cada vez que se obtiene un documento relevante. Vamos sumando la precisión de los documentos relevantes que van apareciendo y al final calculamos su media. Este valor favorece los sistemas que obtienen antes los documentos relevantes. Pero hay que tener cuidado con él porque puede darse el caso de una alta precisión media y sin embargo tener un pobre resultado en cuanto exhaustividad.

4.1.2.2 Precisión-R

La idea es generar un valor resumen de la clasificación tomando la precisión de la posición R, siendo R el número total de documentos relevantes para la consulta actual. Por ejemplo si R es 10, se tomaría la media de la precisión de los 10 primeros documentos que han aparecido.

Este parámetro se utiliza para observar el comportamiento de un algoritmo para cada consulta individual en un experimento.

4.1.2.3 Histogramas de precisión

La precisión-R para varias consultas puede usarse para comparar la evolución en la recuperación de dos algoritmos. Si denotamos $RP_A(i)$ y $RP_B(i)$ la precisión-R de los algoritmos *A* y *B* para la consulta *i*. Se define la diferencia:

$$RP_{A/B}(i)=RP_A(i)-RP_B(i)$$

Un valor 0 de esta medida indica que los dos algoritmos tienen equivalentes resultados para la consulta *i*. Un valor positivo indica un mejor resultado de recuperación para el algoritmo A.

Se suele representar mediante un diagrama de barras, como el de la figura 9, llamado histograma de precisión que permite rápidamente ver los resultados para dos algoritmos y una serie de consultas.

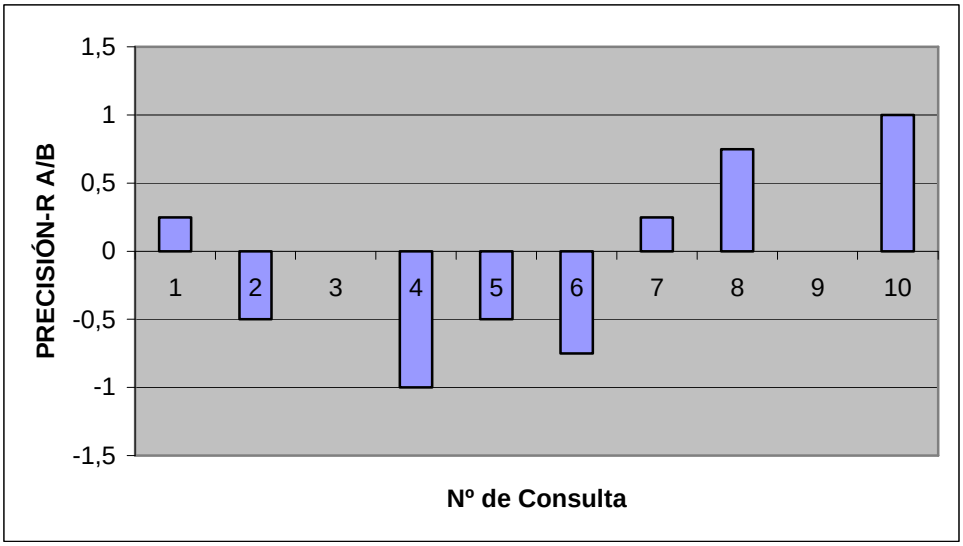

Figura 9. Histograma de precisión para 10 consultas.

4.1.2.4 Tabla resumen estadístico

Las medidas de valores individuales pueden también almacenarse en una tabla para proporcionar un resumen estadístico del conjunto de todas las consultas en una tarea de recuperación. Se podría incluir: número total de documentos relevantes recuperados por las consultas, número de relevantes que podrían haberse recuperado, etc.

4.1.3 Otras medidas alternativas

Aunque precisión y exhaustividad han sido utilizadas muy a menudo hay estudios que revelan problemas [Korfhage, 1997] [Raghavan et al., 1989] [Tague-Sutcliffe, 1992]. Algunos de estos casos, en los que existen mejores alternativas son:

- Colecciones muy grandes de documentos en las que no se conocen todos los elementos.
- Recuperación de consultas en modo interactivo, en las que pueden interesar medir otros factores como el tiempo, la facilidad de uso, etc.
- Cuando los documentos recuperados no se presentan ordenados.

Por ello, a lo largo de los años se han propuesto otras medidas alternativas como son las siguientes.

4.1.3.1 El término harmónico

Se trata de una medida que combina precisión y exhaustividad es el término harmónico F [Shaw Jr. et al., 1997], que se define:

$$F(j) = \frac{2}{\frac{1}{r(j)} + \frac{1}{P(j)}}$$

donde r (j) es la exhaustividad para los j primeros documentos clasificados, y P (j) la precisión para esos mismos documentos. La función F toma valores en el intervalo [0,1]. Vale 0 si no se han recuperado documentos relevantes y 1 cuando se han recuperado todos los relevantes. Toma valores altos cuando ambos valores, precisión y exhaustividad, son altos.

4.1.3.2 La medida E

Otra medida que combina exhaustividad y precisión es la medida de evaluación E [Rijsbergen, 1979]. Esta medida trata de recoger la opinión del usuario con respecto a qué medida le es más interesante, la exhaustividad o la precisión, mediante un parámetro b, que debe especificar. Se define:

$$E(j) = 1 - \frac{1 + b^2}{\frac{b^2}{r(j)} + \frac{1}{P(j)}}$$

donde *r* (*j*) es la exhaustividad para los *j* primeros documentos clasificados, y *P* (*j*) la precisión para esos mismos documentos. Para *b*=1, *E* (*j*) funciona como el complemento del término harmónico *F* (*j*). Valores superiores a 1 indican que el usuario está más interesado en la precisión que en la exhaustividad, y menores de 1 indican que prefiere la exhaustividad.

4.1.3.3 Medidas orientadas al usuario

Hasta ahora en las medidas anteriores se había supuesto que el resultado obtenido de las distintas consultas era el mismo para todos los usuarios, pero puede darse el caso de que distintos usuarios puedan tener distintas interpretaciones de cuáles son los documentos relevantes que se obtienen. Para cuantificar este problema, surgen las medidas orientadas al usuario [Korfhage, 1997], como son: ratio de cobertura, novedad del ratio, exhaustividad relativa y esfuerzo de exhaustividad.

Consideramos una colección de referencia, un ejemplo de solicitud de información *I*, y una estrategia de recuperación a ser evaluada. Sea *R* el conjunto de documentos relevantes para *I* y *A* el conjunto de documentos recuperado. Sea *U* el subconjunto de *R* que es conocido por el usuario, cuyo número de elementos es |*U*|. La intersección de los conjuntos *A* y *U* contiene los documentos conocidos por el usuario que serán relevantes cuando se recuperen. El número de documentos de este conjunto será |*Rk*|, mientras que |*Ru*| será el número de documentos relevantes desconocidos por el usuario que serán recuperados. El ratio de cobertura (*coverage ratio*) se define:

$$\text{cov}erage = \frac{|Rk|}{|U|}$$

La novedad del ratio se define:

$$novedad = \frac{|Ru|}{|Ru| + |Rk|}$$

Un valor alto del ratio de cobertura indica que el sistema busca más documentos relevantes que los que el usuario esperaba ver. Un valor alto de novedad indica que el sistema está revelando al usuario muchos nuevos documentos de los que desconocía.

La exhaustividad relativa viene dada por el ratio entre el número de documentos relevantes encontrados y el número de relevantes que el usuario esperaba encontrar.

El esfuerzo de exhaustividad es dado por el ratio entre el número de documentos relevantes que el usuario esperaba encontrar y el número de documentos examinados en un intento por buscar los documentos relevantes esperados.

4.2 Colecciones de evaluación

A la hora de evaluar un proceso de recuperación de evaluación necesitamos comparar los documentos relevantes que obtiene el sistema con los que realmente debería recuperar, para ello es necesario conocer cuáles deben ser los documentos relevantes y esto sólo se consigue realizando el análisis sobre una colección de evaluación.

Estas colecciones están formadas por: un conjunto de documentos con sus índices, un conjunto de consultas y un conjunto de documentos relevantes para cada una de las consultas. Estos documentos suelen haber sido seleccionados por expertos.

4.2.1 Las primeras colecciones

Se trata de colecciones que son pequeñas y por tanto necesitan mucho menos tiempo de instalación y preparación que las colecciones que se manejan hoy día, pero que han sido ampliamente utilizadas.

Algunas colecciones utilizadas para estudios experimentales [Fox, 1983] son:

- ADI. Documentos en ciencias de la información. Consta de 82 documentos y 35 consultas.
- CACM. (*Communications of ACM*). 3204 documentos y 52 consultas. Todos los artículos publicados en *Communications of the ACM* desde 1958 a 1979. Cubren aspectos relacionados con la informática.

  Junto con los documentos se proporciona información organizada en campos como: autor, fecha, palabras contenidas en el título, referencias directas entre artículos, etc.

  Para cada consulta, se incluye también el conjunto de sus documentos relevantes.
- INSPEC. Resúmenes de electrónica, informática y física.
- CISI o simplemente ISI (*Institute of Scientific Information*). 1460 artículos de biblioteconomía y 76 consultas. Para cada documento incluye tres campos: autor, palabras contenidas en el título y en el resumen y número de citaciones conjuntas por cada par de artículos.
- *Medlars (Medical articles)*. 1033 artículos médicos y 30 consultas.

Otras colecciones también muy utilizadas han sido:

- *Cranfield.* Contiene 1400 documentos sobre aeronáutica, con 225 consultas y sus correspondientes documentos relevantes.
- CF *(Cystic Fibrosis Collection).* Compuesta de 1239 documentos sobre la fibrosis cística, y 100 consultas realizadas por expertos con más de 20 años de experiencia. Está obtenida a partir de la base de datos MEDLINE. Cada una de las consultas tiene cuatro juicios de relevancia (con valores 0 para no relevante, 1 para marginalmente relevante y 2 para muy relevante) aportados por 3 expertos y por un bibliógrafo médico, respectivamente.

4.2.2 La colección *TREC* (*Text REtrieval Conference*)

Durante bastantes años la evaluación de la recuperación de la información fue difícil porque no se disponía de colecciones grandes con las que probar los algoritmos, esto hacía que las comparaciones entre resultados pudieran ser bastante dispares. A principios de los años 90, Donna Harman, del Instituto Nacional de Estándares y Tecnología (NIST) de Maryland, comienza a organizar unas conferencias, llamadas TREC [http://trec.nist.gov], dedicadas a la experimentación con grandes colecciones de incluso millones de documentos. En cada conferencia se han ido diseñando conjuntos de experimentos de referencia que después se han utilizado para comparar los sistemas de recuperación.

Además de NIST, también copatrocina estas conferencias, DARPA (*Defense Advanced Research Projects Agency*), como parte del programa TIPSTER.

Los participantes en las conferencias utilizan gran variedad de técnicas que desean evaluar, todos ellos trabajan con la misma colección, en un principio de unos 2 *Gb*. de texto (TREC-3), con más de 1 millón de documentos y conjuntos de preguntas (*topics*). De esta forma se pueden comparar de forma efectiva los resultados.

Un ejemplo de documento TREC y otro de pregunta (obtenidos de [Harman, 1995], son los mostrados en las figuras 10 y 11.

```
<doc>
<docno>WSJ880406-0090</docno>
<hl>AT&T Unveils Services to Upgrade Phone Networks Under Global Plan</hl>
<author>Janet GuyonWSJ Staff)</author>
<dateline>New York</dateline>
<text>
American Telephone & Telegraph Co. introduced the first of a new generation of
phone services with broad implication for computer and communication equipment
markets.
AT & T said it is the first national long distance carrier to announce prices for
specificic services under a world-wide standarization
...
</text>
</doc>
```

Figura 10. Ejemplo de documento TREC.

<top>
<num> Number: 168
<title> Topic: Financing AMTRAK

<desc> Description:
A document will address the role of the Federal Governement in financing the operation of the National Railroad Transportation Corporation (AMTRAK).

<narr> Narrative**:** A relevant document must provide information on the governmet's responsibility to make AMTRAK an economically viable entity. It could also discuss the privatization of AMTRAK as an alternative to continuing government subsidies. Documents comparing government subsidies given to air and bus transportation with those provided to AMTRAK would also be relevant.
</top>

Figura 11. Ejemplo de consulta en la colección TREC.

4.2.3 *CLEF* (*Cross Language Evaluation Forum*)

En la sociedad de la información actual cada día toma más relevancia la utilización de documentos que están escritos en distintos idiomas. Se buscan sistemas que permitan tanto la recuperación como la consulta, de forma independiente de los lenguajes originales de los documentos. Es lo que se conoce como CLIR (*Cross Language Information Retrieval*), recuperación de la información multilingüe.

El Foro Europeo de evaluación multilingüe (CLEF), persigue dos objetivos básicos en relación con las librerías digitales globales. Por una parte desarrollar una infraestructura que permita la prueba, calibración y evaluación de los sistemas de recuperación de la información que se utilizan con los lenguajes de la unión europea. Por otra parte, crear colecciones de tests, con datos reutilizables, que puedan ser utilizados por los desarrolladores e investigadores como conjuntos de prueba.

De esta forma, se facilita la colaboración entre distintos grupos, que se enfrentan a problemas similares y que tienen intereses comunes, pero además fomenta el intercambio de ideas y de resultados. Así se consigue aunar esfuerzos y competir en el mercado global al igual que se hace desde Estados Unidos o Asia.

CLEF nació en enero de 2000, como una evolución de una línea de estudio que se había formado en TREC junto con un grupo de voluntarios europeos, entre 1997 y 1999, para el estudio de los lenguajes multilingües europeos [Peters y Braschler, 2001].

4.2.4 *NTCIR* (*NII-NACSIS Test Collection for IR Systems*)

Son una serie de talleres de evaluación para facilitar la búsqueda en el acceso a la información, incluyendo Recuperación de la Información, obtención de respuestas, resúmenes de textos, extracción, etc. Fueron patrocinados conjuntamente por JSPS (*Japan Society for Promotion of Science*) y por NACSIS (*National Center of Science Information Systems*) desde 1997, añadiéndose RCIR/NII (*Research Center for Information Resources at National Institute of Informatics*) a partir del 2000.

Sus principales objetivos son:

- Estimular las tecnologías para el acceso a la información, proporcionando colecciones de test reutilizables para la experimentación que permitan la comparación entre lenguajes.
- Proporcionar un foro a los grupos de investigación sobre estos temas.
- Investigar métodos de evaluación, para técnicas y métodos sobre acceso a la información, que permitan manejar conjuntos de datos reutilizables en los experimentos.

5 OPERACIONES CON TEXTO

5.1 Introducción

Una de las operaciones más importantes en recuperación de la información es el preprocesado de los documentos. Se debe determinar cuáles son las palabras que se van a utilizar como términos índice, y es un proceso en el que pueden intervenir muchos factores. Se pueden incluir determinados tipos de palabras, ya que unas aportan más significación que otras, como pueden ser los nombres. Se puede optar por incluir todas las palabras, con lo que nos aseguraríamos que no se van a producir resultados de consultas inusuales para el usuario final, o podríamos elegir soluciones intermedias como utilizar listas de palabras vacías, reducir palabras a su lexema, construir tesauros o utilizar un vocabulario controlado.

Pero además del preprocesado del documento, son necesarias otras tareas como puede ser la construcción de un tesauro representando la relación entre términos y el agrupamiento de los documentos relacionados. Si bien en otros contextos como son las librerías digitales hay otros factores que son críticos como es la rapidez en la recuperación, dejando al margen la precisión en la recuperación.

Para favorecer la rapidez en las transmisiones por la Web en ocasiones se recurre a la compresión de documentos para que disminuya la transmisión, surgiendo un nuevo problema como es el tiempo de la comprensión/descompresión. Aunque últimamente las nuevas técnicas de compresión son muchos más rápidas y en muchas ocasiones permiten operar con los documentos sin tener que descomprimirlos.

Por último otra operación que cada día está siendo más utilizada, sobre todo a raíz del comercio electrónico, es la encriptación, junto con las ya habituales de protección mediante claves. En ambos casos las operaciones de encriptación de texto son muy importantes y constituyen un amplio campo de estudio.

5.2 Preprocesado de documentos

El preprocesado de documentos suele dividirse en las siguientes 5 etapas [Baeza-Yates y Ribeiro-Neto, 1999].

5.2.1 Análisis léxico del texto

Es el proceso de convertir el texto de entrada, entendido como una secuencia de caracteres, en un conjunto de palabras, de las cuáles se extraerán los términos índice. El punto de partida es determinar cuáles son los separadores entre palabras, pero además hay que tener en cuenta [Fox, 1992]: los dígitos, guiones, signos de puntuación y la distinción entre letras mayúsculas y minúsculas.

Los números, en general, no son buenos términos índice porque generalmente su significado viene asociado a un contexto, por lo que ellos tienen una significación muy vaga. En principio se podría prescindir de ellos, si bien en algunos casos, como cuando vienen unidos a una palabra, podrían ser bastante determinantes, o con determinados formatos, como un DNI o una tarjeta de crédito. En estos últimos casos podrían utilizarse pero con formatos unificados.

Los guiones son otro caso complicado de tratar, porque pueden aparecer en distintos contextos, como separadores de palabras en determinadas frases, o como parte integral de palabras. La solución es adoptar una regla general y especificar las excepciones caso por caso.

En el caso de los símbolos de puntuación suelen eliminarse en esta fase de análisis léxico. En algunas ocasiones pueden estar integrados en las propias palabras, pero no tiene mayor trascendencia eliminarlos, porque en las consultas también serán eliminados con lo que no deberían afectar a la recuperación. Aunque siempre podría darse el caso de que fuera necesaria una lista de excepciones.

La distinción entre mayúsculas y minúsculas tampoco suele ser un problema para los analizadores, que normalmente lo que hacen es convertir todo el texto o a mayúsculas o a minúsculas.

Todos estos aspectos tratados son delicados y en muchas ocasiones requieren tomar decisiones de diseño, que habrá que adoptar cuidadosamente, ya que los resultados finales pueden verse afectados.

5.2.2 Eliminación de palabras vacías

A la hora de elegir las palabras que van a formar parte de los índices interesan términos que sean discriminantes, es decir, que nos permitan separar unos documentos de otros. Por esta razón, las palabras muy frecuentes, que aparecen en todos los documentos e incluso varias veces en cada uno de ellos no son de utilidad para este propósito. Suelen denominarse palabras vacías y filtrarse para no ser utilizadas. Las más habituales son los artículos, preposiciones y conjunciones, aunque en algunas ocasiones pueden incluirse verbos, adverbios y adjetivos, o al menos alguno de ellos.

El principal beneficio que aporta su eliminación es la reducción del tamaño de las estructuras que mantienen los índices, que estaría en torno al 40%. Pero, por el contrario, puede perderse exhaustividad en la recuperación. Imaginemos una consulta como “El sí de las niñas”, podría verse reducida a niñas, con lo que los documentos recuperados no serían demasiado precisos. Por esta razón en muchos buscadores Web, se utiliza el texto completo como índice.

5.2.3 Lematización (*Stemming*)

Lematizar, según la Real Academia Española de la Lengua, es: “En un diccionario o repertorio léxico, elegir convencionalmente una forma para remitir a ella todas las de su misma familia por razones de economía”.

En el caso de la recuperación de la información se trata de sustituir las palabras por su lema (*stem*), esto hará que disminuyan las variantes de las palabras adoptándose un único concepto, lo que traerá como consecuencia que disminuya el tamaño de las estructuras índice. Aunque en ocasiones se utilizan indistintamente los términos raíz y

lema, ya que son próximos y en muchos casos coinciden, existen diferencias, por lo que utilizaremos lema.

La lematización trabaja en el nivel morfológico del lenguaje natural. La morfología es el área de la lingüística que se encarga de la estructura interna de las palabras, y aunque en ocasiones no se le ha dado mucha importancia, su efecto se va a notar en el rendimiento. La morfología suele dividirse en dos subclases: inflexión y derivación. La inflexión (o flexión) describe los cambios predecibles que las palabras sufren como resultado de la sintaxis, por ejemplo el plural en los nombres o las formas del pasado en los verbos. Son cambios que no afectan al significado general de las palabras, mientras que la derivación, procedimiento por el cual se forman vocablos alterando la estructura de otros mediante formantes no flexivos como los sufijos, puede o no afectar al significado.

Puede haber distintas formas de lematizar [Kroventz, 1993], por una parte puede utilizarse como un método de expansión de preguntas, sustituyendo los términos por sus lemas, por otra parte puede verse como una técnica de *clustering*, en la que los *clusters* están basados en reglas de unión y por último como una forma de normalizar los conceptos utilizados en las consultas. Estas formas han dado lugar a multitud de algoritmos.

Un algoritmo de lematización es un procedimiento computacional por el que se reducen las palabras a su forma común, quitando los sufijos de derivación y flexión [Lovins, 1968]. De esta forma se pueden reducir los ficheros de índices introduciendo los lemas de los términos y realizar las búsquedas utilizando también lemas, con lo que se aumentan los éxitos de asociar las preguntas con los documentos.

Aunque en muchas ocasiones, lematizar es una práctica habitual, no existen estudios concluyentes que determinen su beneficio [Frakes y Baeza-Yates, 1992]. También son muchas las clasificaciones de los algoritmos de lematización. Una de las más utilizadas, la de Frakes, distingue cuatro tipos de estrategias de lematización:

- **Consulta de tabla**. Consiste en mirar la raíz de las palabras en una tabla. Simple, pero puede requerir mucho espacio.
- **Variedad de sucesor**. Basado en la determinación del límite de los morfemas (unidad mínima significativa del análisis gramatical).

- **N-gramas**. Basada en la identificación de digramas y trigramas (combinaciones de 2 y 3 letras). Utilizado más para agrupar términos que como procedimiento de lematización.
- **Extracción de afijos**. Es un método intuitivo, simple y que puede ser implementado eficientemente, lo que hace que sea muy utilizado. Lo más importante en el proceso es extraer el sufijo, ya que las principales variantes de las palabras se producen mediante sufijos. Se suele utilizar una lista de sufijos, que se van aplicando a las palabras, para transformarlas, mediante unas reglas.

Los dos algoritmos más importantes de lematización, para el idioma inglés, son el de Lovins [Lovins, 1968] y el de Porter [Porter, 1980]. El primero de ellos elimina unos 260 sufijos distintos clasificados por tamaño, mientras que el segundo elimina unos 60 mediante un método de aproximación en varias iteraciones. Uno de los problemas que surgen al lematizar es la pérdida o cambio de significado de las palabras que puede producirse al acortarlas, por eso Kroventz, en su algoritmo, utiliza lematización flexiva, indexando los documentos por significados, en lugar de por palabras. El proceso que sigue es eliminar los plurales, los participios y los gerundios y chequear el resultado con un diccionario para verificar que el resultado es válido, él mismo compara sus resultados con los de Lovins y Porter concluyendo que son mejores.

También existen lematizadores para otros idiomas, como francés [Savoy, 1999], español [Figuerola et al., 2002], árabe, holandés, griego esloveno o latín. Son precisamente en idiomas morfológicamente más complejos como el griego [Kalamboukis, 1995] donde se obtienen mejores resultados, a diferencia del inglés que por su simplicidad la mejora de la exhaustividad no compensa la pérdida de precisión.

5.2.4 Selección de términos índice

La primera disyuntiva que se puede plantear es si se va a utilizar el texto completo para indexar o bien se van a seleccionar un conjunto de términos. En el caso de elegir sólo algunas palabras puede hacerse de forma automatizada o bien seleccionarlas, mediante algún experto en el tema.

En los procesos de selección automatizados se suelen intentar seleccionar los nombres, ya que como se ha visto son los que más significación conllevan y por tanto permiten una mejor recuperación de los textos. Para seleccionarlos el proceso es eliminar el resto de tipos de palabras, como artículos, conjunciones, etc. Cuando aparecen dos o más nombres juntos suelen dejarse como un único término índice y se denominan grupos de nombres. Estos grupos son conjuntos de nombres que están separados por un número de palabras que no excede un valor umbral que se determina y que suele ser 3.

5.2.5 Construcción de tesauros

Un tesauro consiste en una lista de palabras importantes en una determinada área de conocimiento, que para cada término almacenado mantiene un conjunto de palabras relacionadas con ese término. Lo habitual es que este conjunto de palabras relacionadas para un término, tenga establecida una relación de sinonimia.

Los principales propósitos de un tesauro básicamente serían [Foskett, 1997]:

- Proporcionar un vocabulario estándar.
- Ayudar a los usuarios en la localización de términos para formular consultas adecuadas.
- Proporcionar jerarquías clasificadas que permitan ampliar o limitar las consultas de los usuarios según sus necesidades.

Se utiliza un tesauro cuando interesa mantener un vocabulario controlado, con las ventajas que ello conlleva como la normalización de los términos indexados, la reducción de ruido, o la recuperación basada en conceptos en lugar de palabras. Suele estar asociado a áreas de conocimiento con amplio vocabulario acumulado, como puede ser la medicina, pero en otras ocasiones no está tan clara su utilización como es el caso de la Web, debido a sus continuos cambios.

A la hora de recuperar documentos, primero se decide qué es lo que se quiere recuperar y después se plantea una consulta en el lenguaje del sistema de recuperación

de la información. Esta consulta contendrá una serie de términos, que dependiendo de la experiencia del usuario podrá dar mejores o peores resultados, pero suele darse el caso de que se obtengan pobres resultados, debido en muchos casos a la ambigüedad del lenguaje, a las distintas formulaciones dependiendo de los usuarios y a figuras como la polisemia y la sinonimia [Furnas et al., 1987]. Es en estos casos cuando más útil sería el tesauro porque permitiría mediante los términos relacionados llevar a cabo una recuperación más amplia. Incluso podría intervenir el usuario modificando los términos de la consulta inicial, lo que conllevaría un nuevo cálculo de los pesos de los términos, dando lugar a lo que se conoce como expansión de consultas.

Aunque los procesos de creación de tesauros manual y automatizado, como es lógico, tienen sus diferencias, comparten muchas tareas relacionadas con la frecuencia y la capacidad de discriminación de los términos, como son [Peña et al., 2002]:

- Eliminar términos de alta frecuencia y baja discriminación.
- Formar términos compuestos por términos de baja frecuencia y baja discriminación para que resulten más específicos que utilizados por separado.
- Agrupar términos de buena discriminación y frecuencias media en clases para aumentar la especifidad frente a la completitud de la respuesta.
- Controlar que la frecuencia de los términos agrupados en una clase sea similar, para mejorar las comparaciones de las preguntas con los documentos.
- Equilibrar el tamaño de las clases, para una mejor movilidad por el árbol del tesauro.

Si nos centramos en la creación automatizada de tesauros, estos serían los métodos más utilizados [Han et al., 1995]:

- Basados en información sintáctica. Las relaciones entre términos son generadas en base al conocimiento lingüístico y a la coocurrencias estadísticas [Grefenstette, 1992] [Ruge, 1992]. Se utiliza una gramática y un diccionario para sacar una lista de términos relacionados con uno dado. Las consultas se expanden añadiendo estos términos. Esto produce

unos resultados sólo algo mejores que los de las consultas originales [Salton, 1971b].

- Basados en la información relevante. La información relevante se utiliza para construir estructuras globales de información, como pseudotesauros [Salton, 1980] o árboles de mínima expansión [Smeaton y van Rijsbergen, 1983]. Las consultas se expanden con los términos de estas estructuras. El inconveniente de este método es que no siempre se cuenta con información sobre la relevancia.
- Clasificación automática de términos. La similitud de términos se realiza primero basándose en la hipótesis de asociación (si un término es buen discriminante de documentos relevantes y no relevantes, un término íntimamente asociado a él también lo será), y después se utiliza para clasificar términos poniendo un valor umbral de similitud [Lesk, 1969] [Minker et al., 1972] [Spark-Jones y Barber, 1971]. Las consultas son expandidas añadiendo todos los términos de las clases que contienen los términos de la consulta. Según varios autores esta idea es demasiado pobre para ser útil [Minker et al., 1972] [Peat y Willett, 1991] [Spark-Jones, 1991].
- Utilización de clasificación de documentos. Los documentos primero son clasificados y los términos infrecuentes encontrados en una clase de documentos son agrupados para formar las clases de un tesauro [Croch, 1990]. La indexación de los documentos y de las consultas es incrementada o bien remplazando un término por una clase del tesauro o añadiendo un tesauro al índice de datos. Se trata de un método cuyos resultados dependen de parámetros que son muy difíciles de determinar [Croch y Yong, 1992], a la vez que muy costoso.
- Expansión de consulta basada en conceptos [Qiu y Frei, 1993] (tesauros de similitud). Un tesauro de similitud es una matriz de términos frente a términos construida basándose en cómo están indexados los términos de la colección. Cada término de la colección se caracteriza por los documentos en los que aparece. Se utiliza un método probabilístico para

estimar la probabilidad de que un término sea similar a los dados en la consulta. Este método mejora sustancialmente el rendimiento de la recuperación.

- Expansión de consultas basadas en frases [Jing y Croft, 1994]. Se construye un tesauro de asociación de términos frente a frases para plasmar la relación existente entre términos y frases. Para expandir las consultas se añaden las frases más relacionadas con la consulta mediante el tesauro. Al igual que el caso anterior, también se mejora el rendimiento de la recuperación.

6 BÚSQUEDA DE INFORMACIÓN EN LA WEB

6.1 Introducción

Un caso especial de recuperación de la información es la Web, por varios motivos, primero por su tamaño, es difícil encontrar bases de datos de sus dimensiones, del orden de los Terabytes y en continuo crecimiento. Segundo por su dinamismo, ya que continuamente está cambiando, no sólo creciendo, sino modificando y eliminando contenidos. Tercero por la heterogeneidad de elementos: textos en distintos formatos, audio, imágenes y vídeo.

Pero si ahondamos en los datos aún se pueden detectar más problemas:

- La distribución de los datos, por multitud de redes y subredes, sin estructuras predefinidas.
- La falta de estructura.
- La redundancia de los datos, que pueden estar repetidos en diversos servidores.
- La heterogeneidad de la calidad de la información, ya que al no haber ningún filtrado, pueden aparecer informaciones falsas, obsoletas, malintencionadas, mal escrita o con errores de diversos tipos. Existen algoritmos específicos para cuando la información es incompleta o con ruido, pero se conoce la tasa de error [Aggarwal y Yu, 2008].

Otros tipos de problemas son los relacionados con los usuarios, y serían los siguientes:

- La forma de especificar lo que se desea recuperar, el lenguaje de consulta.
- La forma de procesar los datos que se obtienen como resultado. Aquí estaría incluida la forma en que se clasifican los resultados, la forma en que se presentan determinados documentos, que pueden ser, por ejemplo, excesivamente grandes.

Se pueden distinguir tres formas de buscar en la Web:

- Utilizar motores de búsqueda que indexan una parte de los documentos de la Web como si se tratara de bases de datos que manejan el texto completo.
- Utilizar directorios Web, en los que aparecen los documentos clasificados por temas.
- Mediante la utilización de hiperenlaces.

6.2 Características de la Web

Realmente es difícil ofrecer cifras sobre la Web por los continuos cambios que se producen, por su evolución, por las elevadas cifras que se manejan y por su vasta distribución geográfica.

Como datos globales (2002) podemos hablar de más de 40 millones de servidores Web, más de tres mil millones de páginas, y más de 600 mil millones de archivos. Casi la mitad de estas páginas estarían en inglés, aunque existirían páginas en más de 100 lenguas.

En cuanto a la Web española (datos del año 2000) [Baeza-Yates, 2002], tendría 37.672 dominios, 47.788 sitios Web, 7,3 millones de páginas, 4,1 millones de palabras distintas y un volumen de 25,5 Gb. En cuanto a la distribución de los dominios sería: .es (28,1%), .com (59,5%), .org (5,9%), .net (5,9%) y otros (0,6%).

El mismo autor [Baeza-Yates et al., 2005], actualizando sus datos a octubre de 2004, estima que en España ya se puede hablar de 300.000 sitios (6 veces más) y más de 16 millones de páginas (algo más del doble) en tan solo cuatro años de diferencia.

La estructura de la Web, véase figura 12, tendría un núcleo fuertemente enlazado desde el que se podría navegar de unas páginas a otras. En la parte izquierda estarían las páginas nuevas, desde las que se podría llegar al núcleo, pero no regresar. En la parte derecha estarían las páginas desactualizadas o viejas a las que se podría llegar desde el núcleo, pero tampoco volver. Existirían también páginas aisladas en islas, pasarelas

entre las páginas nuevas y algunas antiguas, y entradas y salidas aisladas, que serían los tentáculos.

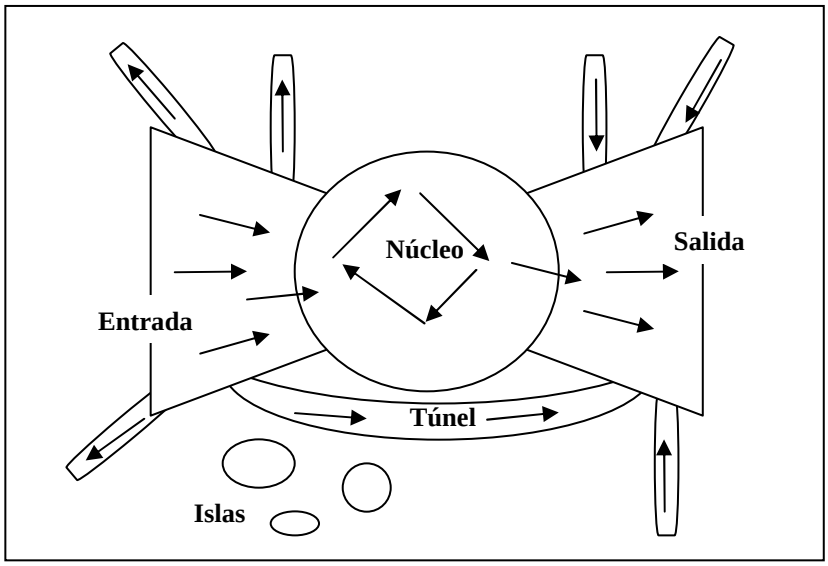

Figura 12. Estructura topológica de la Web [Baeza-Yates, 2002].

En España, el núcleo está formado por el 71% de las páginas, las entradas son el 12%, las salidas el 10% y el resto lo conforman el 7%.

Las palabras más comunes en los servidores españoles son: copyright (15.3%), información, general, España, página, servicios, Madrid, mail, home, Internet (9.7%)

Las palabras más buscadas: gratis, fotos, España, Madrid, juegos, móviles, música, historia, vídeo, viajes, mp3 y turismo. La búsqueda de frases representa un 12%.

Los formatos, o tipos de ficheros más populares son: HTML para documentos Web, GIF y JPG para imágenes, ASCII, PostScript y PDF para texto, MP3 para audio, MPG y AVI para vídeo.

El tamaño medio de una página Web en HTML, suele ser de 5 a 7 KBytes, con pocas imágenes, en torno a 5, en muchas ocasiones para definir el aspecto de la página, con botones, bolas, líneas, etc. Contienen también entre 5 y 15 enlaces, la mayoría de ellos a la propia página o a otras páginas en el propio servidor.

6.3 Buscadores en la Web

Existen dos procedimientos básicos de búsqueda en la Web: los directorios (*Open Directory*) y los motores de búsqueda. En muchas ocasiones desde la misma página Web se ofrecen ambos servicios.

Los directorios son estructuras jerarquizadas por temas. En un primer nivel aparecerían una serie de temas generales, que permitirían acceder a cualquiera de ellos, que a su vez se subdividiría en temas más específicos, y así sucesivamente se iría bajando de nivel. Se trata de una forma de acceso bastante intuitiva, sobre todo cuando el usuario no sabe muy bien qué es lo que está buscando. Al usuario no iniciado le resulta sencillo moverse por la estructura, pero de fondo está el problema de la clasificación de los documentos. Es necesario ir anexando cada documento a la jerarquía.

La otra forma de acceso son los motores de búsqueda, que utilizan el método de recuperación de texto completo. En este caso el usuario debe plantear la consulta mediante la utilización de texto, normalmente introduciendo palabras o frases de búsqueda. Estos motores de búsqueda, previamente han tenido que indexar todas las palabras para poder realizar las recuperaciones. Este proceso de indexación es lo más complicado porque deben recorrer la Web en busca de información, añadiendo las nuevas páginas. Se trata de un proceso continuo, debido a la movilidad y dinamismo de la Web.

6.3.1 Arquitectura de los buscadores

Podemos hablar de dos tipos de arquitecturas referidas a los buscadores: arquitectura centralizada y arquitectura distribuida.

La arquitectura centralizada, representada en la figura 13, utiliza robots o rastreadores que se mueven por la Web en busca de nuevas páginas que va enviando al servidor central para ser indexadas. A pesar de su nombre, no se desplazan por la red,

sino que se están ejecutando en una máquina que envía solicitudes a los servidores repartidos por la Web, para recabar información.

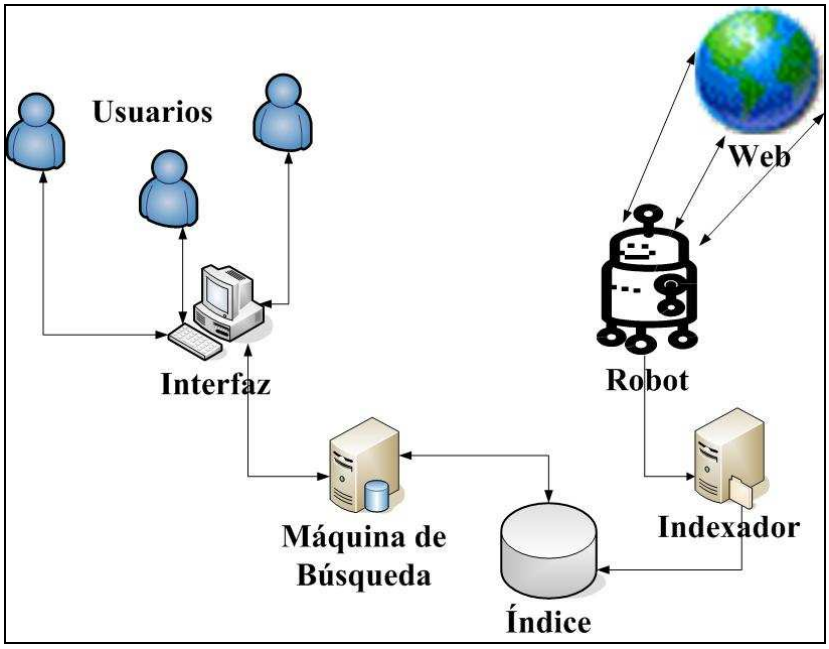

Figura 13. Arquitectura centralizada de un buscador.

Si observamos la arquitectura, se aprecia otra parte diferenciada que hace referencia a las consultas que el usuario, mediante una interfaz, formula al sistema y éste responde mediante los resultados clasificados obtenidos.

Los principales problemas de esta arquitectura son: por una parte, el proceso de mantenimiento de los datos, ya que la Web cambia, los servidores tienen mucha carga y a veces existen dificultades en las transmisiones, y por otra parte el gran volumen de datos que se almacenan.

El segundo tipo de arquitectura que se puede dar, más eficiente, es la arquitectura distribuida *Harvest* [Bowman et al., 1994]. Su principal inconveniente es que necesita la coordinación de varios servidores Web, pero permite abordar algunos de los problemas que presentaba la arquitectura centralizada, como eran:

- Los servidores Web recibían solicitudes de distintos robots, incrementando su carga.

- El tráfico en la red crecía porque los robots recuperaban objetos completos, cuando la mayoría de su contenido se descartaba.
- La información era recopilada independientemente por cada robot, sin coordinación entre los distintos buscadores.

La forma de paliar estos problemas es introduciendo dos nuevos elementos: recolectores (*gatherers*) y agentes (*brokers*). Un recolector recopila e indexa información de uno o varios servidores Web. Se establecen en el sistema unos tiempos periódicos para recolectar información. Un agente proporciona el mecanismo de indexación y el interfaz para las consultas de los datos recolectados. Los agentes recuperan la información de uno o más recolectores o de otros agentes, actualizando incrementalmente sus índices, sin necesidad de que los documentos (objetos completos) circulen por la red.

Se pueden dar distintas configuraciones entre agentes y recolectores, dando lugar a distintas prestaciones y consiguiendo distintas mejoras, ya que se trata de una arquitectura muy flexible, un posible esquema genérico sería el de la figura 14.

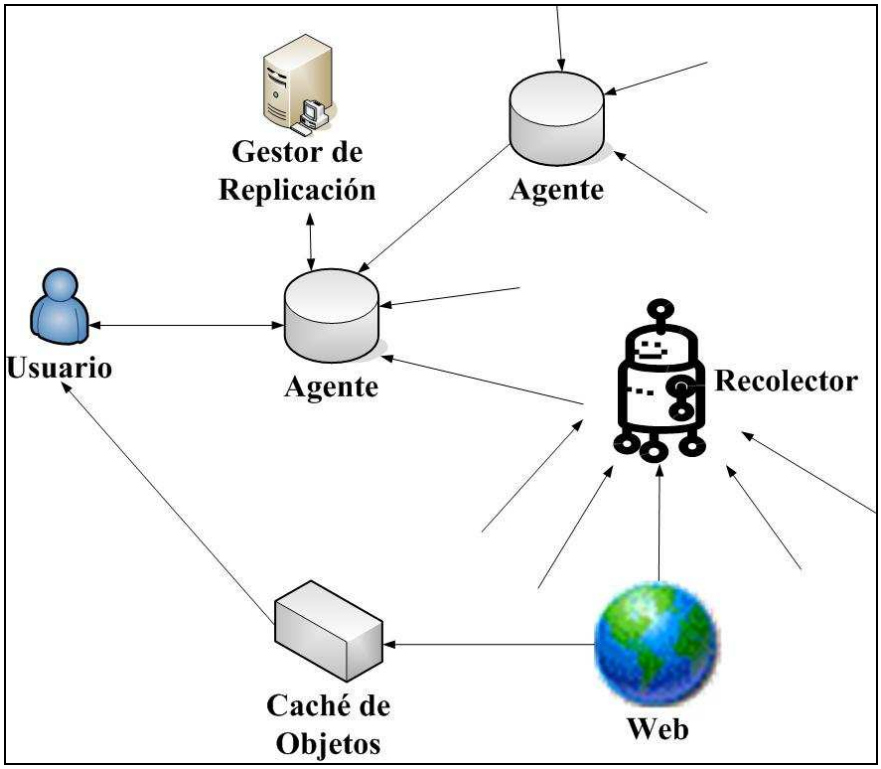

Figura 14. Arquitectura distribuida Harvest.

Otras de las mejoras que introduce esta arquitectura es la caché de objetos y la posibilidad de replicación. El replicar, por ejemplo en distintas zonas geográficas, permite accesos más rápidos. También se puede utilizar la replicación para dividir el proceso de recolección entre varios servidores Web. Por su parte la caché de objetos reduce la carga de los servidores y en general de la red, y mejora la latencia cuando se accede a las páginas Web.

Un ejemplo práctico de aplicación de esta arquitectura, en España, es el sistema DESITAS, impulsado por el Ministerio de Administraciones Públicas [Villagrá et al., 1999]. Se trata de un sistema de intermediación electrónica entre los usuarios y el entorno de las Administraciones Públicas, que permite la localización de la información que necesitan, mediante la búsqueda por palabras clave, evitando tener que conocer la dirección exacta de la unidad administrativa que contiene la información buscada.

Actualmente, [Harvest, 2004] es un sistema ampliamente extendido, distribuido mediante licencia GPL, que permite recopilar información mediante una interfaces Web, tanto en Internet como en intranet, utilizando http, ftp, o nntp. Además debido a su diseño modular ha ido incorporando todo tipo de formatos: HTML, TeX, DVI, PS, mail, páginas man, news, troff, RTF, Word, Excel, WordPerfect, código C, PDF y será fácilmente adaptable a otros nuevos.

6.3.2 Interfaz de usuario

Se distingue una interfaz de usuario para formular las consultas y otra para presentar los resultados al usuario. La interfaz de las consultas básica, suele consistir en una caja de texto en la que se escriben palabras. Dependiendo de los buscadores interpretan estas palabras de distinta forma, unos buscan las páginas que contenga alguna de las palabras, mientras que otros buscan las que contengan todas las palabras. Casi todos ellos permiten búsquedas más complejas, con operadores *booleanos*, búsquedas de frases, búsqueda por proximidad, etc.

Los resultados que se obtienen pueden ser filtrados de distinta forma, modificando la búsqueda con nuevas palabras, o excluyendo alguna palabra, o por tipo de documento, por idioma, por URL, por título, por fechas, etc.

La forma en que se presentan los resultados suele ser una lista, en la que cada entrada suele tener la URL, tamaño, formato, título, fecha, y dos o tres líneas bien con el comienzo, bien con un fragmento donde aparecen las palabras de la búsqueda. Dentro de la lista guardan un orden de relevancia, pero también se pueden dar distintos criterios.

6.3.3 Orden de relevancia de los resultados

La mayoría de los buscadores utilizan variaciones de los modelos booleano y vectorial para ordenar los documentos recuperados, en muchos casos sin necesidad de tener que acceder a los documentos sino sólo mediante los índices, no obstante los buscadores comerciales no facilitan información sobre las técnicas que utilizan, por lo que es difícil comparar sus resultados.

Además del clásico algoritmo de clasificación del modelo vectorial, basado en la estrategia de asignar pesos a los términos (también conocido como esquema *tf-idf term frequency-inverse document frecuency),* se pueden añadir los siguientes [Yuwono y Lee, 1996]:

- Dispersión *booleano*.
- Dispersión vector.
- Más citado.

Los dos primeros corresponden a los métodos utilizados por el modelo *booleano* y vectorial extendido, que incluyen las páginas apuntadas por una página de la respuesta, o las que apuntan a una página de la respuesta. El tercero está basado sólo en los términos incluidos en páginas que tengan un enlace a páginas en la respuesta.

Como los resultados obtenidos pueden ser dispares dependiendo del tipo de algoritmo aplicado y de las peculiaridades de los datos a los que se apliquen, es frecuente la investigación y adaptación de algoritmos, por ejemplo desarrollando algoritmos para grandes conjuntos de datos [Ganti et al., 2006] o para datos en la Web. Algunos de los nuevos algoritmos de clasificación, relacionados con la Web, utilizan la información de los hiperenlaces. Se consideran como páginas más populares y de mayor

calidad las que tienen más hiperenlaces que apuntan hacia ellas. Además si dos páginas tienen muchos enlaces comunes denota una relación entre ellas. Algunos ejemplos de este tipo de técnicas son:

- WebQuery [Carriere y Kazman, 1997]. La forma de trabajar es recopilando un conjunto de páginas, como pueden ser las de una consulta, y se ordenan basándose en cómo está conectada cada una de las páginas. Como segundo paso se amplía este conjunto buscando las páginas más conectadas con el conjunto original.

- HITS (*Hypertext Induced Topic Search*) [Kleinberg, 1998]. El esquema de ordenación depende de las consultas y parte del conjunto de páginas S, que apuntan a las páginas de la respuesta o son apuntadas por páginas de la respuesta. Las páginas que tienen muchos enlaces a ellas en S se llaman autoridades, contienen por tanto contenidos relevantes, mientras que las que tienen muchos enlaces hacia fuera se denominan *hubs*, y apuntan a contenidos similares. Existe una realimentación en ambos sentidos, entre estos dos tipos de páginas. Se define $H(p)$ y $A(p)$ los conjuntos de páginas *hubs* y autoridades normalizados de una página p, y se cumple la siguiente ecuación:

$$H(p) = \sum_{u \in S \mid p \to u} A(u), \qquad A(p) = \sum_{v \in S \mid v \to p} H(v)$$

  Estos valores pueden ser determinados mediante un algoritmo iterativo y convergen al principal vector característico de la matriz de enlaces de S. Hay que tener cuidado en el caso de la Web porque se puede disparar el tamaño del conjunto S, con lo que se suele definir un número de páginas.

- *PageRank*. Es parte del algoritmo que utiliza el buscador *Google* [Brin y Page, 1998]. Simula un usuario navegando aleatoriamente por la Web, que salta a una página de probabilidad q o sigue un enlace en la página actual con probabilidad 1-q. Se asume que este usuario nunca vuelve hacia una página que ya ha visitado. Este proceso puede modelarse con una cadena de Markov, calculando la probabilidad de permanecer en cada página, este valor es usado como parte del mecanismo de

ordenación de *Google*. Si $C(a)$ es el número de enlaces a páginas externas de la página a y se supone que a es apuntada por las páginas p_1 a p_n, entonces *PageRank* de a se define como:

$$PR(a) = q + (1-q)\sum_{i=1}^{n} PR(p_i)/C(p_i)$$

donde q debe ser puesto por el sistema (un valor típico es 0,15). *PR* puede ser calculado usando un algoritmo iterativo.

6.3.4 Rastreando la Web

La forma más simple de recorrer la red es tomar un conjunto de URLs y a partir de ellas ir ampliando el conjunto recursivamente a través de sus enlaces, bien a lo ancho bien en profundidad. Una segunda forma sería partir de un conjunto seleccionado de URLs, frecuentemente visitadas, ya que ellas podrían facilitar información a partir de sus solicitudes. En ambos casos se pueden realizar recorridos efectivos con un robot, pero si queremos utilizar varios conjuntamente, se presentan dificultades para no recorrer una página más que una vez. Una solución es dividir el espacio de búsqueda en zonas y asignar cada una a un robot. Estas zonas podrían ser asignadas mediante códigos de países o nombre de Internet.

Otro problema importante es la frecuencia con la que se actualizan los índices, ya que debido al continuo dinamismo de las páginas, nos podemos encontrar con enlaces que hacen referencia a páginas que ya no existen. Por ello se adoptan diferentes soluciones, la mayoría optan por incluir en la entrada la fecha de indexación, en algunos casos se actualizan después de unos días o semanas, otros optan por rastrear de nuevo todas sus páginas, mientras que algunos seleccionan sólo algunas para verificarlas.

Otro aspecto es la forma en que se realiza el recorrido, primero a lo ancho o primero en profundidad. Una política primero a lo ancho, recorre todas las páginas apuntadas por la página actual. Es un sistema bueno para páginas estructuradas por temas relacionados. Una política primero en profundidad visita la primera página apuntada desde la página actual y desde ahí baja de nivel a la primera apuntada en un

nivel inferior, así sucesivamente hasta que se llega al límite de profundidad, volviendo recursivamente.

Como los robots pueden sobrecargar los servidores con peticiones y pueden consumir un ancho de banda importante de Internet, se desarrollaron un conjunto de instrucciones [Koster, 1993] para su uso. Una de las instrucciones era colocar un fichero especial en la raíz de cada servidor indicando las restricciones de ese sitio, en particular las páginas que no deberían ser indexadas, o las que utilizaban marcos o mapas de sitio o contraseñas.

6.3.5 Índices

La mayoría de los índices utilizan variaciones de los ficheros invertidos, que son listas de palabras ordenadas, cada una de ellas con un conjunto de punteros a las páginas donde aparecen. Se pueden eliminar las palabras vacías para reducir el tamaño del índice y realizar tareas de normalización, como puede ser eliminar los signos de puntuación y el exceso de espacios en blanco o convertir a mayúsculas o minúsculas.

Para facilitar al usuario las tareas de búsqueda, en el índice se complementa con descripciones de la página, como el título, la fecha, el tamaño, la primera línea, etc. El proceso que se sigue cuando realiza una consulta, es realizar una búsqueda binaria en la lista de palabras del fichero invertido. Si se solicitan varias palabras, los resultados tienen que ser combinados para generar la respuesta final, que será eficiente si las palabras utilizadas no son demasiado frecuentes.

Los ficheros invertidos también pueden apuntar a las ocurrencias de una palabra dentro de un documento (inversión completa), aunque esto es demasiado costoso en cuanto a espacio para la Web, ya que cada puntero tiene que indicar la página y la posición dentro de ésta.

El apuntar a páginas o a posiciones de palabras es un indicador de granularidad del índice. Los índices pueden ser menos densos si apuntan a bloques lógicos en lugar de a páginas. De esta forma se reduce la variación del tamaño de los distintos documentos, haciendo que todos los bloques más o menos tengan el mismo tamaño.

Esto reduce el tamaño de los punteros y el número de ellos y de hecho se ha utilizado en sistemas comerciales como *Glimpse* [Manber y Wu, 1994].

6.4 Directorios Web

El ejemplo más claro de este tipo de directorios, durante años, ha sido Yahoo!, pero ha habido, y hay, muchos otros, con distinto éxito en diferentes países. Hoy día se mezclan habitualmente estos directorios con los buscadores, y muchos de ellos como *AltaVista*, *Lycos*, o *Google* permiten formular consultas pero a la vez permiten indagar en una estructura arborescente de directorios (aunque en realidad es un grafo acíclico dirigido).

Quizá sean más interesantes, por el cometido específico que tienen, los sitios especializados en temas como: negocios, noticias, educación, empleo, entretenimiento, reseñas bibliográficas, etc.

La estructura de los directorios Web es jerárquica, basándose en clasificaciones del conocimiento humano. Las categorías del primer nivel suelen estar entre 12 y 26, pero también se suele ofrecer en el primer nivel las subcategorías del segundo nivel, para agilizar la navegación, apareciendo normalmente más de 70 entradas adicionales.

La ventaja de los directorios es que si encontramos lo que buscamos el resultado nos será útil en la mayoría de los casos. La desventaja es que la clasificación no está especializada suficientemente y que no todas las páginas están clasificadas.

6.5 Búsquedas utilizando hiperenlaces

6.5.1 Lenguajes de consulta Web

Se trataría de poder realizar consultas en las que no sólo interviniera el contenido de las páginas, sino que se pudiera incluir la estructura de los enlaces que conecta las páginas. De esta forma se podrían hacer consultas, por ejemplo de páginas con una imagen, tres enlaces y un vídeo. Para conseguirlo se han tenido que utilizar modelos de datos distintos a los habituales. El más importante utiliza un modelo gráfico etiquetado para representar las páginas y los enlaces (las páginas serían nodos y los enlaces los arcos) y un modelo de datos semiestructurado para representar el contenido de las páginas.

La primera generación de lenguajes de consulta Web tuvo como meta combinar contenido con estructura, utilizaban por una parte patrones que aparecen en los documentos y por otra parte consultas gráficas describiendo la estructura de enlaces. Algunos de ellos eran: *W3QL* [Konopnicki y Shmueli, 1995], *WebSQL* [Mendelzon et al., 1997], *WebLog* [Lakshmanan et al., 1996] y *WQL* [Li et al., 1998].

La segunda generación, denominados lenguajes de manipulación de datos Web, mantenía el énfasis en los datos semiestructurados, pero extendía el modelo proporcionando acceso a la estructura de las páginas Web y permitiendo la creación de nuevas estructuras como resultado de una consulta. En estos lenguajes se encontrarían: STRUQL [Fernández et al., 1997], FLORID [Himmeroder et al., 1997] y WebOQL [Arocena y Mendelzon, 1998].

Tanto los lenguajes de primera, como los de segunda generación han sido utilizados por aplicaciones, pero nunca por el usuario final, aunque algunos de ellos llegaran a presentar una interfaz. Las últimas tendencias han sido extender estos lenguajes para que realizaran otras tareas en la Web, como extraer e integrar información desde las páginas y construir y reestructurar sitios Web.

6.5.2 Búsqueda dinámica y agentes software

La búsqueda dinámica en la Web es equivalente a la búsqueda secuencial en un documento. Se inicia una búsqueda online partiendo de una página y recorriendo sus enlaces. La ventaja es que se utiliza la red real y no los índices almacenados, con los consiguientes problemas de actualización. El inconveniente de este tipo de búsqueda es que es lenta, por lo que se suele utilizar en subconjuntos reducidos de la Web.

La primera heurística diseñada fue la búsqueda pez (*fish search*) [De Bra y Post, 1994] que explota la intuición de que los documentos relevantes tienen, a menudo, vecinos (documentos enlazados) que también son relevantes. Por tanto, la búsqueda es guiada por los enlaces de los documentos relevantes. Fue perfeccionada mediante la búsqueda tiburón (*shark search*) [Hersovici et al., 1998], que da una mejor valoración de la relevancia de las páginas adyacentes. La idea principal de este algoritmo es seguir los enlaces con alguna prioridad, comenzando con una página y una consulta dada. En cada paso se analiza la página de mayor prioridad. Si es relevante, una heurística decide si seguir los enlaces de esa página. Si es así, se añaden nuevas páginas a la lista de prioridades en las posiciones adecuadas.

Algunos trabajos relacionados incluyen agentes para buscar información específica en la Web [Ngu y Wu, 1997] [LaMacchia, 1997]. Esto implica negociar con fuentes de información heterogéneas, que deben ser combinadas. Un punto importante es cómo se determinan las fuentes relevantes y cómo se mezclan los resultados recuperados.

7 CLASIFICACIÓN EN LA RECUPERACIÓN DE DOCUMENTOS

7.1 Introducción

Las áreas de aplicación del *clustering* son de lo más variadas, algunas de las más importantes son: procesamiento de señales en los datos y detección de patrones [Frey y Dueck, 2007], la segmentación de imágenes, la recuperación de la información, la clasificación de documentos, la asociación de reglas en minería de datos, el análisis del tráfico de redes [Erman et al., 2006] y la detección automática de intrusos [Portnoy, 2009], el seguimiento del uso de la Web y el análisis de transacciones. Últimamente también se está aplicando a nuevas tecnologías emergentes como las redes sociales [Handcock et al., 2007], la bioinformática [Pavlopoulos et al., 2009] o el intercambio de datos estructurados en las redes P2P [Kantere et al., 2009].

Nos centraremos en la clasificación de documentos y en la recuperación de la información.

Una secuencia típica del proceso de *clustering* [He et al., 2004] sería:

- Representación de un patrón o modelo, podría incluso incluir extracción de características y/o selección de las mismas.
- Definición de una medida de proximidad en el patrón para el dominio de los datos.
- *Clustering* o agrupación de los puntos de los datos de acuerdo a la representación del patrón elegida y las medidas de proximidad.

La representación del modelo se refiere a la observación del elemento que vamos a tratar, a la abstracción del problema de aprendizaje, incluyendo el tipo, el número y la escala de sus características y el formato de la representación. La selección de las características consistirá en determinar cuál va a ser el subconjunto más representativo para ser usado por el ordenador. Por extracción de características entendemos el proceso de convertir las observaciones de las características naturales en un formato entendible por el ordenador.

El formato de representación es muy importante porque de él puede depender que los *clusters* que se obtengan sean los esperados por el usuario. Los resultados pueden ser diferentes utilizando por ejemplo coordenadas cartesianas o coordenadas polares. Los otros dos aspectos también muy importantes son la selección y extracción de las características ya que afectan a la calidad y a la eficiencia de los resultados. Un conjunto de características superfluas no mejoran la calidad de los resultados y por el contrario aumentan la complejidad computacional del sistema, pero tampoco podremos conformarnos con un conjunto demasiado reducido porque podríamos hacer disminuir la exactitud y producir una pérdida en los *clusters* de salida.

Dado un conjunto de documentos, el *clustering* ha sido utilizado tradicionalmente para agrupar o clasificar los documentos de forma que cada grupo de documentos obtenido represente un tema o tópico. Este planteamiento supone el poder obtener un conjunto de temas que pueden refinarse a distintos niveles y dar una estructura jerárquica en forma de árbol. Pero no siempre es habitual que los documentos puedan clasificarse en un único tema, sobre todo a medida que descendemos en la jerarquía y hacemos un mayor refinamiento, por lo que han sido necesarias otras aproximaciones en las que se han tenido en cuenta el grado de implicación de los documentos en los temas.

En muchas ocasiones no se va a buscar una clasificación perfecta de los documentos en los temas, sino más bien repartir los documentos de colecciones en grupos, de forma que puedan ser aplicados a problemas prácticos como puede ser la agrupación de documentos obtenidos en las búsquedas Web.

Lo habitual es que se asuma que un objeto pertenece a un grupo sólo si está más próximo a otro objeto de ese grupo que a un objeto de otros grupos, como es el caso del *clustering* jerárquico o el particional. Pero no siempre es así, en la práctica se observan ejemplos reales en los que se aprecia que algún documento puede ser más similar a otro documento de otro grupo, que incluso a los de su propio grupo [Steinbach et al., 2000], lo que ha dado lugar a trabajos [Ertöz et al., 2004] que utilizan el número de vecinos próximos que comparte un documento [Jarvis y Patrick, 1973].

La medida que nos indica lo acertado que ha estado el resultado obtenido al formular una consulta, con respecto a un conjunto de documentos, es su similitud

(*similarity*). Los documentos más similares con respecto a la consulta, deberían ocupar una posición más destacada a la hora de presentárselos al usuario. En muchas ocasiones no es sencillo establecer una medida de este tipo debido a lo ambiguas que pueden ser las consultas. Habitualmente la consulta se reduce a una, dos, o como mucho, tres palabras, que en ocasiones pueden tener distintos significados dependiendo del contexto. El usuario fácilmente sería capaz de situarla en el contexto de su interés, luego parece razonable pensar, que una clasificación por temas de los documentos obtenidos, podría mejorar la presentación de los resultados. Sería aquí donde entraría en juego la clasificación (*clustering*). Esta técnica puede ser muy útil, como primera medida, para clasificar grandes cantidades de información que manejan habitualmente los directorios de muchos portales Web.

Se utilizan medidas de similitud para clasificar colecciones de documentos en grupos. Los documentos que pertenecen al mismo grupo tienen un grado de similitud mucho mayor que la que tienen documentos que pertenecen a distintos grupos.

Como el tipo, el rango y el formato de las características de entrada son definidas durante el proceso de representación del modelo, sería en ese momento cuando también habría que elegir el modelo de medida de proximidad que se va a utilizar, ya que para ser una buena medida debería basarse solamente en las características clave de los datos del dominio. Una ayuda suele ser estudiar cuidadosamente las relaciones entre los elementos para determinar una medida de similitud conveniente. La elección será muy importante, porque distintas medidas pueden dar lugar a muy diferentes *clusters* de salida.

Se conoce como hipótesis de clasificación (*cluster hypothesis*) [Rijsbergen, 1979]: dada una clasificación adecuada de una colección de documentos, si un documento es válido para el usuario, es probable que también lo sean otros documentos que pertenecen al mismo grupo de ese documento. Esta hipótesis ha sido verificada experimentalmente [Cutting et al., 1992] [Schütze y Silverstein, 1997] [Zamir y Etzioni, 1999].

7.2 Distintos enfoques

Para analizar los distintos enfoques en la utilización de *clustering*, recogidos en [Chakrabarti, 2003], partimos de una colección de documentos que pueden ser caracterizados por alguna propiedad interna, como podría ser el caso del modelo vectorial, o bien por alguna propiedad externa, por ejemplo una medida de distancia (*dissimilarity*) $\delta(d_1, d_2)$ o una medida de afinidad (*similarity*) $\rho(d_1, d_2)$, especificada entre pares de documentos.

El desarrollo que se va a llevar a cabo en las próximas secciones se resume en la figura 15.

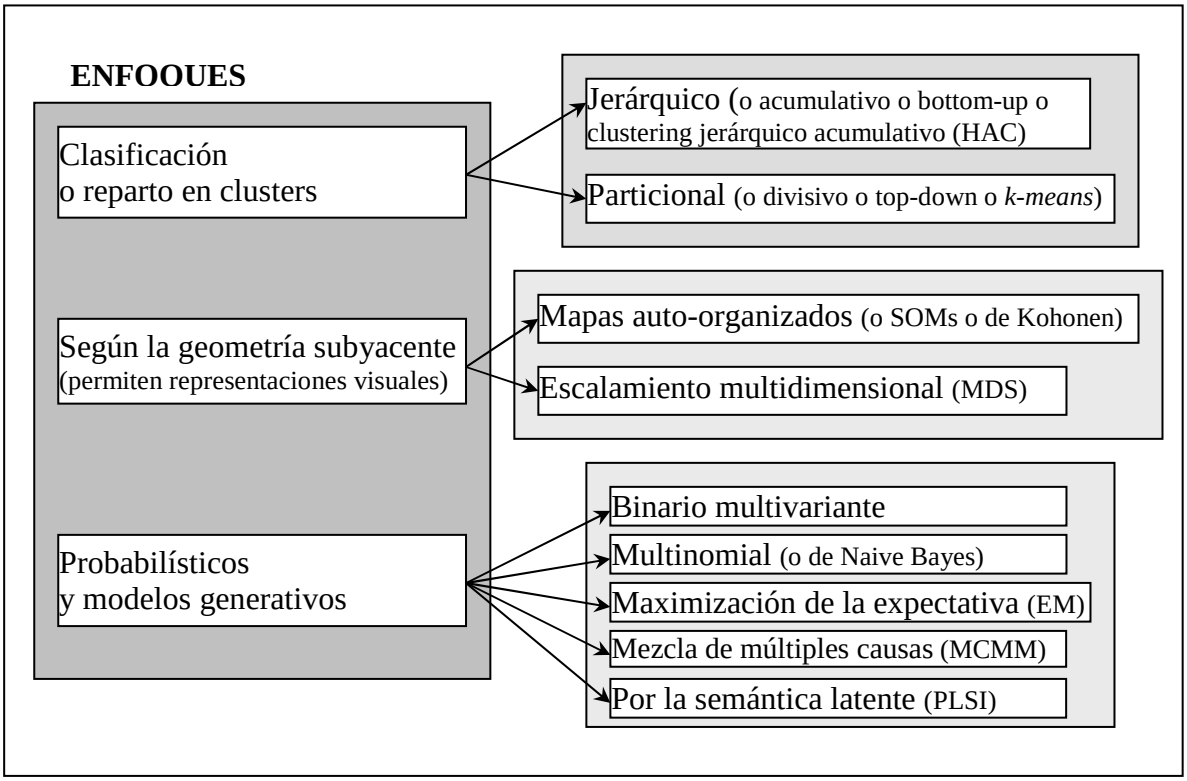

Figura 15. Enfoques en la utilización de *clustering*.

7.2.1 Enfoque de clasificación o reparto en grupos (*clusters*)

Se trataría de dividir un conjunto de documentos *D*, en *k* subconjuntos o *clusters*, $D_1,\ldots, D_k$, de forma que la distancia entre *clusters* sea mínima:

$$\sum_i \sum_{d_1,d_2 \in D_i} \delta(d_1, d_2)$$

o bien la semejanza sea máxima:

$$\sum_i \sum_{d_1,d_2 \in D_i} \rho(d_1, d_2)$$

En el caso de que se disponga de una representación interna de los documentos, como es el caso del modelo vectorial, es habitual especificar los *clusters* con el mismo modelo utilizado. Por ejemplo un *cluster* de documentos podría ser representado por el centroide (término medio) de los vectores de los documentos. En este caso se trataría de minimizar:

$$\sum_i \sum_{d \in D_i} \delta(d, \vec{D}_i)$$

o bien maximizar:

$$\sum_i \sum_{d \in D_i} \rho(d, \vec{D}_i)$$

donde $\vec{D}_i$ es la representación del vector correspondiente al *cluster i*.

7.2.2 Enfoque según la geometría subyacente

La idea consiste en aprovechar los casos en los que existe una clasificación natural de los datos, para trasladar los datos como puntos a espacios de dos o tres dimensiones, de forma que se mantengan las propiedades de clasificación, pero obteniendo mapas que nos faciliten las tareas de clasificación. Una técnica que utiliza esta idea serían los mapas autoorganizados (*self-organizing maps*), que colocan los

clusters en una rejilla de un plano, y va colocando los documentos de forma iterativa en las regiones del plano.

Otra técnica, llamada escalamiento multidimensional (*multidimensional scaling*, *MDS*), utiliza como entradas al sistema, parejas de similitud (*similarity*), disparidad (*dissimilarity*) entre documentos. Mediante un algoritmo se transforman los documentos en puntos del espacio de dos o tres dimensiones, con la mínima distorsión en la distancia de los pares indicados.

Tanto los mapas autoorganizados como el escalamiento multidimensional, por su naturaleza, son heurísticas, por lo que no se garantiza que todas las colecciones puedan ser bien representadas.

La indexación por semántica latente (*latent semantic indexing*), es otra técnica relacionada con el álgebra lineal para descomponer la matriz de términos de los documentos. Los factores pueden ser usados para obtener una representación en una baja dimensión tanto de los documentos como de los términos.

7.2.3 Enfoques probabilísticos y modelos generativos

En las técnicas anteriores, la medida de similitud-disparidad era fijada por el usuario, lo que podía provocar que una mala medida pudiera dañar la calidad de la clasificación. Las técnicas probabilísticas buscan modelar la colección de documentos, siendo generados por un proceso aleatorio, seguido de un conjunto específico de distribuciones.

Una forma de hacerlo podría ser asumiendo que cada *cluster* que buscamos está asociado con una distribución sobre los términos de nuestro léxico. Dada la colección se puede estimar el número de distribuciones y los parámetros que definen estas distribuciones. El problema de clasificación que se planteará será estimar estas distribuciones. Se suele partir considerando que cada documento es generado por exactamente una distribución, si bien en la práctica un tema corresponde a un *cluster*, es difícil que un documento sólo trate de un tema, lo normal será que trate varios.

Estimar una distribución de términos sobre los documentos presenta dificultades. Sobre todo, si queremos considerar la distribución entre términos o entre secuencias de términos debido al gran número de términos en el vocabulario, por eso en muchos casos, se simplifica, y se considera que los términos son independientes.

7.3 Paradigmas de particionamiento jerárquicos: acumulativos (*Bottom-Up*) y particionales (*Top-Down*)

Una primera técnica acumulativa de *clustering* puede consistir en agrupar repetidamente grupos de documentos similares hasta que se logre el número de *clusters* deseados, para pasar a continuación a una técnica descendente que de forma iterativa refine la asignación de los documentos al número de *clusters* programados. Este método es algo lento pero puede ser válido con pocos elementos del corpus, como semilla para más tarde aplicar otro algoritmo.

7.3.1 *Clustering* acumulativo

Consiste en partir del conjunto de todos los documentos e ir combinándolos de forma sucesiva en grupos, de forma que en cada grupo aparezcan los documentos que tengan una gran similitud. Se podrá descender hasta que se obtengan tantos grupos como se deseen. Este método se conoce como acumulativo (*agglomerative*), *bottom-up*, o *clustering* jerárquico acumulativo (*hierarchical agglomerative clustering HAC*). Una revisión de este método junto con una discusión de los algoritmos que se pueden aplicar a bases de datos de tamaño no trivial se recogen en [Willett, 1988].

Esta técnica produce una secuencia anidada de particiones, en la parte superior una única partición que contiene todos los documentos y en la inferior se podría llegar

hasta obtener particiones con un único documento. En el medio se puede ver cada nivel como una combinación de las dos particiones de nivel inferior o bien como una división de la partición superior.

El proceso de combinación jerárquico lleva a un árbol que se suele representar gráficamente mediante un *dendrogram*. Se van uniendo los grupos con mayor similitud, hasta que finalmente se obtiene un único grupo, pero el usuario puede detener el proceso cuando alcanza un número de grupos (*clusters)* deseados.

Se trata de un método que permite una fácil revisión por parte del usuario, que puede recorrer la estructura del árbol y ver la clasificación obtenida con distintos niveles de detalle [Maarek et al., 2000]. Se pueden presentar distintas vistas con diferentes niveles de abstracción, lo que lo hace muy adecuado para exploraciones interactivas y para la visualización. Además en muchos casos, cuando los *clusters* tienen *subclusters*, la estructura jerárquica se ajusta perfectamente al dominio de la aplicación real subyacente.

Una medida de similitud, habitual para unir dos *cluster*, es la *self-similarity*. Se define la *self-similarity* de un grupo de documentos Φ, como la similitud media entre parejas de documentos en Φ.

$$s(\Phi) = \frac{1}{\binom{|\Phi|}{2}} \sum_{d_1, d_2 \in \Phi} s(d_1, d_2) = \frac{2}{|\Phi|(|\Phi|-1)} \sum_{d_1, d_2 \in \Phi} s(d_1, d_2)$$

Donde como valor de *s* $(d_1, d2)$ se toma el coseno de *tf-idf*.

Si los documentos están normalizados en la norma L_2, *s* $(d_1, d2)$ se reduce al producto escalar $\langle d_1, d_2 \rangle$.

Por cada grupo Φ de documentos, se mantiene un vector perfil de grupo no normalizado, $p(\Phi) = \sum_{d \in \Phi} \vec{d}$ que es la suma de los vectores de los documentos que pertenecen al grupo, junto con el número de documentos del grupo.

Se verifica que:

$$s(\Phi) = \frac{\langle p(\Phi), p(\Phi) \rangle - |\Phi|}{|\Phi|(|\Phi|-1)}$$

y

$$p(\Gamma U\Delta) = \left\langle p(\Gamma), p(\Gamma)\right\rangle + \left\langle p(\Delta), p(\Delta)\right\rangle + 2\left\langle p(\Gamma), p(\Delta)\right\rangle$$

por tanto para calcular $s(\Gamma \cup \Delta)$ desde $p(\Gamma)$ y $p(\Delta)$ se reduce a hacer unos pocos productos escalares.

Se asume que los documentos y los vectores perfiles de grupo están inmersos en un espacio de dimensiones fijadas, aunque pueden ser un número elevado, pudiendo llegar a varios miles. El tiempo de cálculo de los productos escalares no es proporcional al número de dimensiones, sino sólo al número de coordenadas que no son cero. Por eso el cálculo de los productos escalares cerca de las hojas de un *dendrogram* sería rápido, pero se haría más lento a medida que nos acercamos a la raíz, porque los vectores de los perfiles se hacen más densos. Los tiempos de ejecución de los algoritmos que se utilizan suelen ser del orden de n^2 o n^3 [Zamir y Etzioni, 1999].

Como solución para reducir el tiempo de computación se truncan los documentos y los vectores de perfiles en un número fijo. Esto puede hacer que la salida de *clusters* producida sea diferente a si se hubiera llegado a la representación total pero hay trabajos, como [Schütze y Silverstein, 1997], que demuestran que la calidad de los *clusters* obtenidos no se ve afectada. Esto significa que la técnica de *clustering* puede funcionar correctamente incluso si los vectores han sido reducidos con considerable pérdida de información, debido a que se considera una tarea de menos grano fino y que requiere menos precisión a la hora de determinar la distancia entre los objetos. Existen sistemas como *Scatter/Gather* [Cutting et al., 1992], para mostrar a los usuarios de los motores de búsqueda la información devuelta de sus consultas clasificada en grupos de documentos acordes a su similitud, haciendo uso de esta técnica y que llegan a iguales conclusiones.

7.3.2 *Clustering* particional (*k-means*)

El *clustering* acumulativo, aplicado directamente, utiliza tiempos cuadráticos y consume espacio por lo que no es muy práctico para grandes colecciones de documentos. Si el usuario puede predefinir un pequeño número *k* de *clusters*, será más

eficiente utilizar una estrategia particional o divisiva (*top-down*). El algoritmo más utilizado para llevar a cabo este planteamiento es el k-medios (*k-means*) [McQueen, 1967]. Su fundamento es minimizar la distancia dentro de los *clusters* de salida en términos de la suma del error cuadrado.

Se pueden hacer dos tipos de asignaciones de documentos a los *clusters*: las "*hard*", que utilizan sólo dos valores (0 ó 1), y las "*soft*", que utilizan un valor fraccionario entre 0 y 1 para determinar la pertenencia a un *cluster*.

La asignación "*hard*" representa los documentos mediante vectores y el *cluster* mediante el centroide (media de los vectores) de los documentos que pertenecen al *cluster*. Se parte de una asignación arbitraria (o guiada por una heurística) de los documentos en k grupos y sus correspondientes centroides, para adentrarse en un proceso iterativo en el que se van cambiando los documentos de *cluster* hacia otros más similares, se vuelven a recalcular los centroides en base a los movimientos, mientras que se produzcan movimientos y hasta que los centroides se muevan distancias despreciables.

La asignación "*soft*", en lugar de asignar los documentos a los *clusters*, representa cada *cluster* mediante un vector. Como no hay asignación explicita de los documentos a los *clusters*, no está directamente relacionada con los documentos. El objetivo es buscar un vector μ_c, por cada *cluster c*, de forma que minimice el error de cuantización, $\sum_d \min_c | d - \mu_c |^2$.

Una de las características de este algoritmo es que puede no ser determinista: evaluará diferentes agrupamientos cada vez que se ejecute y puede obtener distintos resultados [Steinbach et al., 2000]. Por tanto, aun siendo más rápido que los algoritmos jerárquicos, es importante la selección inicial de los centroides [Bradley y Fayyad, 1998], ya que si la selección es mala también lo será la clasificación obtenida.

Se han desarrollado alternativas a este algoritmo como el algoritmo *Bisecting K-Means* [Steinbach et al., 2000], demostrando que se consiguen mejores resultados [Zhao y Karypis, 2001], que los algoritmos jerárquicos y el propio *K-Means*, con tiempos de cálculo lineales con la cantidad de elementos a clasificar. Se han realizado experimentos para estudiar la validez de este tipo de *clustering* midiendo la aproximación de los

resultados obtenidos con respecto a esquemas de clasificación elaborados por expertos, de forma que se comparen los *clusters* y su representación con respecto a las clases [Šilić et al., 2008].

7.3.3 Comparación de los métodos jerárquicos y combinación de métodos

Tradicionalmente muchas de las soluciones de *clustering* jerárquico se han obtenido utilizando algoritmos acumulativos en los que los documentos son inicialmente asignados a sus propios *clusters* que se van mezclando de dos en dos hasta que se forma el árbol completo, algunos de estos algoritmos son: CURE [Guha et al., 1998], ROCK [Guha et al., 1999] o Chameleon [Karypis et al, 1999].

Por otra parte también se han utilizado algoritmos particionales para obtener soluciones de *clustering* jerárquico mediante la repetición de sucesivas divisiones [Jain y Dubes, 1988], [Ng y Han, 1994], o [Boley, 1998].

Son muchos los estudios que han comparado ambas tendencias. En los últimos años varios investigadores [Cutting et al., 1992] [Steinbach et al., 2000], han reconocido que los algoritmos particionales funcionan bien con grandes colecciones de documentos, debido a sus relativamente bajos requerimientos de computación. Si bien la creencia común era, en cuanto a calidad del *clustering*, que los algoritmos particionales son inferiores y menos efectivos que sus opuestos acumulativos. Un estudio muy completo y pormenorizado [Zhao y Karypis, 2002], ya que evalúa nueve algoritmos acumulativos y seis algoritmos particionales, cada uno de ellos sobre doce conjuntos de datos, revela que los métodos particionales funcionan mejor que los acumulativos. Las limitaciones de los métodos acumulativos se deben a los errores que pueden ser introducidos durante las decisiones iniciales de mezcla, especialmente en los casos en los que hay un gran número inicial de alternativas igualmente buenas para cada *cluster*. Al no disponerse de una visión global es difícil en cada caso tomar una decisión acertada. Estos errores iniciales pueden propagarse y multiplicarse en el progreso de unión para generar el árbol jerárquico.

Se han propuesto nuevos métodos, como el "*clustering* acumulativo restringido" (*constrained agglomerative clustering*) [Zhao y Karypis, 2002], que busca mejores resultados tomando las ventajas y desventajas de los dos métodos anteriores. La característica de los métodos particionales es que utilizan información sobre la colección de documentos entera, cuando quieren hacer divisiones en *clusters*, mientras que los acumulativos tienen un punto de vista local. Pero esto a la vez puede presentar ventajas y desventajas dependiendo de las características de los documentos. Si por ejemplo los documentos forman pequeños y razonables grupos cohesivos será sencillo agruparlos pero sin embargo los algoritmos particionales podrían fallar si las colecciones son grandes y se producen divisiones demasiado tempranas.

El planteamiento del método de *clustering* acumulativo restringido es combinar ambos métodos. Se utiliza el *clustering* particional para construir el espacio sobre el que se tomarán las decisiones acumulativas, de forma que cada documento sólo podrá mezclarse con otros documentos que formen parte de su mismo *cluster*. El primer paso será aplicar un algoritmo particional para obtener un número *k* de soluciones o caminos. Cada uno de estos *clusters* obtenidos es tratado por separado como si se tratara de una colección, aplicando un algoritmo acumulativo y obteniendo su correspondiente árbol jerárquico. Finalmente los *k* árboles obtenidos son mezclados en un único árbol con un algoritmo acumulativo tratando los documentos de cada subárbol como un *cluster* que ya ha sido formado durante la unión. Con esto se consiguen obtener las ventajas de la vista global pero también del punto de vista local, pero además se mejora la complejidad algorítmica. Los resultados experimentales muestran que este método siempre mejora las soluciones de los métodos sólo acumulativos sin importar el algoritmo particional que se utilice para obtener los *clusters* intermedios y en muchos casos supera a su correspondiente algoritmo particional.

Otro de los objetivos que se persigue en ocasiones es preservar la privacidad de los clusters, ya que a veces en una organización resulta interesante la utilización de datos de distintas fuentes a fin de producir un conocimiento más preciso y útil, lo que conlleva conflictos sobre la privacidad y preservación de la información. En estos casos se busca poder llevar a cabo una extracción de conocimiento sin desvelar los datos individuales, necesitando utilizar algoritmos más complejos o variantes de los protocolos [Jagannathan y Wright, 2005].

En ocasiones se recurre a una primera aproximación obteniendo un conjunto de *clusters*, que después se refinan modificando su contenido, o bien se vuelven aplicar técnicas de *cluster* a las categorías obtenidas, mediante técnicas de agregación de clusters (*clustering aggregation*), utilizando algoritmos evolucionados, conformando un método de meta-cluster [Gionis et al., 2007].

7.4 Métodos de *clustering* que permiten representaciones visuales

Los métodos tratados en el apartado anterior no presentan mecanismos para poder representar de forma visual los *clusters*, y este aspecto, de visualización de los resultados, es importante en la recuperación de la información. A continuación se presentan otros métodos que permiten la representación de los documentos como puntos en un espacio de dimensiones determinado, normalmente dos o como mucho tres, de forma que puedan ser visualizados directamente.

7.4.1 Mapas auto-organizados (*SOMs*)

Se denominan mapas auto-organizados (*self-organizing maps*) o mapas de Kohonen [Kohonen, 1997]. Se trata de una familia de redes neuronales auto-organizadas que se ha utilizado extensamente para el *clustering* y la visualización. Se aproximan mucho al planteamiento del algoritmo k-medios, pero en lugar de preocuparse sólo de asociar documentos a *clusters*, SOMs lo que hace es encajar los *clusters* desde el principio en un espacio de pocas dimensiones, colocando próximos entre sí los que guardan relación. El mapa se va formando asociando un vector μ_c con cada *cluster c*, y dicho vector se irá refinando de forma iterativa. Además cada *cluster* se representará como un punto en el espacio, o un nodo en una cuadrícula.

Durante el aprendizaje la red actualizará no sólo los pesos de los ganadores sino también los pesos de sus vecinos ganadores. Para cada *cluster* se define su vecindad *N(c)*, de forma que pueda contener los nodos que estén a dos pasos de c. También se define una función de proximidad $h(\gamma, c)$, que nos indica la distancia entre un nodo γ y otro *c*. Siendo $h(c, c) = 1$, e irá decayendo este valor a medida que aumente la distancia entre los nodos. Cuando sea 0 entonces $\gamma \notin N(c)$.

Los SOM son una clase de redes neuronales donde los ítems de datos que representan los documentos activan la neurona asociada al documento y algunas otras relacionadas por su vecindad. El algoritmo global inicializa todos los vectores con valores aleatorios y repetidamente escoge un documento aleatorio de la colección y actualiza el modelo de cada neurona hasta que el modelo de los vectores deja de cambiar de forma significativa.

Hay gran número de variaciones sobre este algoritmo, principalmente determinadas por la dimensión del mapa, la definición del conjunto de vecinos, la definición de la función de proximidad y el paradigma que se utilice para el reajuste.

El inconveniente de este método es que su aprendizaje se ve afectado por los valores de inicialización de la red y por el orden de presentación de los datos de entrada.

7.4.2 Escalamiento multidimensional (*MDS*)

En los casos anteriores, del algoritmo k-medios y de SOM, los documentos se representaban mediante el modelo vectorial, pero en otras aplicaciones los documentos pueden ser caracterizados sólo por la distancia a otros documentos. Incluso en los casos en los que se dispone de una representación interna, puede usarse para generar distancias entre pares de documentos. Hacer esto puede ayudar a incorporar realimentación por parte del usuario, utilizando una medida de distancia que sustituye a la calculada por la representación interna.

El objetivo de MDS (*multidimensional scaling*), es representar los documentos como puntos en espacios de dos o tres dimensiones, de forma que la distancia euclídea

entre cada par de puntos se acerque, en la medida de lo posible, a la distancia entre ellos especificada por la entrada. Si $d_{i,j}$ es una medida de distancia o disimilaridad, definida por el usuario, entre los documentos *i* y *j*, y $\hat{d}_{i,j}$ la distancia euclídea entre los puntos que representan esos documentos dados por el algoritmo MDS, se define como

$$stress = \frac{\sum_{i,j}(\hat{d}_{i,j} - d_{i,j})^2}{\sum_{i,j} d_{i,j}^2}$$

Se pretende minimizar *stress*, aunque no es sencillo. Se comienza asignando aleatoriamente, o mediante alguna heurística, coordenadas a todos los puntos. A continuación se van moviendo los puntos una pequeña distancia en una dirección de forma que se reduzca *stress*. Si comenzamos con n puntos, el proceso implica un tiempo de computación de orden *n* para mover cada punto, y de orden n^2 para cada iteración. Existe una variación llamada FastMap [Faloutsos y Lin, 1995] que consigue mejores tiempos. Para ello pretende que los documentos sean verdaderos puntos en algún espacio multidimensional con bastantes dimensiones para buscar una proyección en un espacio con una dimensión reducida *k*, consiguiendo un tiempo de computación de *nk*.

7.5 Aproximación probabilística al *clustering*

Aunque la representación del modelo vectorial ha sido muy utilizada en recuperación de la información, en el caso del *clustering* deja algunos problemas sin resolver. En el caso, por ejemplo, del *clustering* jerárquico acumulativo (HAC), el documento y el perfil de los vectores del grupo eran determinados mediante el valor *idf* (inverso de la frecuencia en el documento), antes del proceso acumulativo, cuando quizá fuera más lógico calcular *idf* después, ya con $\Gamma \cup \Delta$.

Dado un corpus con varios temas, los documentos probablemente incluirán términos muy indicativos de uno de los temas, junto con otros términos muy ruidosos seleccionados de un conjunto de palabras vacías. Por ello una mejor función de *idf* sería la que minimizara las palabras ruidosas, aunque obtendríamos el mismo efecto si

distinguiéramos que un documento está compuesto por distribuciones separadas y aplicáramos similitud sólo a los términos no ruidosos.

Se van a estudiar procesos aleatorios que generan los documentos y caracterizan el *clustering* descubriendo procesos y los parámetros asociados que son más probables hayan sido generados por una colección dada de documentos. Varios aspectos deseados son:

- No se necesita *idf* para determinar la importancia de un término.
- Algunos de los modelos que se estudiarán pueden directamente capturar la idea de palabras vacías frente a palabras con contenido relevante.
- No es necesario definir distancias o similitud entre entidades.
- La asignación de entradas a los *cluster* no es "*hard*", sino probabilística.

7.5.1 Distribuciones generativas de los documentos

El reconocimiento estadístico de patrones y los algoritmos de recuperación de la información son dados con la premisa de que los modelos que observamos son generados por procesos aleatorios que siguen distribuciones específicas. La observación nos permite especificar parámetros relativos a esas distribuciones que nos facilitan las operaciones de clasificación o indexación, pero resulta complicado aplicarlas en el terreno del lenguaje natural, y los algoritmos que se obtienen consumen mucho tiempo de computación, por lo que la técnica habitual es reducir aspectos de los datos observados. Se suele prescindir, sobre todo, de las dependencias y de la ordenación entre términos.

Suponiendo drásticamente que los términos son independientes y además que no importa el número de veces que aparece un término, podemos utilizar una variable que tome valores 0/1. A este modelo se le conoce como binario multivariante (*multivariate binary model*), o simplemente modelo binario (*binary model*), pero existen variaciones del mismo que utilizan más niveles [Booth et al., 2008]. Un documento se representará

por un vector con un bit (0/1) por cada término en el vocabulario *W*. El bit correspondiente a un término se activará con probabilidad ϕ_t , y se desactivará con probabilidad $1-\phi_t$. Todos los ϕ_t son reunidos en el conjunto de parámetros del modelo, llamado Φ.

Dado Φ, la probabilidad de generar un documento *d* viene dada por:

$$\Pr(d \mid \Phi) = \prod_{t \in d} \phi_{c,t} \prod_{t \in W, t \notin d} (1 - \phi_{c,t})$$

Donde habitualmente $|W| >> |d|$, por lo que los pequeños documentos son despreciados. Pero además si asumimos que $0 < \phi_t < 1$, todos los documentos $2^{|W|}$, tendrían probabilidad positiva, cosa improbable en la vida real.

Otro modelo que intenta paliar estas consecuencias es el modelo multinomial (*multinomial model*), o de Naive Bayes, utilizado en algunas ocasiones para modelos no supervisados con realimentación [Cohn et al., 2008]. Primero se decidirá el número total de términos *L* del documento *d* que van a ser generados desde la correspondiente distribución Pr (*l*), suponiendo que el evento instanciado es l_d. Los documentos son generados mediante palabra-eventos, usando un dado con |W| lados, un lado por cada término del vocabulario. Una vez lanzado el dado la probabilidad de que salga la cara correspondiente al término *t* es $\theta_t (\sum_t \theta_t = 1)$. Se representan por Θ, todos los parámetros necesarios para obtener la longitud de la distribución y todos los $\theta_t s$. El dado se lanza l_d veces y se anotan los términos que van apareciendo. Suponiendo que el término *t* aparece *n (d, t)* veces, con $\sum_\tau n(d,\tau) = l_d$. El evento del documento en este caso consta de l_d y el conjunto de sumas *{n (d, t)}*. La probabilidad de este conjunto de eventos viene dada por:

$$\Pr(l_d, \{n(d,t)\} \mid \Theta) = \Pr(L = l_d \mid \Theta) \Pr(\{n(d,t\} \mid l_d, \Theta) = \Pr(L = l_d \mid \Theta) \binom{l_d}{\{n(d,t)\}} \prod_{t \in d} \theta_t^{n(d,t)}$$

donde $\binom{l_d}{\{n(d,t)\}} = \dfrac{l_d!}{n(d,t_1)!\,n(d,t_2)!\ldots}$ es el coeficiente multinomial.

Este modelo tampoco tiene en cuenta la dependencia de que un término ocurra con otro. Aun así, este modelo, conservando el número de veces que aparece cada término, resulta ser algo mejor para la mayoría de las tareas de minería de texto.

7.5.2 Los modelos mezcla y maximización de la expectativa (*EM*)

Dada una colección de documentos, por ejemplo obtenidos de la Web, es posible estimar Θ_{Web} para esa colección y calcular la probabilidad $\Pr(d \mid \Theta_{Web})$ de todos los documentos *d* con respecto a Θ_{Web}, pero no es suficiente un modelo multinomial para la Web entera. Si nos dieran una serie de temas por adelantado, como arte, ciencias y política, podríamos determinar a qué grupo pertenece cada documento. En lugar de estimar Θ_{Web}, podríamos estimar de forma más especializada, $\Theta_{arte}, \Theta_{ciencias}, \Theta_{política}$ y para un documento del tipo γ, evaluar su $\Pr(d \mid \Theta_{\gamma})$. Parece que esto sería más largo porque se podría determinar qué términos son raros o frecuentes para un tema y compararlos con un documento aleatorio de la red. Esta sería la esencia de la mezcla de modelos para la generación de documentos. Supondríamos que hay *m* temas (o componentes o *clusters*), que el autor de una página tendría que determinar. Se podría hacer usando una distribución selectora *m-way* multinomial, con probabilidades $\alpha_1,...,\alpha_m$, donde $\alpha_1 + ... + \alpha_m = 1$. Una vez que un tema γ es decidido, se utiliza Θ_{γ}, como distribución para ese tema para generar el documento. Se pueden usar distribuciones binomiales o multinomiales para cada término, e incluso diferentes distribuciones para diferentes componentes.

Para simplificar la notación se engloban en Θ, los parámetros $\theta_{y,t}$, uno por cada término *t,* los $\alpha_i s$, y también el número de temas *m*. Es decir:

$$\Theta = (m; \alpha_1,...,\alpha_m; \{\theta_{\gamma,t} \forall \gamma, t\})$$

El algoritmo completo, recibe el nombre de maximización de la expectativa (*expectation maximization*) (EM). Buscar un valor adecuado de *m*, el número de temas o

cluster, no es una tarea trivial. Para algunas aplicaciones puede ser un valor conocido e incluso conocer los documentos que corresponden a un *cluster*. Sería el caso en el que de forma manual se asignan algunos documentos a sus correspondientes *clusters*, es lo que se conoce como aprendizaje semisupervisado [Haldiki et al., 2008] [Kulis et al., 2009]. Cuando *m* no está especificado hay dos formas de obtenerlo. La primera consiste en dejar fuera algunos de los datos, construir el modelo mezcla del resto, y buscar la probabilidad de que los datos que se han quedado fuera den los parámetros mezcla. Este proceso se repetirá mientras aumente el número de *clusters* hasta que la probabilidad deje de crecer. Si no se dejaran datos de entrenamiento fuera, el sistema podría generar un excesivo valor de *m*, que se conoce como sobre adaptación (*overfitting*), estudiado junto con la determinación del valor *m* por [Smyth, 1996].

La segunda forma de obtener *m* sería restringir la complejidad del modelo utilizando una distribución anterior sobre los parámetros del modelo que haga improbables los modelos complejos.

Uno de los inconvenientes que presenta el modelo mezcla es el no considerar que muchos documentos son relevantes para varios temas o *clusters*.

7.5.3 Modelo mezcla de múltiples causas (*MCMM*)

Si un documento trata sobre un tema, pertenece a un *cluster*, tendrá asociadas una serie de palabras características relacionadas con ese tema cuya probabilidad de aparecer será mayor que la del resto de palabras. Si *c* es el tema o *cluster* y *t* los términos asociados, $\gamma_{c,t} (0 \le \gamma_{c,t} \le 1)$ será una medida normalizada de que los términos *t* estén en el *cluster c*. Suponiendo que $a_{d,c} (0 \le a_{d,c} \le 1)$ representa cómo es activado el *cluster c* escribiendo un documento *d*, entonces se cree que los términos *t* aparecerán en el documento *d* por una disyunción *soft* (*soft disjunction*), también llamada *noisy OR*:

$$b_{d,t} = 1 - \prod_{c} (1 - a_{d,c} \gamma_{c,t})$$

Esto indica que los términos no aparecen sólo si no son activados por alguna de las clases consideradas. Permite que el documento *d* sea representado por el modelo binario *n (d, t),* el número de veces que los términos *t* aparecen en el documento, es decir 0 ó 1.

Este modelo es simple y flexible, por lo que puede ser utilizado tanto en aprendizajes de tipo supervisado como no supervisado. El inconveniente que presenta es su velocidad, ya que la representación de la matriz de acoplamiento es densa y su ascenso es lento. Para hacerse una idea, con unos cientos de términos, unos miles de documentos y alrededor de 10 *clusters* se tardarían unos minutos con un modelo supervisado, mientras que con uno no supervisado se necesitarían horas.

7.5.4 Modelo probabilístico de indexación por la semántica latente (*PLSI*)

Se trata de un modelo generativo [Hofmann, 1999], utilizado cuando los documentos corresponden a varios temas. Se parte de una colección de documentos de la que se conoce el número total de términos representada mediante una matriz en la que cada término viene dado por *n(d,t),* que representa la frecuencia del término *t* en el documento *d.* Por otra parte cada par *(d, t)* tiene asociado un evento binario. El número de veces que ocurre el evento viene dado por el dato *n (d, t).* El total de todos los contadores de eventos se fija de antemano, y los eventos *(d, t)* deben distribuir este número total.

Cuando un autor redacta un documento induce una distribución de probabilidad Pr(*c*) sobre los temas o *clusters*. Diferentes *clusters* causan eventos *(d, t)* con probabilidades diferentes. Para obtener la probabilidad total Pr (*d, t*), se suma la de todos los *clusters*:

$$\Pr(d,t) = \sum_{c} \Pr(c) \Pr(d,t \mid c)$$

La aproximación principal en este modelo es asumir la independencia condicional entre *d* y *t*, dado *c*, quedando:

$$\Pr(d,t) = \sum_c \Pr(c)\Pr(d \mid c)\Pr(t \mid c)$$

7.6 Filtrado colaborativo (*CF*)

En lugar de preocuparnos de clasificar documentos en función de sus términos, podemos utilizar la relación entre documentos y términos en el sentido de poder desear clasificar términos basados en los documentos en los que aparecen. En términos generales siempre que tengamos una fuente de datos bipartita, agrupando dos clases de entidades, se pueden utilizar técnicas parecidas a EM, LSI O PLSI para construir modelos que se ajusten a los datos. Una forma interesante de datos bivalentes son las preferencias de los usuarios. Las personas forman un conjunto de entidades y los ítems de lo que les gusta o disgusta forman el otro. En lugar de la relación "documento contiene término", utilizamos "a las personas le gusta ítem".

La entrada al sistema es una matriz *Y*, que no está completa, en la que en las filas se representan las personas y en las columnas los ítems que les gustan, supongamos por ejemplo películas. Las entradas que tienen valores asignados Y_{ij}, tendrán un 1 ó un 0, indicando si a una persona le gusta una película o no. El aspecto central del filtrado colaborativo (CF) es, dada una nueva fila (persona) con unas cuantas entradas disponibles (películas que le gustan), ¿podemos determinar de forma eficiente y con precisión el resto de valores (columnas) omitidos, utilizando el resto de las experiencias de las demás personas recogidas en la matriz? Si fuera así, nos permitiría seleccionarle las películas con las que disfrutaría.

Los primeros sistemas de filtrado colaborativo representaban cada persona como un vector con sus preferencias sobre las películas, y clasificaban a la gente utilizando métodos ya vistos, como k-means, pero no utilizaban la simetría en la relación entre la gente y los ítems. Este nuevo modelo que se presenta extiende una mezcla de modelos con dos tipos de entidades:

- Permitir que la matriz *Y* tenga *m* personas y *n* películas. Se asume que cada persona puede ser clasificada en *m' clusters* y las películas en *n' clusters*.
- La probabilidad de que una persona aleatoria pertenezca al *cluster* de personas *i'* es $p_{i'}$.
- La probabilidad de que una película aleatoria pertenezca al *cluster* de las películas *j'* es $p_{j'}$.
- La probabilidad de que una persona pertenezca al *cluster* de las personas *i'* y que le guste una película del *cluster* de películas *j'*, es $p_{i'j'}$.

Estos parámetros pueden ser estimados utilizando el método de Maximización de la Expectativa (EM) o de Monte Carlo. La estimación de $\pi_{i\to i'}(\pi_{j\to j'})$ depende de las estimaciones actuales de pi', pj', pi'j' y la fila (columna) correspondiente a la entrada seleccionada. Si consideramos el caso en el que la persona i ha sido escogida aleatoriamente, correspondiendo a la fila i de la matriz y consideramos la película correspondiente a la columna j en esta fila. Suponemos que la película j está asignada al *cluster* de películas $\ell(j)$. Entonces $\pi_{i\to i'}$ puede ser estimada como:

$$\pi_{i\to i'} = p_{i'}\prod_j \begin{cases} p_{i'\ell(j)} & if \quad Y_{i,j}=1 \\ 1-p_{i'\ell(j)} & if \quad Y_{i,j}=0 \end{cases}$$

Una fórmula simétrica se utiliza si una película es seleccionada aleatoriamente en lugar de una persona. Utilizando esta asignación de personas y películas a los *clusters*, es relativamente sencillo refinar las estimaciones de $p_{i'}$, $p_{j'}$ y $p_{i'j'}$ para todos los *i', j'*.

7.7 Otras técnicas aplicadas al *clustering* de documentos

7.7.1 *Clustering* de documentos basado en enlaces

Esta metodología [Noel et al., 2004], utiliza una nueva clase de distancia entre documentos que trata de utilizar la información inherente en la estructura de enlaces de una colección. Las distancias se calculan a través de los procesos de asociación que surgen de la identificación de los conjuntos de ítems (*itemsets*) que son enlazados de forma conjunta a menudo. Si consideramos que un enlace equivaldría en el terreno literario a una citación, los conjuntos de ítems habría que entenderlos, en un orden superior, haciendo referencia a citaciones conjuntas o co-citaciones [Small, 1973]. Una pareja de documentos estaría relacionada, si ambos son citados de forma conjunta por otros documentos. Tradicionalmente el análisis de documentos basados en este tipo de enlaces ha constituido una medida de similitud para el *clustering*. El objetivo que se buscaba era encontrar grandes colecciones de documentos que se correspondieran con campos individuales de estudio.

Esta técnica presenta la ventaja de permitir una rápida representación de los enlaces mediante grafos y una rápida formulación en forma de matriz binaria de adyacencia. Las filas se corresponderían con los documentos que citan a otros y las columnas con los documentos que son citados. Un elemento $a_{ij} = 1$ indica que el documento *i* cita al documento *j*.

Se conoce como contador o recuento de citaciones (*citation count*), de una pareja dada de documentos, al número de documentos que los citan de forma conjunta, y este valor se utiliza como medida de similitud. Utilizando la matriz de adyacencia *A*, el contador de citaciones será una cantidad calculada para cada pareja de columnas de la matriz, y se define como:

$$c_{j,k} = \sum_i a_{i,j} a_{i,k} = a_j \bullet a_k = A^T A$$

donde a_j y a_k son vectores columna de A, *i* son las filas indexadas y A^T la transpuesta de *A*. Estos valores se pueden normalizar haciendo que tomen valores en el intervalo [0,1], y se pueden convertir en disimilaridad.

El análisis de los enlaces simples en el *clustering* de documentos se utiliza por su baja complejidad computacional en grandes colecciones de documentos, pero uno de sus inconvenientes es el posible efecto en cadena que se puede producir cuando documentos que no están relacionados se agrupan en el mismo *cluster* debido a los enlaces con algún documento intermedio. Por ello se han desarrollado criterios alternativos más fiables como son el encadenamiento o enlazado medio y el encadenamiento completo. Estos criterios se aplican en una heurística acumulativa en la que se van mezclando los *cluster* con una distancia más corta entre ellos En el caso de los enlaces simples, esta distancia entre dos *clusters* es la menor distancia posible entre objetos en *clusters* separados. Para el encadenamiento medio, la distancia es la media entre las distancias entre objetos de *clusters* separados. Para el encadenamiento completo, la distancia es la mayor distancia entre objetos en *clusters* separados. Estos tres tipos de encadenamiento dan lugar a tres criterios de *clustering*: débil, intermedio y fuerte.

Tradicionalmente se ha aplicado el *clustering* débil, por su baja complejidad computacional, pero últimamente debido al desarrollo de los equipos, se hace factible aplicar criterios de *clustering* fuerte en el análisis de citaciones. En la metodología de [Noel et al., 2004] utilizan un criterio particularmente fuerte tomado de la minería de asociación. Incluye esencialmente las co-citaciones de orden superior, es decir, entre conjuntos de documentos de cardinalidad arbitraria. Esto añade el beneficio de desarrollar un tipo de *clustering* orientado al usuario, ya que es el usuario quien proporciona una retroalimentación iterativa para ayudar a guiar el proceso, basado en el conocimiento del dominio de la aplicación. Con la distancia entre parejas, el usuario puede orientar el *clustering* con pesos sobre las distancias para varios pares de documentos, aplicando mayores pesos a los pares cuya similitud es más importante. Se ha intentado que la asignación de estos pesos se pueda llevar a cabo de forma automatizada, mediante la utilización de nuevas formulas y mediante la variación de los pesos en las diferentes iteraciones, a medida que se van produciendo *clusters* [Huang et al., 2005], [Andersen et al, 2007]. Se han estudiado algunas variaciones en dos partes, en las que primero se determinan los núcleos o centroides de clusters y después se asignan nodos a los clusters, consiguiendo mejores tiempos de computación [Avrachenkov et al., 2008].

7.7.2 *Clustering* de consultas en el contexto de la Web

El *clustering* de consultas es una clase de técnicas que intentan agrupar consultas en un repositorio, relacionadas con la semántica de los usuarios, que se completa con la interacción entre los usuarios y el sistema. Este tipo de técnicas se ve impulsado por los requerimientos de las modernas búsquedas web. Las principales aplicaciones del *clustering* de consultas [Wen y Zhang, 2004] son:

- **Detección de preguntas frecuentes (FAQs)**. Al contrario que los sistemas de búsqueda tradicionales, estos nuevos sistemas tratan de entender las preguntas de los usuarios para sugerir preguntas similares que otros usuarios han planteado y que el sistema tiene ya la respuesta. En la mayoría de los casos estas respuestas han sido preparadas y chequeadas por editores humanos. Por tanto lo que se necesitan son herramientas automáticas que ayuden a los editores a identificar las preguntas, y aquí es donde interviene el *clustering*, ya que se trata de agrupar preguntas. Las dificultades son muchas y de lo más variadas, como preguntas mal formuladas, o como preguntas similares formuladas con palabras muy distintas.
- **Elección de términos índice**. Un problema constatado es la elección de los términos índice de forma que representen el contenido del documento, ya que muchos motores de búsqueda todavía sólo cuentan con las palabras clave contenidas en las páginas web para construir los índices. Este es uno de los factores clave que afecta a la precisión de los motores de búsqueda. En muchas ocasiones se produce una desigualdad entre las palabras utilizadas en la consulta y las que contiene el documento por lo que no se obtienen buenos resultados. Una forma de paliarlo es el *clustering* de consultas, que trata de agrupar en un mismo *cluster* consultas similares que después se utilizan como fuente para seleccionar los términos índice para los documentos.
- **Reformulación de consultas**. Son muchas las dificultades que presenta la Web en cuanto a la formulación de las consultas. Se parte de multitud de

páginas creadas por diferentes autores, que utilizan los más variados vocabularios, pero además el propio lenguaje presenta ambigüedades en cuanto al significado de las palabras. Por ello muchos de los buscadores intentan identificar las intenciones de los usuarios y sugieren listas de términos alternativos para que se reformule la consulta, por supuesto haciendo uso de técnicas de *clustering*.

El principal problema del *clustering* de consultas es determinar una función de similitud adecuada para que realmente agrupe adecuadamente las consultas mediante algún algoritmo de *clustering*. Básicamente hay dos categorías o métodos para calcular la similitud entre consultas: una está basada en el contenido de las consultas y la otra en la sesión de las consultas. Se investigan variantes, como tener en cuenta la medida de la autoridad de los sitios web, añadiéndola como una característica a tener en cuenta [Liu et al., 2009].

Es mucho el interés que los buscadores comerciales tienen en este tipo de *clustering*, ya que puede contribuir a ofrecer mejores resultados al usuario final [Mateos et al., 2008]. Se han llevado a cabo estudios que analizan la incidencia de su utilización frente a métodos tradicionales mediante el desarrollo de herramientas web que presentan los resultados de las consultas en forma de árbol navegable [Ferragina y Gulli, 2005] [Mateos y García-Figuerola, 2007]. Se han desarrollado también buscadores especializados, o herramientas de indexación por determinados campos, como pueden ser los autores de los documentos, de forma que se puedan obtener de forma precisa referencias y artículos de un determinado autor, evitando las posibles variantes del nombre de un mismo autor o las abreviaturas en alguno de sus valores [Han et al., 2005].

7.7.2.1 Clustering de consultas basado en el contenido

Se trata del punto de vista clásico en recuperación de la información: si dos consultas tienen los mismos, o similares términos, quiere decir que precisan de la misma, o similar información. Sin embargo este planteamiento presenta desventajas bien conocidas derivadas de las limitaciones de las palabras clave, sobre todo cuando las consultas son muy cortas, cosa bastante habitual.

Dependiendo de las diferentes formas de representar las consultas —palabras clave, palabras en su orden, y frases— se aplican distintas medidas de similitud.

Similitud basada en palabras clave o en frases. Se trata del caso más habitual en Recuperación de la Información. Los documentos se representan por vectores en un espacio vectorial compuesto por todas las palabras clave y la medida típica de similitud es el coseno, como ya se ha descrito ampliamente en el apartado 3.1.2. Esta medida puede ser fácilmente extendida a las frases, ya que el significado de una frase es más preciso que el de una única palabra. Si se consigue identificar frases en una consulta, podemos conseguir un cálculo más preciso de la similitud. Se puede intentar mediante dos métodos. El primero sería utilizando un reconocedor de sustantivos en las frases basado en reglas sintácticas y estadísticas. El otro sería utilizar un diccionario de frases. En cualquier caso, como las consultas suelen ser muy cortas, resulta difícil deducir la semántica únicamente de las frases, por lo que se trata de métodos que no proporcionan una base fiable en el *clustering* de consultas.

Similitud basada en las operaciones, a nivel de palabra, para hacer coincidir cadenas. Muchas de las medidas basadas en términos clave descartan muchos términos clasificándolos como palabras vacías, que sin embargo pueden aportar mucha información en el caso particular de las consultas, en el que se trabaja con muy pocos términos. En estos casos se deberían tener en cuenta todos los términos y utilizar una medida de similitud que se aplicara a todos los términos. Esta medida es el número de operaciones de edición (inserción, borrado o sustitución de palabras), para unificar o hacer coincidir dos cadenas (consultas). Para normalizar este valor, en el rango [0, 1], se utiliza el máximo de los números de palabras (*np*) de las consultas:

$$similaridad_{edición}(q1, q2) = 1 - \frac{dist\_edic(q1, q2)}{\max(np(q1), np(q2))}$$

La ventaja de este método es que tiene en cuenta, por una parte, el orden de las palabras, y por otra parte, determinadas palabras —que antes se descartaban como palabras vacías— que nos permiten clasificar el tipo de consulta cuando aparece una de estas palabras, como “dónde”, “porqué” o “quién”.

Similitud basada en las operaciones, a nivel carácter, para hacer coincidir cadenas. Es frecuente que se produzcan errores ortográficos en las consultas, bien

porque no se conozca bien el término o bien a la hora de teclear las palabras. En estos casos el *clustering* de consultas se utiliza para detectar y corregir esta clase de errores ortográficos. La medida de similitud entre dos cadenas (consultas) es la del caso anterior pero en este caso el número de operaciones de edición (inserción, borrado, o sustitución) se refieren a caracteres en lugar de a palabras.

7.7.2.2 Clustering de consultas basado en la sesión

Debido a la corta longitud de las consultas y la ambigüedad de las palabras que se suelen emplear, los métodos de *clustering* basados en el contenido normalmente no agrupan de forma clara las consultas por su semántica. Para obtener mejores resultados, la idea es extender el concepto de consulta al concepto de sesión de la consulta, entendiéndose por tal, a la consulta y las acciones que el usuario lleva a cabo a continuación, es decir, las páginas web visitadas en respuesta a su solicitud, ya que se considera que estas acciones forman parte de la semántica de la consulta y permiten eliminar la ambigüedad. Estas sesiones pueden obtenerse de los registros de anotaciones de los motores de búsqueda, que permiten conocer la consulta formulada y cuáles han sido las páginas que el usuario ha seleccionado, luego contamos con una información adicional, eso sí, suponiendo que las actividades que se desarrollan después de la consulta son relevantes para la consulta.

La extracción de la sesión de la consulta lleva dos pasos: la identificación del usuario y la identificación de la sesión. La identificación de usuario consiste en aislar de los registros de anotaciones las actividades asociadas con ese usuario. La forma más sencilla sería mediante la IP, pero no siempre sería efectivo (*cachés*, *proxys*, *firewalls*) por lo que en ocasiones se recurre a distintas heurísticas [Cooley et al., 1999] [Pirolli et al., 1996]. Otra solución adoptada es la utilización de cookies, aunque también presentan algunos inconvenientes, como el que pueden ser borradas. Un método más seguro es la autentificación del usuario, pero este proceso a veces es rechazado por usuarios recelosos. El segundo paso del proceso es la identificación de la sesión, que consistiría en separar todas las operaciones que ha realizado el usuario que ya tenemos definido. El inicio de la sesión no presenta dificultades porque es la primera operación que ha realizado, consistente en lanzar una consulta al navegador. El fin de sesión es

más difícil de determinar, uno de los métodos más utilizados consiste en asignar un tiempo máximo, que si se sobrepasa se considera que la sesión ha concluido y ha comenzado otra.

La función de similitud basada en sesión, más simple, es utilizar las páginas en las que ha hecho clic como descripción de la semántica de las consultas del usuario y considerar cada página aisladamente. Se basa en el principio de que: si los usuarios hacen clic en las mismas páginas web para diferentes consultas es porque estas consultas son similares. Si $D(q_1)$ es el conjunto de páginas que el sistema ha devuelto al usuario como resultado de su consulta q_1, e igualmente $D(q_2)$ para q_2. El conjunto de páginas sobre las que el usuario ha hecho clic para las consultas q_1 y q_2 puede ser visto como:

$$DC(q_1) = \{d_{q_{11}}, d_{q_{12}}, ..., d_{q_{1i}}\} \subseteq (q_1)$$

$$DC(q_2) = \{d_{q_{21}}, d_{q_{22}}, ..., d_{q_{2j}}\} \subseteq (q_2)$$

Si $DC(q_1) \cap DC(q_2) = \{d_1, d_2, ..., d_k\} \neq \phi$ entonces las páginas d_1, d_2,…, d_k representan los temas comunes de las páginas q_1 y q_2. La similitud entre la consulta q_1 y q_2 está determinada por $DC(q_1) \cap DC(q_2)$. La similitud entre las dos consultas es proporcional al número de clic compartidos, tomado individualmente así:

$$similaridad_{sesión-sencilla}(q_1, q_2) = \frac{DN(q_1, q_2)}{\max(dn(q_1), dn(q_2))}$$

siendo $dn(q_1)$ el número de páginas sobre las que se ha hecho clic para la consulta q_1 (igual para q_2) y $DN(q_1, q_2)$ el número de páginas sobre las que se ha hecho clic en común.

A pesar de su simplicidad, esta medida presenta una sorprendente habilidad para clasificar consultas semánticamente relacionadas que contienen diferentes palabras. Pero no siempre es demasiado efectiva, funciona bien cuando las páginas sobre las que se hace clic están restringidas a un estrecho rango y muchos usuarios hacen clic en las mismas páginas, pero falla cuando dos consultas conducen a dos páginas similares pero separadas, ya que la similitud entre estas dos consultas no es detectada. Para superar este problema y obtener un *cluster* de consultas con una mayor cobertura, se puede

utilizar el *cluster* de documentos para agrupar primero las páginas web similares, de esta forma, dos consultas apuntando a dos páginas web diferentes pero que pertenecen al mismo *cluster* pueden considerarse como consultas similares. Hay dos formas de combinar el *cluster* de consultas y el *cluster* de documentos: el *clustering* unidireccional y el bidireccional.

***Clustering* unidireccional**. Dentro de este tipo también podemos distinguir dos caminos. El primero de ellos consiste en repartir todas las páginas web en *n clusters*, utilizando para ello algún algoritmo de *clustering* de documentos. Para cada sesión de consultas, todas las páginas sobre las que se ha hecho clic son sustituidas por sus correspondientes *clusters* en los que están incluidas. La similitud entre dos consultas es proporcional al número de *cluster* compartidos, utilizando la misma función de similitud anterior, pero siendo ahora $DN(q_1, q_2)$ el número de *clusters* comunes. El segundo de los caminos consiste en incorporar una medida de similitud de documentos en el cálculo de la similitud de las consultas. Sea $s(d_i, d_j)$ la función que calcula la similitud entre los documentos, *di*, *dj* los documentos sobre los que se ha hecho clic por las consultas q_1 y q_2 respectivamente, y $dn(q_1)$ y $dn(q_2)$ el número de documentos sobre los que ha hecho clic cada consulta, la función de similitud se define como:

$$similaridad_{sesión+docum}(q_1, q_2) = \frac{1}{2} \times \left(\frac{\sum_{i=1}^{m} (\max_{j=1}^{n} s(d_i, d_j))}{dn(q_1)} + \frac{\sum_{j=1}^{n} (\max_{i=1}^{m} s(d_i, d_j))}{dn(q_2)} \right)$$

***Clustering* bidireccional**. El cruce de referencias entre consultas y páginas web se puede utilizar también para clasificar documentos similares, ya que si varias personas seleccionan un conjunto común de documentos para la misma consulta, quiere decir que este conjunto es similar. Se trata de una hipótesis dual basada en la hipótesis de las consultas basadas en sesión que ha sido utilizada en varios motores de búsqueda. Una vez que un documento es localizado el sistema devuelve un conjunto de documentos que otras personas han visitado junto con ese documento.

En definitiva, mediante las sesiones de las consultas, la distancia entre dos consultas puede ser evaluada examinando las páginas web a las que se ha hecho clic y la distancia entre dos páginas web puede ser evaluada examinando sus consultas relacionadas. Por tanto, los procedimientos de *clustering* de consultas y el de

documentos (páginas web) pueden ser combinados y reforzados para construir un método de *clustering* bidireccional. Un algoritmo de este tipo se analiza en [Beeferman y Berger, 2000].

7.7.2.3 Combinando el clustering de consultas basado en el contenido y el basado en la sesión

Como ya se indicó antes, el *clustering* de consultas basado en el contenido no es suficiente para determinar la relación de las consultas basada en la semántica. Principalmente por dos motivos, primero porque las palabras no tienen un único significado y segundo porque palabras muy distintas pueden hacer referencia a un mismo concepto. Esta situación se agrava aún más en el caso de consultas que son muy cortas. Por otra parte el *clustering* de consultas basado en la sesión también presenta dificultades y no es una solución perfecta, ya que el hecho de que una persona haga clic en una página no tiene porqué ser un juicio relevante, por diversos motivos: puede desviar su atención y hacer clic en páginas no relacionadas o por otra parte una página puede tratar diversos temas dando lugar a diferentes consultas. Por lo tanto puede dudarse de la plena efectividad de las páginas visitadas en las sesiones, pero eso no quiere decir que sea un problema que haga desechar el método. Se tiene evidencia de que los resultados mejoran y que la efectividad de las páginas recuperadas se incrementa significativamente. Otro aspecto que se podría considerar es el número de clic que pueden proporcionar una base fidedigna para el *clustering* de consultas, ya que la mayoría de las consultas van seguidas de una o dos selecciones. Combinando el contenido de las palabras y los clics de las páginas se pueden obtener medidas con lo mejor de las dos estrategias, la más sencilla sería linealmente de esta forma:

$$similaridad = \alpha \times similaridad_{basada-contenido} + \beta \times similaridad_{basada-sesión}$$

Algunos trabajos utilizan esta medidas combinadas [Wen et al., 2002], el principal problema es determinar los valores de α y β, que debe hacerse de forma experimental.

Surgen nuevos retos y nuevas necesidades en las consultas, una práctica habitual es buscar información sobre una persona en Internet, pero esta tarea es dificultosa teniendo en cuenta la extensión global de la red y la posible duplicidad de nombres,

sobre todo cuando éstos pueden ser muy comunes. Además se puede hacer aún más compleja la búsqueda cuando existe algún personaje relevante cuyo nombre coincide con el que estamos buscando. Son muchos los grupos de investigación que están aplicando técnicas de clustering con nuevos algoritmos a este caso de estudio, desarrollándose campañas para analizar el problema [Artiles et al., 2009].

8 PROPUESTA DE UN MODELO DE CLASIFICACIÓN MEDIANTE LA APLICACIÓN DE TÉCNICAS DE *CLUSTERING*

8.1 Introducción

Se pretende aplicar las reglas de *clustering* para clasificar de forma automatizada grandes cantidades de información que manejan habitualmente los directorios de muchos portales web, buscando repartir los documentos de colecciones en grupos, de forma que puedan ser aplicados posteriormente a otros problemas prácticos como puede ser la agrupación de documentos obtenidos en las búsquedas web, o la visualización de directorios.

De esta forma se aborda un tema de gran interés para el usuario de las tecnologías de la información y la comunicación como es la mejora en la localización de contenidos ante la creciente avalancha de datos y su temporalidad. Por otra parte se busca brindar a los portales web nuevas formas de poder presentar la información que complementen a las ya existentes.

Para ello, se analiza la herramienta utilizada para llevar a cabo el proceso, se estudian los formatos soportados, tanto de entrada como de salida, así como sus elementos configurables, y se definen las fases que se van a seguir, adaptadas a las características de los documentos web.

8.2 Descripción del entorno experimental

Son diversos los entornos que se pueden manejar para realizar pruebas con *clusters*, por lo que ha sido necesario analizar sus características para utilizar el que mejor se adapta a nuestros propósitos. Algunas aplicaciones incluyen el tratamiento de *clusters* como una parte más de sus funciones; es el caso de diversas clases de software estadístico como CMLIB, APSTAT, GENERAL, MULTI, o MULTIV. Otras aplicaciones se centran en algún algoritmo determinado, como puede ser el de Kohonen (LVQ *Learning vector quantization*), el escalamiento multidimensional (Netlib/MDS),

o el *clustering* borroso (conjunto de programas de Borgelt, o FuzzyK). Y son muy pocos los paquetes especializados en el análisis de datos utilizando *clusters*. De ellos algunos son comerciales, como es el caso de Clustan, por lo que nos centraremos en los de libre distribución: Cluster y Cluto.

Cluster, es un entorno gráfico multiplataforma, desarrollado originalmente por Michael Eisen en la Universidad de Stanford [Eisen et al., 1998], para el análisis de datos desde microvectores de ADN u otros conjuntos de datos relacionados con la gnómica. Soporta distintas técnicas de *cluster* como jerárquico, *k-means*, mapas autoorganizados y análisis de componentes principales (PCA). Se complementa con la herramienta *TreeView* que permite visualizar los resultados obtenidos con Cluster. Su principal área de aplicación, en la que han sido validados los resultados, ha sido la gnómica, a diferencia de Cluto, que se presenta más versátil habiéndose utilizado en muy distintos campos como ciencias, biología, transacciones electrónicas, sistemas de información geográfica, y sobre todo, los que nos interesan: recuperación de la información y análisis de webs.

8.2.1 Características técnicas de Cluto

CLUTO [Karypis, 2001], es un paquete de software. Su nombre viene de "*CLUstering TOolkit*", que permite realizar *clustering* de conjuntos de datos de cualquier dimensión, y análisis de las características de los *clusters* obtenidos. Permite manejar tres tipos de algoritmos de *clustering* que corresponden a los paradigmas acumulativo, particional y basado en representaciones gráficas.

La mayoría de los algoritmos que utiliza tratan el *clustering* como un proceso de optimización, intentando maximizar o minimizar alguna función que se haya definido como criterio, bien de forma global o local, en el espacio de soluciones. Permite elegir entre siete funciones criterio para los tipos de *clustering* tanto particional como acumulativo, especialmente indicadas para el *cluster* de documentos, ya que permiten obtener soluciones de calidad aun trabajando con conjuntos de datos con muchas dimensiones o con muchos elementos. Además incorpora la mayoría de las funciones criterio locales tradicionales utilizadas en el contexto del *clustering* acumulativo como:

enlace-simple (*single-link)*, enlace-completo (*complete-link*) y UPGMA (*Unweighted Pair Group Method with Arithmatic Mean*).

Uno de los aspectos a destacar de CLUTO es el método que utiliza para optimizar estas funciones criterio, basado en un algoritmo incremental aleatorio con muy bajos requerimientos computacionales pero que obtiene soluciones de mucha calidad.

Al tratarse de un paquete o *kit* de software, también proporciona herramientas para analizar los *cluster*s obtenidos, para entender las relaciones entre los objetos asignados a cada *cluster* y las relaciones entre *clusters*, y herramientas para visualizar las soluciones obtenidas.

8.2.2 Conjuntos de datos utilizados

Se han utilizado dos conjuntos de datos, uno mucho más manejable sobre el que se han realizado todas las pruebas en todos los apartados y otro mucho más extenso para medir los casos extremos. Como punto de partida se necesitaba una web lo suficientemente grande como para que la clasificación pudiera extenderse a portales de mediano tamaño, pero además sería útil tener una idea de su contenido para ir constatando si los resultados que se iban obteniendo se ajustaban a la realidad, por ello se recurrió a la web de la Universidad de Salamanca. Se obtuvieron dos variantes que se detallan a continuación.

8.2.2.1 Caso de estudio A.

Se descarga el portal Web de la Universidad de Salamanca, mediante un capturador web (también conocidos como aspiradoras, arañas, o navegadores *offline*), a partir de su url. Las características de los documentos obtenidos son:

- Tamaño completo de la descarga: 304 MBytes.
- 6547 ficheros.

- 250 carpetas, de las cuales 24 eran de primer nivel. De estas 24, 12 no tenían subcarpetas, 4 tenían 1, 3 tenían 2, 1 tenía 3, 2 tenían 6, 1 tenía 18 y la más numerosa tenía 19.

8.2.2.2 Caso de estudio B.

Se recopila de forma sistemática todo el portal de la Universidad de Salamanca, recogiendo distintos servidores y organizando su contenido en carpetas. Se genera una estructura arborescente con las siguientes características:

- Tamaño completo 1,80 GBytes.
- 11 carpetas que contienen cada una otras 100 carpetas.
- Cada carpeta de último nivel contiene aproximadamente 100 ficheros, dando un total de 109.175 documentos html.

8.3 Proceso seguido para la clasificación

Para poder clasificar un conjunto de documentos en formato *html* es necesario preparar los datos haciéndolos pasar por una serie de fases que los vayan adaptando a las herramientas que se van a utilizar, en este caso CLUTO. El proceso llevado a cabo se ilustra en la figura 16 y se detalla a continuación.

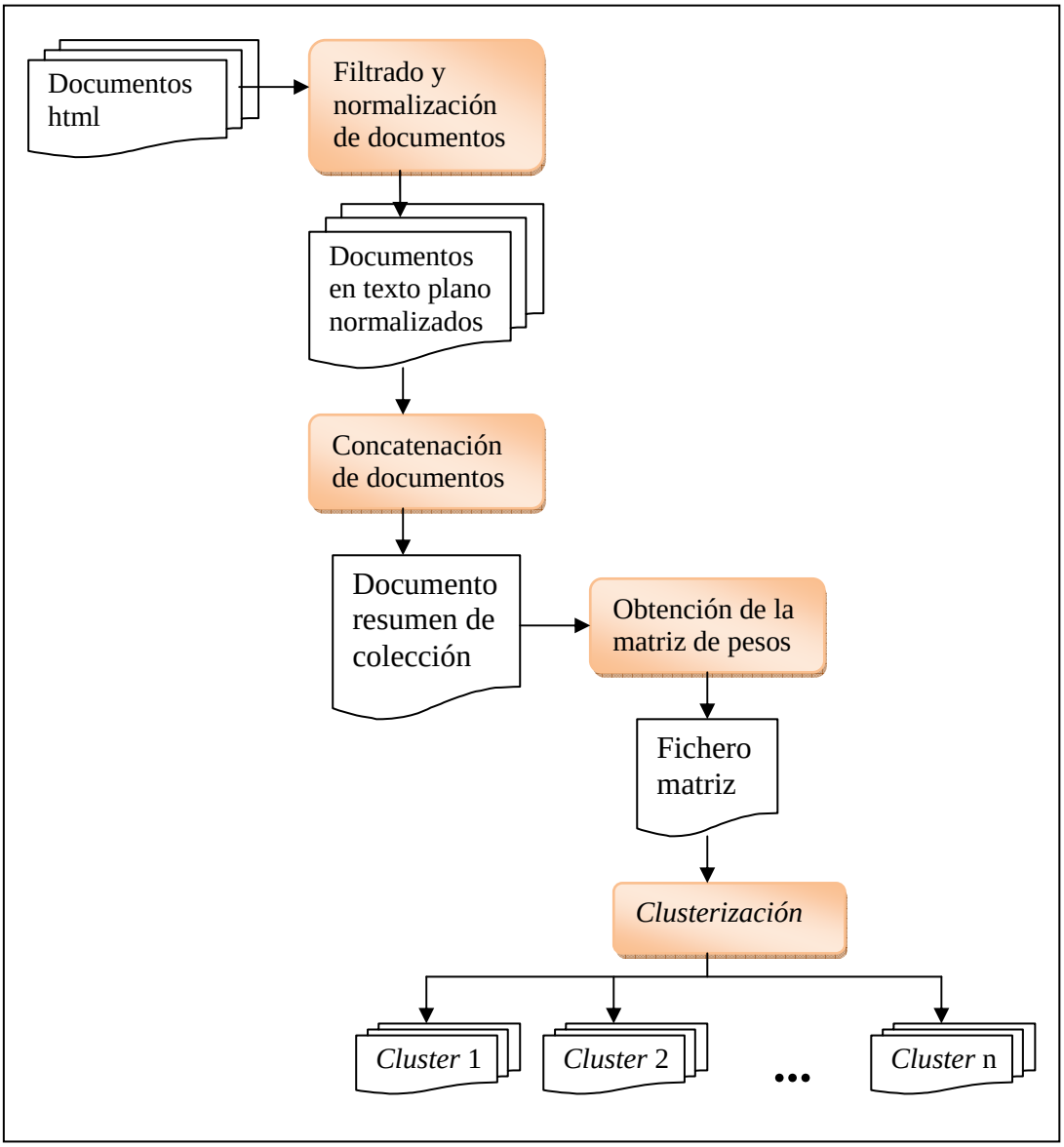

Figura 16. Proceso de Clasificación de documentos.

8.3.1 Fase de filtrado y normalización

Partimos de un conjunto de páginas *html* que vamos a manejar como si se trataran de ficheros escritos en texto plano, por lo que será necesario filtrar todas las

etiquetas correspondientes al lenguaje *html*, que no nos van a aportar datos, sino todo lo contrario. Además de las etiquetas puede aparecer texto correspondiente a sentencias de lenguajes de programación inmersos y comentarios relacionados, que también habrá que intentar eliminar. Pero además en esta primera fase, podemos normalizar los caracteres dependientes de las fuentes, codificar los acentos, sustituir los separadores por espacios en blancos y a su vez reducir las secuencias de espacios en blanco.

Este proceso hay que hacerlo para cada uno de los documentos, por lo que hay que automatizarlo, obteniendo como salida un fichero normalizado compuesto únicamente por palabras útiles. (*previo.pl* es un *script* en *perl* que se encarga de este proceso, anexo D).

8.3.2 Fase de concatenación

Consiste en unir los ficheros de texto que se han ido obteniendo en un único documento resumen de la colección, de forma que cada línea de este documento resumen corresponda a las palabras de una página *html* original. (Se utilizan los ficheros por lotes *concatena.bat* y *procesa.bat*). Esta forma de proceder viene condicionada por la herramienta que se va a utilizar para la obtención de las frecuencias de cada uno de los términos. Un fragmento de un fichero de salida obtenido es el de la figura 17. Se observa que cada documento viene precedido por un número que lo identifica. El script procesa.bat, se encarga además de ir generando las etiquetas XML, que recogen las características individuales de cada documento, en concreto: el número identificativo del documento, su URL, y su ruta local. Estos datos se van añadiendo al documento final XML, al que después se irán añadiendo datos en etapas sucesivas.

```
...
102  noticias universidad de salamanca body archivo fotografico personas ed actos genericas
todas mostrando fotos de la a la de un total de anterior siguiente gabinete de comunicacion
y protocolo patio de escuelas n salamanca
103  gabinete de comunicacion body buscador de noticias introduzca el texto que quiera
buscar en la caja de texto puede utilizar tanto mayusculas como minusculas no olvide poner
los acentos oportunos haga click sobre la lupa gabinete de comunicacion y protocolo patio de
escuelas n salamanca
104  gabinete de comunicacion body comunicacion cientifica hay un total de notas de prensa
en este año la universidad invertira millones de euros en un proyecto para combatir
enfermedades geneticas el de los universitarios de castilla y leon consume alcohol durante
el fin de semana los jovenes fumadores sufren perdida de masa osea lo que predispone a
padecer osteoporosis investigadores de la universidad descubren al unico reptil conocido que
poliniza plantas investigadores analizan los factores que determinan la calidad del queso de
oveja mas de expertos mundiales asisten a la iv conferencia internacional sobre
nanotecnologia los antibioticos en la dieta de los animales propician la aparicion de
resistencia en salmonella investigadores de la universidad abren una nueva linea de
investigacion sobre el autismo la universidad evalua los factores que inciden en la calidad
del lechazo de castilla y leon la universidad abre una nueva linea de investigacion sobre
enfermedades infecciosas la universidad analiza un parasito ampliamente expandido en
animales domesticos la universidad estudiara las causas de la desert la universidad pionera
...
```

Figura 17. Fragmento de fichero conteniendo las palabras de un documento por línea.

8.3.3 Fase de obtención de la matriz de pesos

A partir del fichero con los documentos en cada una de sus líneas (ejemplo de la figura 17), se debe obtener una matriz con el formato espacio-vectorial que utiliza CLUTO. Las columnas de la matriz serían todos los términos distintos del conjunto de documentos, mientras que las filas representarían cada documento. Para cada fila se marcarían las columnas (términos), correspondientes al documento asociado a esa fila, teniendo en cuenta que un mismo término puede aparecer varias veces en el mismo documento. Para simplificar las dimensiones de la matriz y reducir su tamaño sólo se almacena información sobre los cruces de filas y columnas que no son nulos.

Por lo tanto el formato que genera consiste en almacenar en un fichero de texto plano la matriz *A*, con *n+1* filas y *m* columnas, utilizando la primera fila como una cabecera de información y las *n* filas restantes asociadas a cada uno de los documentos del fichero total. La primera fila contiene tres números enteros, el primero es el número de filas en la matriz (*n*), el segundo es el número de columnas (*m*) y el tercero es el número total de entradas en la matriz que son distintas de cero. Cada una de las filas obtenidas contiene una pareja de valores para cada una de las entradas distintas de cero de su correspondiente fila en la matriz de documentos. Cada pareja estará formada por el número de columna y por su valor correspondiente, siempre no nulo, porque si no, no aparecería ese valor de columna.

Un fragmento de matriz y su representación gráfica se muestra en la figura 18. Se puede apreciar que cada fila puede tener distinta longitud, así la fila dos tendría dos valores, la fila tres tendría tres pero la fila ocho tendría 78 valores. El significado de la fila 2, que es 1 1 2 1, sería que el término de la columna 1 aparece una vez y que el término de la columna 2 aparece también 1 vez en el primer documento.

```
3116 17145 162038
1 1 2 1
3 1 1 1 2 1
1 1 2 1
3 1 1 1 2 1
4 1 5 1 6 1 7 1 8 1 9 1 10 1 11 1 12 1 13 1 14 1 15 1 16 1 17 1
1 1 2 1 3 1
18 2 19 2 20 2 11 1 21 1 22 2 23 4 24 1 25 1 26 1 27 1 28 1 29 1 30 3
31 1 32 1 33 1 34 1 35 1 36 2 37 1 38 1 39 3 40 3 41 1 42 1 43 3 44 3
45 2 46 1 47 1 48 1 49 1 50 1 51 1 52 1 53 1 54 1 55 1 56 4 57 1 58 1
59 1 60 1 61 1 62 1 63 1 64 7 8 1 65 2 66 1 67 1 68 1 69 1 70 1 71 1
72 1 73 4 74 1 75 1 4 1 76 1 6 1 77 1 78 1 79 1 80 1 81 1 82 1 83 1 84
1 85 1 86 1 87 1 88 1 89 1 90 1 91 1
18 2 19 3 92 1 20 2 11 1 21 2 22 2 23 7 24 1 93 1 94 1 25 1 26 1 27 1
28 1 29 1 30 3 31 1 33 1 34 1 35 1 36 3 38 1 41 1 40 4 39 3 42 1 43 3
44 5 45 3 46 1 47 1 95 1 48 1 96 1 49 1 97 2 51 1 52 1 53 1 54 1 55 1
57 1 56 4 59 1 58 1 60 2 62 1 98 1 63 2 64 8 66 1 8 1 65 2 67 1 68 1
69 1 99 1 71 1 100 1 73 4 72 1 74 1 75 1 76 1 4 1 6 1 77 2 78 1 79 2
101 1 102 1 80 2 82 1 81 2 84 1 103 1 87 2 86 1 88 1 89 1 90 1 91 1
104 1
105 1 106 1 29 1 107 1 4 1 108 1 109 2 91 1
3 1 1 1 2 1
5 1 498 1 40 1 499 1 497 1
...
```

Figura 18. Ejemplo de formato de matriz de pesos.

La generación de la matriz se hace con *doc2mat*, una herramienta desarrollada en *perl* por el propio autor de CLUTO. Como argumento de entrada se facilita el fichero que contiene los documentos que quieren ser convertidos al modelo espacio-vectorial y como argumento de salida genera un fichero con la matriz de términos-documento. Permite lematización de términos y eliminación de palabras vacías. Se puede utilizar su propia lista de palabras, utilizar una nueva o ambas. La eliminación de las palabras se hace antes de la lematización por los que las palabras deben estar completas, sin reducirse. Otros valores que permite configurar es la longitud mínima de los *tokens* a tener en cuenta, la eliminación de los valores numéricos de las palabras o la eliminación de identificadores al principio de los documentos.

También genera un fichero con el nombre de salida elegido y con extensión *.clabel*, que contiene los términos correspondientes a las columnas, las etiquetas, un término por línea. Ejemplo de fragmento en la figura 19.

```
apache
www
moved
document
server
port
found
noches
webmaster
fonseca
escuelas
english
concursos
web
oposiciones
academico
directorio
pruebas
fax
plazo
preinscripcion
patio
reconocimiento
presenta
programacion
abad
mecenazgo
abre
tlf
virtual
internacional
```

Figura 19. Fragmento de fichero de salida con cabeceras de columna, tipo clabel.

8.3.4 Fase de obtención de los *clusters*

A partir de la matriz obtenida en la fase anterior (FicheroMatriz), se agrupan los documentos en tantos *clusters* como se especifique en el último de los argumentos (NúmeroClusters). El programa encargado de ello es *vcluster*, cuyo formato es el siguiente:

vcluster [parámetros opcionales] FicheroMatriz NúmeroClusters

Si la ejecución es correcta se muestra por pantalla información sobre la calidad de los *clusters* obtenidos y estadísticas del tiempo empleado en el proceso. También se obtiene un fichero de salida llamado FicheroMatriz.clustering.NúmeroClusters, con la solución. Por ejemplo al ejecutar:

vcluster MiMat 10

obtendríamos: MiMat.clustering.10

El fichero de salida puede tener dos formatos distintos dependiendo de los parámetros opcionales especificados. El primer formato, que se obtiene por defecto, corresponde al vector de *clustering*. Se trata de un fichero con tantas líneas como la matriz de entrada, en cada línea un único número que indica el *cluster* al que pertenece el elemento *i* correspondiente de la matriz de entrada. Los números de *cluster* van desde cero hasta el nº de *clusters* menos uno. En el caso de que algún objeto no haya podido ser asignado a ningún *cluster* aparecerá con el valor -1, como se aprecia en la figura 20.

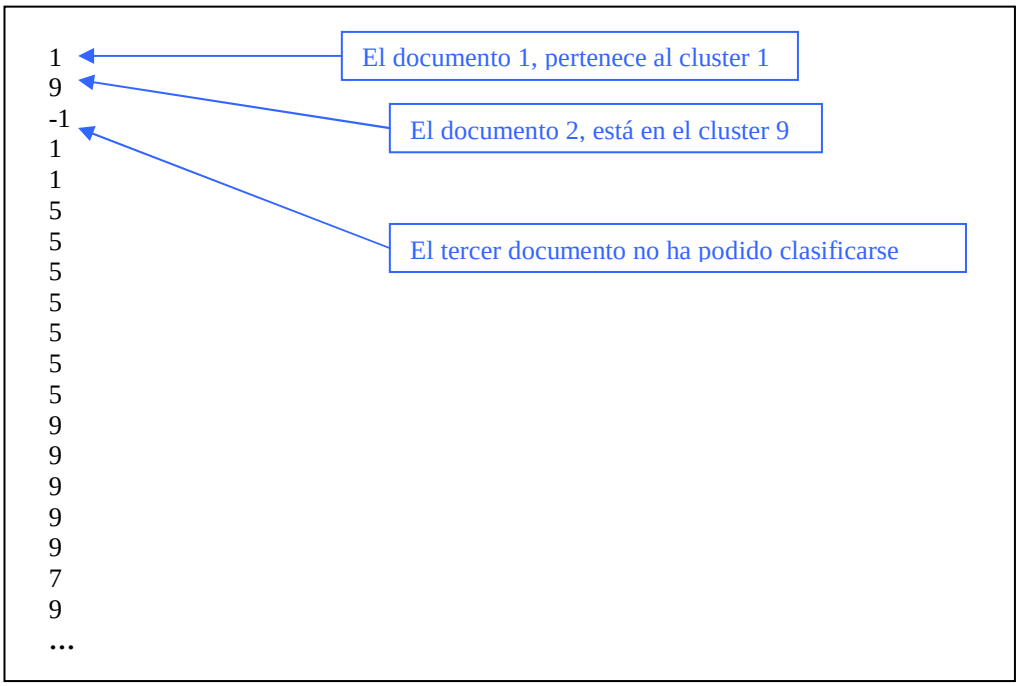

Figura 20. Ejemplo de fichero de salida, simple, clasificado.

El segundo formato que puede tener el fichero de salida, que se produce cuando se ha elegido algún tipo de *clustering* jerárquico acumulativo o alguno particional en el que se recoja la estructura de árbol, consiste en un vector que permite simular la forma de árbol. Si el número de *clusters* es *k*, el fichero contiene *2k-1* líneas, en la línea *i* aparece un número que representa el padre del nodo *i* del árbol. El nodo raíz va en la última línea y su valor va puesto a -1. En este formato, en cada línea aparecen otros dos valores, el primero es la similitud media entre los hermanos de cada nodo del árbol, el segundo es el cambio en el valor de la función criterio resultante de la combinación del

par de *clusters*. Estos dos valores están a cero para los *clusters* finales. Véase la figura 21.

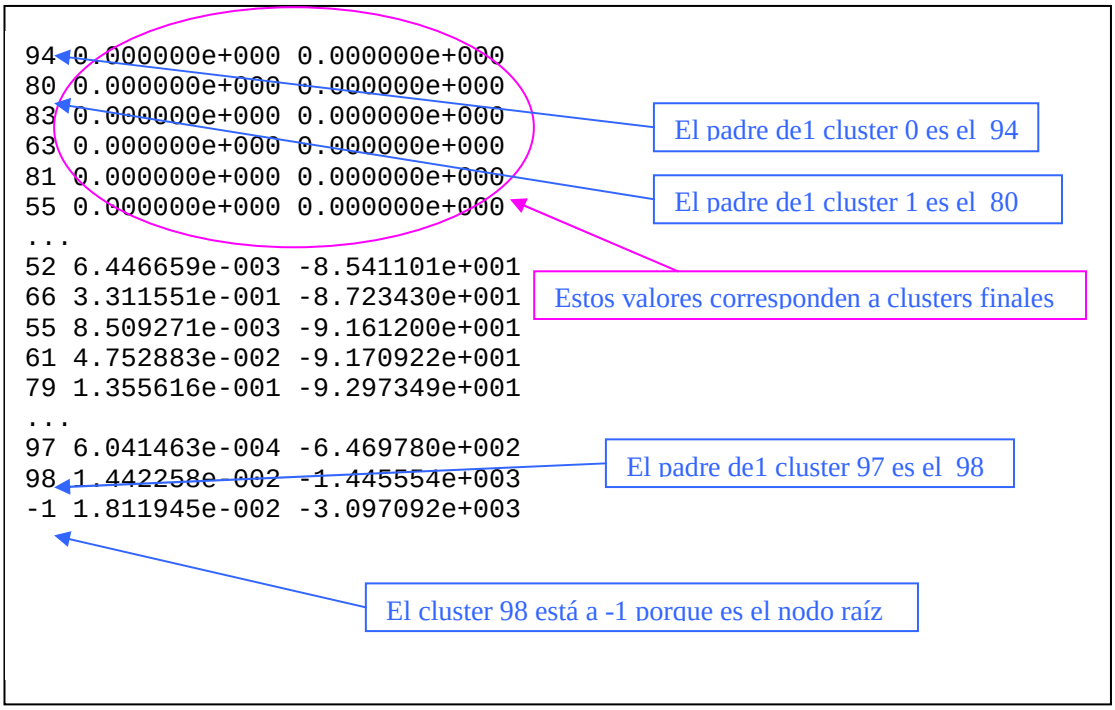

Figura 21. Ejemplo de fichero de salida, para estructura en forma de árbol.

Los parámetros opcionales pueden dividirse en tres grupos. El primer grupo controla aspectos relacionados con el algoritmo de *clustering*, el segundo grupo se encarga del tipo de análisis e informe que se hace de los *clusters* y el tercer grupo controla la visualización de los *clusters*. Se detallan en la siguiente sección.

8.4 Opciones controlables en la obtención de los *clusters*

Todas las opciones que se pueden controlar se introducen mediante los "parámetros opcionales". Existen multitud de parámetros en distintos ámbitos: relacionados con el algoritmo de *clustering*, relacionados con el tipo de análisis de los *clusters*, con el tipo de informe, con el formato de visualización de los *clusters*, etc.

El parámetro más importante es el método de *clustering* que se selecciona. Los tipos posibles son:

rb (*Repeated Bisections*). Para obtener los *k clusters* se realizan *k-1* bisecciones repetidas. Se divide la matriz en dos, se selecciona una de las parte y así sucesivamente hasta que se obtiene el número deseado de *clusters*. En cada paso, para realizar la división se busca optimizar una función, que se puede elegir entre un conjunto preestablecido. De esta forma la función criterio está optimizada de forma local, aunque no global. También se puede elegir el *cluster* que se selecciona para continuar con las divisiones.

rbr (*Repeated Bisections Refinement*). El proceso seguido es igual al del caso anterior, pero al final la solución en conjunto es optimizada globalmente.

direct Se buscan los *k clusters* de forma simultánea. Para valores pequeños de *k* (10-20) los resultados son mejores que realizando bisecciones.

agglo (*Agglomerative*). Se utiliza el paradigma acumulativo. Se detiene el proceso de acumular cuando se obtienen los *k clusters*.

graph Se hace un modelo de los objetos mediante un gráfico de proximidad en el que cada objeto es un vértice que se conecta con los objetos más similares. Este gráfico se va fraccionando hasta obtener los *k clusters* mediante un algoritmo de tipo "*min-cut*".

bagglo (*Bisections agglomerative*). Se calcula de forma similar al método *agglo*. El proceso de agrupamiento es influido por una solución particional (*rb*) que es obtenida en el conjunto de datos inicial.

Es posible seleccionar la función de similitud que se utiliza. Los posibles valores se especifican a continuación, si bien las dos últimas funciones sólo se aplican si el método de *clustering* utilizado es *graph*:

cos Función coseno. Es la que utiliza la aplicación por defecto si no se especifica ninguna otra.

corr Coeficiente de correlación.

dist Se busca que la similitud entre los objetos sea inversamente proporcional a la distancia euclídea.

jacc Coeficiente de Jaccard extendido.

Otro parámetro determinante, que también se puede seleccionar, es la función criterio de *clustering* utilizada para buscar los *clusters*. Dispone de 7 funciones criterio propias, que denota como: **i1, i2, e1, g1, g1p, h1 y h2.** Además permite utilizar otras funciones criterio tradicionales, manejadas por otras muchas herramientas, como son: **slink** (*single-link*), **wslink (***single-link* con pesado de *clusters*), **clink** (*complete-link*), **wclink** (*complete-link* con pesado de *clusters*), **upgma** (*Unweighted Pair Group Method with Arithmetic Mean*).

8.5 Información obtenida sobre la calidad de los *clusters*

Tanto o más importante que los *clusters* obtenidos en el fichero de salida, es la información proporcionada por pantalla sobre la calidad de la solución obtenida. El formato más simple es el recogido en la figura 22. La calidad de los *clusters* se mide por la función criterio que se utiliza y por la similitud entre los objetos de cada *cluster*.

```
*******************************************************************************
vcluster (CLUTO 2.1.1) Copyright 2001-03, Regents of the University of Minnesota

Matrix Information ------------------------------------------------------------
  Name: salida1.mat, #Rows: 3116, #Columns: 19117, #NonZeros: 304725

Options -----------------------------------------------------------------------
  CLMethod=RB, CRfun=I2, SimFun=Cosine, #Clusters: 10
  RowModel=None, ColModel=IDF, GrModel=SY-DIR, NNbrs=40
  Colprune=1.00, EdgePrune=-1.00, VtxPrune=-1.00, MinComponent=5
  CSType=Best, AggloFrom=0, AggloCRFun=I2, NTrials=10, NIter=10

Solution ----------------------------------------------------------------------

------------------------------------------------------------------------
10-way clustering: [I2=1.21e+003] [3114 of 3116]
------------------------------------------------------------------------
cid  Size  ISim  ISdev   ESim  ESdev  |
------------------------------------------------------------------------
  0   132 +0.968 +0.106 +0.009 +0.001 |
  1    28 +0.849 +0.200 +0.008 +0.003 |
  2   230 +0.636 +0.132 +0.021 +0.011 |
  3    96 +0.562 +0.113 +0.003 +0.003 |
  4    49 +0.408 +0.140 +0.015 +0.005 |
  5   175 +0.330 +0.153 +0.040 +0.009 |
  6   156 +0.252 +0.084 +0.010 +0.003 |
  7   730 +0.077 +0.015 +0.015 +0.005 |
  8   844 +0.074 +0.018 +0.029 +0.006 |
  9   674 +0.058 +0.012 +0.031 +0.007 |
------------------------------------------------------------------------

Timing Information ------------------------------------------------------------
   I/O:                                0.296 sec
   Clustering:                         2.625 sec
   Reporting:                          0.079 sec
*******************************************************************************
```

Figura 22. Ejemplo de salida de información para 10 *clusters*.

La primera información que aparece se refiere a los datos de entrada suministrados, nombre del fichero matriz, nº de filas y de columnas y los distintos valores que han tomado las opciones configurables. A continuación viene claramente separada la parte correspondiente a la solución. En esta parte, en la primera línea, a continuación del número de *clusters*, aparece el valor global de la función criterio para la solución calculada, en el ejemplo sería I2=1.21e+003, en la misma línea figura también el número de objetos del total que han podido ser clasificados.

A continuación, en forma de tabla se recogen estadísticas para cada uno de los *clusters*, que aparecen, uno en cada línea. Los valores de las columnas son:

- **cid** identificador de *cluster*, un número que empieza en 0.
- **Size** nº de objetos del *cluster*.
- **ISim** similitud media entre los objetos del *cluster* (Interna).
- **ISdev** desviación estándar con respecto a la similitud interna media.
- **ESim** similitud media de los objetos del *cluster* y el resto de los objetos.
- **ESdev** desviación estándar de la similitud externa.

Se puede ampliar esta información analizando cada uno de los *clusters* y especificando los conjuntos de características que mejor describen y los que discriminan para cada uno de los *clusters*. Para ello es necesario especificar la opción *–showfeatures* en la línea de ejecución. El tipo de información es el que se muestra en la figura 23.

```
*******************************************************************************
vcluster (CLUTO 2.1.1) Copyright 2001-03, Regents of the University of Minnesota

Matrix Information ----------------------------------------------------------
  Name: salida1.mat, #Rows: 3116, #Columns: 19117, #NonZeros: 304725

Options ---------------------------------------------------------------------
  CLMethod=RB, CRfun=I2, SimFun=Cosine, #Clusters: 10
  RowModel=None, ColModel=IDF, GrModel=SY-DIR, NNbrs=40
  Colprune=1.00, EdgePrune=-1.00, VtxPrune=-1.00, MinComponent=5
  CSType=Best, AggloFrom=0, AggloCRFun=I2, NTrials=10, NIter=10

Solution --------------------------------------------------------------------

------------------------------------------------------------------------
10-way clustering: [I2=1.21e+003] [3114 of 3116]
------------------------------------------------------------------------
cid  Size  ISim  ISdev   ESim  ESdev  |
------------------------------------------------------------------------
  0   132 +0.968 +0.106 +0.009 +0.001 |
  1    28 +0.849 +0.200 +0.008 +0.003 |
  2   230 +0.636 +0.132 +0.021 +0.011 |
  3    96 +0.562 +0.113 +0.003 +0.003 |
  4    49 +0.408 +0.140 +0.015 +0.005 |
  5   175 +0.330 +0.153 +0.040 +0.009 |
  6   156 +0.252 +0.084 +0.010 +0.003 |
  7   730 +0.077 +0.015 +0.015 +0.005 |
  8   844 +0.074 +0.018 +0.029 +0.006 |
  9   674 +0.058 +0.012 +0.031 +0.007 |
------------------------------------------------------------------------
-------------------------------------------------------------------------------
10-way clustering solution - Descriptive & Discriminating Features...
-------------------------------------------------------------------------------
Cluster   0, Size:   132, ISim: 0.968, ESim: 0.009
      Descriptive:  col00007 12.5%, col00010 12.3%, col00017 12.2%, col00016 12.2%, col00009 12.1%
   Discriminating:  col00007  6.5%, col00010  6.5%, col00017  6.4%, col00016  6.4%, col00009  6.3%

Cluster   1, Size:    28, ISim: 0.849, ESim: 0.008
      Descriptive:  col00610 20.0%, col00607 18.7%, col01411 15.6%, col02631 13.6%, col00131 11.5%
   Discriminating:  col00610 10.3%, col00607  9.7%, col01411  8.1%, col02631  7.0%, col00131  5.9%

Cluster   2, Size:   230, ISim: 0.636, ESim: 0.021
      Descriptive:  col17814 14.7%, col17818  6.1%, col00073  5.0%, col00609  3.8%, col17813  3.7%
   Discriminating:  col17814  8.5%, col00635  6.4%, col17818  3.5%, col00073  2.4%, col17813  2.1%

Cluster   3, Size:    96, ISim: 0.562, ESim: 0.003
      Descriptive:  col00001 30.1%, col00002 24.1%, col00003 22.6%, col18832  2.4%, col18831  2.4%
   Discriminating:  col00001 15.3%, col00002 12.3%, col00003 11.5%, col00635  4.8%, col00044  3.5%

Cluster   4, Size:    49, ISim: 0.408, ESim: 0.015
      Descriptive:  col18973 13.3%, col03794 12.7%, col17439 11.2%, col16719 10.7%, col00073 10.3%
   Discriminating:  col18973  7.5%, col03794  7.0%, col17439  6.3%, col16719  6.0%, col00635  5.3%

Cluster   5, Size:   175, ISim: 0.330, ESim: 0.040
      Descriptive:  col00635 29.2%, col00044 17.9%, col00798  7.0%, col06840  6.9%, col06825  6.6%
   Discriminating:  col00635  7.5%, col06840  5.4%, col06825  5.2%, col00797  4.2%, col00044  3.7%

Cluster   6, Size:   156, ISim: 0.252, ESim: 0.010
      Descriptive:  col00798 31.5%, col01397 12.0%, col01238 10.8%, col00611  8.8%, col00005  8.1%
   Discriminating:  col00798 14.7%, col01397  6.4%, col01238  5.8%, col00635  5.4%, col00611  4.5%

Cluster   7, Size:   730, ISim: 0.077, ESim: 0.015
      Descriptive:  col15272  6.0%, col01926  5.9%, col03043  5.8%, col02641  5.6%, col00134  5.6%
   Discriminating:  col00635  8.3%, col00044  6.1%, col15272  4.0%, col01926  3.9%, col03043  3.8%

Cluster   8, Size:   844, ISim: 0.074, ESim: 0.029
      Descriptive:  col00635 15.4%, col00044  9.5%, col00691  3.4%, col00045  2.7%, col00534  2.3%
   Discriminating:  col00635  3.6%, col00691  2.8%, col00798  2.5%, col17814  2.5%, col00006  1.7%

Cluster   9, Size:   674, ISim: 0.058, ESim: 0.031
      Descriptive:  col00635  6.5%, col00022  4.1%, col00044  4.0%, col00084  3.3%, col00045  2.0%
   Discriminating:  col17814  2.7%, col00798  2.6%, col00022  2.4%, col00084  1.8%, col00005  1.7%
-------------------------------------------------------------------------------

Timing Information ----------------------------------------------------------
   I/O:                          0.437 sec
   Clustering:                   2.875 sec
   Reporting:                    0.266 sec
*******************************************************************************
```

Figura 23. Salida obtenida con la opción *-showfeatures*.

La información general numérica que se obtiene para cada *cluster* es la misma que con la versión simplificada, pero se añade información sobre los términos más importantes para cada *cluster.* Como se puede apreciar en la figura 23, esta información añadida contiene el número de las columnas de los términos que mejor describen cada *cluster* y de igual forma los términos discriminantes aparecen también como número de columna con lo que su análisis es casi inútil. Por ello, es aconsejable introducir información sobre el nombre de cada columna (correspondería a una palabra del documento), y esto es posible mediante el parámetro *-clabelfile*, seguido del nombre del fichero con esta información. De esta forma se obtiene una salida como la de la figura 24.

```
--------------------------------------------------------------------------------
10-way clustering solution - Descriptive & Discriminating Features...
--------------------------------------------------------------------------------
Cluster   0, Size:   132, ISim: 0.968, ESim: 0.009
      Descriptive:  inexistent 12.5%, encontrado 12.3%, consult 12.2%, administrador 12.2%, servidor 12.1%
   Discriminating:  inexistent  6.5%, encontrado  6.5%, consult  6.4%, administrador  6.4%, servidor  6.3%

Cluster   1, Size:    28, ISim: 0.849, ESim: 0.008
      Descriptive:  href 20.0%, html 18.7%, alert 15.6%, seleccionado 13.6%, dispon 11.5%
   Discriminating:  href 10.3%, html  9.7%, alert  8.1%, seleccionado  7.0%, dispon  5.9%

Cluster   2, Size:   230, ISim: 0.636, ESim: 0.021
      Descriptive:  background 14.7%, relx  6.1%, write  5.0%, index  3.8%, absolut  3.7%
   Discriminating:  background  8.5%, scrollbar  6.4%, relx  3.5%, write  2.4%, absolut  2.1%

Cluster   3, Size:    96, ISim: 0.562, ESim: 0.003
      Descriptive:  click 30.1%, move 24.1%, page 22.6%, arrastr  2.4%, mover  2.4%
   Discriminating:  click 15.3%, move 12.3%, page 11.5%, scrollbar  4.8%, color  3.5%

Cluster   4, Size:    49, ISim: 0.408, ESim: 0.015
      Descriptive:  screen 13.3%, codigo 12.7%, option 11.2%, select 10.7%, write 10.3%
   Discriminating:  screen  7.5%, codigo  7.0%, option  6.3%, select  6.0%, scrollbar  5.3%

Cluster   5, Size:   175, ISim: 0.330, ESim: 0.040
      Descriptive:  scrollbar 29.2%, color 17.9%, foto  7.0%, venu  6.9%, transito  6.6%
   Discriminating:  scrollbar  7.5%, venu  5.4%, transito  5.2%, fotografico  4.2%, color  3.7%

Cluster   6, Size:   156, ISim: 0.252, ESim: 0.010
      Descriptive:  foto 31.5%, antartida 12.0%, expedicion 10.8%, sin  8.8%, documento  8.1%
   Discriminating:  foto 14.7%, antartida  6.4%, expedicion  5.8%, scrollbar  5.4%, sin  4.5%

Cluster   7, Size:   730, ISim: 0.077, ESim: 0.015
      Descriptive:  pag  6.0%, volver  5.9%, categoria  5.8%, publicado  5.6%, listado  5.6%
   Discriminating:  scrollbar  8.3%, color  6.1%, pag  4.0%, volver  3.9%, categoria  3.8%

Cluster   8, Size:   844, ISim: 0.074, ESim: 0.029
      Descriptive:  scrollbar 15.4%, color  9.5%, hora  3.4%, la  2.7%, gabinet  2.3%
   Discriminating:  scrollbar  3.6%, hora  2.8%, foto  2.5%, background  2.5%, pagina  1.7%

Cluster   9, Size:   674, ISim: 0.058, ESim: 0.031
      Descriptive:  scrollbar  6.5%, lo  4.1%, color  4.0%, que  3.3%, la  2.0%
   Discriminating:  background  2.7%, foto  2.6%, lo  2.4%, que  1.8%, documento  1.7%
--------------------------------------------------------------------------------
```

Figura 24. Fragmento de salida mejorada introduciendo etiquetas para las columnas.

Para cada uno de los *clusters* aparece una línea con los 5 términos más descriptivos y otra línea con los 5 más discriminantes, este valor puede modificarse mediante la opción *-nfeautures*. A la derecha de cada término aparece el porcentaje que

representa ese término con respecto a la similitud media del *cluster*. Si se especifica la opción *-showsummaries*, además se analizarán las características más descriptivas de cada *cluster* y se tratará de identificar el conjunto de características que ocurren de forma conjunta en los objetos, de esta forma se pueden determinar *sub-clusters*, el resultado obtenido para el caso anterior es el recogido en la figura 25.

```
-------------------------------------------------------------------------------
10-way clustering solution - Cluster Summaries using Cliques...
-------------------------------------------------------------------------------
Cluster   0, Size:   132, ISim: 0.968, ESim: 0.009
  98.33%    inexistent encontrado administrador consult servidor

Cluster   1, Size:    28, ISim: 0.849, ESim: 0.008
  92.86%    href html alert seleccionado dispon

Cluster   2, Size:   230, ISim: 0.636, ESim: 0.021
  97.04%    background relx write index absolut

Cluster   3, Size:    96, ISim: 0.562, ESim: 0.003
  59.72%    click arrastr mover
  73.61%    click move page

Cluster   4, Size:    49, ISim: 0.408, ESim: 0.015
  70.61%    screen codigo option select write

Cluster   5, Size:   175, ISim: 0.330, ESim: 0.040
  65.83%    scrollbar color foto venu transito

Cluster   6, Size:   156, ISim: 0.252, ESim: 0.010
  77.44%    foto antartida expedicion sin documento

Cluster   7, Size:   730, ISim: 0.077, ESim: 0.015
  98.11%    pag volver categoria publicado listado

Cluster   8, Size:   844, ISim: 0.074, ESim: 0.029
  97.54%    scrollbar color hora la gabinet

Cluster   9, Size:   674, ISim: 0.058, ESim: 0.031
  93.38%    scrollbar lo color que la
```

Figura 25. Fragmento de salida en la que se han incluido sumarios para cada *cluster*.

Aparece información añadida para cada *cluster*, puede aparecer un único sumario o varios, en este último caso es habitual que se produzca solapamiento entre los *sub-clusters*, sería el caso del *cluster* 3 de la figura 25.

9 DESARROLLO DEL MODELO DE CLASIFICACIÓN

9.1 Introducción

Se ha procedido a la verificación del modelo mediante la aplicación en cada una de sus fases de un método eminentemente inductivo. Mediante la realización de sucesivos experimentos se han ido ajustando los parámetros para conseguir resultados cada vez más precisos, para posteriormente poder generalizar y aplicar directamente el método a cualquier conjunto de valores. Se comienza por analizar cada una de las fases, estudiándose los pormenores que se producen en cada una de ellas y se exponen las adaptaciones que han sido necesarias para ir mejorando los resultados.

Se continúa analizando cada uno de los métodos de clustering en combinación con cada una de las funciones criterio para determinar cuáles son los valores óptimos para el caso de estudio de los portales web.

Se determinan también los tiempos de computación, la incidencia del número de *clusters* y se detalla cómo se comportan las características descriptivas, en relación con los métodos de *clustering* y las funciones criterio, centrándose en las combinaciones seleccionadas por sus mejores resultados.

9.2 Actuaciones en la fase de filtrado y normalización de los documentos

9.2.1 Filtrado de etiquetas

Los resultados que se obtienen en primera instancia habiendo realizado un filtrado básico consistente en la normalización de los caracteres dependientes de las fuentes, codificado los acentos y eliminado las etiquetas correspondientes al lenguaje *html,* se pueden observar en la figura 26.

```
10-way clustering solution - Cluster Summaries using Cliques...
--------------------------------------------------------------------------------
Cluster   0, Size:   132, ISim: 0.968, ESim: 0.009
  98.33%    inexistent encontrado administrador consult servidor

Cluster   1, Size:    28, ISim: 0.849, ESim: 0.008
  92.86%    href html alert seleccionado dispon

Cluster   2, Size:   230, ISim: 0.636, ESim: 0.021
  97.04%    background relx write index absolut

Cluster   3, Size:    96, ISim: 0.562, ESim: 0.003
  59.72%    click arrastr mover
  73.61%    click move page

Cluster   4, Size:    49, ISim: 0.408, ESim: 0.015
  70.61%    screen codigo option select write

Cluster   5, Size:   175, ISim: 0.330, ESim: 0.040
  65.83%    scrollbar color foto venu transito

Cluster   6, Size:   156, ISim: 0.252, ESim: 0.010
  77.44%    foto antartida expedicion sin documento

Cluster   7, Size:   730, ISim: 0.077, ESim: 0.015
  98.11%    pag volver categoria publicado listado

Cluster   8, Size:   844, ISim: 0.074, ESim: 0.029
  97.54%    scrollbar color hora la gabinet

Cluster   9, Size:   674, ISim: 0.058, ESim: 0.031
  93.38%    scrollbar lo color que la
--------------------------------------------------------------------------------
```

Figura 26. Primer resultado con un filtrado básico.

Se aprecia en algunos *clusters*, términos propios de lenguajes de programación que una vez analizados se constata que no forman parte del contenido de las páginas *web*. Por otra parte aparecen términos que claramente son palabras vacías en castellano, como "*lo*", "*que*" y "*la*". Por último se constata que existe lematización, cuestionándose hasta qué punto será adecuada.

Una primera actuación consiste en refinar el filtrado de las etiquetas, se observa que al aparecer etiquetas anidadas, con el método empleado no se eliminaban todas ellas, por lo que actuando sobre el fichero previo.pl (*script* en perl, anexo D), se lleva a cabo un filtrado selectivo de las etiquetas, primero las que hacen referencias a *scripts*, después las de tipo *dochdr* y por último el resto. Los *clusters* que se obtienen ahora son los de la figura 27.

```
--------------------------------------------------------------------------------
10-way clustering solution - Cluster Summaries using Cliques...
--------------------------------------------------------------------------------
Cluster   0, Size:    58, ISim: 0.820, ESim: 0.024
  100.00%    function fotografico archivo swapimgrestor preloadimag

Cluster   1, Size:   164, ISim: 0.685, ESim: 0.008
  83.05%    error inexistent encontrado consult servidor

Cluster   2, Size:   196, ISim: 0.718, ESim: 0.064
  95.20%    gaceta local volver listado seccion

Cluster   3, Size:   161, ISim: 0.575, ESim: 0.067
  96.52%    tribuna volver listado seccion actualidad

Cluster   4, Size:    97, ISim: 0.486, ESim: 0.003
  46.05%    click arrastr mover
  59.79%    move page

Cluster   5, Size:   304, ISim: 0.362, ESim: 0.011

  88.36%    document write usal function els

Cluster   6, Size:   183, ISim: 0.313, ESim: 0.006
  73.41%    foto antartida expedicion
  53.37%    foto venu transito

Cluster   7, Size:   365, ISim: 0.318, ESim: 0.045
  88.93%    adelanto volver listado seccion actualidad

Cluster   8, Size:   763, ISim: 0.071, ESim: 0.018
  78.22%    comunicacion gabinet hora la sala

Cluster   9, Size:   823, ISim: 0.049, ESim: 0.020
  86.15%    lo que comunicacion la gabinet
```

Figura 27. Resultado depurando las etiquetas.

9.2.2 Refinamiento en la eliminación de código

Al volver sobre los términos representativos de cada *cluster*, todavía siguen apareciendo términos propios de un lenguaje de programación como son: *function*, *swapimgrestor*, *preloadimag*, *move*, *page*, *document*, *write* o *else*. Todos ellos corresponden a código que no va dentro de etiquetas, por lo que de nuevo hay que hacer una nueva actuación sobre previo.pl. En este caso se modifica su contenido para que pueda eliminar el código de las funciones, los comentarios de programación, las declaraciones de variables, el código relacionado con los condicionales, así como los bloques correspondientes a cualquiera de los elementos anteriores. Los *cluster* que se obtienen con estas modificaciones aparecen en la figura 28.

```
10-way clustering solution - Cluster Summaries using Cliques...
--------------------------------------------------------------------------------
Cluster   0, Size:   165, ISim: 0.679, ESim: 0.009
  82.55%    error inexistent encontrado consult servidor

Cluster   1, Size:   197, ISim: 0.709, ESim: 0.064
  94.52%    gaceta local volver listado seccion

Cluster   2, Size:   102, ISim: 0.545, ESim: 0.015
  63.40%    foto generica mostrando
  69.93%    venu transito foto

Cluster   3, Size:    96, ISim: 0.495, ESim: 0.003
  46.18%    click arrastr mover
  60.42%    move page

Cluster   4, Size:    92, ISim: 0.386, ESim: 0.008
  90.43%    admiss exam entranc usal univers

Cluster   5, Size:   137, ISim: 0.333, ESim: 0.010
  91.53%    foto antartida expedicion sin titulo

Cluster   6, Size:   524, ISim: 0.337, ESim: 0.031
  99.54%    volver listado seccion actualidad dentro

Cluster   7, Size:   140, ISim: 0.304, ESim: 0.021
  87.57%    usal acceso prueba investigacion tlf

Cluster   8, Size:   809, ISim: 0.069, ESim: 0.018
  90.98%    comunicacion gabinet hora la impres

Cluster   9, Size:   827, ISim: 0.049, ESim: 0.020
  82.97%    lo que la comunicacion gabinet
```

Figura 28. Resultando eliminando el código de programación.

9.2.3 Eliminación de palabras vacías del castellano

El siguiente punto de actuación serán las palabras vacías del castellano. Se observa en los *clusters* 8 y 9 de la figura 28, que aparecen términos que clasifican como "*lo*", "*que*", "*la*", dando lugar a los *clusters* con mayor número de documentos, pero que no aportan información sobre el contenido de estos *clusters*. Es necesario aportar una nueva lista de palabras vacías para el castellano (vacias.txt), que unida a la que tiene de forma interna doc2mat (palabras vacías en inglés), eliminará tanto unas como otras, ya que cabe la posibilidad, y de hecho así es en nuestro caso de estudio, de que haya páginas tanto en español como en inglés. Los términos de los nuevos *clusters* son los de la figura 29.

```
10-way clustering solution - Cluster Summaries using Cliques...
--------------------------------------------------------------------------------
Cluster   0, Size:   197, ISim: 0.699, ESim: 0.059
  94.52%    gaceta local volver listado seccion

Cluster   1, Size:   170, ISim: 0.641, ESim: 0.006
  80.35%    error inexistent encontrado consult servidor

Cluster   2, Size:   102, ISim: 0.546, ESim: 0.016
  63.40%    foto generica mostrando
  69.93%    venu transito foto

Cluster   3, Size:    96, ISim: 0.494, ESim: 0.002
  46.18%    click arrastr mover
  60.42%    move page

Cluster   4, Size:    93, ISim: 0.381, ESim: 0.008
  90.54%    admiss exam entranc usal univers

Cluster   5, Size:   138, ISim: 0.314, ESim: 0.010
  91.45%    foto antartida expedicion titulo documento

Cluster   6, Size:   530, ISim: 0.311, ESim: 0.028
  98.72%    volver listado seccion actualidad me

Cluster   7, Size:   144, ISim: 0.298, ESim: 0.019
  89.03%    usal acceso prueba investigacion tlf

Cluster   8, Size:   845, ISim: 0.063, ESim: 0.015
  94.27%    comunicacion gabinet hora impres formato

Cluster   9, Size:   774, ISim: 0.036, ESim: 0.017
  66.20%    comunicacion gabinet curso universidad premio
--------------------------------------------------------------------------------
```

Figura 29. Resultado aportando una lista de palabras vacías.

Ahora se aprecia que los *clusters* 8 y 9 son sustancialmente diferentes, sobre todo el 9 que no aportaba más que un término significativo. Se observan términos carentes de significación como "*me*" en el *cluster* 6, o muy ambiguos como "*admiss*" o "*univers*" en el *cluster* 4. La siguiente actuación consiste en no utilizar lematización para intentar recuperar estos valores, el resultado obtenido es el de la figura 30.

```
10-way clustering solution - Cluster Summaries using Cliques...
--------------------------------------------------------------------------------
Cluster   0, Size:    58, ISim: 0.915, ESim: 0.008
  96.21%    venus transito fotos body universidad

Cluster   1, Size:   165, ISim: 0.677, ESim: 0.006
  83.15%    error consulte servidor inexistente administrador

Cluster   2, Size:   197, ISim: 0.696, ESim: 0.058
  94.52%    gaceta local volver listado seccion

Cluster   3, Size:    96, ISim: 0.493, ESim: 0.002
  46.18%    click mover arrastre
  60.42%    moved page

Cluster   4, Size:    93, ISim: 0.373, ESim: 0.007
  90.54%    admission exams entrance usal university

Cluster   5, Size:   528, ISim: 0.314, ESim: 0.027
  99.02%    volver listado seccion actualidad mes

Cluster   6, Size:   137, ISim: 0.288, ESim: 0.008
  89.34%    antartida expedicion foto titulo fotos

Cluster   7, Size:   153, ISim: 0.260, ESim: 0.015
  86.67%    usal acceso pruebas investigacion tlf

Cluster   8, Size:   191, ISim: 0.110, ESim: 0.016
  26.49%    genericas mostrando fotografico archivo anterior

Cluster   9, Size:  1471, ISim: 0.040, ESim: 0.007
  87.83%    comunicacion gabinete horas universidad impresion
--------------------------------------------------------------------------------
```

Figura 30. Resultado aportando lista de palabras vacías y sin lematización.

Con estos cambios han aparecido términos más comprensibles, "*me*" se ha transformado en "*mes*", "*admiss*" en "*admission*" y "*univers*" en "*university*". Otros *clusters* cuyos términos no presentaban reducción han quedado sin sufrir ninguna alteración. Las siguientes actuaciones modificarán la lista de las palabras vacías para ver su incidencia sobre los *clusters* resultantes, el primer paso será incluir como palabras vacías, los términos "*universidad*", "*university*" y "*usal*", por su gran número de apariciones en las páginas, lo que hace que forme parte de 4 de los 10 *clusters* actuales. El resultado obtenido es el de la figura 31.

```
10-way clustering solution - Cluster Summaries using Cliques...
--------------------------------------------------------------------------------
Cluster   0, Size:    56, ISim: 0.981, ESim: 0.007
  100.00%    venus transito fotos body salamanca

Cluster   1, Size:   165, ISim: 0.677, ESim: 0.005
  83.15%    error consulte servidor inexistente administrador

Cluster   2, Size:   195, ISim: 0.709, ESim: 0.058
  95.49%    gaceta local volver listado seccion

Cluster   3, Size:    96, ISim: 0.493, ESim: 0.001
  46.18%    click mover arrastre
  60.42%    moved page

Cluster   4, Size:    94, ISim: 0.345, ESim: 0.005
  83.62%    admission exams entrance registration courses

Cluster   5, Size:   530, ISim: 0.311, ESim: 0.027
  98.72%    volver listado seccion actualidad mes

Cluster   6, Size:   129, ISim: 0.292, ESim: 0.014
  93.18%    acceso pruebas investigacion mapa agenda

Cluster   7, Size:   138, ISim: 0.284, ESim: 0.007
  88.84%    antartida expedicion foto titulo fotos

Cluster   8, Size:   254, ISim: 0.078, ESim: 0.015
  19.45%    genericas mostrando fotografico archivo siguiente

Cluster   9, Size:  1432, ISim: 0.040, ESim: 0.007
  87.68%    comunicacion gabinete horas impresion formato
--------------------------------------------------------------------------------
```

Figura 31. Resultado con la lista de palabras vacías, vacias1.txt.

Se han modificado los cuatros *clusters* en los que aparecían esos términos y además un *cluster* más, el número 8 que ha incrementado considerablemente el número de documentos. Los cinco *clusters* restantes no han sufrido ningún cambio, ni en sus términos, ni en el número de documentos de cada uno de ellos. Analizando de nuevo los términos representativos de los *clusters* se observa, que aparecen las palabras "*siguiente*" y "*volver*", habituales en las páginas web como elementos para pasar a la siguiente página o volver a la anterior. También se observa en el *cluster* con más elementos, el 9, que casi contiene tantos documentos como la suma del resto de *clusters*, que los términos "*comunicación*" y "*protocolo*" se refieren al "*gabinete de comunicación y protocolo*", al que pertenecen la mayoría de las páginas. Eliminando estos cuatro términos se obtienen los *clusters* de la figura 32.

```
10-way clustering solution - Cluster Summaries using Cliques...
--------------------------------------------------------------------------------
Cluster   0, Size:   165, ISim: 0.677, ESim: 0.005
  83.15%    error consulte servidor inexistente administrador

Cluster   1, Size:   217, ISim: 0.610, ESim: 0.050
  91.80%    gaceta local listado seccion actualidad

Cluster   2, Size:   103, ISim: 0.540, ESim: 0.010
  62.78%    fotos genericas mostrando
  69.90%    venus transito fotos

Cluster   3, Size:    97, ISim: 0.484, ESim: 0.001
  46.05%    click mover arrastre
  59.79%    moved page

Cluster   4, Size:    94, ISim: 0.345, ESim: 0.005
  83.62%    admission exams entrance registration courses

Cluster   5, Size:   125, ISim: 0.303, ESim: 0.014
  93.76%    acceso pruebas investigacion mapa agenda

Cluster   6, Size:   138, ISim: 0.286, ESim: 0.007
  88.99%    antartida expedicion foto titulo fotos

Cluster   7, Size:   508, ISim: 0.296, ESim: 0.026
  98.66%    listado seccion actualidad mes dia

Cluster   8, Size:   452, ISim: 0.063, ESim: 0.012
  40.09%    sala prensa rueda rectorado retratos

Cluster   9, Size:  1190, ISim: 0.026, ESim: 0.010
  51.75%    facultad horas curso formato impresion
--------------------------------------------------------------------------------
```

Figura 32. Resultado con la lista de palabras vacías, vacias2.txt.

Con el conjunto de documentos normalizados de esta forma y con las palabras vacías eliminadas como se ha indicado anteriormente se dispone del documento resumen base que es el que se va a utilizar en las siguientes fases del proceso.

9.2.4 Tiempos de computación de la fase de filtrado y normalización y de la de concatenación

La fase de filtrado y normalización guarda una estrecha relación con la fase de concatenación, porque una vez que un documento se procesa pasa a formar parte del fichero resultante de resultados antes de pasar a procesar el siguiente documento.

Estas primeras fases son de las que mayores tiempos de computación producen, junto con alguna opción determinada de algún método concreto de obtención de *clusters* que se estudiará más adelante. Los elevados tiempos se deben al gran número de operaciones de entrada/salida que se producen sobre disco. Cada uno de los ficheros html hay que procesarlo individualmente y secuencialmente. Se extraen los comentarios y el código, se normalizan los caracteres, se eliminan espacios, tabuladores y saltos de líneas y se obtiene un documento abreviado en única línea. Estos

documentos que se van obteniendo se añaden a un fichero de texto, de forma que el resultante de cada documento html será una nueva línea del fichero global, que eran separadas por saltos de línea.

Este proceso se puede aprovechar para guardar distintas características de cada documento html en diversos ficheros para posteriores utilizaciones, como puede ser el título del documento, o su ruta completa o alguna asociación de tipo número-documento. Los tiempos que se producen en esta fase son proporcionales al número de documentos, en la tabla 3 se recogen los tiempos para nuestros casos de estudio.

Tabla 3. Tiempos de computación de la fases de filtrado y normalización, y concatenación.

	Tiempo	**Tamaño del fichero resultante**
Caso de estudio A	4 min. y 29 seg.	20,2 KBytes
Caso de estudio B	3 horas y 12 seg.	217.833 KBytes

9.3 Análisis de la incidencia de los métodos de *clustering* y de las funciones criterio en los resultados

9.3.1 Calidad de los métodos de *clustering* y de las funciones criterio

Partiendo del fichero matriz de pesos aceptado como resultado de las fases anteriores, se pasa a estudiar cómo afecta en el tamaño y calidad de los *clusters* que se obtienen, el método de *clustering* utilizado. También se analiza la influencia de las distintas funciones criterio que pueden ser aplicadas a cada método.

Las pruebas se dividen en dos grupos, unas las que recogen los métodos particionales (*rb*, *rbr* y *direct*) y otras las que se refieren a los métodos acumulativos

(*agglo* y *bagglo*). En el caso de los métodos particionales las posibles funciones criterio han sido: i1, i2, e1, g1, g1p, h1 y h2. Para los métodos particionales, además de las anteriores funciones que también son válidas, se han utilizado las funciones específicas: *slink*, *clink* y *upgma*. El significado de las funciones que valen para los dos tipos de *clustering* se recoge en la tabla 4. La notación empleada es: *k* es el número total de *clusters*, *S* es el número total de objetos que van a ser clasificados, S_i es el conjunto de objetos del *cluster i*, n_i es el número de objetos en el *cluster i*, *v* y *u* son dos objetos y *sim(v, u)* es la similitud entre estos dos objetos.

Tabla 4. Definiciones de las funciones criterio.

i1	maximiza $\sum_{i=1}^{k} \frac{1}{n_i} \left(\sum_{v,u \in S_i} sim(v,u) \right)$
i2	maximiza $\sum_{i=1}^{k} \sqrt{\sum_{v,u \in S_i} sim(v,u)}$
e1	minimiza $\sum_{i=1}^{k} n_i \frac{\sum_{v \in S_i, u \in S} sim(v,u)}{\sqrt{\sum_{v,u \in S_i} sim(v,u)}}$
g1	minimiza $\sum_{i=1}^{k} \frac{\sum_{v \in S_i, u \in S} sim(v,u)}{\sum_{v,u \in S_i} sim(v,u)}$
g1p	minimiza $\sum_{i=1}^{k} n_i^2 \frac{\sum_{v \in S_i, u \in S} sim(v,u)}{\sum_{v,u \in S_i} sim(v,u)}$
h1	maximiza $\frac{i1}{e1}$
h2	maximiza $\frac{i2}{e1}$

El significado de las funciones específicas para el *clustering* acumulativo se recoge en la tabla 5. Estas funciones miden la similitud entre dos *clusters* S_i y S_j.

Tabla 5. Definiciones de las funciones criterio específicas del *clustering* acumulativo.

slink	$sim_{slink}(Si, Sj) = \max_{d_i \in S_i, d_j \in S_j}\{\cos(d_i, d_j)\}$
clink	$sim_{clink}(Si, Sj) = \min_{d_i \in S_i, d_j \in S_j}\{\cos(d_i, d_j)\}$
upgma	$sim_{upgma}(Si, Sj) = \frac{1}{n_i n_j} \sum_{d_i \in S_i, d_j \in S_j} \cos(d_i, d_j)$

Los resultados obtenidos para los métodos particionales, obteniendo 10 *clusters*, se recogen en la tabla 6, mientras que los resultados para métodos acumulativos se recogen en la tabla 7.

Tabla 6. Resultados de los métodos particionales, obteniendo 10 *clusters* (caso A).

	i1		i2		e1		g1		g1p		h1		h2	
	Size	**ISim**	**Size**	**ISim**	**Size**	**ISim**	**Size**	**ISim**	**Size**	**ISim**	**Size**	**ISim**	**Size**	**ISim**
rb	56	0,981	166	0,670	161	0,707	30	1,000	288	0,405	131	0,980	195	0,701
	46	0,990	197	0,683	222	0,596	46	0,990	453	0,331	178	0,774	180	0,569
	131	0,970	103	0,537	488	0,310	134	0,942	326	0,186	137	0,658	176	0,488
	58	0,919	97	0,484	242	0,216	58	0,919	320	0,135	102	0,544	337	0,303
	38	0,855	91	0,364	291	0,148	59	0,894	339	0,099	96	0,493	151	0,235
	175	0,784	123	0,309	182	0,084	38	0,855	253	0,091	154	0,251	259	0,202
	137	0,658	137	0,290	472	0,054	88	0,370	294	0,072	404	0,259	252	0,124
	154	0,251	530	0,287	431	0,047	725	0,311	198	0,068	209	0,200	421	0,060
	387	0,272	437	0,061	336	0,042	134	0,299	342	0,054	266	0,100	737	0,037
	1907	0,024	1208	0,025	264	0,032	1777	0,025	276	0,051	1412	0,023	381	0,033
	0,236	0,67	**0,216**	0,394	**0,187**	0,224	**0,221**	0,661	**0,16**	0,149	**0,228**	0,428	**0,198**	0,275
	Size	**ISim**	**Size**	**ISim**	**Size**	**ISim**	**Size**	**ISim**	**Size**	**ISim**	**Size**	**ISim**	**Size**	**ISim**
rbr	55	1,000	165	0,677	229	0,569	30	1,000	305	0,340	129	1,000	195	0,697
	46	0,990	197	0,683	483	0,314	46	0,990	481	0,314	178	0,774	178	0,481
	131	0,970	103	0,537	281	0,243	133	0,955	387	0,138	137	0,658	228	0,360
	58	0,919	97	0,484	338	0,123	58	0,919	359	0,110	102	0,544	346	0,295
	38	0,855	92	0,359	330	0,119	59	0,894	337	0,101	96	0,493	282	0,173
	175	0,784	530	0,287	362	0,072	38	0,855	263	0,087	404	0,259	245	0,109
	137	0,658	157	0,240	234	0,054	90	0,367	275	0,078	189	0,192	298	0,090
	154	0,251	148	0,244	200	0,050	725	0,311	200	0,070	258	0,148	401	0,065
	392	0,269	429	0,061	383	0,051	134	0,298	194	0,066	554	0,044	272	0,048
	1903	0,024	1171	0,026	249	0,051	1776	0,025	288	0,062	1042	0,028	644	0,041
	0,236	0,672	**0,215**	0,36	**0,166**	0,165	**0,221**	0,661	**0,152**	0,137	**0,224**	0,414	**0,186**	0,236

	i1		i2		e1		g1		g1p		h1		h2	
	Size	**ISim**	**Size**	**ISim**	**Size**	**ISim**	**Size**	**ISim**	**Size**	**ISim**	**Size**	**ISim**	**Size**	**ISim**
direct	129	1,000	195	0,697	309	0,429	163	0,693	722	0,314	129	1,000	198	0,685
	46	0,990	103	0,537	403	0,346	103	0,537	367	0,151	56	0,981	218	0,402
	306	0,561	97	0,484	283	0,243	96	0,493	322	0,106	98	0,473	521	0,294
	96	0,493	253	0,365	326	0,132	90	0,367	252	0,107	175	0,492	274	0,183
	87	0,378	527	0,290	295	0,119	730	0,307	254	0,085	544	0,331	257	0,112
	111	0,357	122	0,265	359	0,073	131	0,306	264	0,087	231	0,180	238	0,095
	177	0,291	157	0,240	290	0,062	143	0,253	313	0,084	255	0,153	208	0,082
	404	0,261	363	0,070	201	0,058	273	0,093	218	0,075	380	0,051	361	0,071
	128	0,192	316	0,041	252	0,054	593	0,039	204	0,074	439	0,041	477	0,046
	1605	0,023	956	0,029	371	0,051	767	0,025	173	0,068	782	0,031	337	0,046
	0,222	0,455	**0,201**	0,302	**0,164**	0,157	**0,2**	0,311	**0,148**	0,115	**0,207**	0,373	**0,181**	0,202

Cada uno de los valores de las columnas *size*, representa el número de documentos de un *cluster*. Cada uno de los valores de las columnas *ISim*, representa la similitud media interna de un *cluster*. El último valor de la columna *ISim*, es la ISim media de los valores que tiene por encima. El último valor de la columna Size, contiene la media ponderada de los valores ISim con respecto al número de documentos por *cluster*. Aunque en un principio pudiera pensarse en la media de la similitud interna de los *clusters* obtenidos para un método, como medida de su calidad, no es un dato preciso aunque sí nos puede dar una orientación. Así por ejemplo, si un método obtuviera varios *clusters* con un único documento, su similitud interna (ISim) sería 1 para cada uno de ellos, lo que haría que la media para ese método fuera un valor muy elevado, sin que por ello se hubiera dado una buena clasificación, es el caso que se produce en el método slink, de la tabla 7. Es por ello, que se necesita que intervenga de alguna forma el número de elementos clasificados por *cluster*, por eso se toma como medida la media ponderada de la similitud de cada *cluster* con respecto a los elementos clasificados.

Tabla 7. Resultados de los métodos acumulativos, obteniendo 10 *clusters*, caso A.

	i1		i2		e1		g1		g1p		h1		h2		slink		clink		upgma	
	Size	**ISim**	**Size**	**ISim**	**Size**	**ISim**	**Size**	**ISim**	**Size**	**ISim**	**Size**	**ISim**	**Size**	**ISim**	**Size**	**ISim**	**Size**	**ISim**	**Size**	**ISim**
agglo	2	1,000	2	1,000	2	1,000	2	1,000	2	1,000	561	0,027	542	0,041	2	1,000	192	0,192	2	1,000
	565	0,282	140	0,352	459	0,042	58	0,919	298	0,077	129	1,000	311	0,067	30	1,000	58	0,919	96	0,493
	129	1,000	535	0,294	203	0,093	743	0,297	757	0,276	165	0,275	536	0,294	1	1,000	112	0,353	728	0,309
	153	0,850	129	1,000	502	0,054	131	0,980	374	0,060	153	0,850	381	0,073	1	1,000	24	0,160	483	0,134
	55	1,000	1413	0,028	423	0,064	57	0,956	281	0,058	1198	0,031	129	1,000	1	1,000	8	0,386	267	0,191
	58	0,919	382	0,062	363	0,049	38	0,855	405	0,055	58	0,919	216	0,101	2	0,986	8	0,348	12	0,127
	46	0,990	137	0,278	231	0,329	31	0,943	291	0,088	55	1,000	168	0,755	1	1,000	5	0,292	5	0,357
	118	0,341	183	0,680	236	0,069	86	0,370	310	0,190	205	0,197	483	0,039	1	1,000	5	0,414	55	0,053
	180	0,219	58	0,919	362	0,376	1741	0,026	179	0,116	139	0,632	139	0,260	1	1,000	28	0,368	1350	0,030
	1783	0,023	110	0,282	308	0,412	202	0,212	192	0,067	426	0,250	184	0,102	3049	0,033	2649	0,038	91	0,100
	0,225	0,662	**0,21**	0,49	**0,151**	0,249	**0,207**	0,656	**0,134**	0,199	**0,227**	0,518	**0,188**	0,273	**0,046**	0,902	**0,082**	0,347	**0,144**	0,279
	Size	**ISim**	**Size**	**ISim**	**Size**	**ISim**	**Size**	**ISim**	**Size**	**ISim**	**Size**	**ISim**	**Size**	**ISim**	**Size**	**ISim**	**Size**	**ISim**	**Size**	**ISim**
bagglo	137	0,658	938	0,033	315	0,074	96	0,493	434	0,135	470	0,039	407	0,330	2	1,000	568	0,071	97	0,484
	129	1,000	197	0,687	403	0,334	723	0,313	299	0,049	131	0,980	335	0,073	103	0,537	97	0,484	724	0,312
	193	0,708	199	0,259	289	0,083	103	0,537	249	0,113	503	0,057	276	0,042	1	1,000	6	0,246	384	0,197
	148	0,515	132	0,967	446	0,056	136	0,914	261	0,041	197	0,687	553	0,035	1	1,000	4	0,365	240	0,226
	103	0,537	488	0,035	313	0,079	92	0,357	326	0,045	237	0,230	333	0,104	1	1,000	141	0,264	102	0,103
	135	0,296	352	0,287	285	0,222	136	0,293	286	0,074	727	0,029	309	0,072	1	1,000	90	0,119	132	0,084
	138	0,269	135	0,296	284	0,455	157	0,228	276	0,092	137	0,658	284	0,455	1	1,000	103	0,537	56	0,254
	96	0,493	175	0,492	240	0,120	173	0,127	237	0,104	330	0,075	249	0,091	1	1,000	2	0,624	165	0,036
	209	0,236	161	0,218	245	0,049	132	0,084	284	0,455	148	0,515	132	0,967	1	1,000	1354	0,028	221	0,045
	1801	0,023	312	0,061	269	0,043	1341	0,026	437	0,311	209	0,236	211	0,157	2977	0,033	724	0,312	968	0,037
	0,227	0,474	**0,208**	0,334	**0,154**	0,152	**0,204**	0,337	**0,15**	0,142	**0,203**	0,351	**0,181**	0,233	**0,053**	0,857	**0,148**	0,305	**0,159**	0,178

Comparando los métodos particionales frente a los acumulativos (tabla 6 frente a tabla 7) se obtienen los mejores resultados dentro de los particionales. Si analizamos los resultados para las siete funciones criterio que pueden ser utilizadas tanto con los métodos particionales como con los acumulativos, para cada una de estas funciones, siempre están por delante dos de los tres métodos particionales, *rb* y *rbr*, pero no así el tercero, *direct*, sino que se pone por delante el método acumulativo *agglo* en 5 de las 7 ocasiones, véase figura 33.

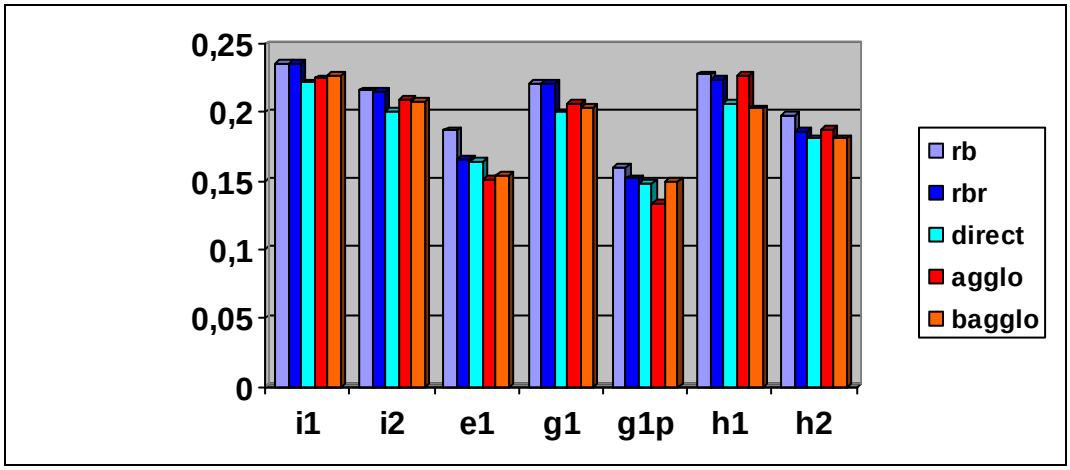

Figura 33. Gráfico de incidencia de los métodos de *clustering* (10 *clusters*, caso A).

Incluso en una ocasión, el método *agglo*, se pone por delante del método *rbr*, pero detrás de *rb*, es cuando se utiliza la función criterio h1.

Si estudiamos ahora el comportamiento de las funciones criterio para cada método, las que mejor comportamiento presentan son i1, h1 e i2, en este orden.

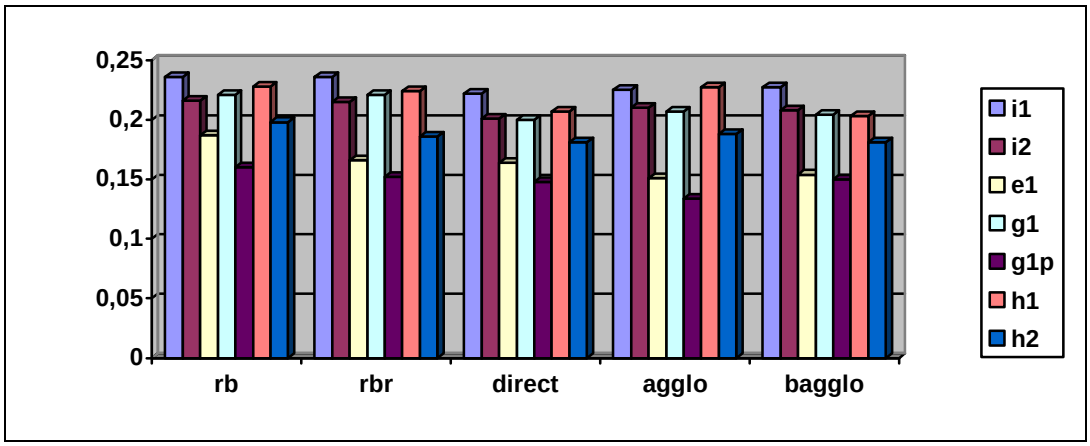

Figura 34. Gráfico de incidencia de las funciones criterio comunes para todos los métodos (10 *clusters*, caso A).

Podemos ver en la figura 34, que la función que mejores resultados presenta es i1, está por delante en todo los métodos excepto en *agglo*, que es superada por h1. En segundo lugar se encuentra h1.

Por otra parte es necesario examinar el comportamiento de las funciones específicas para los métodos acumulativos, como son *slink*, *clink* y *upgma*, junto con las funciones que se pueden utilizar para todos los métodos de *clustering*. En la figura 35 se recoge su comportamiento.

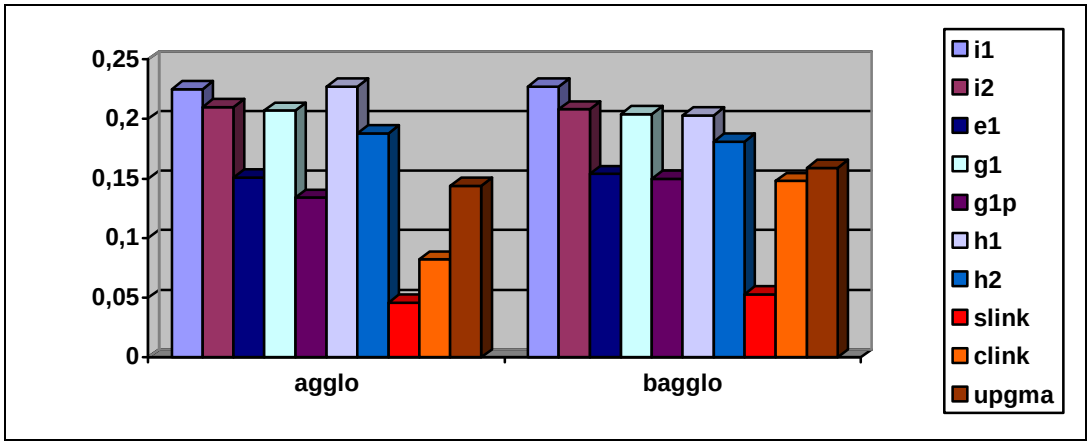

Figura 35. Gráfico de incidencia de las funciones criterio para métodos acumulativos (10 *clusters*, caso A).

Como se puede apreciar, estas funciones no destacan por su comportamiento con respecto al resto de funciones, es más, tanto *slink* como *clink* ofrecen unos resultados

muy pobres, no así *upgma*, que se pone a la altura de alguna de las siete funciones restantes, pero sin superar en ningún caso a las ya destacadas i1, i2 y h1.

9.3.1.1 Caso de estudio B

Como ya se ha indicado este caso se caracteriza por el elevado número de páginas. El comportamiento en cuanto a los métodos de *clustering* sólo puede analizarse para los métodos particionales ya que los acumulativos presentan unos tiempos y requisitos inviables como se indican en el apartado siguiente. Los métodos particionales presentan un comportamiento similar a los del caso de estudio A, destacan los métodos rb y rbr sobre direct para todas las funciones criterio. Los resultados se recogen en la tabla 8.

Tabla 8. Resultados de los métodos particionales, obteniendo 10 *clusters* (caso B).

	i1		i2		e1		g1		g1p		h1		h2	
rb	**Size**	**ISim**	**Size**	**ISim**	**Size**	**ISim**	**Size**	**ISim**	**Size**	**ISim**	**Size**	**ISim**	**Size**	**ISim**
	3857	1,000	3858	0,999	3861	0,998	95244	3,890	11654	0,768	3857	1,000	3866	0,995
	3856	1,000	3860	0,998	3856	1,000	1	1,000	15603	0,733	3856	1,000	3856	1,000
	3859	0,999	3865	0,990	4327	0,813	1	1,000	34256	0,269	3859	0,999	3844	0,999
	3843	1,000	3877	0,963	11573	0,777	1	1,000	5040	0,605	3844	0,999	11574	0,777
	3798	0,999	11589	0,775	15367	0,747	2	1,000	5157	0,575	3804	0,997	11667	0,791
	3852	1,000	34767	0,262	30718	0,313	3	1,000	7951	0,262	3852	1,000	4487	0,752
	7714	0,869	12886	0,667	10467	0,055	152	0,989	6584	0,088	7715	0,869	34805	0,262
	7696	0,846	6749	0,097	14929	0,019	351	0,565	6539	0,036	7821	0,831	9460	0,066
	28993	0,333	6328	0,042	7160	0,005	86	0,145	5330	0,028	31017	0,308	15680	0,019
	39297	0,012	18976	0,014	4497	0,044	10914	0,045	8641	0,022	37130	0,012	7516	0,005
	0,435	0,806	**0,404**	0,581	**0,397**	0,477	**3,479**	1,063	**0,364**	0,339	**0,433**	0,802	**0,405**	0,567
rbr	**Size**	**ISim**	**Size**	**ISim**	**Size**	**ISim**	**Size**	**ISim**	**Size**	**ISim**	**Size**	**ISim**	**Size**	**ISim**
	3857	1,000	3883	0,987	4004	0,940	95289	3,822	29459	0,328	3857	1,000	3868	0,996
	3857	1,000	3891	0,984	3926	0,955	1	1,000	16556	0,666	3860	0,999	3863	0,973
	3860	0,999	3864	0,990	11575	0,777	1	1,000	12935	0,640	3861	0,999	3922	0,968
	3843	1,000	3915	0,947	15337	0,749	1	1,000	5580	0,500	3843	1,000	4320	0,816
	3798	0,999	11613	0,773	30081	0,321	2	1,000	5796	0,469	3804	0,997	11629	0,794
	3853	0,999	12547	0,699	4917	0,645	3	1,000	6441	0,387	3853	0,999	11577	0,777
	7714	0,869	32737	0,286	7029	0,081	152	0,989	7076	0,078	7715	0,869	31127	0,307
	7662	0,850	6675	0,099	9790	0,020	344	0,581	5561	0,029	7728	0,843	10678	0,052
	29002	0,333	6429	0,041	6067	0,054	56	0,330	7157	0,026	35298	0,257	17204	0,008
	39309	0,012	21201	0,015	14025	0,011	10906	0,045	10194	0,016	32936	0,011	8567	0,044
	0,435	0,806	**0,408**	0,582	**0,394**	0,455	**3,42**	1,077	**0,356**	0,314	**0,428**	0,797	**0,41**	0,574
direct	**Size**	**ISim**	**Size**	**ISim**	**Size**	**ISim**	**Size**	**ISim**	**Size**	**ISim**	**Size**	**ISim**	**Size**	**ISim**
	589	0,950	31598	0,301	19344	0,703	362	0,528	30324	0,318	621	0,913	19472	0,697
	700	0,887	19302	0,654	30044	0,321	1417	0,523	21516	0,588	1870	0,562	30688	0,313
	30743	0,658	20153	0,657	19289	0,654	474	0,467	20900	0,567	19247	0,708	19291	0,654
	7716	0,723	2559	0,343	3423	0,133	1002	0,437	4961	0,092	30191	0,320	3863	0,142
	2842	0,404	1264	0,302	4568	0,108	2425	0,376	6776	0,058	19288	0,654	2293	0,099
	26702	0,356	5406	0,090	2759	0,069	344	0,284	4396	0,033	2792	0,189	5845	0,076
	981	0,195	3098	0,019	4734	0,060	2118	0,213	3515	0,067	8518	0,021	8671	0,020
	597	0,096	3400	0,018	4930	0,020	925	0,061	5827	0,030	5036	0,015	4250	0,014
	22980	0,007	9021	0,017	8463	0,022	15655	0,010	3295	0,014	10234	0,010	5551	0,011
	12905	0,050	10954	0,032	9201	0,007	82033	-1,000	5245	0,012	8958	0,044	6831	0,050
	0,362	0,433	**0,354**	0,243	**0,352**	0,21	**-0,738**	0,19	**0,334**	0,178	**0,363**	0,344	**0,353**	0,208

En cuanto a las funciones el mejor comportamiento de nuevo es para i1 y h1, seguido de i2, los valores de calidad con respecto al caso son más elevados para las mismas condiciones, es decir fijando un método y una función. Es debido a que en este caso de estudio existen más páginas repetidas.

Un comportamiento anormal lo presenta la función g1, ya que obtiene valores de similitud superior a uno para los métodos rb y rbr e inferiores a cero (negativos) para el método direct. Valores no posibles, ya que el rango de valores es [0,1]. Suponemos que puede tratarse de un problema de desbordamiento en alguno de los tipos de variables en la codificación del algoritmo. Para valores superiores del número de *clusters* los resultados concuerdan con los indicados, recogiéndose en el apéndice B para 20 y 30 *clusters*.

9.3.2 Estudio de los tiempos de computación *clustering*

La complejidad computacional en los métodos particionales suele ser muy baja, la clasificación binaria de un conjunto de documentos suele ser lineal con el número de documentos, y en la mayoría de los casos el número de iteraciones necesarias para los algoritmos más codiciosos es pequeña, menores a 20 y se puede considerar independiente del número de documentos. Si se asume que durante cada bisección, cada *cluster* tiene una fracción de los documentos originales, entonces para las n-1 bisecciones el tiempo será $O(n \log n)$.

En los métodos acumulativos el proceso lleva dos pasos. En el primer paso es obtener la similitud entre parejas del conjunto de documentos, que tiene $O(n^2)$. El segundo paso es la selección repetida de las parejas de *clusters* que mejor optimizan la función criterio. Una forma sencilla de hacerlo es volver a calcular las ganancias que se producen al mezclar cada par de *clusters* en cada nivel de acumulación y seleccionar el par más prometedor. Durante el paso l, esto requiere una complejidad de $O((n-l)^2)$, llevando a una complejidad total de $O(n^3)$.

Este proceso puede mejorarse para las funciones i1, i2, e1, g1 y g1p, pero no para h1 y h2. Para estas funciones la similitud de una pareja archivada al mezclar un par de *clusters i* y *j* no cambia durante los diferentes pasos de acumulación con tal de que *i* y *j* no sean seleccionados para la mezcla. Por lo que para cada par de *clusters* se puede almacenar su similitud en una cola de prioridades. Cuando un par de *clusters i* y *j* es seleccionado para mezclarse en un *p*, la prioridad encolada se actualiza para que cualquier ganancia correspondiente a los *clusters* implicados sea eliminada, y las ganancias de la mezcla del resto de *clusters* con el nuevo *cluster p* sean insertadas. Si la prioridad de la cola se implementa utilizando una pila binaria, la complejidad de esta operación es $O((n-l) \log(n-l))$ y la complejidad total para los *n-l* pasos acumulativos es $O(n^2 \log n)$.

Para las funciones h1 y h2, la complejidad no puede reducirse porque la mejora en el valor global de las funciones criterio cuando se mezclan dos *clusters* tiende a cambiar para todos los pares de *clusters*, por lo que no pueden calcularse a priori para almacenarlos en la cola. Por lo tanto el inconveniente que presentan las funciones h1 y h2 para métodos acumulativos es su alta complejidad, $O(n^3)$, siendo *n* el número de objetos, mientras que para el resto de funciones la complejidad es $O(n^2 \log n)$, lo que para nuestro caso de estudio, que no es una muestra excesivamente grande, supone pasar de unos 700 segundos a uno o dos segundos. Los tiempos de computación experimentales de estas funciones, para 10, 20 y 30 *clusters*, se recogen en la tabla 9 (para 20, sombreados en gris y para 30, en azul). Como se puede apreciar todos los métodos particionales tienen menores tiempos de computación que los acumulativos, ya que su complejidad es $O(n \log n)$, aproximadamente entre cuatro y ocho veces menores. Dentro de los particionales, el método *direct* tiene unos tiempos casi dobles a *rb* y *rbr*. En los acumulativos los tiempos del método *agglo* son ligeramente inferiores a los de *bagglo*.

Tabla 9. Tiempos de computación para 10, 20 y 30 *clusters* (caso A).

		rb	rbr	direct	agglo	bagglo
i1	**10**	0,703s	0,703s	1,062s	7,000s	7,625s
	20	1,171s	1,234s	2,109s	6,906s	7,593s
	30	1,485s	1,578s	3,266s	7,062s	7,625s
i2	**10**	0,593s	0,656s	1,156s	7,391s	8,046s
	20	0,890s	1,046s	2,484s	7,359s	8,015s
	30	1,031s	1,344s	3,672s	7,391s	8,046s
e1	**10**	0,828s	0,953s	1,797s	7,828s	8,453s
	20	1,109s	1,406s	3,562s	7,875s	8,484s
	30	1,218s	1,719s	5,484s	7,875s	8,468s
g1	**10**	0,672s	0,719s	1,610s	9,406s	10,297s
	20	1,203s	1,391s	2,750s	9,265s	10,234s
	30	1,281s	1,578s	4,156s	9,359s	10,203s
g1p	**10**	0,859s	0,984s	1,468s	7,265s	7,906s
	20	1,062s	1,312s	2,875s	7,172s	7,859s
	30	1,203s	1,578s	4,344s	7,297s	7,875s
h1	**10**	0,781s	0,953s	2,015s	10m 6,578s	10m 1,546s
	20	1,406s	1,687s	4,078s	10m 6,437s	10m 1,375s
	30	1,609s	2,203s	6,234s	10m 6,609s	10m 2,687s
h2	**10**	0,875s	0,969s	1,828s	9m 41,422s	9m 17,250s
	20	1,125s	1,484s	3,718s	9m 21,547s	9m 17,078s
	30	1,312s	1,828s	5,719s	9m 35,375s	9m 19,812s
slink	**10**				6,140s	7,187s
	20				6,046s	6,984s
	30				6,234s	6,906s
clink	**10**				6,375s	7,469s
	20				6,375s	7,296s
	30				6,343s	7,172s
upgma	**10**				6,328s	7,390s
	20				6,343s	7,234s
	30				6,359s	7,141s

Se puede observar también que al aumentar el número de *clusters*, los tiempos de los métodos particionales aumentan, al doblar el número de *clusters* (de 10 a 20) los tiempos también son casi dobles, mientras que en los métodos acumulativos la tendencia es la contraria, al aumentar el número de *clusters*, menor es el número de niveles que hay que ir mezclando, obtener 20 en lugar de 10 supone evitarnos un paso, por lo que los tiempos están muy próximos y hay muy poca variación. La tendencia de ambos métodos para una misma función se muestra en la figura 36.

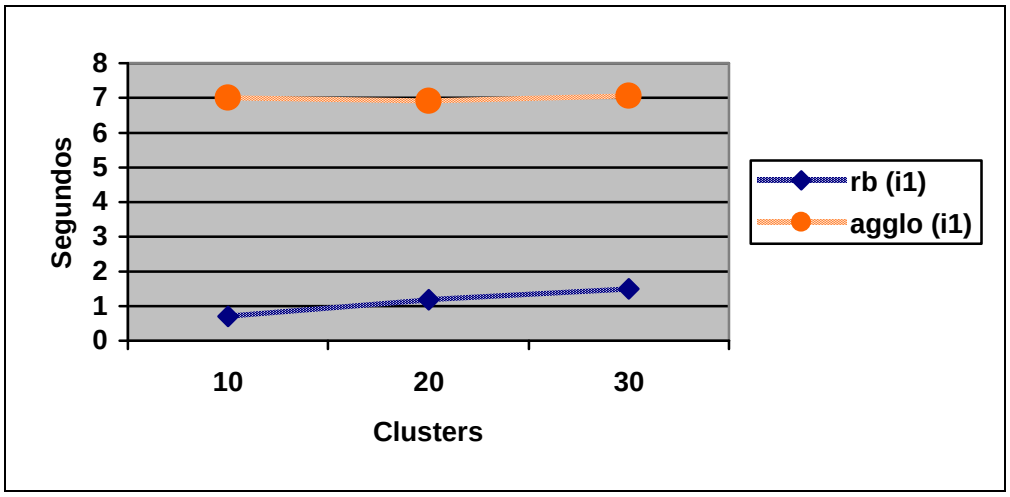

Figura 36. Evolución de los tiempos de computación.

Para cualquiera de estos métodos la memoria física del ordenador es una característica primordial, ejecutar los algoritmos en un ordenador con el doble de RAM supone reducir los tiempos de computación a la mitad (dato obtenido al calcular en dos máquinas una con 512 Mbytes y otra con 1024 Mbytes).

Para el caso de la clasificación, los tiempos de los métodos acumulativos con las funciones h1 y h2 son tiempos asumibles, ya que el proceso de obtención de los *clusters*, llámense directorios, se puede hacer de forma programada, pero no son tiempos aceptables cuando se espera una respuesta inmediata, como sería el caso de las consultas.

Si aumentamos considerablemente el número de páginas web a procesar, sería el caso de estudio B, los tiempos de computación y los requisitos hardware toman un cariz muy distinto, llegando a ser determinantes. En la tabla 10 se recogen los tiempos para el caso B, pero sólo de los métodos particionales, ya que el consumo de memoria de los métodos acumulativos, es tal, que se aborta automáticamente la aplicación solicitando más de 4 GBytes de memoria.

Tabla 10. Tiempos de computación de métodos particionales para 10, 20 y 30 *clusters* (caso B).

		rb	rbr	direct
i1	**10**	1m 7,094s	1m 9,609s	2m 14,078s
	20	1m 43,750s	2m 6,593s	5m 22,765s
	30	2m 0,750s	2m 43,640s	9m 40,719s
i2	**10**	1m 5,578s	1m 12,813s	2m 20,265s
	20	1m 25,265s	1m 54,500s	5m 31,593s
	30	1m 38,688s	2m 31,532s	9m 57,109s
e1	**10**	1m 15,593s	1m 33,031s	3m 14,641s
	20	1m 49,500s	2m 30,578s	7m 30,359s
	30	2m 27,828s	3m 26,500s	13m 15,125s
g1	**10**	1m 5,234s	1m 9,125s	2m 59,672s
	20	1m 9,500s	1m 46,968s	7m 0,281s
	30	1m 13,406	2m 18,156s	12m 27,156s
g1p	**10**	1m 39,906s	1m 46,235s	3m 2,531s
	20	2m 1,500s	2m 36,219s	7m 5,969s
	30	2m 27,828s	3m 35,391s	12m 36,015
h1	**10**	1m 5,562s	1m 23,906s	3m 20,234s
	20	2m 1,500s	2m 44,797s	7m 50,594s
	30	2m 22,719s	3m 34,485s	13m 48,156
h2	**10**	1m 23,734s	1m 41,391s	3m 14,234s
	20	1m 47,234s	2m 28,968s	7m 34,109s
	30	2m 12,906s	3m 21,953s	13m 20,688

9.3.3 Incidencia del número de *clusters* seleccionados

Si aumentamos el número de *clusters* obtenidos pasando de 10 a 20 o a 30 no se producen cambios sustanciales ni en cuanto a los métodos ni en cuanto a las funciones criterio. Los resultados obtenidos para los métodos particionales se recogen en la tabla 11, mientras que los resultados para métodos acumulativos se recogen en la tabla 12. En el apéndice A se recogen los resultados para 30 *clusters*.

Tabla 11. Resultados de los métodos particionales, obteniendo 20 *clusters* (caso A).

	i1		i2		e1		g1		g1p		h1		h2	
	Size	**ISim**	**Size**	**ISim**	**Size**	**ISim**	**Size**	**ISim**	**Size**	**ISim**	**Size**	**ISim**	**Size**	**ISim**
rb	30	1,000	30	1,000	161	0,707	1	1,000	145	0,618	30	1,000	46	0,990
	56	0,981	46	0,990	138	0,646	1	1,000	194	0,441	56	0,981	131	0,980
	46	0,990	57	0,956	222	0,596	30	1,000	288	0,405	46	0,990	57	0,956
	131	0,970	58	0,919	102	0,544	30	1,000	114	0,367	131	0,980	195	0,701
	58	0,919	136	0,916	175	0,492	46	0,990	139	0,240	58	0,919	139	0,642
	38	0,855	39	0,817	102	0,390	57	0,956	157	0,238	38	0,855	97	0,484
	27	0,836	197	0,683	150	0,273	66	0,975	163	0,223	178	0,774	176	0,488
	175	0,784	137	0,658	140	0,263	28	0,920	326	0,186	137	0,658	49	0,397
	22	0,714	179	0,474	141	0,262	68	0,911	186	0,170	33	0,455	103	0,387
	137	0,658	91	0,364	73	0,264	38	0,855	153	0,155	176	0,488	156	0,243
	30	0,625	123	0,309	105	0,100	41	0,454	107	0,108	90	0,371	151	0,235
	147	0,518	137	0,290	100	0,093	176	0,488	91	0,105	103	0,387	95	0,251
	85	0,395	214	0,227	228	0,095	47	0,388	94	0,098	124	0,328	155	0,134
	111	0,357	83	0,174	182	0,084	206	0,405	253	0,091	119	0,331	199	0,110
	102	0,389	176	0,126	144	0,089	343	0,388	94	0,086	125	0,209	162	0,068
	124	0,328	114	0,088	187	0,062	134	0,299	67	0,083	233	0,112	222	0,066
	81	0,238	178	0,075	160	0,055	178	0,188	88	0,081	132	0,088	170	0,063
	184	0,139	465	0,043	244	0,055	457	0,045	121	0,068	412	0,044	241	0,055
	136	0,083	314	0,043	176	0,054	605	0,043	154	0,063	376	0,040	211	0,044
	1369	0,024	315	0,030	159	0,026	537	0,028	155	0,059	492	0,022	334	0,044
	0,306	0,59	**0,295**	0,459	**0,249**	0,258	**0,266**	0,617	**0,213**	0,194	**0,303**	0,502	**0,278**	0,367
	Size	**ISim**	**Size**	**ISim**	**Size**	**ISim**	**Size**	**ISim**	**Size**	**ISim**	**Size**	**ISim**	**Size**	**ISim**
rbr	30	1,000	30	1,000	205	0,658	1	1,000	241	0,526	30	1,000	154	0,717
	55	1,000	46	0,990	143	0,628	1	1,000	182	0,463	129	1,000	194	0,703
	46	0,990	57	0,956	176	0,488	30	1,000	188	0,461	56	0,981	138	0,651
	131	0,970	133	0,955	218	0,394	29	1,000	131	0,307	46	0,990	176	0,488
	58	0,919	58	0,919	106	0,375	46	0,990	300	0,220	58	0,919	111	0,353
	38	0,855	39	0,817	77	0,250	57	0,956	195	0,158	38	0,855	104	0,246
	27	0,836	195	0,690	182	0,184	66	0,975	201	0,156	179	0,767	154	0,230
	174	0,786	138	0,651	177	0,184	67	0,937	208	0,142	136	0,661	167	0,215
	22	0,714	176	0,488	195	0,150	29	0,917	61	0,132	176	0,487	106	0,234
	133	0,674	92	0,359	185	0,120	38	0,855	190	0,121	91	0,365	132	0,193

	i1		i2		e1		g1		g1p		h1		h2	
	30	0,625	140	0,282	240	0,111	194	0,703	183	0,114	108	0,374	173	0,160
	145	0,525	137	0,273	174	0,088	40	0,466	212	0,108	132	0,290	151	0,141
	85	0,395	216	0,228	143	0,091	179	0,476	105	0,105	156	0,242	176	0,126
	111	0,357	98	0,142	123	0,088	50	0,378	100	0,103	121	0,216	109	0,101
	105	0,387	156	0,144	97	0,092	131	0,306	116	0,102	312	0,082	120	0,086
	121	0,337	118	0,084	120	0,082	354	0,284	95	0,099	163	0,056	144	0,077
	100	0,213	212	0,064	154	0,078	146	0,247	86	0,097	188	0,054	197	0,077
	166	0,151	438	0,047	119	0,072	300	0,084	89	0,096	416	0,048	179	0,070
	129	0,091	320	0,043	166	0,069	549	0,042	74	0,087	326	0,044	184	0,060
	1383	0,024	290	0,032	89	0,049	782	0,024	132	0,090	228	0,038	220	0,060
	0,307	0,592	**0,296**	0,458	**0,225**	0,213	**0,275**	0,632	**0,21**	0,184	**0,298**	0,473	**0,247**	0,249
direct	**Size**	**ISim**	**Size**	**ISim**	**Size**	**ISim**	**Size**	**ISim**	**Size**	**ISim**	**Size**	**ISim**	**Size**	**ISim**
	30	1,000	30	1,000	193	0,708	30	1,000	251	0,503	30	1,000	155	0,708
	129	1,000	133	0,955	163	0,544	133	0,955	202	0,419	129	1,000	194	0,703
	58	0,919	58	0,919	175	0,492	58	0,919	267	0,369	46	0,990	176	0,488
	20	0,788	39	0,817	197	0,446	38	0,855	263	0,261	181	0,761	184	0,470
	177	0,776	194	0,703	136	0,319	22	0,686	181	0,178	137	0,658	136	0,319
	136	0,661	103	0,537	64	0,214	194	0,703	190	0,174	96	0,493	213	0,226
	103	0,529	177	0,485	171	0,195	103	0,537	198	0,156	176	0,487	157	0,223
	174	0,494	92	0,359	244	0,174	183	0,465	164	0,128	91	0,365	105	0,223
	38	0,406	134	0,279	119	0,181	90	0,367	162	0,125	129	0,298	173	0,159
	85	0,395	348	0,293	203	0,123	121	0,337	178	0,131	139	0,316	150	0,145
	48	0,386	154	0,247	177	0,129	118	0,325	118	0,117	189	0,269	174	0,130
	121	0,337	143	0,158	184	0,099	352	0,283	95	0,116	86	0,233	93	0,136
	148	0,264	86	0,129	114	0,095	64	0,223	111	0,111	169	0,135	109	0,104
	161	0,287	114	0,108	115	0,089	121	0,188	109	0,113	181	0,072	118	0,089
	38	0,216	117	0,079	144	0,089	99	0,124	91	0,104	153	0,062	119	0,087
	108	0,214	178	0,074	105	0,089	96	0,097	105	0,096	157	0,060	102	0,083
	87	0,144	157	0,061	116	0,079	228	0,063	111	0,090	180	0,052	155	0,084
	106	0,098	315	0,054	141	0,073	298	0,041	91	0,097	372	0,051	120	0,082
	451	0,048	257	0,045	209	0,073	327	0,039	69	0,095	248	0,047	277	0,061
	871	0,018	560	0,032	119	0,065	414	0,030	133	0,097	200	0,043	179	0,059
	0,285	0,449	**0,252**	0,367	**0,228**	0,214	**0,277**	0,412	**0,209**	0,174	**0,277**	0,37	**0,241**	0,229

Tabla 12. Resultados de los métodos acumulativos, obteniendo 20 *clusters* (caso A).

	i1		i2		e1		g1		g1p		h1		h2		slink		clink		upgma	
	Size	**ISim**	**Size**	**ISim**	**Size**	**ISim**	**Size**	**ISim**	**Size**	**ISim**	**Size**	**ISim**	**Size**	**ISim**	**Size**	**ISim**	**Size**	**ISim**	**Size**	**ISim**
agglo	2	1,000	2	1,000	2	1,000	271	0,348	2	1,000	226	0,044	134	0,085	2	1,000	188	0,200	2	1,000
	100	0,368	222	0,234	162	0,077	1	1,000	132	0,075	129	1,000	100	0,117	30	1,000	58	0,919	96	0,493
	129	1,000	80	0,392	222	0,361	67	0,962	131	0,062	119	0,338	237	0,069	1	1,000	112	0,353	728	0,309
	153	0,850	129	1,000	107	0,101	32	0,909	84	0,093	153	0,850	132	0,068	1	1,000	24	0,160	9	0,294
	55	1,000	38	0,855	102	0,079	57	0,956	164	0,140	169	0,223	129	1,000	1	1,000	8	0,386	267	0,191
	58	0,919	197	0,051	148	0,307	38	0,855	337	0,401	58	0,919	218	0,412	2	0,986	8	0,348	41	0,067
	46	0,990	137	0,278	231	0,329	31	0,943	113	0,093	55	1,000	168	0,755	1	1,000	5	0,292	5	0,357
	118	0,341	183	0,680	236	0,069	86	0,370	310	0,190	119	0,299	126	0,066	1	1,000	5	0,414	7	0,201
	1595	0,023	58	0,919	160	0,067	156	0,242	179	0,116	139	0,632	139	0,260	1	1,000	28	0,368	86	0,109
	99	0,756	56	0,981	209	0,127	46	0,99	196	0,158	46	0,990	144	0,220	1	1,000	2	0,581	5	0,243
	38	0,855	125	0,285	184	0,143	138	0,255	166	0,068	190	0,415	161	0,272	1	1,000	699	0,320	87	0,101
	30	1,000	140	0,628	201	0,090	340	0,266	261	0,379	267	0,034	254	0,048	1	1,000	2	1,000	5	0,294
	80	0,392	46	0,990	227	0,053	940	0,029	127	0,159	298	0,066	121	0,223	1	1,000	65	0,780	5	0,442
	92	0,718	586	0,026	160	0,811	1	1,000	149	0,137	86	0,366	157	0,496	1	1,000	1514	0,026	3	0,419
	38	0,774	173	0,459	158	0,125	26	0,963	167	0,170	573	0,038	82	0,468	1	1,000	98	0,172	396	0,187
	90	0,382	30	1,000	140	0,629	64	1,000	126	0,138	38	0,855	90	0,171	1	1,000	6	0,350	1	1,000
	105	0,186	60	0,229	96	0,270	132	0,885	159	0,279	30	1,000	156	0,103	1	1,000	6	0,328	3	0,641
	15	1,000	536	0,038	112	0,261	24	0,594	94	0,111	67	0,452	58	0,684	1	1,000	162	0,700	9	0,190
	26	0,688	97	0,188	96	0,109	63	0,200	100	0,146	127	0,073	242	0,048	1	1,000	4	0,648	101	0,104
	220	0,201	194	0,062	136	0,080	576	0,036	92	0,080	200	0,044	241	0,046	3039	0,033	95	0,069	1233	0,032
	0,299	0,672	**0,287**	0,515	**0,213**	0,254	**0,268**	0,64	**0,188**	0,2	**0,29**	0,482	**0,247**	0,281	**0,049**	0,951	**0,199**	0,421	**0,158**	0,334
bagglo	**Size**	**ISim**	**Size**	**ISim**	**Size**	**ISim**	**Size**	**ISim**	**Size**	**ISim**	**Size**	**ISim**	**Size**	**ISim**	**Size**	**ISim**	**Size**	**ISim**	**Size**	**ISim**
	137	0,658	112	0,215	137	0,658	96	0,493	234	0,321	103	0,537	211	0,157	2	1,000	616	0,356	97	0,484
	129	1,000	197	0,687	250	0,095	103	0,102	170	0,120	131	0,980	179	0,142	103	0,537	97	0,484	724	0,312
	106	0,234	327	0,048	135	0,081	103	0,537	155	0,230	182	0,115	176	0,034	1	1,000	6	0,246	75	0,091
	148	0,515	132	0,967	152	0,049	136	0,914	152	0,086	246	0,051	152	0,055	1	1,000	4	0,365	240	0,226
	103	0,537	103	0,537	149	0,237	92	0,357	213	0,228	109	0,233	137	0,658	1	1,000	141	0,264	102	0,103
	135	0,296	224	0,056	159	0,081	136	0,293	133	0,082	276	0,042	172	0,124	1	1,000	90	0,119	132	0,084
	138	0,269	135	0,296	152	0,059	157	0,228	130	0,089	137	0,658	129	0,086	1	1,000	103	0,537	56	0,254
	96	0,493	175	0,492	110	0,231	173	0,127	147	0,517	261	0,045	110	0,231	1	1,000	2	0,624	165	0,036
	128	0,675	161	0,218	245	0,049	132	0,084	134	0,323	148	0,515	132	0,967	1	1,000	135	0,296	161	0,079
	88	0,377	135	0,081	196	0,119	214	0,227	200	0,146	146	0,252	197	0,687	1	1,000	128	0,087	117	0,068
	157	0,139	162	0,134	193	0,708	160	0,080	224	0,593	132	0,646	248	0,050	1	1,000	108	0,218	138	0,889
	103	0,387	137	0,658	147	0,517	197	0,687	156	0,037	134	0,299	147	0,517	1	1,000	8	0,376	146	0,049
	65	0,942	96	0,493	166	0,036	180	0,034	147	0,057	194	0,110	154	0,229	1	1,000	90	0,361	132	0,056

	i1		i2		e1		g1		g1p		h1		h2		slink		clink		upgma	
	146	0,088	150	0,086	154	0,142	56	0,254	128	0,054	96	0,493	163	0,078	1	1,000	5	0,439	70	0,075
	121	0,094	88	0,377	133	0,955	157	0,055	156	0,139	148	0,087	133	0,301	1	1,000	178	0,034	83	0,085
	156	0,055	191	0,032	134	0,062	154	0,047	121	0,061	100	0,393	147	0,058	1	1,000	113	0,064	156	0,055
	92	0,118	160	0,054	137	0,290	175	0,492	115	0,070	159	0,055	153	0,048	1	1,000	4	0,648	69	0,096
	165	0,036	103	0,387	100	0,393	137	0,658	137	0,658	65	0,942	100	0,393	1	1,000	136	0,914	180	0,118
	56	0,254	150	0,048	120	0,061	345	0,040	108	0,096	153	0,048	123	0,06	1	1,000	367	0,051	95	0,338
	820	0,028	151	0,050	120	0,400	186	0,054	129	0,278	169	0,036	126	0,290	2967	0,034	758	0,034	151	0,239
	0,268	0,36	**0,265**	0,296	**0,246**	0,261	**0,263**	0,288	**0,224**	0,209	**0,259**	0,327	**0,246**	0,258	**0,057**	0,929	**0,217**	0,326	**0,211**	0,187

Siguen estando los métodos particionales por encima de los acumulativos, continúa estando el método acumulativo *agglo* por encima del método particional *direct*, pero se produce un cambio dentro de los dos particionales mejor clasificados, poniéndose por delante el método *rbr* sobre el método *rb*, cuando se utiliza con las funciones i1, i2 y g1. Véase figura 37.

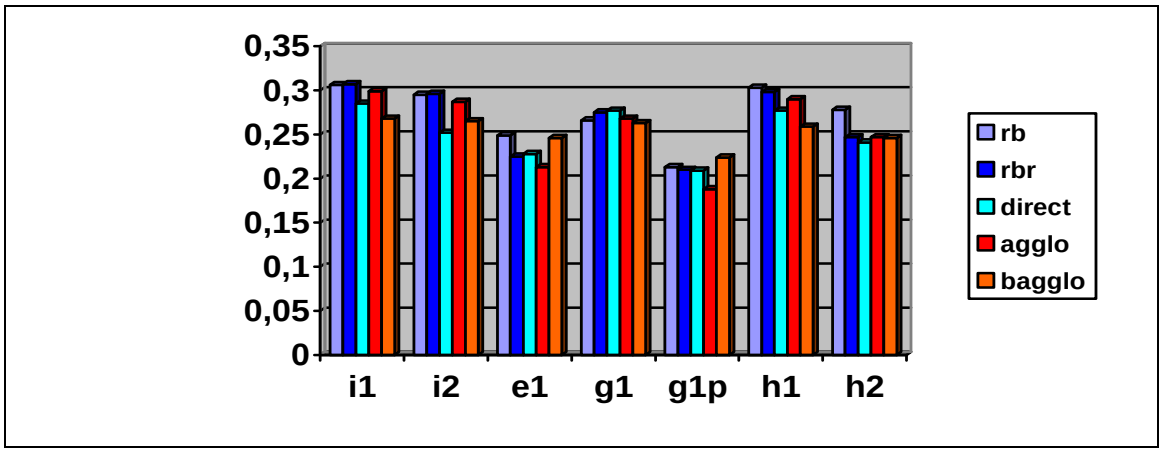

Figura 37. Gráfico de incidencia de los métodos de *clustering* (20 *clusters*, caso A).

En cuanto a las funciones criterios comunes para los dos métodos el comportamiento no cambia al aumentar el número de *clusters*. Sigue presentando el mejor comportamiento i1, seguida de h1.Véase figura 38, en las que ya sólo aparecen las funciones de mejores resultados.

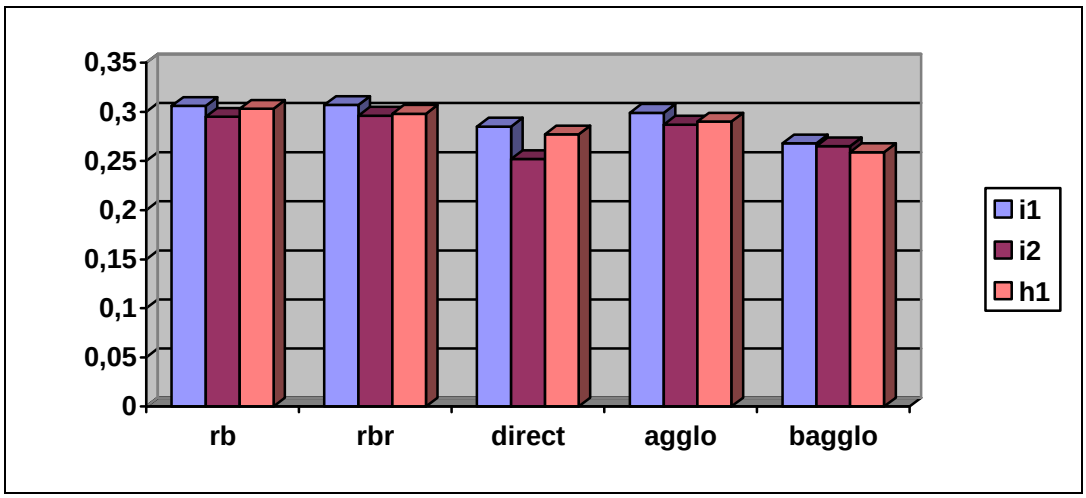

Figura 38. Gráfico de incidencia de las funciones criterio comunes para todos los métodos (20 *clusters*, caso A).

Por último en este apartado, dentro de los métodos acumulativos, hay que comparar el comportamiento de las funciones criterio válidas para todos los métodos, con las funciones criterio específicas de los métodos acumulativos. Siguen proporcionando mejores valores las funciones no específicas. Dentro de las específicas se produce un cambio, pasando a ser la función destacada *clink*, en detrimento de la función *upgma*. Véase figura 39.

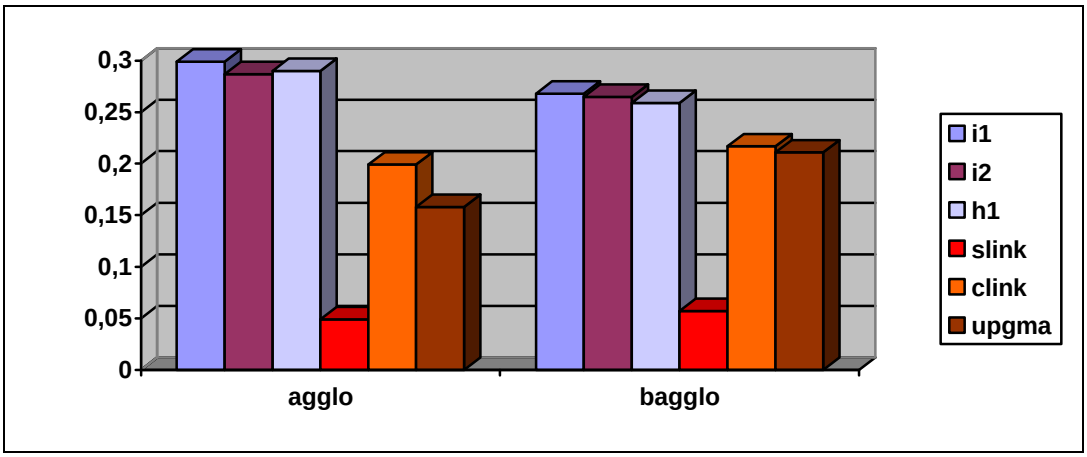

Figura 39. Gráfico de incidencia de las funciones criterio para métodos acumulativos (20 *clusters*, caso A)

Si comparamos los resultados, fijando un método y una función criterio y observamos el comportamiento de los *clusters* al aumentar el número de *clusters* especificados, vemos que a medida que aumentamos el número de *clusters*, la medida de calidad utilizada aumenta. Esto es debido a que los *clusters* que presentan gran similitud entre sus elementos persisten, es decir, se mantienen sin sufrir modificaciones aunque queramos obtener nuevos *clusters*. Cuanto mayor es la similitud de un *clusters* mayor es la posibilidad de encontrarlo tal cual al hacer nuevas divisiones. Si nos fijamos en los 10 clusters obtenidos con el método rb y la función i1, vemos que los 5 primeros que tienen similitud superior a 0,8 aparecen sin modificaciones al obtener 20 clusters y también al obtener 30 clusters. Los clusters 6 y 7 que tienen similitudes de 0,784 y 0,658 permanecen cuando pasamos a obtener 20 clusters pero ya se dividen al obtener 30 clusters.

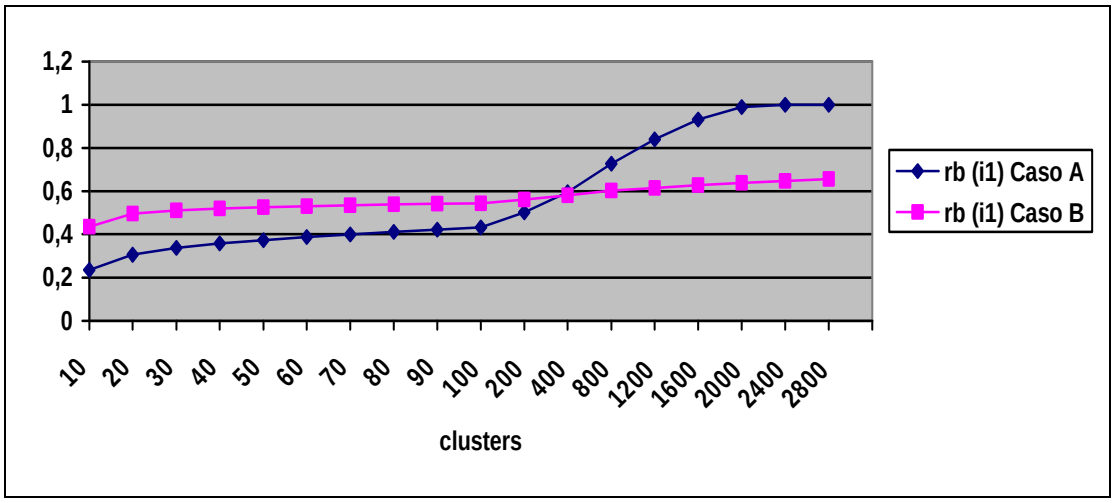

Figura 40. Variación de la calidad de los *clusters* con respecto al número de *clusters* (Caso A).

Lo habitual es que al dividirse un *cluster* se obtengan otros *clusters* con mayor similitud que la del *cluster* del que partían, por este motivo aumentan la calidad de la solución global, hasta que lleguemos a un punto en que todos los *clusters* tengan similitud 1, que no tienen porqué ser *cluster* unitarios con un único documento, como se puede apreciar en el gráfico de la figura 40. Se observa también que la recuperación de las curvas se produce mucho más tarde cuanto mayor es el número de elementos del caso de estudio.

9.4 Análisis de las características descriptivas (*features*)

Las características descriptivas de un *cluster* son los términos más representativos de ese *cluster*, las palabras que más contribuyen a la similitud media entre los objetos de cada *cluster*. Dentro de la matriz de pesos, las filas corresponderían a los documentos, las columnas a las dimensiones o características de los documentos. Igual que podemos elegir las características que mejor contribuyen a la similitud interna también podemos seleccionar las características que mejor discriminan ese *cluster* con respecto al resto de *cluster*. Suele ser habitual que exista solapamiento entre estos conjuntos de términos, por lo que nos vamos a centrar en las descriptivas.

A continuación se presenta la tabla 13, que recoge las 5 características más descriptivas de cada *cluster*, para el caso de 10 *clusters*, para los métodos que mejor comportamiento han presentado –rb, rbr y agglo– y cada uno de estos métodos evaluados para las tres funciones criterio destacadas: i1, i2 y h1. Se ha establecido un código de colores, de forma que se puede identificar un mismo *cluster* en distintos métodos y con distintas funciones criterio.

Tabla 13. Estudio de características descriptivas para 10 *clusters*, caso A.

	i1	i2	h1
rb	Cluster 0, Size: 56, ISim: 0.981, ESim: 0.005 60.71% venus transito fotos nubes descargar Cluster 1, Size: 46, ISim: 0.990, ESim: 0.015 97.83% genericas mostrando fotografico archivo anterior Cluster 2, Size: 131, ISim: 0.970, ESim: 0.007 98.47% inexistente servidor encontrado consulte administrador Cluster 3, Size: 58, ISim: 0.919, ESim: 0.001 77.59% moved page click Cluster 4, Size: 38, ISim: 0.855, ESim: 0.004 100.00% ctrl mover arrastre acercar haga Cluster 5, Size: 175, ISim: 0.784, ESim: 0.056 97.71% gaceta local listado seccion actualidad Cluster 6, Size: 137, ISim: 0.658, ESim: 0.060 100.00% tribuna listado seccion actualidad mes Cluster 7, Size: 154, ISim: 0.251, ESim: 0.007 75.71% antartida expedicion foto titulo fotos Cluster 8, Size: 387, ISim: 0.272, ESim: 0.036 89.04% adelanto listado seccion actualidad mes Cluster 9, Size: 1907, ISim: 0.024, ESim: 0.005 54.68% horas facultad protocolo escuelas lugar	Cluster 0, Size: 166, ISim: 0.670, ESim: 0.005 82.65% error consulte servidor inexistente administrador Cluster 1, Size: 197, ISim: 0.683, ESim: 0.052 94.42% gaceta local listado seccion actualidad Cluster 2, Size: 103, ISim: 0.537, ESim: 0.008 62.78% fotos genericas mostrando 69.90% venus transito fotos Cluster 3, Size: 97, ISim: 0.484, ESim: 0.001 46.05% click mover arrastre 59.79% moved page Cluster 4, Size: 91, ISim: 0.364, ESim: 0.005 85.49% admission exams entrance registration courses Cluster 5, Size: 123, ISim: 0.309, ESim: 0.014 93.66% acceso pruebas investigacion mapa agenda Cluster 6, Size: 137, ISim: 0.290, ESim: 0.007 89.49% antartida expedicion foto fotos titulo Cluster 7, Size: 530, ISim: 0.287, ESim: 0.025 98.72% listado seccion actualidad mes dia Cluster 8, Size: 437, ISim: 0.061, ESim: 0.011 41.51% sala prensa rueda rectorado retratos Cluster 9, Size: 1208, ISim: 0.025, ESim: 0.009 42.57% facultad horas curso protocolo salon	Cluster 0, Size: 131, ISim: 0.980, ESim: 0.007 99.08% consulte inexistente servidor encontrado administrador Cluster 1, Size: 178, ISim: 0.774, ESim: 0.056 97.42% gaceta local listado seccion actualidad Cluster 2, Size: 137, ISim: 0.658, ESim: 0.060 100.00% tribuna listado seccion actualidad mes Cluster 3, Size: 102, ISim: 0.544, ESim: 0.008 63.40% fotos genericas mostrando 69.93% venus transito fotos Cluster 4, Size: 96, ISim: 0.493, ESim: 0.001 46.18% click mover arrastre 60.42% moved page Cluster 5, Size: 154, ISim: 0.251, ESim: 0.007 75.71% antartida expedicion foto titulo fotos Cluster 6, Size: 404, ISim: 0.259, ESim: 0.035 88.02% adelanto listado seccion actualidad mes Cluster 7, Size: 209, ISim: 0.200, ESim: 0.009 43.06% admission exams 56.62% acceso pruebas investigacion Cluster 8, Size: 266, ISim: 0.100, ESim: 0.013 53.76% sala rectorado retratos 66.04% rueda sala prensa Cluster 9, Size: 1412, ISim: 0.023, ESim: 0.008 38.24% facultad actividades protocolo curso 46.69% facultad horas protocolo curso
rbr	Cluster 0, Size: 55, ISim: 1.000, ESim: 0.005 100.00% venus transito fotos Cluster 1, Size: 46, ISim: 0.990, ESim: 0.015	Cluster 0, Size: 165, ISim: 0.677, ESim: 0.005 83.15% error consulte servidor inexistente administrador	Cluster 0, Size: 129, ISim: 1.000, ESim: 0.007 100.00% inexistente servidor encontrado consulte administrador

	i1	i2	h1
	97.83% genericas mostrando fotografico archivo anterior Cluster 2, Size: 131, ISim: 0.970, ESim: 0.007 98.47% inexistente servidor encontrado consulte administrador Cluster 3, Size: 58, ISim: 0.919, ESim: 0.001 77.59% moved page click Cluster 4, Size: 38, ISim: 0.855, ESim: 0.004 100.00% ctrl mover arrastre acercar haga Cluster 5, Size: 175, ISim: 0.784, ESim: 0.056 97.71% gaceta local listado seccion actualidad Cluster 6, Size: 137, ISim: 0.658, ESim: 0.060 100.00% tribuna listado seccion actualidad mes Cluster 7, Size: 154, ISim: 0.251, ESim: 0.007 75.71% antartida expedicion foto titulo fotos Cluster 8, Size: 392, ISim: 0.269, ESim: 0.036 88.93% adelanto listado seccion actualidad mes Cluster 9, Size: 1903, ISim: 0.024, ESim: 0.005 54.82% horas facultad protocolo escuelas lugar	Cluster 1, Size: 197, ISim: 0.683, ESim: 0.052 94.42% gaceta local listado seccion actualidad Cluster 2, Size: 103, ISim: 0.537, ESim: 0.008 62.78% fotos genericas mostrando 69.90% venus transito fotos Cluster 3, Size: 97, ISim: 0.484, ESim: 0.001 46.05% click mover arrastre 59.79% moved page Cluster 4, Size: 92, ISim: 0.359, ESim: 0.005 85.43% admission exams entrance registration courses Cluster 5, Size: 530, ISim: 0.287, ESim: 0.025 98.72% listado seccion actualidad mes dia Cluster 6, Size: 157, ISim: 0.240, ESim: 0.007 85.10% antartida expedicion foto fotos titulo Cluster 7, Size: 148, ISim: 0.244, ESim: 0.013 88.11% acceso pruebas investigacion mapa directorio Cluster 8, Size: 429, ISim: 0.061, ESim: 0.011 41.54% sala prensa rueda rectorado retratos Cluster 9, Size: 1171, ISim: 0.026, ESim: 0.009 29.19% facultad horas curso salon acto	Cluster 1, Size: 178, ISim: 0.774, ESim: 0.056 97.42% gaceta local listado seccion actualidad Cluster 2, Size: 137, ISim: 0.658, ESim: 0.060 100.00% tribuna listado seccion actualidad mes Cluster 3, Size: 102, ISim: 0.544, ESim: 0.008 63.40% fotos genericas mostrando 69.93% venus transito fotos Cluster 4, Size: 96, ISim: 0.493, ESim: 0.001 46.18% click mover arrastre 60.42% moved page Cluster 5, Size: 404, ISim: 0.259, ESim: 0.035 88.02% adelanto listado seccion actualidad mes Cluster 6, Size: 189, ISim: 0.192, ESim: 0.007 70.90% antartida expedicion foto fotos titulo Cluster 7, Size: 258, ISim: 0.148, ESim: 0.009 35.66% admission exams 52.07% acceso pruebas investigacion Cluster 8, Size: 554, ISim: 0.044, ESim: 0.010 33.72% prensa sala rueda rectorado retratos Cluster 9, Size: 1042, ISim: 0.028, ESim: 0.009 42.03% facultad actividades curso protocolo 51.30% facultad horas curso protocolo
agglo	Cluster 0, Size: 2, ISim: 1.000, ESim: 0.000 100.00% index Cluster 1, Size: 565, ISim: 0.282, ESim: 0.023 99.36% listado seccion actualidad mes dia Cluster 2, Size: 129, ISim: 1.000, ESim: 0.007 100.00% inexistente servidor encontrado consulte administrador Cluster 3, Size: 153, ISim: 0.850, ESim: 0.061 98.69% gaceta local listado seccion actualidad Cluster 4, Size: 55, ISim: 1.000, ESim: 0.005	Cluster 0, Size: 2, ISim: 1.000, ESim: 0.000 100.00% index Cluster 1, Size: 140, ISim: 0.352, ESim: 0.007 46.19% fotos genericas mostrando 50.95% venus transito fotos Cluster 2, Size: 535, ISim: 0.294, ESim: 0.024 100.00% listado seccion actualidad mes dia Cluster 3, Size: 129, ISim: 1.000, ESim: 0.007 100.00% inexistente servidor encontrado consulte administrador	Cluster 0, Size: 561, ISim: 0.027, ESim: 0.011 5.39% sans apache tomcat font 8.73% aprobacion Cluster 1, Size: 129, ISim: 1.000, ESim: 0.007 100.00% inexistente servidor encontrado consulte administrador Cluster 2, Size: 165, ISim: 0.275, ESim: 0.009 75.52% fotos antartida expedicion foto titulo Cluster 3, Size: 153, ISim: 0.850, ESim: 0.061 98.69% gaceta local listado seccion actualidad

	i1	i2	h1
	100.00% venus transito fotos Cluster 5, Size: 58, ISim: 0.919, ESim: 0.001 77.59% moved page click Cluster 6, Size: 46, ISim: 0.990, ESim: 0.015 97.83% genericas mostrando fotografico archivo anterior Cluster 7, Size: 118, ISim: 0.341, ESim: 0.008 97.29% antartida expedicion foto fotos titulo Cluster 8, Size: 180, ISim: 0.219, ESim: 0.010 46.67% admission exams 53.70% acceso pruebas investigacion Cluster 9, Size: 1783, ISim: 0.023, ESim: 0.005 40.61% facultad protocolo actividades 47.18% horas facultad protocolo lugar	Cluster 4, Size: 1413, ISim: 0.028, ESim: 0.006 32.73% horas sala prensa lugar 39.56% horas facultad lugar Cluster 5, Size: 382, ISim: 0.062, ESim: 0.012 11.26% actividades 36.65% acceso pruebas investigacion tlf Cluster 6, Size: 137, ISim: 0.278, ESim: 0.008 92.41% antartida expedicion foto fotos titulo Cluster 7, Size: 183, ISim: 0.680, ESim: 0.056 93.55% gaceta local listado seccion actualidad Cluster 8, Size: 58, ISim: 0.919, ESim: 0.001 77.59% moved page click Cluster 9, Size: 110, ISim: 0.282, ESim: 0.005 27.73% family tomcat 72.73% admission exams entrance	Cluster 4, Size: 1198, ISim: 0.031, ESim: 0.008 20.33% facultad actividades 39.87% horas sala lugar 42.04% horas facultad lugar Cluster 5, Size: 58, ISim: 0.919, ESim: 0.001 77.59% moved page click Cluster 6, Size: 55, ISim: 1.000, ESim: 0.005 100.00% venus transito fotos Cluster 7, Size: 205, ISim: 0.197, ESim: 0.010 44.39% admission exams 56.42% acceso pruebas investigacion Cluster 8, Size: 139, ISim: 0.632, ESim: 0.062 97.12% tribuna listado seccion actualidad mes Cluster 9, Size: 426, ISim: 0.250, ESim: 0.035 87.32% listado seccion adelanto actualidad mes

En unas ocasiones la coincidencia de los *clusters* es total, coincide exactamente el número de documentos de ambos *clusters*. Sería el caso del método rb en el que el *cluster* 7 con i1 es idéntico al 5 con h1. En otras ocasiones las características descriptivas coinciden, pero el número de documentos varía, por ejemplo el *cluster* 7 con i1 y el 6 con i2, uno tiene 154 documentos y el otro 137. Esto es debido a que todas las características del *cluster* contribuyen a la similitud interna, el seleccionar 5 no quiere decir que sean todas, aunque podrían serlo si los documentos no tuvieran más que esas, sino que se muestran las que en mayor porcentaje contribuyen.

Siempre queda como último *cluster* –podrían ser varios- un cajón de sastre a donde van a parar los documentos que más dificultades presentan para clasificar por presentar un comportamiento disperso con respecto a los ya definidos. Nos referiremos a sus elementos como, *sin clasificar*.

El método rb obtiene 5 *clusters* con correspondencia en las tres funciones criterio, es decir, estos 5 *clusters* se obtienen independientemente de la función criterio. Se ampliarían a 7 los *clusters* que coinciden si comparamos los datos de la función criterio i1 con h1. Con i1, quedarían 1783 documentos sin clasificar, con i2, 2175 (530+437+1208) y con h1 1678 (266+1412).

El método rbr obtiene 5 *clusters* con correspondencia en las tres funciones criterio. Se ampliarían a 7 los *clusters* que coinciden si comparamos los datos de la función criterio i1 con h1. Con i1 quedarían 1903 documentos sin clasificar, con i2 1171 y con h1 1042.

Por último el método agglo obtiene 5 *clusters* comunes a las tres funciones, que pasa a 6 si se comparan i1 con h1. Con i1 quedarían 1783 documentos sin clasificar, con i2 serían 1905 (1413+382+110) y con h1 serían 1759 (1198+561).

Viendo en conjunto los resultados, la combinación que más documentos clasifica de forma equilibrada sería el método rbr con la función criterio h2.

De igual forma se ha seguido el mismo procedimiento para las pruebas realizadas con 20 *clusters*, los resultados figuran en la tabla 14.

Tabla 14. Estudio de características descriptivas para 20 *clusters*, caso A.

	i1	i2	h1
rb	Cluster 0, Size: 30, ISim: 1.000, ESim: 0.004 100.00% tomcat apache sans family report	Cluster 0, Size: 30, ISim: 1.000, ESim: 0.004 100.00% tomcat apache sans family report	Cluster 0, Size: 30, ISim: 1.000, ESim: 0.004 100.00% tomcat apache sans family report
	Cluster 1, Size: 56, ISim: 0.981, ESim: 0.005 60.71% venus transito fotos nubes descargar	Cluster 1, Size: 46, ISim: 0.990, ESim: 0.015 97.83% genericas mostrando fotografico archivo anterior	Cluster 1, Size: 56, ISim: 0.981, ESim: 0.005 60.71% venus transito fotos nubes descargar
	Cluster 2, Size: 46, ISim: 0.990, ESim: 0.015 97.83% genericas mostrando fotografico archivo anterior	Cluster 2, Size: 57, ISim: 0.956, ESim: 0.005 67.25% venus transito telescopio 75.00% venus transito fotos nubes	Cluster 2, Size: 46, ISim: 0.990, ESim: 0.015 97.83% genericas mostrando fotografico archivo anterior
	Cluster 3, Size: 131, ISim: 0.970, ESim: 0.007 98.47% inexistente servidor encontrado consulte administrador	Cluster 3, Size: 58, ISim: 0.919, ESim: 0.001 77.59% moved page click	Cluster 3, Size: 131, ISim: 0.980, ESim: 0.007 99.08% consulte inexistente servidor encontrado administrador
	Cluster 4, Size: 58, ISim: 0.919, ESim: 0.001 77.59% moved page click	Cluster 4, Size: 136, ISim: 0.916, ESim: 0.006 95.88% consulte servidor inexistente administrador encontrado	Cluster 4, Size: 58, ISim: 0.919, ESim: 0.001 77.59% moved page click
	Cluster 5, Size: 38, ISim: 0.855, ESim: 0.004 100.00% ctrl mover arrastre acercar haga	Cluster 5, Size: 39, ISim: 0.817, ESim: 0.004 97.95% ctrl mover arrastre haga acercar	Cluster 5, Size: 38, ISim: 0.855, ESim: 0.004 100.00% ctrl mover arrastre acercar haga
	Cluster 6, Size: 27, ISim: 0.836, ESim: 0.058 94.07% castro omar adelanto listado seccion	Cluster 6, Size: 197, ISim: 0.683, ESim: 0.052 94.42% gaceta local listado seccion actualidad	Cluster 6, Size: 178, ISim: 0.774, ESim: 0.056 97.42% gaceta local listado seccion actualidad
	Cluster 7, Size: 175, ISim: 0.784, ESim: 0.056 97.71% gaceta local listado seccion actualidad	Cluster 7, Size: 137, ISim: 0.658, ESim: 0.060 100.00% tribuna listado seccion actualidad mes	Cluster 7, Size: 137, ISim: 0.658, ESim: 0.060 100.00% tribuna listado seccion actualidad mes
	Cluster 8, Size: 22, ISim: 0.714, ESim: 0.012 83.64% actividades protocolo null patio escuelas	Cluster 8, Size: 179, ISim: 0.474, ESim: 0.051 98.32% adelanto listado seccion actualidad mes	Cluster 8, Size: 33, ISim: 0.455, ESim: 0.012 89.09% aprobacion procede informe consejo gobierno
	Cluster 9, Size: 137, ISim: 0.658, ESim: 0.060 100.00% tribuna listado seccion actualidad mes	Cluster 9, Size: 91, ISim: 0.364, ESim: 0.005 85.49% admission exams entrance registration courses	Cluster 9, Size: 176, ISim: 0.488, ESim: 0.051 99.55% adelanto listado seccion actualidad mes
	Cluster 10, Size: 30, ISim: 0.625, ESim: 0.048 98.00% zamora opinion listado seccion actualidad	Cluster 10, Size: 123, ISim: 0.309, ESim: 0.014 93.66% acceso pruebas investigacion mapa agenda	Cluster 10, Size: 90, ISim: 0.371, ESim: 0.005 85.33% admission exams entrance registration courses
	Cluster 11, Size: 147, ISim: 0.518, ESim: 0.057 100.00% adelanto listado seccion actualidad mes	Cluster 11, Size: 137, ISim: 0.290, ESim: 0.007 89.49% antartida expedicion foto fotos titulo	Cluster 11, Size: 103, ISim: 0.387, ESim: 0.047 97.28% castilla leon listado seccion actualidad

	i1	i2	h1
	Cluster 12, Size: 85, ISim: 0.395, ESim: 0.005 85.65% admission exams entrance courses registration Cluster 13, Size: 111, ISim: 0.357, ESim: 0.015 99.82% acceso pruebas investigacion agenda mecenazgo Cluster 14, Size: 102, ISim: 0.389, ESim: 0.048 97.25% castilla leon listado seccion actualidad Cluster 15, Size: 124, ISim: 0.328, ESim: 0.007 94.03% antartida expedicion foto titulo fotos Cluster 16, Size: 81, ISim: 0.238, ESim: 0.050 100.00% listado seccion actualidad mes dia Cluster 17, Size: 184, ISim: 0.139, ESim: 0.014 74.28% sala rueda prensa 67.93% retratos sala rectorado Cluster 18, Size: 136, ISim: 0.083, ESim: 0.011 27.94% fac mail seleccione 31.62% fac ext mail 27.70% medios fac ext Cluster 19, Size: 1369, ISim: 0.024, ESim: 0.009 42.70% horas facultad curso protocolo salon	Cluster 12, Size: 214, ISim: 0.227, ESim: 0.042 88.50% castilla listado seccion actualidad mes Cluster 13, Size: 83, ISim: 0.174, ESim: 0.011 46.69% aprobacion procede informe consejo 44.58% actividades Cluster 14, Size: 176, ISim: 0.126, ESim: 0.014 68.41% rectorado retratos sala rueda prensa Cluster 15, Size: 114, ISim: 0.088, ESim: 0.010 27.63% fac ext 24.56% fac seleccione 42.98% medios somos Cluster 16, Size: 178, ISim: 0.075, ESim: 0.013 57.64% fonseca prensa rueda hospederia horas Cluster 17, Size: 465, ISim: 0.043, ESim: 0.012 46.02% facultad horas salon curso actos Cluster 18, Size: 314, ISim: 0.043, ESim: 0.013 37.05% premio espa jose 34.55% acto paraninfo 38.00% acto premio jose Cluster 19, Size: 315, ISim: 0.030, ESim: 0.011 36.51% investigacion 21.27% alumnos estudiantes universitario selectividad	Cluster 12, Size: 124, ISim: 0.328, ESim: 0.007 94.03% antartida expedicion foto titulo fotos Cluster 13, Size: 119, ISim: 0.331, ESim: 0.015 98.15% acceso pruebas investigacion agenda mecenazgo Cluster 14, Size: 125, ISim: 0.209, ESim: 0.045 97.60% listado seccion actualidad mes dia Cluster 15, Size: 233, ISim: 0.112, ESim: 0.013 57.51% sala rectorado retratos 73.61% rueda prensa Cluster 16, Size: 132, ISim: 0.088, ESim: 0.011 48.23% medios somos protocolo 31.44% medios ext 57.20% actividades protocolo Cluster 17, Size: 412, ISim: 0.048, ESim: 0.012 53.70% facultad horas salon actos 53.94% facultad horas salon curso Cluster 18, Size: 376, ISim: 0.040, ESim: 0.013 39.01% acto espa jose 41.49% acto horas paraninfo jose Cluster 19, Size: 492, ISim: 0.022, ESim: 0.010 55.59% investigacion protocolo 34.71% alumnos estudiantes universitario protocolo
rbr	Cluster 0, Size: 30, ISim: 1.000, ESim: 0.004 100.00% tomcat apache sans family report Cluster 1, Size: 55, ISim: 1.000, ESim: 0.005 100.00% venus transito fotos Cluster 2, Size: 46, ISim: 0.990, ESim: 0.015 97.83% genericas mostrando fotografico archivo anterior Cluster 3, Size: 131, ISim: 0.970, ESim: 0.007 98.47% inexistente servidor encontrado consulte	Cluster 0, Size: 30, ISim: 1.000, ESim: 0.004 100.00% tomcat apache sans family report Cluster 1, Size: 46, ISim: 0.990, ESim: 0.015 97.83% genericas mostrando fotografico archivo anterior Cluster 2, Size: 57, ISim: 0.956, ESim: 0.005 67.25% venus transito telescopio 75.00% venus transito fotos nubes Cluster 3, Size: 133, ISim: 0.955, ESim: 0.006	Cluster 0, Size: 30, ISim: 1.000, ESim: 0.004 100.00% tomcat apache sans family report Cluster 1, Size: 129, ISim: 1.000, ESim: 0.007 100.00% inexistente servidor encontrado consulte administrador Cluster 2, Size: 56, ISim: 0.981, ESim: 0.005 60.71% venus transito fotos nubes descargar Cluster 3, Size: 46, ISim: 0.990, ESim: 0.015 97.83% genericas mostrando fotografico archivo

	i1	i2	h1
	administrador	97.74% consulte servidor inexistente encontrado administrador	anterior
	Cluster 4, Size: 58, ISim: 0.919, ESim: 0.001 77.59% moved page click	Cluster 4, Size: 58, ISim: 0.919, ESim: 0.001 77.59% moved page click	Cluster 4, Size: 58, ISim: 0.919, ESim: 0.001 77.59% moved page click
	Cluster 5, Size: 38, ISim: 0.855, ESim: 0.004 100.00% ctrl mover arrastre acercar haga	Cluster 5, Size: 39, ISim: 0.817, ESim: 0.004 97.95% ctrl mover arrastre haga acercar	Cluster 5, Size: 38, ISim: 0.855, ESim: 0.004 100.00% ctrl mover arrastre acercar haga
	Cluster 6, Size: 27, ISim: 0.836, ESim: 0.058 94.07% castro omar adelanto listado seccion	Cluster 6, Size: 195, ISim: 0.690, ESim: 0.052 94.77% gaceta local listado seccion actualidad	Cluster 6, Size: 179, ISim: 0.767, ESim: 0.055 97.32% gaceta local listado seccion actualidad
	Cluster 7, Size: 174, ISim: 0.786, ESim: 0.056 97.82% gaceta local listado seccion actualidad	Cluster 7, Size: 138, ISim: 0.651, ESim: 0.059 99.42% tribuna listado seccion actualidad mes	Cluster 7, Size: 136, ISim: 0.661, ESim: 0.060 100.00% tribuna listado seccion actualidad mes
	Cluster 8, Size: 22, ISim: 0.714, ESim: 0.012 83.64% actividades protocolo null patio escuelas	Cluster 8, Size: 176, ISim: 0.488, ESim: 0.051 99.66% adelanto listado seccion actualidad mes	Cluster 8, Size: 176, ISim: 0.487, ESim: 0.052 99.20% adelanto listado seccion actualidad mes
	Cluster 9, Size: 133, ISim: 0.674, ESim: 0.061 100.00% tribuna listado seccion actualidad mes	Cluster 9, Size: 92, ISim: 0.359, ESim: 0.005 85.43% admission exams entrance registration courses	Cluster 9, Size: 91, ISim: 0.365, ESim: 0.005 100.00% admission exams entrance registration map
	Cluster 10, Size: 30, ISim: 0.625, ESim: 0.048 98.00% zamora opinion listado seccion actualidad	Cluster 10, Size: 140, ISim: 0.282, ESim: 0.007 88.71% antartida expedicion foto titulo fotos	Cluster 10, Size: 108, ISim: 0.374, ESim: 0.047 95.93% castilla leon listado seccion actualidad
	Cluster 11, Size: 145, ISim: 0.525, ESim: 0.057 100.00% adelanto listado seccion actualidad mes	Cluster 11, Size: 137, ISim: 0.273, ESim: 0.014 92.85% acceso pruebas investigacion mapa agenda	Cluster 11, Size: 132, ISim: 0.290, ESim: 0.015 95.61% acceso pruebas investigacion agenda mecenazgo
	Cluster 12, Size: 85, ISim: 0.395, ESim: 0.005 85.65% admission exams entrance courses registration	Cluster 12, Size: 216, ISim: 0.228, ESim: 0.042 88.24% castilla listado seccion actualidad mes	Cluster 12, Size: 156, ISim: 0.242, ESim: 0.007 85.51% antartida expedicion foto fotos titulo
	Cluster 13, Size: 111, ISim: 0.357, ESim: 0.015 99.82% acceso pruebas investigacion agenda mecenazgo	Cluster 13, Size: 98, ISim: 0.142, ESim: 0.011 42.35% aprobacion procede informe consejo 50.00% actividades	Cluster 13, Size: 121, ISim: 0.216, ESim: 0.045 99.34% listado seccion actualidad mes dia
	Cluster 14, Size: 105, ISim: 0.387, ESim: 0.048 97.33% castilla leon listado seccion actualidad	Cluster 14, Size: 156, ISim: 0.144, ESim: 0.014 73.21% rectorado retratos sala rueda prensa	Cluster 14, Size: 312, ISim: 0.082, ESim: 0.012 46.58% sala rectorado retratos 53.63% sala rueda rectorado 60.36% sala prensa rueda
	Cluster 15, Size: 121, ISim: 0.337, ESim: 0.007 95.37% antartida expedicion foto titulo fotos	Cluster 15, Size: 118, ISim: 0.084, ESim: 0.010 27.97% fac ext 24.15% fac seleccione 39.83% medios somos 32.63% medios fac	Cluster 15, Size: 163, ISim: 0.056, ESim: 0.011 32.64% aprobacion procede informe consejo gobierno
	Cluster 16, Size: 100, ISim: 0.213, ESim: 0.046 100.00% listado seccion actualidad mes dia		Cluster 16, Size: 188, ISim: 0.054, ESim: 0.011

	i1	i2	h1
	Cluster 17, Size: 166, ISim: 0.151, ESim: 0.015 74.10% sala rueda prensa 73.29% retratos rectorado sala Cluster 18, Size: 129, ISim: 0.091, ESim: 0.011 29.46% fac mail seleccione 33.33% fac ext mail 28.68% medios fac Cluster 19, Size: 1383, ISim: 0.024, ESim: 0.008 43.07% horas facultad curso protocolo salon	Cluster 16, Size: 212, ISim: 0.064, ESim: 0.012 39.47% fonseca hospederia colegio 44.10% fonseca hospederia prensa rueda Cluster 17, Size: 438, ISim: 0.047, ESim: 0.012 47.49% facultad horas salon curso actos Cluster 18, Size: 320, ISim: 0.043, ESim: 0.013 33.13% premio jose 42.19% acto rector jose 36.98% acto rector paraninfo Cluster 19, Size: 290, ISim: 0.032, ESim: 0.011 33.45% investigacion 20.78% alumnos estudiantes selectividad mensaje	43.09% medios protocolo somos 23.67% medios ext 57.18% actividades protocolo Cluster 17, Size: 416, ISim: 0.048, ESim: 0.012 53.79% facultad horas salon actos 54.03% facultad horas salon curso Cluster 18, Size: 326, ISim: 0.044, ESim: 0.013 40.95% espa rector 35.28% premio espa 36.71% acto paraninfo rector Cluster 19, Size: 228, ISim: 0.038, ESim: 0.010 15.46% seleccione licenciado fac titulacion 28.07% alumnos licenciado titulacion
agglo	Cluster 0, Size: 2, ISim: 1.000, ESim: 0.000 100.00% index Cluster 1, Size: 100, ISim: 0.368, ESim: 0.017 96.80% acceso pruebas investigacion agenda mecenazgo Cluster 2, Size: 129, ISim: 1.000, ESim: 0.007 100.00% inexistente servidor encontrado consulte administrador Cluster 3, Size: 153, ISim: 0.850, ESim: 0.061 98.69% gaceta local listado seccion actualidad Cluster 4, Size: 55, ISim: 1.000, ESim: 0.005 100.00% venus transito fotos Cluster 5, Size: 58, ISim: 0.919, ESim: 0.001 77.59% moved page click Cluster 6, Size: 46, ISim: 0.990, ESim: 0.015 97.83% genericas mostrando fotografico archivo anterior Cluster 7, Size: 118, ISim: 0.341, ESim: 0.008 97.29% antartida expedicion foto fotos titulo Cluster 8, Size: 1595, ISim: 0.023, ESim: 0.007	Cluster 0, Size: 2, ISim: 1.000, ESim: 0.000 100.00% index Cluster 1, Size: 222, ISim: 0.234, ESim: 0.043 89.46% listado castilla seccion actualidad mes Cluster 2, Size: 80, ISim: 0.392, ESim: 0.005 100.00% admission exams entrance registration map Cluster 3, Size: 129, ISim: 1.000, ESim: 0.007 100.00% inexistente servidor encontrado consulte administrador Cluster 4, Size: 38, ISim: 0.855, ESim: 0.004 100.00% ctrl mover arrastre acercar haga Cluster 5, Size: 197, ISim: 0.051, ESim: 0.011 24.11% telefono somos 26.40% telefono mail 23.10% medios somos 21.32% medios fac mail Cluster 6, Size: 137, ISim: 0.278, ESim: 0.008 92.41% antartida expedicion foto fotos titulo Cluster 7, Size: 183, ISim: 0.680, ESim: 0.056 93.55% gaceta local listado seccion actualidad	Cluster 0, Size: 226, ISim: 0.044, ESim: 0.012 22.27% telefono mail ext 18.92% medios mail fac ext Cluster 1, Size: 129, ISim: 1.000, ESim: 0.007 100.00% inexistente servidor encontrado consulte administrador Cluster 2, Size: 119, ISim: 0.338, ESim: 0.008 96.97% antartida expedicion foto fotos titulo Cluster 3, Size: 153, ISim: 0.850, ESim: 0.061 98.69% gaceta local listado seccion actualidad Cluster 4, Size: 169, ISim: 0.223, ESim: 0.046 100.00% listado seccion actualidad mes dia Cluster 5, Size: 58, ISim: 0.919, ESim: 0.001 77.59% moved page click Cluster 6, Size: 55, ISim: 1.000, ESim: 0.005 100.00% venus transito fotos Cluster 7, Size: 119, ISim: 0.299, ESim: 0.016 94.29% acceso pruebas investigacion agenda mecenazgo Cluster 8, Size: 139, ISim: 0.632, ESim: 0.062

	i1	i2	h1
	42.87% horas facultad protocolo lugar curso Cluster 9, Size: 99, ISim: 0.756, ESim: 0.069 100.00% tribuna listado seccion actualidad mes Cluster 10, Size: 38, ISim: 0.855, ESim: 0.004 100.00% ctrl mover arrastre acercar haga Cluster 11, Size: 30, ISim: 1.000, ESim: 0.004 100.00% tomcat apache sans family report Cluster 12, Size: 80, ISim: 0.392, ESim: 0.005 100.00% admission exams entrance registration map Cluster 13, Size: 92, ISim: 0.718, ESim: 0.072 100.00% adelanto listado seccion actualidad mes Cluster 14, Size: 38, ISim: 0.774, ESim: 0.072 79.47% deportes tribuna redaccion listado seccion Cluster 15, Size: 90, ISim: 0.382, ESim: 0.048 79.72% castilla listado seccion norte 90.83% castilla leon listado seccion Cluster 16, Size: 105, ISim: 0.186, ESim: 0.016 27.62% aprobacion procede 73.02% retratos rectorado sala Cluster 17, Size: 15, ISim: 1.000, ESim: 0.014 100.00% actividades protocolo patio escuelas Cluster 18, Size: 26, ISim: 0.688, ESim: 0.046 100.00% zamora opinion listado seccion actualidad Cluster 19, Size: 220, ISim: 0.201, ESim: 0.046 98.36% listado seccion actualidad mes dia	Cluster 8, Size: 58, ISim: 0.919, ESim: 0.001 77.59% moved page click Cluster 9, Size: 56, ISim: 0.981, ESim: 0.005 60.71% venus transito fotos nubes descargar Cluster 10, Size: 125, ISim: 0.285, ESim: 0.015 88.96% acceso pruebas investigacion agenda mecenazgo Cluster 11, Size: 140, ISim: 0.628, ESim: 0.062 97.14% tribuna listado seccion actualidad mes Cluster 12, Size: 46, ISim: 0.990, ESim: 0.015 97.83% genericas mostrando fotografico archivo anterior Cluster 13, Size: 586, ISim: 0.026, ESim: 0.012 28.05% acto premio espa rector horas Cluster 14, Size: 173, ISim: 0.459, ESim: 0.053 97.92% adelanto listado seccion actualidad mes Cluster 15, Size: 30, ISim: 1.000, ESim: 0.004 100.00% tomcat apache sans family report Cluster 16, Size: 60, ISim: 0.229, ESim: 0.012 47.78% aprobacion procede informe 32.50% actividades null Cluster 17, Size: 536, ISim: 0.038, ESim: 0.012 43.38% facultad horas salon tendra 42.58% facultad horas salon curso Cluster 18, Size: 97, ISim: 0.188, ESim: 0.016 69.69% retratos rectorado sala rueda prensa Cluster 19, Size: 194, ISim: 0.062, ESim: 0.013 46.70% fonseca hospederia rueda prensa horas	97.12% tribuna listado seccion actualidad mes Cluster 9, Size: 46, ISim: 0.990, ESim: 0.015 97.83% genericas mostrando fotografico archivo anterior Cluster 10, Size: 190, ISim: 0.415, ESim: 0.052 95.58% adelanto listado seccion actualidad mes Cluster 11, Size: 267, ISim: 0.034, ESim: 0.011 17.98% aprobacion procede informe consejo universitario Cluster 12, Size: 298, ISim: 0.066, ESim: 0.013 38.52% sala retratos rectorado rueda prensa Cluster 13, Size: 86, ISim: 0.366, ESim: 0.006 96.51% admission exams entrance registration map Cluster 14, Size: 573, ISim: 0.038, ESim: 0.012 28.66% facultad curso salon derecho 41.32% facultad horas curso salon Cluster 15, Size: 38, ISim: 0.855, ESim: 0.004 100.00% ctrl mover arrastre acercar haga Cluster 16, Size: 30, ISim: 1.000, ESim: 0.004 100.00% tomcat apache sans family report Cluster 17, Size: 67, ISim: 0.452, ESim: 0.048 82.09% castilla norte listado 83.21% castilla leon mundo listado Cluster 18, Size: 127, ISim: 0.073, ESim: 0.013 30.71% iberoamerica instituto portugal 26.77% actividades vertiente Cluster 19, Size: 200, ISim: 0.044, ESim: 0.013 34.67% acto rector paraninfo 35.50% acto premio espa 31.33% acto premio paraninfo

El método rb obtiene 10 *clusters* con correspondencia en las tres funciones criterio, justo el doble que para el caso de estudio con 10 *clusters*. De estos 10, 7 pueden considerarse idénticos. Se ampliarían a 14 los *clusters* que coinciden si comparamos los datos de la función criterio i1 con h1, de nuevo el doble que los datos obtenidos para 10 *clusters*. Con i1, quedarían 1554 documentos sin clasificar, con i2, 1509 y con h1 1412.

El método rbr obtiene 13 *clusters* con correspondencia en las tres funciones criterio. Se ampliarían a 15 los *clusters* que coinciden si comparamos los datos de la función criterio i1 con h1. En ambos casos, algo más del doble que los datos obtenidos para 10 *clusters*. Con i1 quedarían 1561 documentos sin clasificar, con i2, 1378 y con h1 1158.

Por último el método agglo obtiene 12 *clusters* comunes a las tres funciones que pasa a 13 si se comparan i1 con h1. Con i1 quedarían 1843 documentos sin clasificar, con i2 serían 1513 y con h1 1193.

Viendo en conjunto los resultados, la combinación que más documentos clasifica de forma equilibrada serían el método rbr con la función criterio h1, igual que para 10 *clusters*, pero también el método agglo con la misma función h1.

Si comparamos las dos tablas veremos que al generar nuevos *clusters*, de 10 a 20, primero se dividen aquellos *clusters* que ya estaban formados pero que presentaban *sub-clusters*, por eso siguen apareciendo al final *clusters* residuales con gran número de documentos.

Los datos correspondientes al caso de estudio B se incluyen en el apéndice C.

10 APLICACIÓN DEL MODELO DE CLASIFICACIÓN PARA LA OBTENCIÓN DE DIRECTORIOS WEB DE FORMA AUTOMATIZADA, NO SUPERVISADA

10.1 Introducción

Una vez determinada una metodología de trabajo en el proceso de clasificar el contenido completo de los portales de información, una vez analizadas las principales combinaciones de métodos de clustering y funciones criterio, y después de determinar cuáles son los valores idóneos para el tipo de problema que nos ha ocupado, era necesario aplicar estos resultados para demostrar la posibilidad de poder obtener directorios web de forma automatizada, con las consiguientes ventajas que supone sobre los costosos métodos tradicionales.

Para ello se estudian diferentes alternativas tecnológicas existentes en la actualidad, especialmente indicadas para la web, cono son la tecnología XML/XSLT y el API JAXP, constatando que es posible obtener soluciones desde diferentes enfoques, dejando abierto el camino a nuevas propuestas que vayan apareciendo. Tanto en una solución como en la otra se hace uso de documentos XML para almacenar datos intermedios, verificando lo apropiado de este lenguaje para el intercambio de información entre aplicaciones, máxime en este caso en que se ha buscado separar las interfaces gráficas de la lógica de negocio.

Se hacen dos estudios independientes, uno para directorios de un nivel y otro para directorios jerarquizados multinivel —aplicando en cada uno de ellos una de las técnicas indicadas— ya que son ambos tipos de directorio los que se utilizan hoy día de forma indistinta en diversos portales.

Una vez verificada la utilidad de la aplicación del *clustering* a la obtención de los directorios web, se analizan los resultados con diferentes ejemplos, para finalizar dando unas recomendaciones de la forma que se debe aplicar en la práctica para obtener los mejores resultados.

10.2 Obtención de un directorio de un solo nivel mediante tecnología XML/XSLT

10.2.1 Introducción al proceso de presentación de resultados mediante XML/XSLT

Una vez obtenidos los resultados pasamos a estandarizarlos de forma que puedan ser más fácilmente accesibles, lo más independiente posible de las plataformas, adaptables a distintos dispositivos y configurables, para ello se recurre a la arquitectura formada por XML (*eXtensible Markup Language*)[2,3] y hojas de estilo XSLT (*eXtensible Stylesheet Language Transformations*)[4,5] [Kay, 2001].

Cada vez más se utilizan arquitecturas que separan claramente la parte de procesamiento de los datos, de la parte de presentación, utilizando para ellos diversos patrones, como el MVC (*Model View Controller*) [Reenskaug, 1979] [Burbeck, 1992] o el MVP (*Model View Presenter*) [Potel, 1996] [Bower y McGlashan, 2000]. En ambos casos se persigue separar la lógica de la aplicación de la interfaz de usuario, ya que generalmente esta última es mucho más cambiante. Lo habitual es que se realicen distintas tareas en el servidor (procesamientos, intercambios, modificaciones), siendo uno de los lenguajes preferidos XML, para después mostrar resultados en el cliente, siendo el formato habitual HTML. Por ello una de las técnicas con más auge están teniendo son las transformaciones de tipo XSLT, que permiten convertir los datos XML en otros formatos, como HTML, XHTML, otro formato XML, JPEG, VRML (*Virtual Reality Modeling Language*), SVG (*Scalable Vector Graphics*), código Java, texto plano o incluso PDF. Además de XML/XSL, suele utilizarse algún lenguaje como Java, PHP, o Perl para utilizar las API's que facilitan la interconexión.

Según [Burke, 2002] se pueden definir claramente las funciones de cada lenguaje, Java se utiliza para la lógica empresarial, las consultas y actualizaciones de

[2] *Extensible Markup Language* (XML), W3C , http://www.w3.org/XML/
[3] *Extensible Markup Language* (XML) 1.0 (Second Edition), W3C Recommendation 6 October 2000, http://www.w3.org/TR/REC-xml
[4] *Extensible Stylesheet Language* (XSL) Version 1.0, W3C Recommendation 15 October 2001, http://www.w3.org/TR/xsl/
[5] *XSL Transformations* (XSLT) Version 1.0, W3C Recommendation 16 November 1999, http://www.w3.org/TR/xslt

bases de datos y para crear los datos XML. El XML es responsable de los datos en bruto y XSLT transforma el XML en HTML para que los navegadores lo presenten, pudiendo generar distintas hojas de estilo XSLT, para distintos navegadores, distintos lenguajes o incluso distintos dispositivos.

10.2.2 Justificación de la tecnología XML/XSLT

A lo largo de los años han sido muchas las tecnologías web que se han venido utilizando, sin que ninguna de ellas haya obtenido una supremacía clara en el mercado. Algunas de ellas se han ido quedando en el camino como los CGI (*Common Gateway Interface*), protocolo para crear la interfaz de aplicaciones externas escritos en diferentes lenguajes soportados por los servidores Web, cuyo principal inconveniente es la incompatibilidad entre ellos, su rigidez y la dificultad de herramientas y estándares. Son muchas otras las tecnologías basadas en servidores Web, como PHP, Java junto con *servlets* (se cargan una vez en memoria y se mantienen para posteriores solicitudes) y páginas JSP (código HTML, junto con indicadores que se amplían de forma dinámica mediante el motor JSP produciendo páginas dinámicas), o bien páginas JSP y EJB (*Enterprise JavaBeans*, modelo de componentes estándar del lado del servidor). Todas ellas presentan la dificultad de poder separar la lógica de negocio de la generación de la interfaz.

La aparición de XML, su posterior estandarización, y su gran aceptación han hecho de este formato un medio ideal para representar la estructuración de contenido, llegando a convertirse en uno de los formatos de intercambio en la Web más difundidos. Se trata de un formato para representar todo tipo de información, por lo tanto muy adecuado para el procesamiento de documentos, como recomienda [de la Rosa, 2003] o [Berrocal et al., 2000]. Su aceptación ha sido tal, que incluso se han desarrollado modelos de metadatos basados en XML, como el *Resource Description Framework* (RDF) utilizados para la estructuración de la información en las bibliotecas digitales y para la optimización de la recuperación en Internet [Méndez, 1999].

10.2.3 Fundamentos de la tecnología XML/XSLT

El lenguaje XML es una evolución de los lenguajes SGML (*Standard Generalized Markup Language*) y HTML (*HyperText Markup Language*). SGML se basa en el concepto de tipo de documento, que puede interpretarse como una abstracción de documentos con unas características y objetivos comunes. Además incorpora la idea de la inclusión de marcas en el texto de los documentos para diferenciar estructura y contenido informativo. Se trata de un metalenguaje. Este lenguaje evoluciona hasta convertirse en norma internacional ISO 8879:1986. A su vez se desarrolla el lenguaje HTML, que consiste en una aplicación de SGML (es decir en un lenguaje de marcas para un tipo de documento especial, uno de los muchos que podrían definirse con SGML, en el que se han especificado sus indicadores y en el que no se pueden añadir nuevos elementos).

Por su parte XML también es un metalenguaje por lo que tampoco define ningún indicador propio sino simplemente unas reglas que deben cumplir los documentos para estar bien formados. Aún siendo un perfil de SGML, y por tanto una versión simplificada que eliminaba las características menos utilizadas su difusión y aceptación ha sido mayor precisamente por la reducción de la complejidad.

Para especificar las características de un vocabulario XML se pueden utilizar DTD's (Definición de tipo de Documento) o *Schemas*. Ambos son metalenguajes, si bien el esquema XML ofrece capacidades de validación mucho más sofisticadas que las DTD, que en algunas ocasiones son difíciles o imposible especificar mediante las DTD y por otra parte los esquemas son documentos XML en sí mismos.

Un documento que sigue las reglas especificadas para XML se dice que es un documento bien formado, si además cumple un conjunto de reglas especificadas en un DTD o en un esquema XML, se dice que es un documento válido.

XSLT, es una recomendación oficial del *World Wide Web Consortium* (W3C). Es un poderoso lenguaje de transformaciones de documentos XML en casi cualquier cosa. Se han definido dos familias de estándares para las hojas de estilo. La más antigua y simple es CSS (*Cascading Style Sheets*), un mecanismo para definir propiedades de

aspecto de los elementos. Aunque puede utilizarse con XML, lo habitual es que se haga con documentos HTML, para especificar su aspecto intentando separarlo del contenido. Presentan algunas limitaciones: no pueden cambiar el orden de aparición de los elementos en el documento, no pueden realizar cálculos, y no pueden combinar múltiples documentos, esto es debido a que no fueron diseñadas para este fin, de ahí la aparición del segundo estándar, las hojas XSLT, como lenguaje de transformación de documentos, muy influenciado por los lenguajes de programación funcionales, como *Lisp*, *Scheme* o *Haskell*.

Algunos de los escenarios en que son adecuadas son:

- Sitios web que manejan distintos dispositivos.
- Cuando se necesitan utilizar distintos sistemas de base de datos, con lo que se adopta XML como lenguaje común y después se realizan las transformaciones necesarias.
- Sitios con mucha actividad y rediseño de las presentaciones cada poco tiempo.
- Procesamiento conjunto de diversos documentos, con obtención de características comunes o procesamiento de determinas características.

Para aplicar las hojas de estilo XSLT sobre los documentos XML se recurre a los procesadores XSLT, que suelen estar en conjunción con un analizador sintáctico (*parser*) XML. Son múltiples los procesadores que han ido apareciendo quizá los más populares sean:

- **XT**. Escrito por James Clark, editor de la especificación XSLT, por lo que se adapta escrupulosamente a los estándares.
- **LotusXSL**. Es un procesador Java de IBM *Alphaworks*, donado en 1999 a Apache, para constituir la base de Xalan, aunque se ha mantenido en paralelo.
- **Xalan**. Escrito en Java, por lo que se necesita la máquina virtual java (JVM), como la de Microsoft, Sun o IBM. Necesita el analizador sintáctico Xerces, que se incluye en la distribución.

- **Saxon**. De Michael Key. Ofrece soporte total para la especificación XSLT, está muy actualizado y documentado. También escrito en Java, con su propio analizador sintáctico XML.
- **XsltProc**. Desarrollado para Linux, basado en la biblioteca XML/XSL de Gnome. Se proporciona como un ejecutable por lo que no es necesario instalar Java y es muy rápido.
- **Msxml**. Desarrollado por Microsoft, ofrece soporte total sobre la recomendación XSLT. Viene instalado por defecto en los sistemas operativos Windows.
- **Jaxp**. Una API basada en Java desarrollada por Sun, que pretende ofrecer una interfaz estándar para una amplia gama de analizadores sintácticos y procesadores XSLT. Por defecto incorpora Xalan como procesador predefinido.

Muchos de estos procesadores pueden utilizarse como bibliotecas de clases desde diversos lenguajes de programación, y algunos navegadores los traen implementados desde 2005. Es el caso de *Mozilla*, *FireFox* o *Internet Explorer*. De esta forma las transformaciones pueden aplicarse desde el cliente en lugar de en el servidor, con lo que en ocasiones aporta ventajas, como reducir la carga de trabajo del servidor o reducir el ancho de banda necesario en las comunicaciones ya que basta con enviar el XML una sola vez.

La técnica XSLT se ajusta de forma bastante directa hacia el modelo MVC. En el caso de aplicaciones de servidor, el XML representaría el modelo, el *servlet* representa el controlador y el XSLT la vista, como se puede apreciar en la figura 41. En ocasiones las hojas XSLT pueden contener una parte de lógica, que hace que no sea tan marcada la separación entre vista y controlador. En el caso en que se trabaje sólo desde el lado del cliente, la parte de controlador puede implementarse en las hojas XSLT, mediante la introducción de código en algún lenguaje de scripts, como *jscripts* o *vbscripts*.

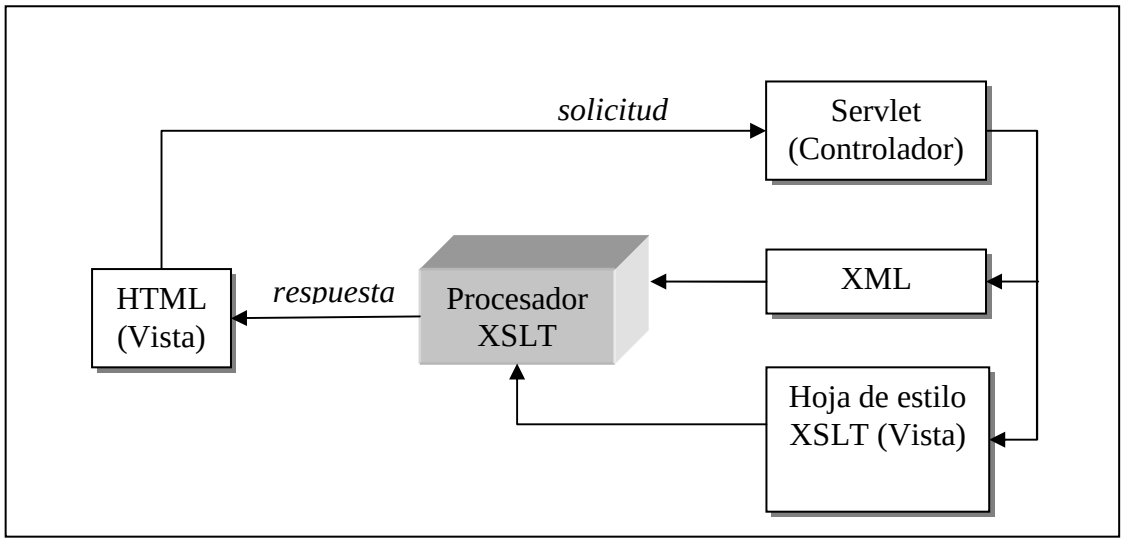

Figura 41. Modelo conceptual XSLT, adaptado de [Burke, 2002].

10.2.4 Proceso de presentación de la vista mediante hojas de estilo XSLT

Como hemos visto, el modelo de datos para una aplicación web, aplicando la tecnología XSLT, se puede obtener de forma independiente generando como primera fase el documento XML. Nuestro documento XML va a tener dos elementos bien diferenciados: por una parte cada uno de los documentos y por otra parte los *clusters* en los que se van a agrupar esos documentos. De los documentos habrá que recoger un identificativo, su título, su localización física local y su dirección web. Para los *clusters*, de nuevo un identificativo único, sus términos característicos y los documentos asociados a cada *cluster*. Estos dos elementos diferenciados van a dar lugar a dos momentos distintos en la elaboración del documento XML. En una primera etapa habría que recoger la información sobre los documentos. Esto podría hacerse en la primera fase del proceso de clasificación descrito en los capítulos 8 y 9, y que se resume en la figura 16, pero de igual forma podría realizarse de forma previa a esa fase o justo a su finalización. La segunda etapa correspondería a la obtención de la información sobre los *clusters*, en este caso es imprescindible haber realizado todo el proceso de clasificación y partir como entrada de los resultados obtenidos en el proceso de clasificación.

El proceso de obtención del documento XML, junto con el proceso de clasificación, quedaría plasmado en la figura 42.

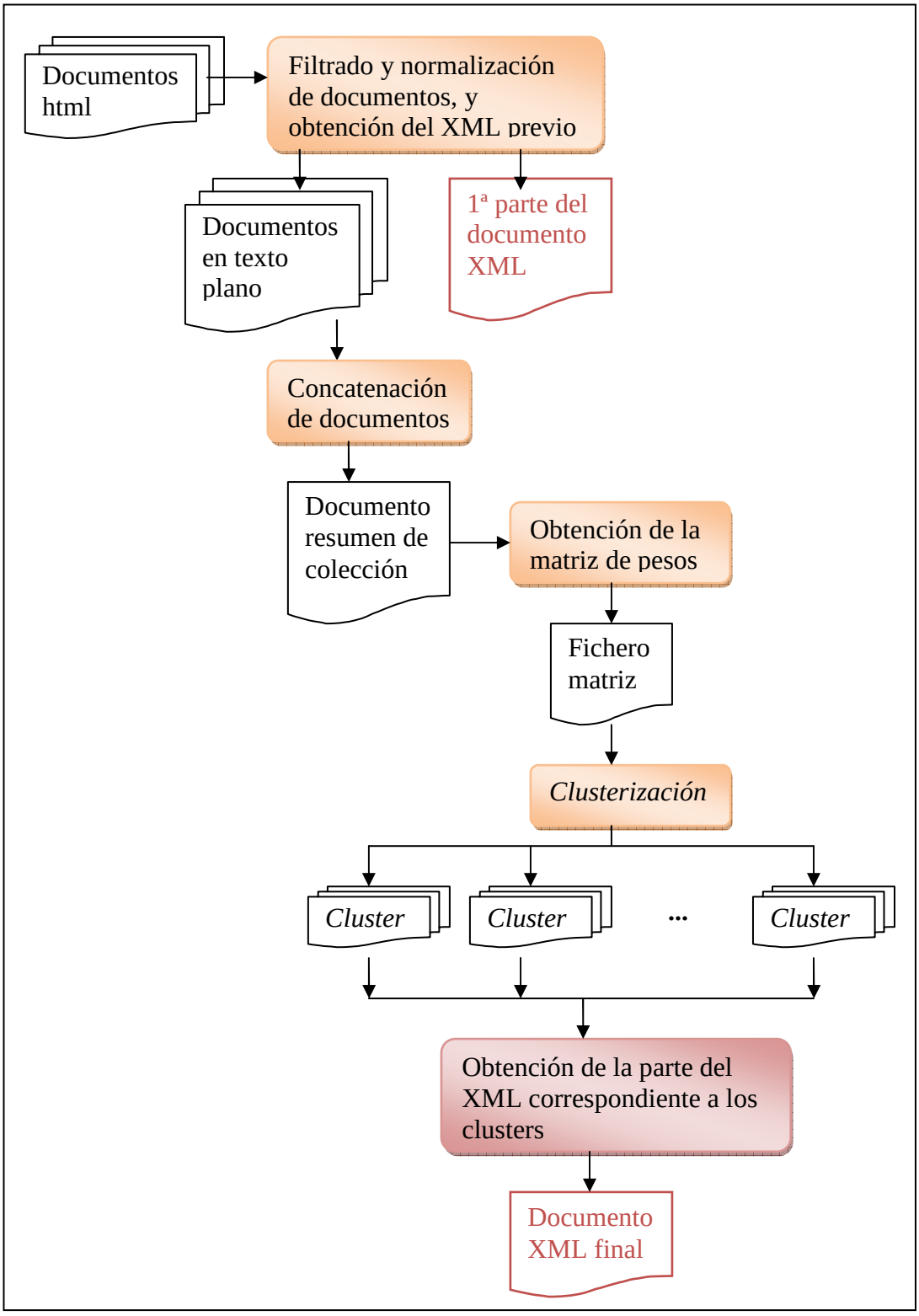

Figura 42. Proceso de obtención del documento XML.

Como ya se ha indicado, el documento XML se va generando en las distintas fases, mediante un proceso incremental, en el que se van añadiendo nuevas etiquetas

XML. El documento final va a constar de dos partes claramente diferenciadas, que se encuentran bajo las etiquetas “paginasBase” y “clusters”. Véase la figura 45. La primera parte del documento, agrupada bajo la etiqueta “paginasBase”, contiene información detallada de cada uno de los documentos individuales: sus características descriptivas (*features*), su URL, y su ruta local. Se obtiene mediante los scripts procesa.bat, y previo.pl. La segunda parte del documento XML, una vez agrupados los documentos en *clusters*, se añade para agregar las etiquetas XML correspondientes a los *clusters*, que van a recoger la información de las características descriptivas resumen de cada *cluster* y la relación de documentos que lo componen. Un detalle del resultado obtenido en esta parte se aprecia en la figura 46. En este caso se procesa mediante el script metercluster.pl, que se recoge en el anexo D.

Una vez generado el XML hay que validarlo, se puede hacer mediante un DTD o mediante un *schema* XML. El DTD utilizado en las primeras fases es el que se recoge en la figura 43, y el esquema W3C final aparece en la figura 44. Un esquema contraído del documento XML aparece en la figura 45, mientras que si expandimos algunos de sus elementos obtendríamos el resultado de la figura 46.

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!ELEMENT IR (paginasBase, clusters)>
      <!ELEMENT paginasBase (elemento*)>
            <!ELEMENT elemento (URL, rutaLocal)>
                  <!ELEMENT URL (#PCDATA)>
                  <!ELEMENT rutaLocal (#PCDATA)>
```

Figura 43. DTD utilizado para validar el documento XML.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified">
	<xs:import namespace="http://www.w3.org/XML/1998/namespace"/>
	<xs:element name="IR">
		<xs:complexType>
			<xs:sequence>
				<xs:element ref="paginasBase"/>
				<xs:element ref="clusters"/>
			</xs:sequence>
		</xs:complexType>
	</xs:element>
	<xs:element name="paginasBase">
		<xs:complexType>
			<xs:sequence>
			<xs:element ref="elemento" minOccurs="0"
maxOccurs="unbounded"/>
			</xs:sequence>
		</xs:complexType>
	</xs:element>

	<xs:element name="elemento">
		<xs:complexType>
			<xs:sequence>
				<xs:element ref="titulo"/>
				<xs:element ref="URL"/>
				<xs:element ref="rutaLocal"/>
			</xs:sequence>
			<xs:attribute name="identif" use="required" type="xs:ID"/>
		</xs:complexType>
	</xs:element>
	<xs:element name="titulo">
		<xs:complexType mixed="true"/>
	</xs:element>
	<xs:element name="URL">
		<xs:complexType mixed="true"/>
	</xs:element>
	<xs:element name="rutaLocal">
		<xs:complexType mixed="true"/>
	</xs:element>
	<xs:element name="clusters">
		<xs:complexType>
			<xs:sequence>
			<xs:element ref="cluster" minOccurs="0"
maxOccurs="unbounded"/>
			</xs:sequence>
		</xs:complexType>
	</xs:element>
	<xs:element name="cluster">
		<xs:complexType>
			<xs:sequence>
				<xs:element ref="features"/>
				<xs:element ref="eltos"/>
			</xs:sequence>
			<xs:attribute name="idcluster" use="required" type="xs:ID"/>
		</xs:complexType>
	</xs:element>
	<xs:element name="features">
		<xs:complexType mixed="true"/>
	</xs:element>
	<xs:element name="eltos">
		<xs:complexType>
			<xs:sequence>
			<xs:element ref="elto" minOccurs="0" maxOccurs="unbounded"/>
			</xs:sequence>
		</xs:complexType>
	</xs:element>
	<xs:element name="elto"/>
</xs:schema>
```

Figura 44. Esquema W3C utilizado para validar el documento XML.

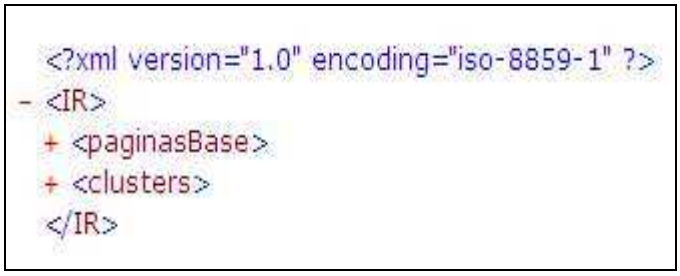

Figura 45. Documento XML resumido.

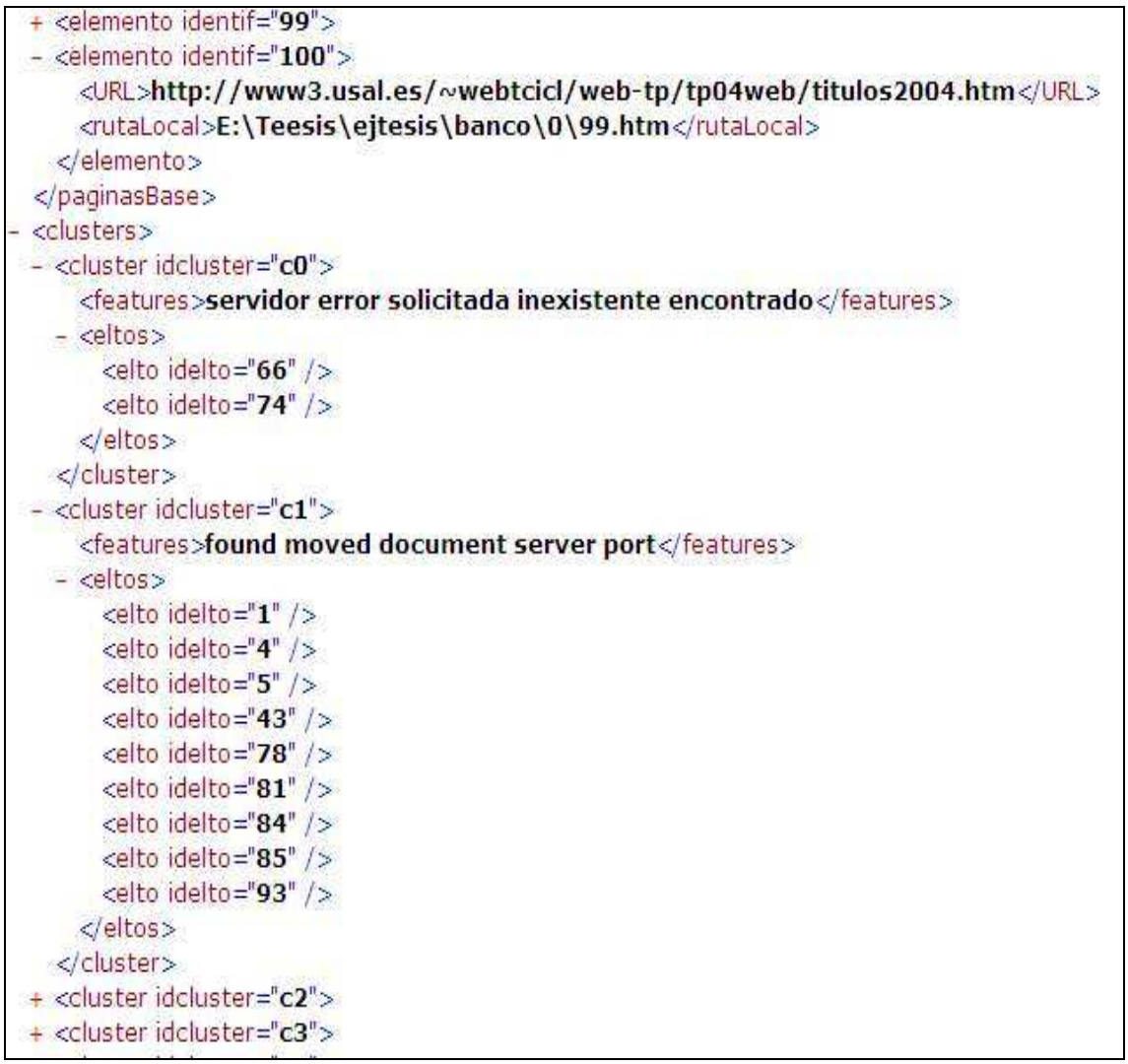

Figura 46. Fragmento de documento XML, para un solo nivel.

Una vez obtenido el documento XML anterior, y haciendo uso como se ha indicado del modelo MVC, por lo tanto ya disponiendo del modelo de datos, se ha desarrollado el controlador y parte de la vista mediante hojas de estilo XSLT. En concreto mediante dos hojas. La primera se encarga de mostrar el esquema general de directorio web, con tantas entradas como *clusters* se hayan obtenido, en el caso de ejemplo han sido 12 *clusters*, y la segunda hoja de estilo se encarga de bajar de nivel y mostrar los resultados del *cluster* que se haya seleccionado. La otra parte de la vista son los documentos HTML que se generan en tiempo de ejecución y que serán los que vea el usuario. El esquema que muestra esta adaptación del modelo MVC, para prescindir de servidor, es el que se recoge en la figura 47.

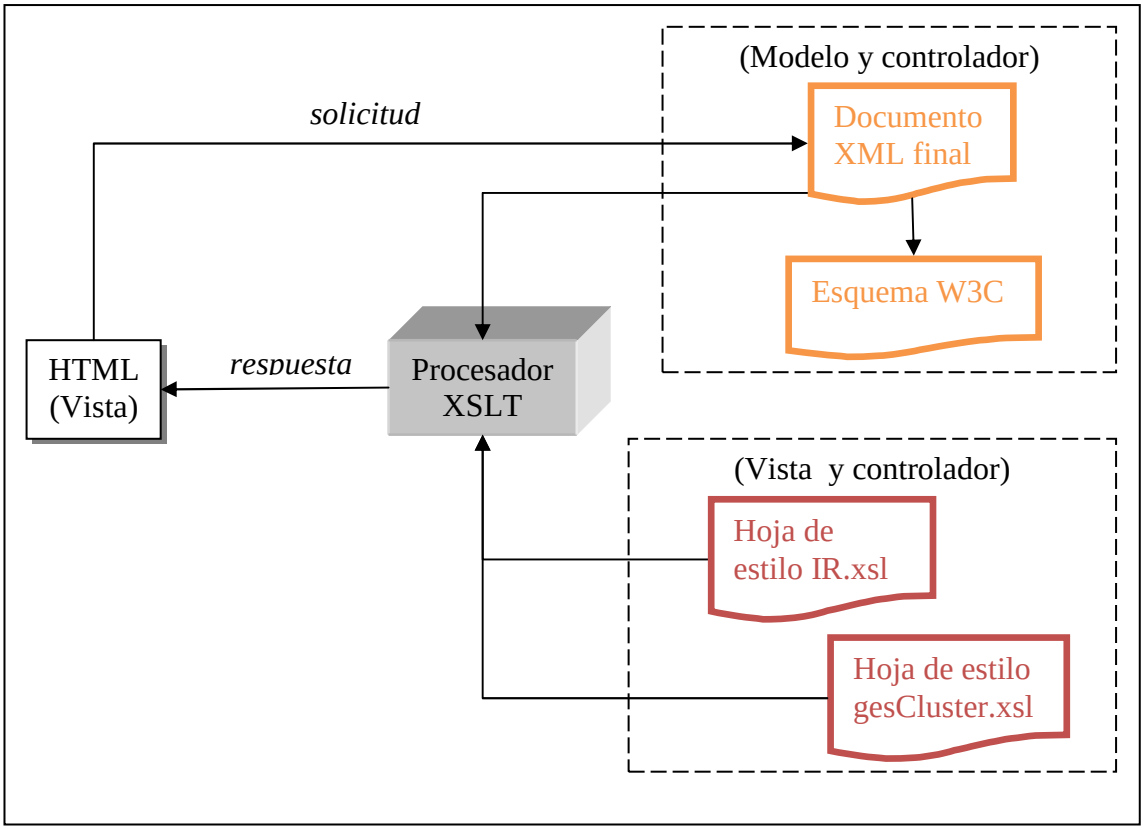

Figura 47. Proceso de generación de la vista del directorio web.

El esquema general de directorio web es el mostrado en la figura 48, y el contenido de la hoja de estilo “IR.xsl”, utilizada para generar de forma dinámica este directorio a partir del documento xml, se muestra en la figura 49.

Figura 48. Directorio web del sitio.

```
<?xml version='1.0' encoding='utf-8'?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:output method="html"/>
<xsl:template match="/">
      <xsl:call-template name="directorio"/>
</xsl:template>

<xsl:template name="directorio">
<html><head/>
<script type="text/javascript" src="switcherPag.js"></script>
      <p>
      <body background="lin_diag.gif" style="width: 365px; height:
313px"/>
      </p>
      <p align="center">
            <img height="216" hspace="0"
src="ESCUDOUSALpara%20pantalla.jpg" align="absMiddle" border="0"
style="width: 277px; position: fixed; height: 216px"/>
      </p>
      <p align="center"> 
            <font face="Vítor" color="#0000ff" size="5">
                  <strong>Directorio &amp;l sitio</strong>
            </font>
      </p>
      <p>
            <table width="80%"  align="center">
                  <tbody>
                        <TR bgColor="#D22020"><img height="1"/></TR>
                              <xsl:for-each
select="IR/clusters/cluster">
                                    <td width="30%">
                              <p>
                              <xsl:variable name="clust"
select="@idcluster" />
                                    <font face="Vítor" color="#0000ff"
size="3">                           <strong><a href="#" onclick =
"cargarHoja('IR.xml','gesCluster.xsl','nclust','{$clust}') ;"> <xsl:value-
of select="$clust"/> </a></strong>
                              </font>
                                    <br/>  <xsl:value-of
select="features"/>
                              </p>
                              </td>
                              <xsl:if test="(position() mod 3) = 0">
                                    <tr>  </tr>
                              </xsl:if>
                              </xsl:for-each>
                              <TR bgColor="#D22020"> <img height="2"/>
                                    <td /><td /><td />
                              </TR>
                        </tbody>
                  </table>
            </p>
            <p align="center"> </p>

</html>
</xsl:template>
</xsl:stylesheet>
```

Figura 49. Contenido de la hoja de estilo IR.xml.

Una vez seleccionado un *cluster*, una de las entradas del directorio del sitio, se accede a las páginas correspondientes a ese subdirectorio, presentando sus páginas correspondientes, como se aprecia en la figura 50.

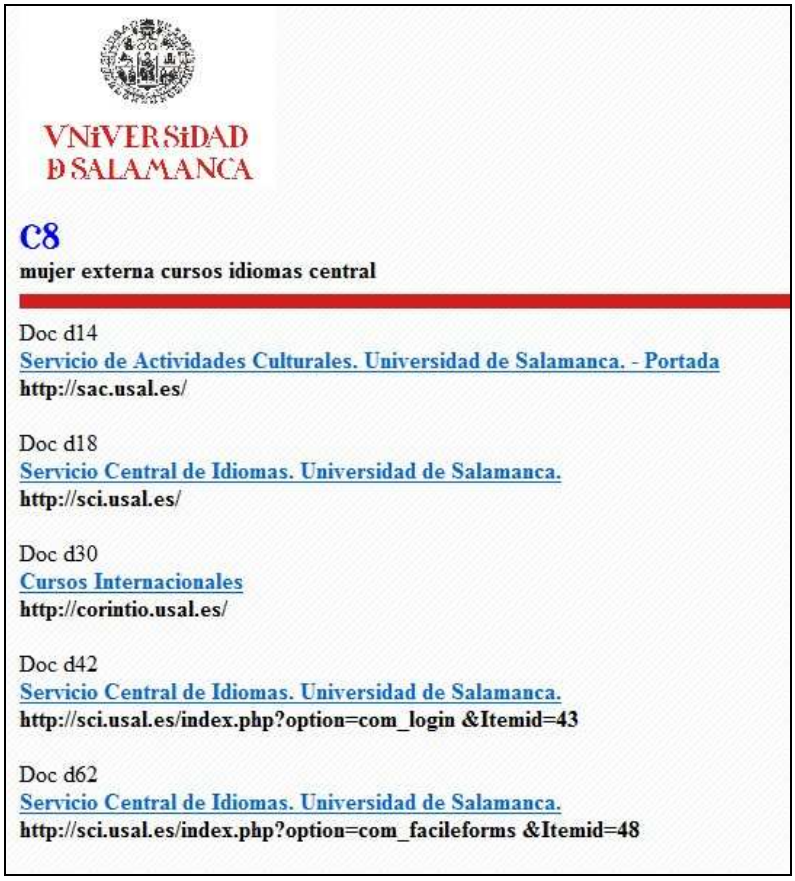

Figura 50. Páginas correspondientes a un *cluster*.

Si nos fijamos en las entradas correspondientes a los clusters 7 y 10 del directorio web de la figura 48, podríamos pensar que los filtros utilizados para quitar el código de programación de las páginas web habría fallado, ya que se observan términos claramente relacionados con la programación Java. Sin embargo si accedemos a estos clusters observamos que se trata de páginas web que contienen documentación sobre el JDK de Java, como vemos en la figura 51. Luego la clasificación ha sido correcta ya que esos términos aparecen como información válida para el usuario en las páginas, no como código, pero además la utilización de las técnicas de *clustering* nos ha permitido

detectar elementos que se escapan de la tónica general del portal y que por tanto podría ser analizada su conveniencia de publicación en la web.

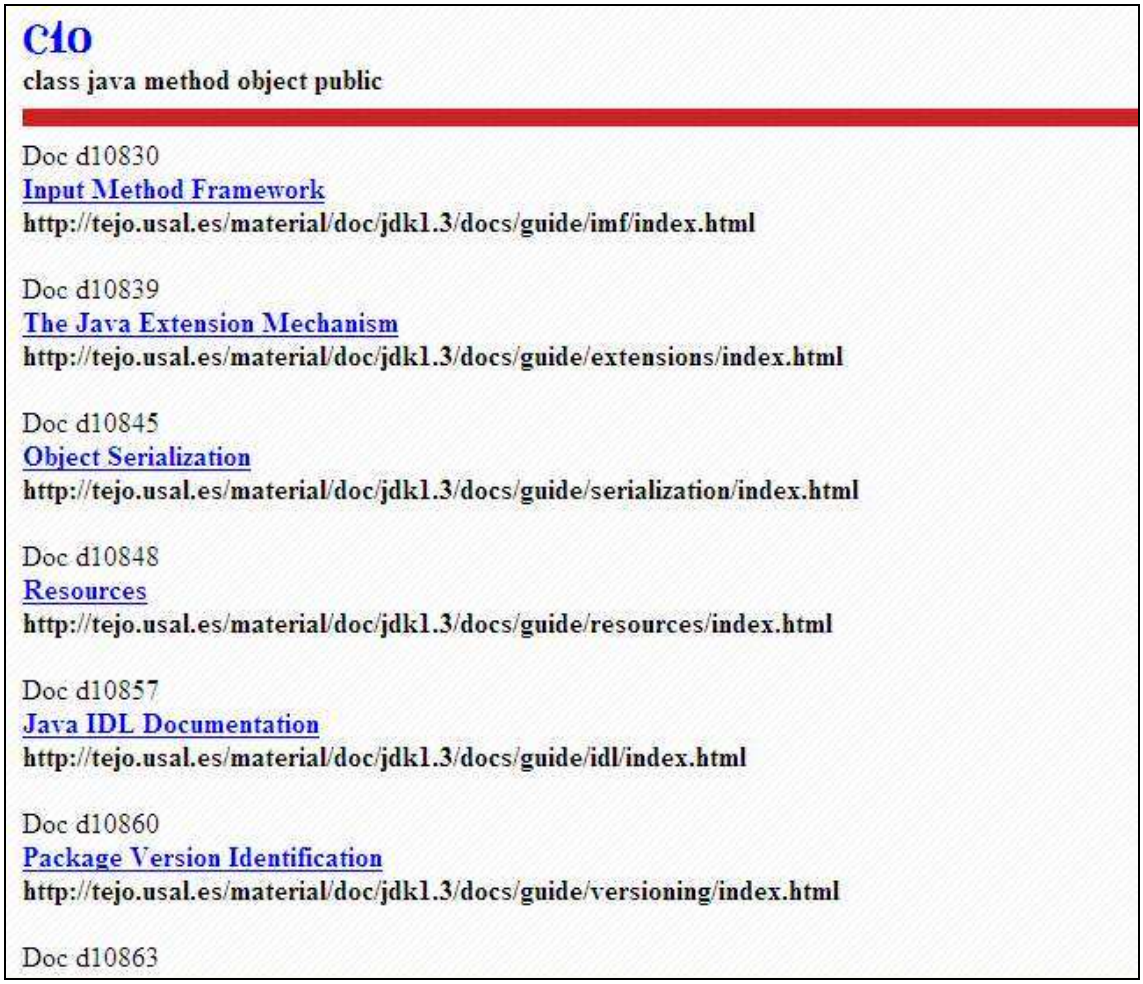

Figura 51. Página correspondiente a un cluster específico de documentación Java.

Como se ha podido constatar el sistema es capaz de clasificar cualquier sitio web, independientemente de su tamaño, habiéndose probado con sitios superiores a las 100.000 páginas, es el caso del sitio *www.usal.es*. Si bien en estos sitios tan extensos es habitual la utilización de distintos servidores con contenidos de lo más dispares, especializándose cada uno de ellos en temas específicos o subapartados bien claros, por lo que en estos casos es más eficiente utilizar la clasificación en cada uno de ellos por separado, pudiendo generar un nivel superior con una entrada hacia cada uno de estos servidores.

A medida que la información está más acotada a entornos más reducidos, los resultados de clasificación que se obtienen se asemejan mucho a los directorios organizados que se ofrecen en ocasiones como directorio o mapa del sitio. En la figura 52 se han superpuesto los resultados de clasificación obtenidos con la aplicación, junto con el directorio y entradas reales, para el portal *lazarillo.usal.es*. Se aprecia que existe una equivalencia entre determinados *clusters* obtenidos y entradas disponibles en el directorio, así por ejemplo, los *clusters* C4 y C5 tratan sobre redes inalámbricas, según se determina por los términos de clasificación, y en el directorio se observan también dos entradas para redes inalámbricas, cubriendo aspectos diferentes, uno etiquetado como “wifi usal” en la parte izquierda y otro como “Conectarse a la Red”, en la parte derecha.

Figura 52. Comparación de *clusters* y directorio de "lazarillo.usal.es".

Si se analizan el resto de *clusters* y entradas de directorio se encuentran nuevas correspondencias, si bien hay que resaltar que no debe tratarse de una correspondencia exacta, ya que los directorios en la mayoría de los casos se elaboran de forma manual, por lo que pueden contener componentes subjetivos. Cuanto más estructurados sean los directorios y más se ajusten a la información del contenido más se ajustarán a los resultados del *cluster*. Otro ejemplo que se aproxima en gran medida es el del sitio *reina.usal.es*, como podemos apreciar en la figura 53 en la que también se ha superpuesto el directorio real a los resultados obtenidos.

Figura 53. Comparación de *clusters* y directorio de "reina.usal.es".

En este último caso, si ajustamos el número de *clusters* al número de entradas del directorio, obtenemos una correspondencia biunívoca entre ambos elementos. En la figura 54 puede observarse, que tres *clusters* son idénticos a tres entradas y los otros dos *clusters* tienen términos que se reconocen inmediatamente al bajar de nivel las otras dos entradas del directorio.

Figura 54. Comparación de *clusters* y directorio de "reina.usal.es", ajustando los *clusters* a las entradas.

Hay que destacar que lo que se obtiene para cada *cluster* son las características (*features*), que son los términos determinantes que contribuyen en mayor medida a la agrupación de documentos en ese *cluster*. Corresponde al usuario etiquetar estos términos bajo un epígrafe común que exprese el significado semántico de esos términos. Si en este último ejemplo obtenemos los términos “enlaces, conferencias, congresos, enlaces, páginas, interés”, quizá examinando las páginas contenidas en este *cluster* podríamos llegar a la conclusión de que corresponden a la “investigación” del grupo de análisis y las podríamos etiquetar de esta forma, aunque igual de válido habría sido ponerles los calificativos de “trabajos de investigación”, “méritos”, “contribuciones y líneas de trabajo”, etc. En este caso al disponer ya de un directorio, hemos podido examinar sus entradas y ver que realmente al entrar en el apartado de “investigación” aparecen habitualmente los términos que hemos obtenido como resultado del *cluster*.

Podemos concluir que la aplicación clasifica los sitios independientemente de su tamaño, aproximándose más a los directorios convencionales a medida que delimitamos el significado del contenido de los servidores, siendo de gran ayuda en el análisis de información para la administración de portales como paso previo para la elaboración de directorios.

10.3 Obtención de un directorio jerarquizado multinivel mediante el API JAXP

10.3.1 Introducción al proceso de presentación de resultados mediante el API JAXP

Como ya hemos justificado anteriormente, es patente la necesidad de separar la parte de procesamiento de datos, de la parte de presentación, por lo que para el procesamiento y almacenamiento de los datos continuaremos usando XML.

Debido al nuevo requerimiento de generar una estructura jerarquizada navegable, proponemos utilizar el API[6] JAXP[7], con el que se puede implementar mediante la utilización de árboles, estructuras similares a las del manejo de directorios tradicionales, pero en nuestro caso siendo los *clusters* los que harán las veces de directorios.

Este API, en conjunto, permite realizar aplicaciones de análisis, transformación, validación, y consultas de documentos XML, permitiendo la abstracción de la implementación del procesador XML. De esta forma se consigue que la aplicación resultante sea lo más independiente posible, de cara a posibles modificaciones y adaptaciones por futuros desarrolladores.

10.3.2 Justificación del API JAXP

Una vez vista la conveniencia de trabajar con documentos XML y probada su validez para el tratamiento de grandes volúmenes de información, como es el caso que nos ocupa, se imponía utilizar alguna herramienta, en este caso un API, que nos facilitara una biblioteca de clases para el tratamiento de estos documentos para poder analizarlos y presentarlos de forma dinámica, jerárquica y navegable.

[6] *Application Programming Interface*
[7] *Java Api for XML Processing,* https://jaxp.dev.java.net/

Son bastantes los APIs de Java para el tratamiento de documentos XML, pudiéndolos agrupar en dos grandes grupos: orientados a documento y orientados a procedimiento. Dentro de los API orientados a documento tendríamos JAXP, encargado de procesar documentos XML usando diversos analizadores, y JAXB[8], utilizado para mapear elementos XML a clases del lenguaje Java. En los APIs orientados a procedimiento estarían, JAXM[9], utilizado para enviar mensajes SOAP[10] sobre Internet de forma estándar, JAXR[11], para acceder a registros de negocios que comparten información, y JAX-RPC[12], para llamadas a métodos remotos SOAP sobre Internet y recepciones de resultados.

La ventaja de utilizar cualquiera de estos APIs está en que todos ellos soportan los estándares de la industria, lo que garantiza la interoperabilidad. Dependiendo del tipo de aplicación que estemos desarrollando deberemos elegir el que mejor se adapte y nos facilite más funcionalidad, en nuestro caso, tratándose de un claro ejemplo de procesamiento de documentos XML, será JAXP, ya que nos va a permitir transformar nuestro documento XML en una estructura jerárquica mediante un analizador (parser).

10.3.3 Fundamentos del API JAXP

Este API soporta los analizadores estándar SAX (*Simple Api for XML parsing*), DOM (*Document Object Model*) y XSLT (*XML Stylesheet Language Transformations*). La finalidad para la que ha sido desarrollado es la flexibilidad, por lo que permite desde las aplicaciones poder utilizar cualquier analizador compatible XML, mediante una capa de conectividad, que también se encarga de poder gestionar diversos procesadores XSL, para poder variar la presentación de los datos. Incorpora el procesador de XSL Xalan y el analizador Xerces.

El API SAX, actualmente es un estándar reconocido por la industria y definido por el grupo XML-DEV, aunque en su origen era sólo válido para Java, actualmente es

[8] *Java Architecture for XML Binding*, https://jaxb.dev.java.net/
[9] *API for XML Messaging*
[10] *Simple Object Access Protocol.* Protocolo estándar que define cómo dos objetos en diferentes procesos pueden comunicarse por medio de intercambio de datos XML.
[11] *Java API for XML Registries*, http://java.sun.com/webservices/jaxr/index.jsp
[12] *APIs for XML based RPC*, https://jax-rpc.dev.java.net/

soportado por distintos lenguajes de programación. Se trata de una especificación que recoge cómo tienen los analizadores XML que pasar información desde los documentos XML a las aplicaciones de software. Está basado en eventos, por lo que cuando recorre un documento y encuentra una sintaxis de construcción se lo notifica a la aplicación mediante la llamada al correspondiente método, por tanto lo va haciendo paso a paso. Presenta la ventaja de trabajar bien con documentos grandes, pero tiene el inconveniente de no poder manipular información una vez procesada, lo que imposibilita volver a nodos ya procesados o poder modificar los datos.

El API DOM, fue definido por el grupo de trabajo DOM de la W3C. Se trata de un conjunto de interfaces para poder construir una representación de objeto de un documento XML, en forma de árbol jerárquico. Permite una vez construido poder manipular el árbol de objetos que encapsula la aplicación, tanto insertar como eliminar nuevos nodos o hacer distintos recorridos o accesos aleatorios. Es ideal para aplicaciones interactivas. Tiene el inconveniente del consumo de memoria, ya que genera toda la estructura del documento en memoria.

Ya que tanto SAX como DOM permiten distintos analizadores de diferentes fabricantes, cada analizador puede presentar sus peculiaridades por lo que es necesario conocer sus clases y métodos, en definitiva su API, que puede diferir de unos a otros, esto hace que las aplicaciones sean específicas para un analizador determinado, violando los principios de incompatibilidad e interoperabilidad, y es aquí donde entra en juego JAXP, ya que se trata, no de un analizador, sino de un grupo de clases integradas a todo analizador, lo que permite que las aplicaciones desarrolladas con JAXP puedan ser migradas fácilmente.

10.3.4 Proceso de presentación de la vista mediante árboles jerarquizados

Las primeras fases del modelo de clasificación utilizado hasta ahora siguen siendo válidas, por lo tanto se repite el proceso de obtención del documento XML, en sus fases, como se presenta en la figura 42. Sí es necesario seleccionar métodos de

clustering que nos permiten obtener árboles que muestran el anidamiento y las dependencias de unos *clusters* con otros.

Se tienen que descartar los métodos jerárquicos acumulativos puros, que para conjuntos reducidos de elementos, como pueden ser los datos de una consulta, presentan buenos resultados, pero que para grandes volúmenes de información, como es el caso de estudio que se presenta, con más de 100.000 páginas, tiene un consumo de memoria tan elevado que no es posible abordarlo con hardware convencional. Por tanto se recurre a los métodos particionales, con los que se han obtenido mejores resultados, como es el rb (*repeated bisections*) con la función criterio i1, pero añadiendo la opción *–fulltree*, que construye un árbol jerárquico completo que mantiene la solución de *cluster* que fue calculada. En este caso los objetos de cada *cluster* forman un sub-árbol, y los diferentes sub-árboles son mezclados para obtener al final un *cluster* que contenga todos los anteriores. De nuevo los requisitos de memoria hacen que no sea posible obtener solución.

La alternativa es utilizar el método particional rb, con la función criterio i1, pero esta vez recogiendo sólo el camino recorrido en forma de árbol hasta obtener los *clusters* solicitados como hojas finales, mediante el parámetro *–showtree*. De esta forma obtenemos un buen resultado consiguiendo un árbol que nos permite posteriormente poder visualizar cómo están relacionados unos *clusters* con otros, y lo que nos interesa, poder navegar por ellos.

Para optimizar aún más los resultados se prueba con una variante del método particional, que combina los métodos particional y acumulativo. En esencia se trata de un método particional, pero indicando no sólo el número de *clusters* final que se desea obtener, sino un parámetro adicional que es un número de *clusters* mayor, de forma que cuando se obtenga este segundo número de *clusters* mediante métodos particionales, se mezclarán mediante técnicas acumulativas para obtener el valor final deseado, más reducido. Esta técnica persigue obtener *clusters* no aislados con pocos elementos.

Se realizan pruebas generando árboles de diferente tamaño. Con valores pequeños, como por ejemplo 50 *clusters* en las hojas mezclando 200 *clusters*, o incluso 100 hojas mezclando 400 *clusters*, se llevan a cabo en unos segundos en el primer caso y escasos minutos en el segundo. Con valores intermedios, como 200 hojas a partir de

800 *clusters*, se efectúa en 8 minutos. Si nos vamos a valores grandes, 1000 hojas a partir de 4000 *clusters*, opera durante horas, observándose un elevado uso de memoria y de CPU, para al final abortar argumentando escasez de memoria.

Árboles tan grandes en la práctica no se utilizan porque hacen que el usuario tenga que recorrer demasiadas ramas, perdiendo la situación y empleando demasiado tiempo. Así por ejemplo, el Directorio Google, utiliza 14 entradas de primer nivel, dando paso a 4 subniveles y en algunos casos 5. En cada uno de estos subniveles hay una media de 44 nuevas entradas, habiendo grandes variaciones, desde las 12 entradas a casos con 99. El árbol obtenido en la prueba realizada para 50 clusters en las ramas sería el que se recoge en la figura 55.

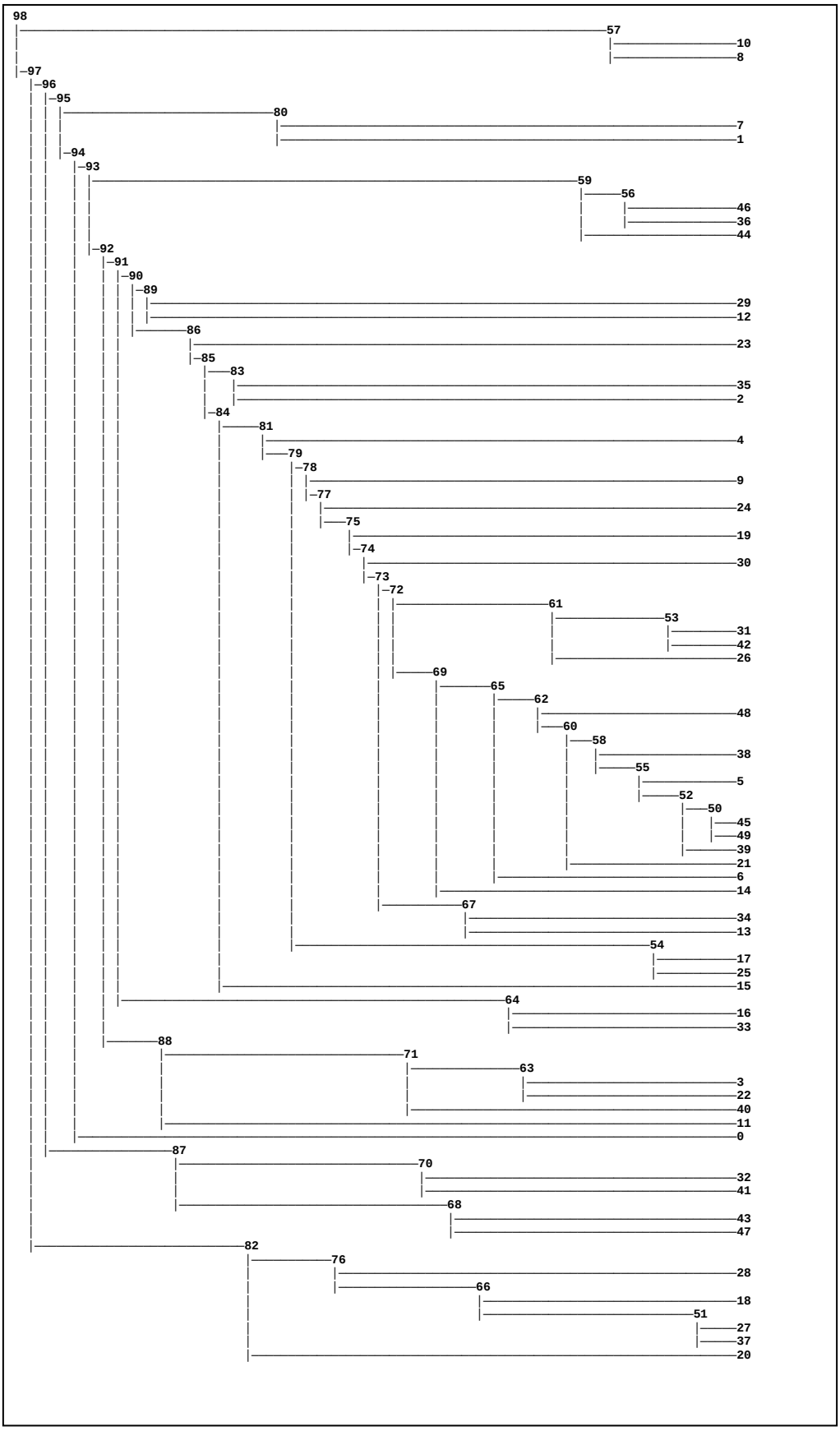

Figura 55. Árbol jerárquico con 50 hojas.

Ha sido necesario desarrollar una nueva aplicación, metercluster2.pl, incluida en el anexo D, para recoger en el documento XML, la relación jerárquica entre los *clusters*. Al tenerse que guardar la dependencia entre unos *clusters* y otros hay que manejar un nuevo fichero que nos proporciona esta información y que se obtiene mediante la utilización de los parámetros proporcionados a CLUTO. El proceso realizado se muestra en la figura 56.

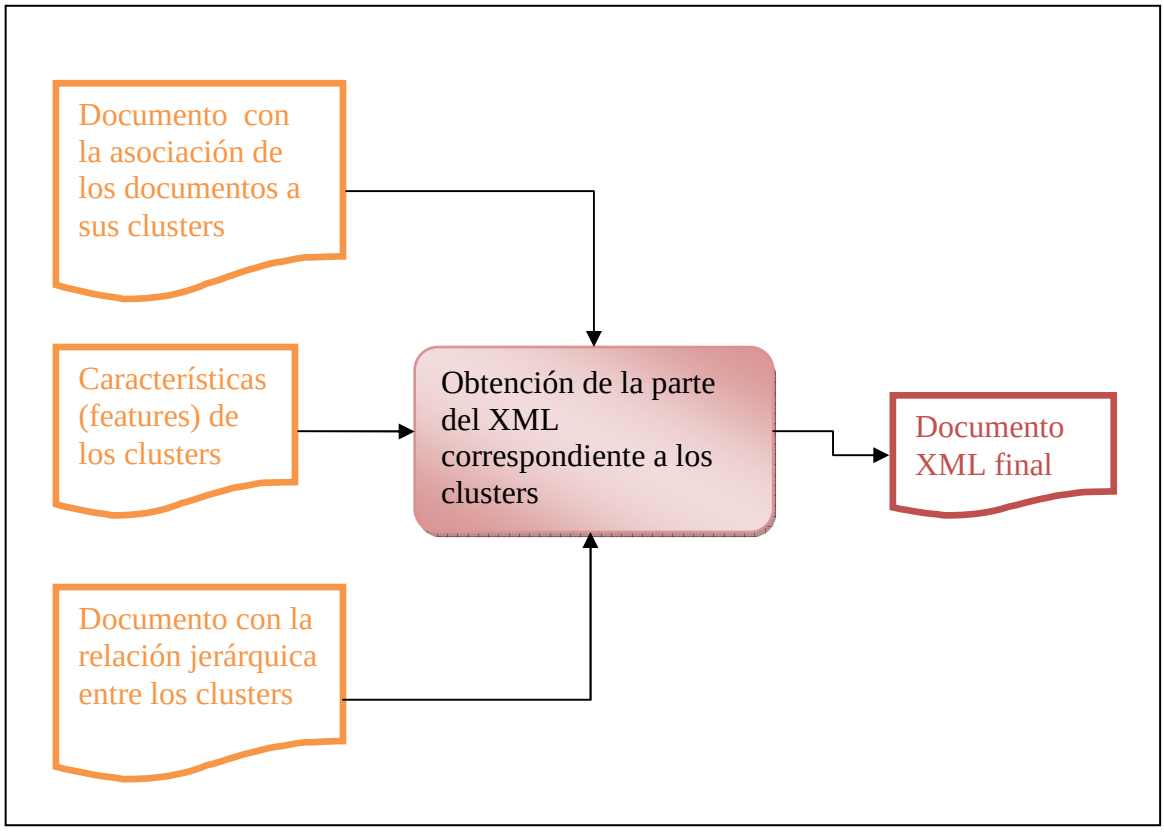

Figura 56. Proceso de obtención del XML asociado con la estructura de árbol de los *clusters*.

Mediante la utilización de estos tres ficheros se pueden obtener los datos necesarios para poder recoger mediante nuevas etiquetas, “padre”, “hijos”, e “hijo”, la relación entre los *clusters*, que formará parte del documento XML final, cuyo aspecto se muestra en la figura 57.

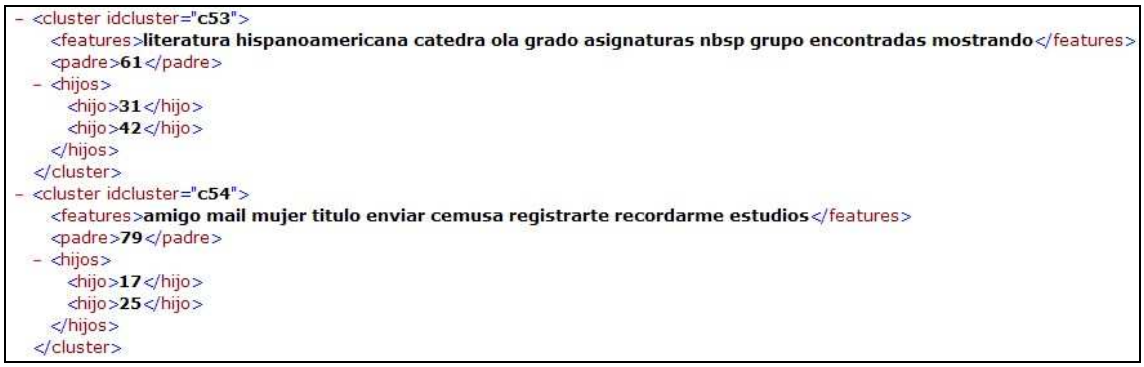

Figura 57. Fragmento de documento XML, con relación jerárquica de *clusters*.

El primero de los documentos, con la asociación de los documentos a sus *clusters*, es un documento en texto plano que tiene tantas líneas como documentos procesados y en cada línea figura el número del *cluster* al que pertenece el documento cuyo número viene dado por el orden de línea que ocupa, como se describe en la figura 20. El segundo documento es un resumen completo de los resultados obtenidos, conteniendo el análisis estadístico de cada *cluster*, sus características (*features*), similares a las recogidas en las figuras 26 a 32 y el árbol gráfico representando la relación de los *clusters*, como el de la figura 55. El tercero de los documentos manejado contiene la relación jerárquica de los *clusters* y sus características ya se han descrito en la figura 21.

Una vez obtenido el documento XML final generado, se analiza para dejarlo limpio de posibles errores sintácticos y se conserva como fuente de datos organizados para el tratamiento con otras aplicaciones. A partir de este documento se busca obtener una estructura jerárquica de directorios navegable que permita moverse de forma más eficiente. Es aquí donde interviene el API DOM, que nos permite cargar todos los datos en memoria y movernos a través de ellos, pero sin olvidar dos aspectos cruciales, la magnitud de los datos que manejamos y su distribución en el documento XML. No olvidemos que en el caso de estudio que estamos manejando hemos organizado cerca de 60000 documentos mediante 99 *clusters*. El problema que se presenta es que el documento XML, aunque contiene la información sobre la jerarquía de los *clusters* no es navegable directamente porque no están organizados en distintos niveles de profundidad.

La disyuntiva que se plantea es cargar el documento en memoria (mediante el API DOM, que a su vez se hace navegable mediante árboles Java) e ir modificándolo a medida que se va navegando por el árbol en tiempo de ejecución, o bien la segunda posibilidad sería transformar previamente el documento XML, en otro nuevo documento XML organizado mediante una estructura de directorios en niveles, para después cargarlo en memoria mediante el API DOM y a su vez también hacerlo navegable mediante árboles Java. Una vez realizadas las pruebas oportunas la primera posibilidad se descarta, porque no es asumible el tiempo de cómputo para el usuario de la aplicación. Se opta por transformar el documento XML en un nuevo documento, pero que ya tenga la estructura de directorios en niveles. Si bien el proceso lleva un tiempo elevado de cálculo, en torno a las 12 horas, se realiza una única vez, para después navegar con eficiencia. Un fragmento de la estructura del nuevo documento XML, se aprecia en la figura 58.

```
                <Documento56711>
                    <Titulo> USAL - Guía de asignaturas de libre elección </Titulo>
                    <URL> http://www.usal.es/~libreleccion/general/alfabetico.php?pagina=14&amp;caracter= <
                </Documento56711>
                <Documento58238>
                    <Titulo/>
                    <URL> http://demos.usal.es/claroline/user/user.php?origin=&amp;column=3&amp;direction=3
                </Documento58238>
            </Documentos>
        </Cluster42>
    </Cluster53>
</Cluster61>
<Cluster69>
    <Features>  browser dokeos doesn page support dsc jpg cise fotograf graduaci discapacidad investigacion
    <Cluster14>
        <Features> browser dokeos doesn page support</Features>
        <Documentos>
            <Documento50220>
                <Titulo> Itinerarios de aprendizaje - Cursos en formato SCORM - Dokeos </Titulo>
                <URL> http://demos.usal.es/claroline/scorm/showinframes.php?openfirst=yes&amp;indexRoute=&a
            </Documento50220>
            <Documento50224>
                <Titulo> Dokeos Documents - 000148143411 - /example_document.html </Titulo>
                <URL> http://demos.usal.es/claroline/document/showinframes.php?cidReq=000148143411&amp;file
            </Documento50224>
            <Documento50247>
```

Figura 58. Estructura del nuevo documento XML, con *clusters* anidados.

Para obtener este nuevo documento se ha implementado una aplicación de transformación en Java, que utiliza el API DOM, tanto para cargar el documento

antiguo, como para generar el nuevo XML, que hace uso además del lenguaje XPath[13], que nos permite construir expresiones (similares a las expresiones regulares) para movernos por el documento cargado en memoria realizando búsquedas. De esta forma podemos localizar, a partir de un *cluster*, sus hijos, que pueden estar en zonas distantes del documento original.

Con este nuevo documento XML ya se puede abordar su navegación. Para ello se ha elaborado un visor en Java, que permite mostrar los *clusters* como si se tratara de una estructura de directorios. Los *clusters* aparecen contraídos y al desplegarlos aparecen sus características descriptivas (*features*) y, o bien los nuevos *clusters* de los que se compone, o bien los documentos que forman parte del *cluster*, según sea el caso. Para cada documento se ha optado por mostrar su título y su URL, que permite recuperar la página. El visor consta de dos paneles, en el de la parte izquierda se lleva a cabo la navegación y en el de la parte derecha se muestra información contextual. Dependiendo de la selección que hagamos, podrá aparecer información de un *cluster* como en la figura 59, o bien de un documento como se puede observar en la figura 60.

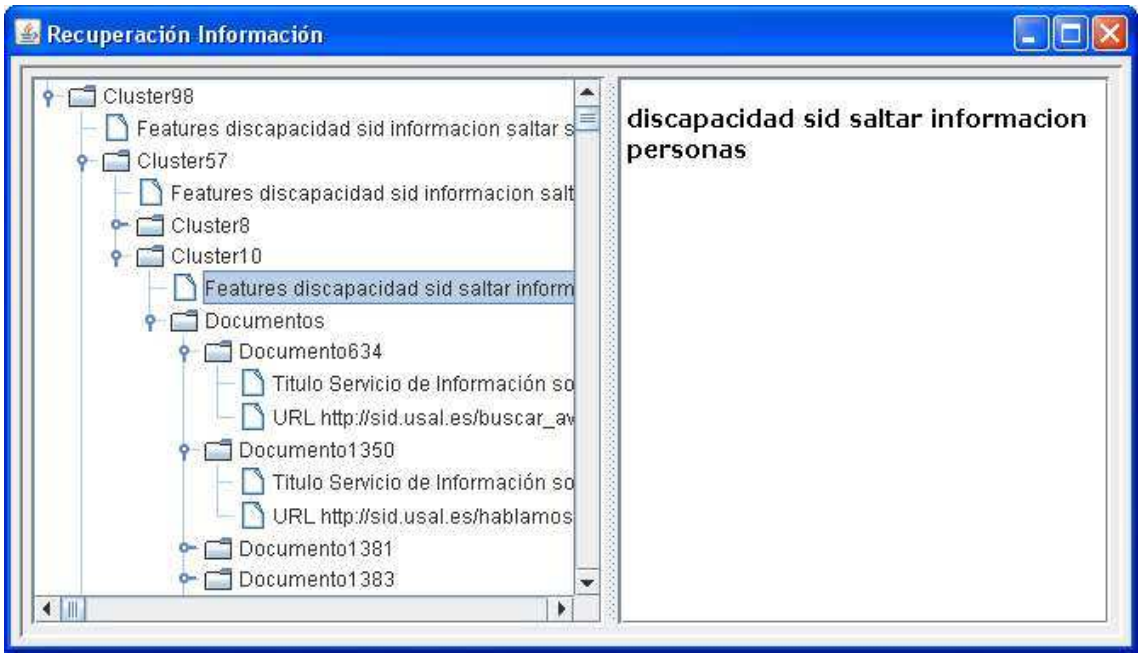

Figura 59. Visor jerárquico mostrando las características de un *cluster*.

[13] *XML Path Language*, http://www.w3.org/TR/xpath

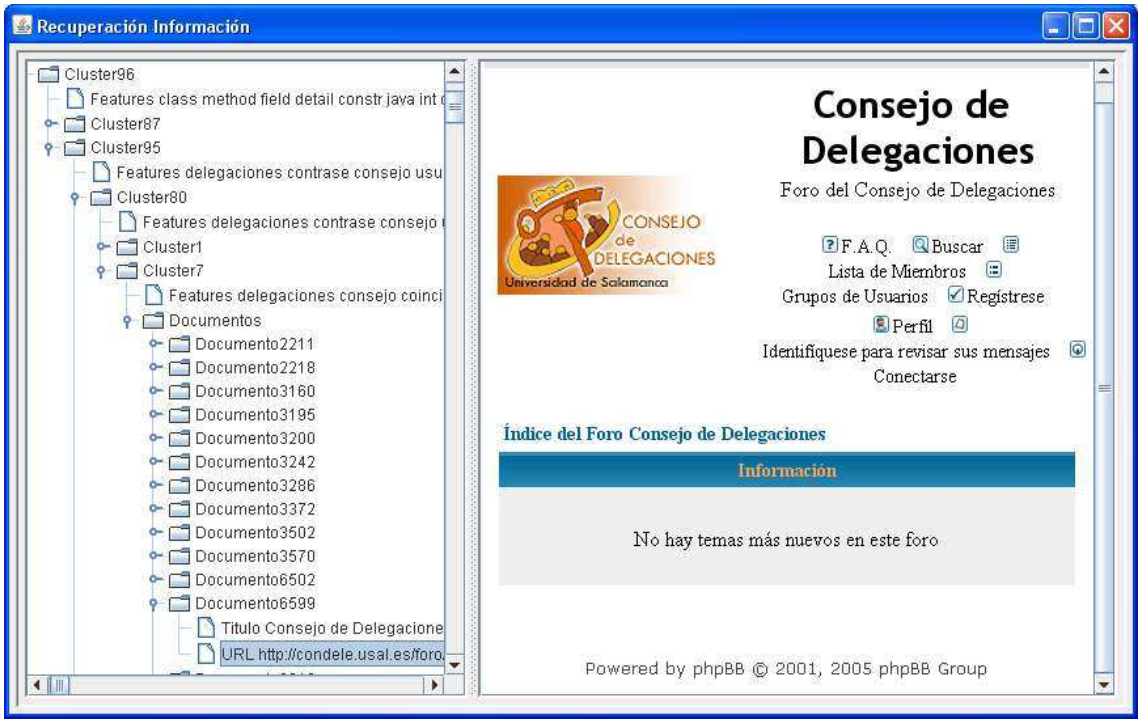

Figura 60. Visor jerárquico, mostrando el contenido de un documento.

El hecho de disponer de documentos resumen en XML, conteniendo toda la información de un sitio Web, ofrece grandes posibilidades ya que permite sin tener que recorrer de nuevo, ni la Web, ni los documentos almacenados, poder alterar la estructura de presentación, sin más que transformar la estructura interna del documento XML, adaptándola a diferentes formatos, presentes o futuros, como podría ser el protocolo Sitemap[14] de Google, utilizado cada vez más como ayuda al rastreo de sitios en distintos buscadores, como Google, Yahoo, MSN o Ask. Una de las ventajas que presenta los documentos XML que hemos generado mediante las técnicas de *clustering*, frente a los que se pueden obtener con herramientas automatizadas para la obtención de mapas de sitio, es la distribución jerarquizada pero atendiendo a la semántica de los documentos, lo que hace que se puedan construir directorios navegables sin supervisión, solventando uno de los principales escollos con los que se encuentran los webmasters y de esta forma servirles de herramienta para el complicado trabajo de elaborar el directorio de un sitio, sobre todo cuando se trata de sitios con gran volumen de información.

[14] Mapas de sitio de Google, http://www.sitemaps.org/es/protocol.php

La elaboración de directorios jerárquicos (*Open Directory*) ha sido una alternativa a las búsquedas tradicionales y un recurso imprescindible para grandes portales. Los buscadores más representativos incorporan este tipo de directorio como alternativa o valor añadido a las búsquedas, es el caso de Google, Yahoo o MSN. En otros casos consideran que los directorios son la clave para construir sistemas de exploración efectivos, cuando son capaces de agrupar los datos y mostrar sus relaciones, sería el caso Dmoz[15], que apuesta por la navegación directa partiendo de un directorio. Esta navegación directa tiene mucha importancia cuando los recursos que estamos buscando son vagos e imprecisos, esto es habitual por ejemplo a la hora de buscar una fotografía, un *blog* o un *post*, en estos casos se recurre al etiquetado (*tagging*), habitual ya en algunos sitios de gran volumen de contenidos como Flickr[16], Delicious[17] o Technorati[18]. De nuevo un problema que surge en los directorios de etiquetas es la cantidad de ellas que pueden aparecer, en algunas casos se recurre a agruparlas alfabéticamente y en otros casos se va más allá y se intentan agrupar por contenidos relacionados y es aquí de nuevo donde el *clustering* puede utilizarse para mejorar los resultados [Begelman et al., 2006] [Brooks y Montanez, 2006], aplicando *clustering* a la clasificación de etiquetas y a la obtención de directorios jerarquizados que permitan la navegación por ellas, surgiendo nuevos algoritmos específicos [Shepitsen et al., 2008] [Ramage et al., 2009].

[15] http://dmoz.org
[16] http://www.flickr.com
[17] http://delicious.com/
[18] http://technorati.com/tag/

11 CONCLUSIONES Y TRABAJO FUTURO

11.1 Conclusiones

La *Recuperación de la Información* ha ido evolucionando, manteniéndose a lo largo de los años como una necesidad para el individuo, que ha ido cambiando a medida que se desarrollaban nuevas tecnologías. Al igual que los avances técnicos han propiciado la evolución de la informática, la cada vez mayor cantidad de información disponible hace que sea necesario investigar en nuevos mecanismos que permitan el tratamiento de grandes volúmenes de información, con técnicas más sofisticadas, de forma que no sólo se recupere información para el usuario sino que además esa información sea lo más relevante posible.

Cuando esta búsqueda de información se hace mediante Internet el problema se agrava porque se amplía exponencialmente el campo de búsqueda, con el consiguiente aumento de la cantidad de valores recuperados, pero también porque se utilizan herramientas de búsqueda que tratan de satisfacer las necesidades de sectores muy amplios, que en ocasiones son manejados de forma muy imprecisa o incluso vaga, por lo que es necesario buscar nuevas formulas que den respuestas eficaces a los usuarios. Las dos líneas de investigación más claras son las consultas Web y la clasificación mediante directorios en la que nos hemos centrado.

Dada la naturaleza de la información que íbamos a procesar, páginas web, ha sido necesario adaptar el proceso de recuperación, que habitualmente se realiza sobre colecciones de documentos bien conocidas, llevando a cabo un análisis documental de las diferentes páginas que nos podemos encontrar en los servidores web.

Analizando la evolución histórica de la recuperación de la información han aparecido muchos elementos con estructura jerárquica en forma de árbol, que han sido ampliamente aceptados, como las tablas de contenidos, los índices, los sistemas de catalogación de las bibliotecas, y otros relacionados con el mundo de la informática también muy utilizados como los directorios, la organización en carpetas, o los menús, por lo que hemos dirigido nuestros esfuerzos a los modelos de navegación, que son los

que el usuario utiliza cuando desea investigar sobre algún tema, o busca referencias, centrándonos en estructuras dirigidas, como los directorios abiertos, agrupando documentos relacionados mediante tópicos.

Debido a la gran profusión de modelos de Recuperación de la Información ha sido necesario establecer una clasificación para dar una visión de conjunto de todos ellos, para después dar a conocer de forma resumida las principales características de cada uno de ellos.

Si bien se han documentado las fases del preprocesado de documentos tradicional, ha habido que adaptarlas debido a la naturaleza de los documentos web, eliminando alguna fase e incorporando otras nuevas. También ha sido determinante el formato soportado por la herramienta del entorno experimental CLUTO. Todo ello ha dado lugar a que se desarrolle una metodología de clasificación, recogida en la figura 16, que nos permitiera definir un proceso de desarrollo con sus etapas, con sus hitos y con sus resultados intermedios. Se partía de distintas experiencias, de lo más variadas con las técnicas de *clustering*, aplicadas a campos muy diversos, pero nos centramos en todas aquellas relacionadas con la Recuperación de la Información, para darle un enfoque diferente, buscábamos su aplicación como herramienta para el procesamiento de información en portales Web. Para algunas de las fases previas al procesamiento con CLUTO, ha sido necesario desarrollar software específico a medida para automatizar los procesos de filtrado y normalización de documentos, y concatenación de documentos.

En la fase de *filtrado y normalización* el escollo que ha habido que solventar ha venido dado por la naturaleza web de las páginas, que presenta graves inconvenientes con respecto al tratamiento habitual de documentos por las múltiples posibilidades de formato y la abundancia de texto relacionada con la presentación visual. Esto ha hecho mucho más dificultosa la extracción de los datos relevantes de cada documento, ya que además del tratamiento habitual de cualquier texto en esta fase, como normalizar los caracteres dependientes de las fuentes, codificar los acentos, o hacer el tratamiento de los espacios en blanco, ha sido necesario eliminar las etiquetas HTML y cualquier rastro de código de lenguaje de programación ajeno a la semántica del documento, como

scripts en distintos lenguajes de programación, o sus comentarios y documentación asociada.

En la fase de *eliminación de palabras vacías del castellano* se ha comprobado que es necesario elaborar una lista de palabras vacías adecuada para obtener unos *cluster* de calidad. Se ha demostrado que es imprescindible en primera instancia incluir las palabras vacía del castellano, en gran medida monosílabos. Como segundo paso mejora los resultados eliminar términos propios del sitio web que se está analizando, como su nombre, slogan o palabras características del sitio. Dedicar un tiempo a revisar los resultados de los clusters obtenidos, compensa, porque fácilmente se puede detectar algún término muy abundante, que puede dar origen a que se genere un cluster, y sin embargo ese término no aporte significado al *cluster*, con lo que fácilmente se podría incluir en la lista de palabras vacías. En cuanto a la lematización presenta la ventaja de mejorar la clasificación pero hace que las características descriptivas que se presentan al usuario estén en algunos casos sesgadas. Por lo tanto si se van a presentar directamente sería mejor no utilizar lematización, mientras que si se van a añadir títulos más significativos para la navegación conviene utilizarla.

En cuanto a los tiempos de computación de las fases de filtrado, normalización y concatenación, se puede afirmar que son tiempos perfectamente aceptables. Son proporcionales al número de páginas web. En el caso de portales pequeños son del orden de varios minutos y en el de grandes sitios web en torno a 3 horas, tiempo habitual para una tarea de mantenimiento de portales o administración de sitios.

En cuanto a la incidencia de los métodos de *clustering* y habiendo realizado los experimentos con cada uno de los métodos de clustering soportados por CLUTO, se ha constado que los métodos de *clustering* particional han ofrecido mejores resultados que los acumulativos, coincidiendo con trabajos recientes en los que se indicaban adecuados para grandes conjuntos de documentos. Se han mostrado más eficientes, coincidiendo con los estudios de [Zhao y Karypis, 2002] [Zhao et al., 2005], aunque ha habido muchos factores distintos entre su caso de estudio y el nuestro: el tipo de documentos era muy distinto, en nuestros caso páginas web, la medida de la calidad de los *clusters* era diferente porque ellos partían de documentos clasificados, mientras que nosotros deseábamos trabajar de forma no supervisada.

Dentro de los métodos particionales los que mejor comportamiento han presentado con respecto a la similitud interna son *rb (repeat bisections)* y *rbr (repeat bisections refinement),* estando por delante siempre *rb* para números de *clusters* en torno a 10 y mejorando *rbr* para valores próximos a 20. En cuanto a los métodos acumulativos el método destacado ha sido *agglo (agglomerative).* En conjunto las mejores opciones son *rb* o *rbr,* dependiendo del número de *clusters*, y en segundo término *agglo,* siempre y cuando, cualquiera de ellos se combinen con las funciones criterio adecuadas.

El análisis de la incidencia de las funciones criterio en los resultados de las pruebas realizadas demuestra que las funciones criterio destacadas, tanto para métodos particionales como acumulativos, son *i1*, *h1* e *i2*, en este orden, destacando el elevado tiempo computacional de *h1* con métodos acumulativos, lo que puede condicionar su utilización.

En lo referente a tiempos de computación, medida considerada determinante en cuanto a rendimiento, se aprecia que el tiempo computacional de los métodos particionales tiene complejidad $O(n \log n)$, frente a la complejidad $O(n^2 \log n)$ de los métodos acumulativos, lo que hace que todos los particionales sean más rápidos que los acumulativos, con valores entre 4 y 8 veces menores. Dentro de los particionales son más rápidos *rb* y *rbr* que *direct*, casi el doble. Un caso especial son las funciones *h1* y *h2* para acumulativos que tienen una complejidad $O(n^3)$, lo que las hace extremadamente lentas. Aumentar el número de *clusters,* en los métodos particionales hace que aumente el tiempo de computación, casi de forma proporcional, mientras que en los métodos acumulativos la tendencia es distinta, ya que las variaciones son mínimas.

Otro factor muy a tener en cuenta es el tamaño de la matriz de entrada ya que para valores muy grandes del número de objetos a clasificar, los métodos acumulativos se hacen inviables por su consumo de memoria física, sería el problema del caso de estudio B, que cuenta con más de 100.000 elementos.

Atendiendo a las características más descriptivas que presentan los tres métodos de mejor comportamiento y las tres funciones criterio más destacadas, vuelven a estar delante los métodos particionales, en cuanto a las funciones destacan *i1* y *h1* y en

combinación la mejor posibilidad es el método *rbr* con la función *h1* porque es el conjunto que más *clusters* de tamaño equilibrado presenta y el que menor número de documentos deja en el *cluster* más amplio.

Al aumentar el número de *cluster* mejora el método *agglo*, pero sin superar a los particionales. La obtención de *clusters* equilibrados también crece de forma proporcional al aumento de nuevos *clusters*, dividiéndose de forma prioritaria aquellos *clusters* que ya presentaban *sub-clusters*.

Una vez establecido el modelo de clasificación y conocidos los mejores resultados en cuanto a métodos y funciones, aplicadas a la clasificación de grandes sistemas de información Web, se han diseñado aplicaciones informáticas que nos han permitido probar la validez práctica del modelo para generar directorios Web de forma automatizada, no supervisada, es decir evitando el proceso habitual, laborioso y con gran coste de recursos humanos, habitual para la clasificación y obtención de este tipo de directorios mediante expertos en la materia correspondiente al sitio. Se han probado con éxito, tanto en la elaboración de directorios de un solo nivel como de directorios jerarquizados multinivel, en ambos casos mediante la generación y utilización de documentos XML, que presentan gran versatilidad para su adaptación a nuevos formatos y herramientas. De esta forma hemos garantizado que los resultados pueden ser abordados desde diferentes interfaces, haciéndolos independientes de la presentación que se pueda hacer de ellos.

En el caso de los directorios web de un solo nivel se ha probado con éxito el acceso mediante hojas de estilo XSLT, adaptándose al esquema MVC. Una vez obtenido el documento XML con el proceso descrito en la figura 42, se ha validado mediante dos técnicas distintas, la utilización de un DTD y la utilización de un *schema*, para ejemplificar el uso y desarrollo de cada uno de estas técnicas. En tiempo de ejecución, la primera parte de presentación de los *clusters* es casi inmediata, aun en el caso de grandes sitios, no así la segunda parte consistente en mostrar las entradas correspondientes a un *clúster*, en este caso se ve afectada por el número de documentos finales procesados. Para portales homogéneos, centrados en un servidor, con contenidos estructurados, la clasificación que se obtiene es totalmente aceptable, se han obtenido entradas muy similares a la distribución en menús que venían presentando. Para grandes

sitios, con contenidos dispares repartidos en distintos servidores, es más eficiente hacer una primera división de los servidores y después aplicar las técnicas de *clustering* por separado a cada uno de los servidores, con lo que estaríamos en el caso anterior con muy buenos resultados.

En el caso de los directorios web jerarquizados multinivel, hemos utilizado diferentes API's como JAXP, DOM, y *Xpath*, para garantizar la utilización de estándares y la interoperabilidad. Debido a la necesidad de jerarquización de los *clusters* y buscando la necesidad de soportar grandes volúmenes de información, los resultados de las pruebas nos hacen descartar los métodos de clustering acumulativo por su excesivos requerimientos de hardware, obteniendo buenos resultados con los particionales, pero para unas combinaciones de parámetros muy precisas, como ya se han indicado. Se ha constatado que es necesario en estos casos obtener un documento XML final que ya posea estructura jerárquica de *clusters*, pudiéndolo elaborar desde las primeras fases o bien obtenerlo mediante un proceso de transformación, como hemos hecho, mediante la utilización del API DOM, que nos permite modificar en memoria un XML cargado para transformarlo en otro, y mediante *Xpath* que nos permite localizar rápidamente elementos en la estructura. Partiendo de este XML jerarquizado, la navegación está asegurada, porque son muchas las estructuradas de manejo de árboles desarrolladas en diversos lenguajes que permiten su generación a partir de ficheros XML, como es el caso de JAVA, lenguaje en el que hemos desarrollado el visor para la navegabilidad real por los directorios web obtenidos.

En ambos casos, tanto en los directorios de un nivel, como en los multinivel, las herramientas aportadas suponen una gran ayuda para los administradores de sitios web, para la detección de anomalías en la distribución de las páginas, como páginas repetidas, páginas sin contenido o páginas con contenidos en distintos idiomas, pero sobre todo para el trabajo de elaboración de directorios de sitios web.

11.2 Trabajo futuro

Un punto de vista diferente que se podría abordar es el tratamiento gráfico que se puede hacer de las técnicas de *cluster*. En este caso se trabajaría con matrices de adyacencia del gráfico de similitud entre los objetos que se quieren clasificar. Se dispondría de un nuevo método de *clustering* denominado *graph*, que maneja gráficos de proximidad, en los que cada objeto se convierte en un vértice y cada vértice se conecta con los que son más similares, para pasar a dividir el gráfico mediante algoritmos del tipo *min-cut*.

Otro aspecto interesante puede ser el desarrollo de herramientas de gestión para webmasters, que haciendo uso de las técnicas de *clustering* descritas y analizadas, aglutinen todas las fases descritas en el modelo de clasificación, pero de forma parametrizada, similar a un IDE, con sus correspondientes menús y ventanas, de forma que pueda ser configurable el directorio raíz donde se encuentre recogidas las páginas web, las múltiples posibilidades de configuración del proceso de *clustering*, el tipo de directorio web que se quiere generar y el formato de salida que permita su publicación y navegabilidad en un sitio web.

Nuevas vías de aplicación de las técnicas de *clustering* vendrían dadas por la aparición y desarrollo masivo de nuevos formatos y tendencias en el desarrollo de Internet. Así podría ser muy útil para resolver algunos de los problemas emergentes en la red, como las citaciones de documentos científicos correspondientes a un autor, consecuencia de la posibilidad de repetición de nombres. Problema muy relacionado también con la repetición en las redes sociales de perfiles de igual nombre pero correspondientes a diferentes personas. Podría ser aplicado también en la clasificación de blogs por contenidos temáticos, campo en el que ya se ha indicado que se está estudiando la clasificación de las etiquetas asociadas a los blogs. Tema del etiquetado interesante ya que abre otras vías de clasificación relacionadas con las imágenes o con la localización de vídeos.

REFERENCIAS

[Aggarwal y Yu, 2008] Aggarwal, C. C. y Yu, P. S. A Framework for Clustering Uncertain Data Streams. En *IEEE 24th International Conference on Data Engineering*, Cancún, México, 150-159, abril de 2008.

[Andersen et al., 2007] Andersen, R.; Chung, F. y Lang, K. Local partitioning for directed graphs using PageRank. En *5th Workshop On Algorithms And Models For The Web-Graph (WAW 2007),* San Diego, CA, USA, 2007.

[Arocena y Mendelzon, 1998] Arocena, G. y Mendelzon, A. WebOQL: Restructuring documents, databases and Webs. En *Int. Conf. on Data Engineering*, Orlando, Florida, USA, 24-33, 1998.

[Artiles et al., 2009] Artiles, J.; Gonzalo, J. y Sekine, S. Weps 2 evaluation campaign: overview of the web people search clustering task. En *2nd Web People Search Evaluation Workshop (WePS 2009), 18th WWW Conference*, Madrid, España, abril de 2009.

[Avrachenkov et al., 2008] Avrachenkov, K.; Dobrynin, V. ; Nemirovsky, D. ; Pham, S. K. y Smirnova, E. PageRank Based Clustering of Hypertext Document Collections. En *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, Singapore, 873-874, 2008.

[Baeza-Yates, 2002] Baeza-Yates, R. The Web of Spain. En *Upgrade*, Vol. III, nº 3, 82-84, junio de 2002.

[Baeza-Yates et al., 2005] Baeza-Yates, R.; Castillo, C. y López, V. Características de la Web de España. En *El profesional de la información*, 15:6-17, junio de 2005

[Baeza-Yates y Navarro, 1996] Baeza-Yates, R. y Navarro, G. Integrating contents and structure in text retrieval. *ACM SIGMOD Record*, 25(1):67-79, marzo de 1996.

[Baeza-Yates y Ribeiro-Neto, 1999] Baeza-Yates, R. y Ribeiro-Neto, B. *Modern Information Retrieval*, ACM Press Series/Addison Wesley, NY, USA, 1999.

[Beeferman y Berger, 2000] Beeferman, D. y Berger, A. Agglomerative clustering of a search engine query log. *Proceedings of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 407-416, *2000.*

[Begelman et al., 2006] Begelman, G.; Keller, P. y Smajda, F. Automated Tag Clustering: Improving search and exploration in the tag space. En *Collaborative Web Tagging Workshop at WWW2006, Edinburgh, Scotland, Citeseer, 2006.*

[Belkin y Croft, 1987] Belkin, N. J. y Croft, B. W. Retrieval Techniques. *ARIST, Annual Review of Information Science and Technology.* Vol. 22, 109-145, 1987.

[Berrocal et al., 2000] Berrocal, José L. A.; García-Figuerola, Carlos y Zazo, Ángel. SGML/XML: Desarrollo en entornos documentales. XVColoquio de la AIB y I Coloquio de la AEB. "Las nuevas formas de la comunicación científica". Universidad de Salamanca. España. 9 al 11 de Mayo de 2000. Disponible en http://reina.usal.es/pub/alonso2000sgml.pdf.

[Boley, 1998] Boley, D. Principal direction divisive partitioning. *Data Mining and Knowledge Discovery*, 2(4), 1998.

[Booth et al., 2008] Booth, J. G.; Casella, G.; y Hobert, J. P. Clustering Using Objective Functions and Stochastic Search. En *Journal of the Royal Statistical Society,* Vol. 70, Nº 1, 119-139, 2008.

[**Bower y McGlashan, 2000] Bower, Andy y McGlashan, Blair.** TWISTING THE TRIAD. The evolution of the Dolphin Smalltalk MVP application framework. [on line] <http://www.object-arts.com/papers/TwistingTheTriad.PDF> [Consulta: 02/09/2008].

[Bowman et al., 1994] Bowman, C. Mic; Danzing, Peter B.; Hardy, Darren R.; Manber, Udi y Schwartz, Michael F. The Harvest information discovery and access system. En *Proc. 2nd Int. WWW Conf.*, 763-771, octubre de 1994.

[Bradley y Fayyad, 1998] Bradley, P. S. y Fayyad, U. M. Refining initial points for k-means clustering. En *J. Shavlik, editor, Proceedings of the fifteenth /international Conference on Machine Learning (ICML '98)*, 91-98, San Francisco, CA, 1988.

[Brin y Page, 1998] Brin, S. y Page, L. The anatomy of a large-scale hypertextual Web search engine. En *Proc. of the 7th Int. WWW Conference*, Brisbane, Australia, abril de 1998.

[Brook y Montanez, 2006] Brook, C. H. y Montanez, N. Improved annotation of the blogosphere via autotagging and hierarchical clustering. En *International World*

Wide Web Conference. Proceedings of the 15th international conference on World Wide Web table of contents. Edinburgh, Scotland, 625-632, 2006.

[Burbeck, 1992] Burbeck, Steve. Applications Programming in Smalltalk-80(TM): How to use Model-View-Controller (MVC) [on line] <http://www.cs.uiuc.edu/users/smarch/st-docs/mvc.html> [Consulta: 02/09/2008].

[Burke, 2002] Burke, Eric M. Java y XSLT. O'Reilly. 2002.

[Burkowski, 1992] Burkowski, F. An algebra for hierarchically organized text-dominated databases. *Information Processing & Management*, 28(3):333-348, 1992.

[Burkowski, 1992] Burkowski, F. Retrieval activities in a database consisting of heterogeneous collections of structured text. En *Proc. of the 15th Annual International ACM/SIGIR Conference on Research and Development in Information Retrieval*, 112-125, Copenhagen, Dinamarca, 1992.

[Canals, 1990] Canals, I. Introducción al hipertexto como herramienta general de información: concepto, sistemas y problemática. En, Revista española de documentación científica, v. 13, n. 2, abril junio 1990.

[Caridad y Moscoso, 1991] Caridad, M; Moscoso, P. *Los sistemas hipertexto e hipermedios: una nueva aplicación en informática documental.* Fundación Germán Sánchez Ruipérez; Pirámide. Madrid. 1991.

[Carriere y Kazman, 1997] Carriere, J. y Kazman, R. WebQuery: Searching and visualizing the Web through connectivity. En *6th Int. WWW Conf.*, Santa Clara, CA, USA, abril de 1997.

[Chakrabarti, 2003] Chakrabarti, S. *Mining the Web. Discovering Knowledge from Hypertext Data*. Morgan Kaufmann Publishers. San Francisco. USA. 81-83, 2003.

[Chen, 1995] Chen, H. Machine Learning for Information Retrieval: Neural Networks, Symbolic Learning, and Genetic Algorithms. En *Journal of the American Society for Information Science*, 46 (3), 194-216, 1995.

[Chaumier, 1979] Chaumier, J. *Techniques documentaries*. París, PUF, 11-13, 1979.

[Cleverdon et al., 1966], Cleverdon, C.; Mills, J. y Keen, E. Factors determining the performance of indexing systems. Design. Vol. I. En *ASLIB Cranfield Research Project.* Cranfield, England. 1966.

[Cleverdon y Keen, 1966], Cleverdon, C. y Keen, E. Factors determining the performance of indexing systems. Test results. Vol. II. En *ASLIB Cranfield Research Project.* Cranfield, England. 1966.

[Cohn et al., 2008] Cohn, D. ; Caruana, R. y McCallum, A. Semi-supervised clustering with user feedback. En *Constrained Clustering: Advances in Algorithms, Theory, and Applications.* Chapman & Hall / CRC, 2008.

[Cooley et al., 1999] Cooley, R. ; Mobasher, B. y Srivastava, J. Data preparation for mining World Wide Web browsing patterns. En *Journal of Knowledge and Information Systems*, vol. 1, nº 1, 1999.

[Courrier, 1976] Courrier, Y. Analyse et languages documentaires, documentaliste. Vol. 13, nº 5-6, 1, 1976.

[Croch, 1990] Croch, C. J. An approach to the automatic construction of global thesaurui. *Information processing and management*, 26(5), 629-640, 1990.

[Croch y Yong, 1992] Croch, C. J. y Yong, B. Experiments in Automatic Statistical Thesaurus Construction. SIGIR'92, 77-87, 1992.

[Cuadra-Elsevier, 1990] Cuadra-Elsevier. *Directory Of Portable Databases: Number 2, December, 1990 (directory Of Portable Databases, Vol. 2),* 1990.

[Cunha, 1987] Cunha, I. F. Análise documentária. En *Análise documentária: a análise de sintese*, Brasilia: IBICT, 38-40, 1987.

[Cutting et al., 1992] Cutting, D. R.; Karger, D. R.; Pedersen, J. O. y Tukey, J. W. Scatter/Gather: A cluster-based approach to browsing large document collections. En *Annual International Conference on Research and Development in Information Retrieval (SIGIR)*, Copenhagen, Denmark, 1992.

[De Bra y Post, 1994] De Bra, P. M. E. y Post, R. D. J. Searching for arbitrary information in the WWW: The fish search for Mosaic. En *Proc. Of the 2nd Int. WWW Conference*, Chicago, USA, octubre de 1997. http://archive.ncsa.uiuc.edu/SDG/IT94/Proceedings/Searching/debra/article.html

[de la Rosa, 2003]. De la Rosa, Antonio. Introducción a XML para Documentalistas [on line]. "Hipertext.net", núm. 1, 2003. <http://www.hipertext.net> [Consulta: 29/08/2008]. ISSN 1695-5498.

[Deerwester et al., 1990] Deerwester, S. C.; Dumais, S. T; Landauer, T. K.; Furnas, G. W. y Harshman, R. A. Indexing by Latent Semantic Analysis. *Journal of the American Society of Information Science*, 41 (6), 391-407, 1990.

[Díaz et al., 1996] Díaz, P.; Catenazzi, N.; Aedo, I. *De la multimedia a la hipermedia*, Ra-ma, Madrid, 1996.

[Eisen et al., 1998] Eisen, M.B.; Spellman, P. T.; Brown, P. O. y Botstein, D. Cluster analysis and display of genome-wide expression patterns. En *Proceedings of the National Academy of Sciences of the United States of America*. Vol. 95, nº 25, 14863-14868, diciembre de 1998.

[Ellis, 1990] Ellis, D. *New horizons in Information Retrieval*. Library Association, London, 1990.

[Erman et al., 2006] Erman, J.; Arlitt, M. y Mahanti, A. Traffic classification using clustering algorithms. En *Joint International Conference on Measurement and Modeling of Computer Systems. Proceedings of the 2006 SIGCOMM workshop on Mining network data*, Pisa, Italia, 281-286, 2006.

[Ertöz et al., 2004] Ertöz, L.; Steinbach, M. y Kumar, V. Finding Topics in Collections of Documents: A Shared Nearest Neighbour Approach. En *Information Retrieval and Clustering.* W. Wu, H. Xiong and S. Shekhar (Eds.), 83-104, 2004.

[Fernández et al., 1997] Fernández, Mary; Florescu, Daniela; Levy, Alon y Suciu, Dan. A query language for a Web-site management system. *SIGMOD Record*, 26(3):4-11, septiembre de 1997.

[Ferragina y Gulli, 2005] Ferragina, P. y Gulli, A. A personalized search engine based on web-snippet hierarchical clustering. En *Special interest Tracks and Posters of the 14th international Conference on World Wide Web* (Chiba, Japan, 10 – 14 de mayo, 2005). WWW '05. ACM, New York, NY, 801-810. 2005.

[FID, 1958] Federación Internacional de Documentación. Documentation terminology. *Revue de Documentation*. Vol. 25, nº 2, 38-39, 1958.

[Figuerola et al., 2002] Figuerola, C. G.; Gómez, R.; Rodríguez, A. F. Z. y Berrocal, J. L. A. Spanish Monolingual Track: The Impact of Stemming on Retrieval. En Peters, C., Braschler, M., Gonzalo, J., and Kluck, M. editors. *Evaluation of Cross-Language Information Retrieval Systems, CLEF 2001*. Vol. 2406 of LNCS, 253-261, 2002.

[Foskett, 1997] Foskett, D. J. Thesaurus. En K. Sparck Jones and P. Willet, editors, *Readings in Information Retrieval*, 111-134. Morgan Kaufmann Publishers, Inc., 1997.

[Fox, 1983] Fox, E. A. *Extending the Boolean and Vector Space Models of Information Retrieval with P-Norm Queries and Multiple Concept Types.* PhD thesis, Cornell University, Ithaca, N. Y., http://www.ncstrl.org, 1983.

[Fox, 1992] Fox, C. Lexical analysis and stoplists. En W. Frakes and R. Baeza-Yates, editors, *Information Retrieval: Data Structures & Algorithms*, 102-130. Prentice Hall, Englewood Cliffs, NJ, USA, 1992.

[Frakes y Baeza-Yates, 1992] Frakes, W. B. y Baez-Yates, R. *Information Retrieval: Data Structures & Algorithms*, Prentice Hall, Englewood Cliffs, NJ, USA, 1992.

[Frey y Dueck, 2007] Frey, B. J. y Dueck, D. Clustering by Passing Messages Between Data Points. En *Science*, Vol. 315, Nº 5814, 972, 2007.

[Furnas et al., 1987] Furnas, G. W.; Landauer, T. K.; Gómez, L. M. y Dumais, S. T. The vocabulary problem in human-system communication. *Communications of the ACM.* 30(11):964-971, november 1987.

[Furnas et al., 1988] Furnas, G. W.; Deerwester, S. T.; Dumais, S. T.; Landauer, T. K.; Harshman, R. A. ; Streeter, L. A. y Lochbaum, K. E. Information retrieval using a singular value decomposition model of latent semantic structure. En *Proc. of the 11th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval.* 465-480, 1988.

[Ganti et al., 2006] Ganti, V.; Ramakrishnan, R.; Gehrke, J. ; Powell, A. y French, J. Clustering Large Datasets in Arbitrary Metric Spaces. *Virginia Univ Charlottesville Dept of Computer Science*, 2006.

[García Gutiérrez, 1984] García Gutiérrez, A. L. *Lingüística documental*, Madrid: Mitre, 79, 1984.

[Gardin et al., 1964] Gardin, J. C. ; Grolier, E. ; Levery, F. *L'organisation de la documentation scientifique*. París: Gauthier Villars, 12, 1964.

[Gimeno Perelló, 1995] Gimeno Perelló, J. Sistemas de indización aplicados en bibliotecas: Clasificaciones, tesauros y encabezamientos de materias. En Magán Walls, J. A. coordinador. *Tratado básico de Biblioteconomía.* Madrid: editorial complutense, 2ª edic., 201-231, 1995.

[Gionis et al., 2007] Gionis, A.; Mannila, H. y Tsaparas, P. Clustering Aggregation. En *ACM Transactions on Knowledge Discovery from Data (TKDD)*, Vol. 1, Nº 1, ACM, 2007.

[Gómez, 2003] Gómez Díaz, R. La evaluación en recuperación de la información [on line]. "Hipertext.net", núm. 1, 2003. <http://www.hipertext.net> [Consulta: 06/04/2007]

[Grefenstette, 1992] Grefenstette, G. Use of syntactic context to produce term association lists for retrieval. *SIGIR'92*, 89-97, June 1992.

[Grossman y Frieder, 1998] Grossman, D. A.; Frieder, O. *Information Retrieval. Algorithms and Heuristics*. Kluwer. 1998.

[Guha et al., 1998] Guha, S.; Rastogi, R. y Shim, K. CURE: An efficient clustering algorithm for large databases. *En Proc. of 1998 ACM-SIGMOD Int. Conf. on Management of Data*, 1998.

[Guha et al., 1999] Guha, S.; Rastogi, R. y Shim, K. ROCK: a robust clustering algorithm for categorical attributes. *En Proc. of the 15th Int'l Conf. on Data Eng.*, 1999.

[Haldiki et al., 2008] Haldiki, M.; Gunopulis, D.; Vazirgiannis, M.; Kumar, N. y Domeniconi, C. A clustering framework based on subjective and objective validity criteria. En *ACM Transactions on Knowledge Discovery from Data (TKDD)*, Vol. 1, Nº 4, Art. 4, 2008.

[Han et al., 1995] Han, C.; Fujii, H. y Croft, W. B. Automatic query expansion for Japanese text retrieval. *Technical Report UM-CS-1995-011, Department of Computer Science, Lederle Graduate Research Center, University of Massachusetts*, 1995.

[Han et al., 2005] Han, H.; Zha, H. y Giles, C. L. Name disambiguation in author citations using a K-way spectral clustering method. En *Proceedings of the 5th*

ACM/IEEE-CS joint conference on Digital libraries, (Denver, CO, USA), 334-343, 2005.

[Handcock et al., 2007] Handcock, M. S.; Raftery, A. E. y Tantrum, J. M. Model-based clustering for social networks. En *Journal of the Royal Statistical Society-Series A*, Vol. 170, part 2, 301-354, 2007.

[Harman, 1992] Harman, D. Ranking Algorithms. En W. B. Frakes and R. Baeza-Yates, editors. *Information Retrieval: Data Structures & Algorithms*, Prentice Hall, Englewood Cliffs, 363-392, NJ, USA, 1992.

[Harman, 1995] Harman, D. Overview of the third text retrieval conference. En *Proc. of the 3rd Text Retrieval Conference (TREC-3)*, 1-19, Gaithersburg, USA, 1995.

[Harold, 2002] Harold, Elliotte Rusty. Processing XML with Java: A Guide to SAX, DOM, JDOM, JAXP, and TrAX. Pearson Education, Inc. 2002.

[Harvest, 2004] Harvest**:** A Distributed Search System. http://harvest.sourceforge.net/

[He et al., 2004] He, J.; Tan, A.; Tan, C. y Sung S. On Quantitative Evaluation of Clustering Systems. En *Information Retrieval and Clustering.* W. Wu, H. Xiong and S. Shekhar (Eds.), 105-134, 2004.

[Hersovici et al., 1998] Hersovici, M. ; Jacobi, M. ; Maarek, Y. S.; Pelleg, D.; Shtalhaim M. y Ur, S. The shark-search algorithm. An application: tailored Web site mapping. En *7th WWW Conference*, Brisbane, Australia, abril de 1998.

[Himmeroder et al., 1997] Himmeroder, Rainer ; Lausen, Georg ; Ludascher, Bertram y Schelepphorst Christian. On a declarative semantics for Web queries. En *Proc. of the Int. Conf. Deductive and Object-Oriented Database(DOOD)*, 386-398, Singapore, diciembre de 1997.

[Hofmann, 1999] Hofmann, T. Probabilistic latent semantic indexing. In *Proceedings of the 22nd International Conference on Research and Development in Information Retrieval (SIGIR'99)*,1999.

[Hooper, 1965] Hooper, R. S. *Indexer consistency tests-origin, measurements, results and utilization.* Bethesda, MD, 1965.

[Huang et al., 2005] Huang, J.Z. Ng, M.K. Hongqiang Rong Zichen Li. Automated variable weighting in k-means type clustering. En *Pattern Analysis and Machine Intelligence, IEEE Transactions on.* Volume: 27, Issue: 5, 657-668, mayo de 2005.

[Jain y Dubes, 1988] Jain, A. y Dubes, R. C. *Algorithms for Clustering Data.* Prentice Hall, 1988.

[Jagannathan y Wright, 2005] Jagannathan, G. y Wright, R. N. Privacy-preserving distributed k-means clustering over arbitrarily partitioned data. En *Proceedings of the Eleventh ACM SIGKDD international Conference on Knowledge Discovery in Data Mining* (Chicago, Illinois, USA, 21 – 24 agosto 2005). KDD '05. ACM, New York, NY, 593-599. 2005.

[Jarvis y Patrick, 1973] Jarvis, R. A. y Patrick, E. A. Clustering using a similarity measure based on shared nearest neighbours. In *IEEE Transactions on Computers*, Vol. C-22, Nº 11, noviembre de 1973.

[Jing y Croft, 1994] Jing, Y. y Croft, W. B. An association thesaurus for information retrieval. *Proceedings of RIAO-94, 4th International Conference "Recherche d'Information Assistee par Ordinateur"*, 146-160, NY, 1994.

[Kalamboukis, 1995] Kalamboukis, T. Suffix stripping with modern Greek. *Program.* 29:313-321, 1995.

[Kantere et al., 2009] Kantere, V.; Tsoumakos, D.; Sellis, T. y Roussopoulos, N. GrouPeer: Dynamic Clustering of P2P Databases. En *Information Systems*, Vol. 34, Nº 1, 62-86, marzo de 2009.

[Karypis et al., 1999] Karypis, G.; Han, E. y Kumar, V. Chameleon: A hierarchical clustering algorithm using dynamic modeling. En *IEEE Computer*, 32(8), 68-75, 1999.

[Karypis, 2001] Karypis, G. CLUTO. A Clustering Toolkit. *Technical Report: #02-01 Department of Computer Science Minneapolis, MN 55455, University of Minnesota*, 2001.

[Kay, 2001] Kay, Michael. XSLT: Programmer's Reference. 2nd Edition. Wrox Press. 2001.

[Kleinberg, 1998] Kleinberg, Jon. Authoritative sources in a hyperlinked environment. En *Proc. of the 9th ACM-SIAM Symposium on Discrete Algorithms*, 668-677. San Francisco, USA, enero de 1998.

[Kohonen, 1997] Kohonen, T. *Self-Organization and Associative Memory*. Springer-Verlag, second edition, Berlín, 1997.

[Konopnicki y Shmueli, 1995] Konopnicki, D. y Shmueli, O. W3QS: A query system for the World Wide Web. En *Proc. of VLDB'95*, 54-65, Zurich, Suiza, septiembre de 1997.

[Korfhage, 1997] Korfhage, Robert R. *Information Storage and Retrieval*. John Wiley & Sons, Inc., 1997.

[Koster, 1993] Koster, M. Guidelines for robot writers, 1993. http://info.webcrawler.com/mak/-projects/robots/guidelines.html

[Kowalski, 1997] Kowalski, G. *Information Retrieval Systems. Theory and Implementation*. Kluwer. 1997.

[Kroventz, 1993] Kroventz, R. Viewing Morphology as an Inference Process. *Proceedings of the 16th ACM/SIGIR Conference*. 191-202, New York, 1993.

[Kulis et al., 2009] Kulis, B.; Basu, S.; Dhillon, I. y Mooney, R. Semi-supervised graph clustering: a kernel approach. En *Machine Learning*. Springer Netherlands, Vol. 74, Nº 1, 1-22, enero de 2009.

[Laksmanan et al., 1996] Laksmanan, Lacks V. S.; Sadri, Fereidoon y Subramanian, Iyer N. A declarative language for querying and restructuring the Web. En *6th Int. Workshop on Research Issues in Data Engineering, RIDE'96.*, Nueva Orleans, febrero de 1996.

[LaMacchia, 1997] LaMacchia, B. The Internet fish construction kit. En *6th Int. WWW Conference*, Santa Clara, CA, USA, abril de 1997.

[Lesk, 1969] Lesk, M. E. Word-word association in document retrieval systems. *American documentation*, 20(1), 27-38, 1969.

[Li et al., 1998] Li, W-S ; Shim, J.; Candan, K. S. y Hara, Y. WebDB: A Web query system and its modeling, language, and implementation. En *Proc. of Advances in Digital Libraries*, Santa Bárbara, CA, USA, abril de 1998.

[Li et al., 2008] Li, Y.; Chung, S. M. y Holt, J. D. Text Document Clustering Based on Frequent Word Meaning Sequences. En *Data & Knowledge Engineering*, Vol. 64, Nº 1, 381-404, enero de 2008.

[Liu et al., 2009] Liu, Y.; Zhang, L. ; Song, R. ; Nie, J-Y. y Wen, J-R. Clustering Queries for Better Document Ranking. En *Proceeding of the 18th ACM conference on Information and knowledge management*, Hong Kong, China, 1569-1572, 2009.

[Liu y Yu, 2005] Liu, Huan; Yu, Lei. Toward Integrating Feature Selection Algorithms for Classification and Clustering. En *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 4, pp. 491-502, abril de 2005.

[López Yepes, 1998] López Yepes, J. Las bases de datos históricas. *Anales de documentación*, Vol. 1. 99-124. 1998.

[López Yepes y Sagredo, 1981] López Yepes, J.; Sagredo Fernández, F. *Estudios de Documentación general e informativa*. Madrid: Seminario Millares Carlo, 413, 1981.

[Lovins, 1968] Lovins, J. B. Development of a Stemming Algorithm. *Mechanical translations and Computational Linguistics*. Vol. 11, 22-31, marzo de 1968.

[Lunch, 1957] Lunch, H. P. A Statistical approach to mechanized encoding and searching of litery information. En *IBM Journal of Research and Development.* 1 (4), 309-313, 1957.

[Maarek et al., 2000] Maarek, Y. S., Fagin; Fagin, R., Ben-Shaul, I. Z. y Pelleg, D. Ephemeral document clustering for Web applications. *IBM Research Report RJ 10186*, abril de 2000.

[Manber y Wu, 1994] Manber, U. y Wu, Sun. GLIMPSE: A tool to search through entire file systems. En *Proc. of USENIX Technical Conference.* 23-32, San Francisco, USA, enero de1994.

[Mateos et al., 2008] Mateos, M.; Beato, E.; Berjón, R.; Fermoso, A. M.; Sánchez, M. A. y García-Figuerola, C. Características para mejorar el clustering de documentos Web. En *Conferência IADIS Ibero-Americana WWW/Internet 2008.* Lisboa, Portugal, diciembre de 2008.

[Mateos y García-Figerola, 2007] Mateos, M. y García-Figuerola, C. Architecture of an hybrid system for experimentation on Web Information Retrieval incorporating clustering techniques. En *Lecture notes computer science. Springer Berlin / Heidelberg.* pp. 427-434. 2007.

[McQueen, 1967] McQueen, J. Some methods for classification and analysis of multivariate observations. 5th Berkeley symposium on mathematics, statistics and probability, 1, 281-298, 1967.

[Mendelzon et al., 1997] Mendelzon, A. ; Mihaila, G. y Milo, T. Querying the World Wide Web. *International Journal on Digital Libraries*, 1(1):54-67, abril de1997.

[Méndez, 1999] Méndez Rodríguez, Eva M.ª. RDF: un modelo de metadatos flexible para las bibliotecas digitales del próximo milenio. En: *7es Jornades Catalanes de Documentació*. Barcelona: COBDC, p. 487-498.

[Minker et al., 1972] Minker, J.; Wilson, G. A. y Zimmerman, B. H. An evaluation of query expansion by the addition of clustered terms for a document retrieval system. *Information storage and retrieval,* 8(6), 329-348, 1972.

[Mizzaro, 1998] Mizzaro, S. How many relevances in information retrieval? En *Interacting With Computers*, 10(3):305-322, 1998.

[Navarro y Baeza-Yates, 1995] Navarro, G. y Baeza-Yates, R. A language for queries on structure and contents of textual databases. En *Proc. of the 18th Annual Int. ACM SIGIR Conference on Research and Development in Information Retrieval*, 93-101, Seattle, USA, Julio de 1995.

[Navarro y Baeza-Yates, 1997] Navarro, G. y Baeza-Yates, R. Proximal nodes: A model to query document databases by content and structure. *ACM Transactions on Office and Information Systems,* 15(4):401-435, 1997.

[Ng y Han, 1994] Ng, R. y Han, J. Efficient and effective clustering method for spatial data mining. En *Proc. of the 20th VLDB Conference*, 144-155, Santiago, Chile, 1994.

[Ngu y Wu, 1997] Ngu, D. y Wu, X. SiteHelper: a localized agent that helps incremental exploration of the World Wide Web. En *6th Int. WWW Conference*, Santa Clara, CA, USA, abril de 1997.

[Nielsen, 1995] Nielsen, J. *Multimedia and hypertext: the Internet and beyond*. Academic Press. Boston. 1995.

[Noel et al., 2004] Noel, S.; Raghavan, V. y Henry Chu, C.-H. Document clustering, visualization, and retrieval via link mining. En *Information Retrieval and Clustering.* W. Wu, H. Xiong and S. Shekhar (Eds.), 161-194, 2004.

[Ogawa et al., 1991] Ogawa, Y.; Morita, T. y Kobayashi, K. A fuzzy document retrieval system using the keyword connection matrix and learning method. *Fuzzy Sets and Systems,* 39:163-179, 1991.

[Otlet, 1935] Otlet, P. *Traité de documentation. Le livre sur le livre. Theorie et practique*. Bruselles: Editions Mundaneum, 1934.

[Pavlopoulos et al., 2009] Pavlopoulos, G.A.; Moschopoulos, C.N.; Hooper, S.D.; Schneider, R. y Kossida, S. jClust: a clustering and visualization toolbox. En *Bioinformatics Applications Note*, Vol. 25, Nº 15, 1994-1996, 2009.

[Pearl, 1988] Pearl, J. *Probabilistic Reasoning in Inteligent Systems: Networks of Plausible Inference.* Morgan Kaufmann Publishers, Inc., 1988.

[Peat y Willett, 1991] Peat, H. J. y Willet, P. The limitation of term co-occurrence data for query expansion in document retrieval system. *J. of the ASIS*, 42(5), 378-383, 1991.

[Peña et al., 2002] Peña, R.; Baeza-Yates, R. y Rodríguez, J. V. *Gestión digital de la información. De bits a bibliotecas digitales y la Web*. RA-MA Editorial, 2002.

[Peters y Braschler, 2001]. Peters, C y Braschler, M. European Research Letter: Cross-Language System Evaluation: The CLEF Campaigns. *Journal of the American Society for Information Science and Technology JASIST*, Vol. 52, Issue 12, 1067-1072, John Wiley & Sons, 2001.

[Pinto, 1993] Pinto Molina, M. *Análisis documental. Fundamentos y procedimientos*. Madrid: Eudema, 2ª ed., 1993.

[Pirolli et al., 1996] Pirolli, P.; Pitkow y Rao, R. Silk from a sow's ear: Extracting usable structures from the Web. En *Proceedings of 1996 Conference on Human Factors in Computing Systems (CHI-96)*, 1996.

[Porter, 1980] Porter, M. An Algorithm for Suffix Sripping. *Program*, Vol. 14(3), 130-137, 1980.

[Portnoy, 2009] Portnoy, L. Intrusion detection with unlabeled data using clustering. En *King Saud University Initial Collection*, 2009.

[Potel, 1996] Potel, Mike. MVP: Model-View-Presenter. The Taligent Programming Model for C++ and Java. [on line] <http://www.wildcrest.com/Potel/Portfolio/mvp.pdf> [Consulta: 02/09/2008].

[Qiu y Frei, 1993] Qui, Y. y Frei, H. P. Concept based query expansion. *SIGIR'93*, 160-169, 1993.

[Raghavan et al., 1989] Raghavan, V. V.; Jung, G. S. y Bollmann, P. A critical investigation of recall and precision as measures of retrieval system performance. *ACM Transactions on Office and Information Systems,*, 7(3):205-229, 1989.

[Ramage et al., 2009] Ramage, D.; Heymann, P. ; Manning, C. D. y García-Molina, H. Clustering the Tagged Web. En *Proceedings of the Second ACM International Conference on Web Search and Data Mining*, Barcelona, España, 54-63, 2009.

[Reenskaug, 1979] Reenskaug, Trygve M. H. MVC XEROX PARC 1978-79 [on line] <http://heim.ifi.uio.no/~trygver/themes/mvc/mvc-index.html> [Consulta: 02/09/2008].

[Ribeiro-Neto et al., 1996] Ribeiro-Neto, Berthier A. y Muntz, Richard. A belief network model for IR. En *Proc. of the 19th Annual Int. ACM SIGIR Conference on Research and Development in Information Retrieval. 253-260*, Zurich, Suiza, 1996.

[Rijsbergen, 1979] van Rijsbergen, C. J. *Information Retrieval, 2ª ed.* Butterworths, Londres, 1979. También en http://www.dcs.gla.ac.uk/Keith/Chapter.1/Ch.1.html

[Roberston y Sparck Jones, 1976] Roberston, S. E. y Sparck Jones, K. *Relevance weighting of search terms. Journal of the American Society for Information Sciences*, 27(3):129-146, 1976.

[Ruge, 1992] Ruge, G. Experiments on linguistically-based term associations. *Information processing and management*, 28(3), 317-332, 1992.

[Salton et al., 1983] Salton, G.; Fox, E. A. y Wu, H. Extended Boolean information retrieval. *Communications of the ACM,* 26(11):1022-1036, noviembre de 1983.

[Salton y Buckley, 1988] Salton, G. y Buckley, C. Term-weighting approaches in automatic retrieval. *Information Processing & Management*, 24(5):513-523, 1988.

[Salton y Lesk, 1968] Salton, G. y Lesk, M. E. Computer evaluation of indexing and text processing. *Journal of the ACM,* 15(1):8-36, enero de 1968.

[Salton y McGill, 1983] Salton G. y McGill M. J. *Introduction to Modern Information Retrieval.* McGraw Hill, N.Y. 1983.

[Salton, 1971a] Salton, G. *The SMART Retrieval System. Experiments in Automatic Document Processing.* Prentice Hall Inc., Englewood Cliffs, NJ, 1971.

[Salton, 1971b] Salton, G. Experiments in automatic thesaurus construction for information retrieval. *Information Processing*, 1, 115-123, 1971.

[Salton, 1980] Salton, G. Automatic term class construction using relevance – a summary of work in automatic pseudoclassification. *Information processing & management*, 16(1), 1-15, 1980.

[Salton, 1989] Salton, G. *Automatic Text Processing.* Addison-Wesley. 1989.

[Saracevic, 1975] Saracevic, T. Relevance: A review of and a framework for the thinking on the notion in information science. En *Journal of the American Society for Information Science*, 26(6):321- 343, 1975.

[Savoy, 1999] Savoy, J. A Stemming Procedure and Stopword List for General French Corpora. *Jounal of the American Society for Information Science.* 50: 944-952.

[Schütze y Silverstein, 1997] Schütze, H. y Silverstein, C. A comparison of projections for efficient document clustering. En *Proceedings of the 20th Annual ACM SIGIR Conference on Research and Development in Information Retrieval*, 74-81, 1997.

[Shaw Jr. et al., 1997] Shaw Jr., W. M.; Burgin, R. y Howell, P. Performance standards and evaluations in IR test collections: Cluster-based retrieval models. *Information Processing & Management,* 33(1):1-14, 1997.

[Shepitsen et al., 2008] Shepitsen, A.; Gemmell, J.; Mobasher, B. y Burke, R. Personalized Recommendation in Social Tagging Systems Using Hierarchical Clustering. En *Proceedings of the 2008 ACM conference on Recommender systems,* Lausanne, Suiza, 259-266, 2008.

[Šilić et al., 2008] Šilić, A.; Žmak, L.; Bašić, B. D. y Moens, M-F. Comparing document classification schemes using k-means clustering. En *Proceedings of the 12th international conference on Knowledge-Based Intelligent Information and Engineering Systems, Part I,* Springer-Verlag, septiembre de 2008.

[Smeaton y van Rijsbergen, 1983] Smeaton, A. F. y van Rijsbergen, C. J. The retrieval effects of query expansion on a feedback document retrieval system. *The computer journal*, 26(3), 239-246, 1983.

[Small, 1973] Small, H. Co-citation in the Scientific Literature: A New Measure of the Relationship Between Two Documents. En *Journal of the American Society of Information Science*, 24, 265-269, 1973.

[Smyth, 1996] Smyth, P. Clustering using Monte Carlo cross-validation. In *Second International Conference on Knowledge Discovery and Data Mining (KDD-96)*, 126-133, Portland, OR, august 1996.

[Spark-Jones y Barber, 1971] Spark-Jones, K. y Barber, E. B. What makes an automatic keyword classification effective?. *J. of the ASIS*, 18, 166-175, 1971.

[Spark-Jones, 1991] Spark-Jones, K. Notes and references on early classification work. *SIGIR Forum*, 25(1), 10-17, 1991.

[Steinbach et al., 2000] Steinbach, M.; Karypis, G. y Kumar, V. A comparison of Document Clustering Techniques. *Technical Report #00-034*. *University of Minnesota*. En *KDD Workshop on Text Mining*, 2000.

[Stubbs et al., 2000] Stubbs., E. A.; Mangiaterra, N. E. y Martínez, A. M. Internal quality audit of indexing: a new application of interindexer consistency. En *Cataloguing & Classification Quarterly*, 28(4):53-70, 2000.

[Tague-Sutcliffe, 1992] Tague-Sutcliffe, J. Measuring the informativeness of a retrieval process. En *Proc. of the 15th Annual Int. ACM SIGIR Conference on Research and Development in Information Retrieval*, 23-36, Copenhagen, Dinamarca, 1971.

[Taube, 1955] Taube, M. *The Uniterms system of indexing operating Manual*. 1955.

[Tittel et al., 1996] Tittel, E.; Gaither, M.; Hassinger, S. y Erwin, M. *Fundamentos de programación: HTML & CGI*, Anaya multimedia, 1996.

[Turtle y Croft, 1990] Turtle Howard y Crof, W. Bruce. Inference networks-based retrieval model. En *Proc. of the 13th Annual Int. ACM SIGIR Conference on Research and Development in Information Retrieval*, 1-24, Bruxelas, Bélgica. 1990.

[Turtle y Croft, 1991] Turtle Howard y Crof, W. Bruce. Evaluation of an inference network-based retrieval model. *ACM transactions in Information Systems*, 9(3):187-222, Julio de 1991.

[Villagrá et al., 1999] Villagrá, V. A.; Berrocal, J.; Moreno, J. I.; López de Vergara, J. E. Desarrollo y validación de un sistema de intermediación electrónica en el entorno de las Administraciones Públicas. En *Actas del I Seminario del Programa Nacional de Aplicaciones y Servicios Telemáticos*

(SPAST-I). OCYT, Dirección General de Enseñanza Superior e Investigación Científica del MEC, Universidad Pública de Navarra, diciembre de 1999.

[Wen et al., **2002] Wen, J.-R.; Nie, J.-Y. y Zhang, H.-J.** Query Clustering Using User Logs. En *ACM Transactions on Information Systems (ACM TOIS)*, vol. 20, nº 1, 59-81, 2002.

[Wen y Zhang] Wen, J. y Zhang, H. Query Clustering in the Web Context. En *Information Retrieval and Clustering.* W. Wu, H. Xiong and S. Shekhar (Eds.), 195-226, 2004.

[Wilkinson y Hingston, 1991] Wilkinson, R. y Hingston, P. Using the cosine measure in a neural network for document retrieval. En *Proc. of the ACM SIGIR Conference on Research and Development in Information Retrieval,* 202-210, Chicago, octubre de 1991.

[Willett, 1988] Willett, P. Recent trends in hierarchic document clustering: A critical review. *Information Processing and Management*, 24(5), 577-597, 1988.

[Wong et al., 1985] Wong, S. K. M.; Ziarko, W. y Wong, P. C. N. Generalized vector space model in information retrieval. En *Proc. 8th ACM SIGIR Conference on Research and Development in Information Retrieval,* 18-25, N.Y., 1985.

[von Luxburg, 2007] von Luxburg, U. A Tutorial on Spectral Clustering. En *Statistics and Computing.* Vol. 17, Nº 4, 395-416, diciembre de 2007.

[Woodhead, 1991] Woodhead, N. *Hypertext and Hypermedia: theory and applications,* Workingham: Addison-Wesley, 1991.

[Yuwono y Lee, 1996] Yuwono, B. y Lee, D. L. Search and ranking algorithms for locating resources on World Wide Web, En *Proc. of the Int. Conference on Data Engineering (ICDE)*, 164-171, Nueva Orleans, USA, 1996.

[Zadeh, 1993] Zadeh, L. A. En D. Dubois, H. Prade, y R. R. Yager, editors, *Readings in Fuzzy Sets for Intelligent Systems*. Morgan Kaufmann, 1993.

[Zamir y Etzioni, 1999] Zamir, O. y Etzioni, O. Grouper: A dynamic clustering interface to Web search results. *Proceeding of the eighth international World Wide Web conference*. Computer Networks and ISDN Systems, 1999.

[Zhao et al., 2005] Zhao, Y.; Karypis, G y Fayyad, U. Hierarchical Clustering Algorithms for Document Datasets, En *Data Mining and Knowledge Discovery.* Springer, Vol 10, Nº2, 141-168, 2005.

[Zhao y Karypis, 2001] Zhao, Y. y Karypis, G. Criterion functions for document clustering, *Technical Report #01-40, Department of Computer Science, University of Massachusetts*, 2001.

[Zhao y Karypis, 2002] Zhao, Y. y Karypis, G. Evaluation of Hierarchical Clustering Algorithms for Document Datasets, En *Proceedings of the 11th Conference of Information and Knowledge Management (CIKM'02)*, 515-524, noviembre de 2002.

[Zimmermann y De Raedt, 2009] Zimmermann, A. y De Raedt, L. Cluster-grouping: from subgroup discovery to clustering. En *Machine Learning*, Springer Netherlands, Vol. 77, Nº 1, 125-159, junio de 2009.

APÉNDICES

A Resultados para el caso de estudio A con 30 *clusters*

Tabla 15. Resultados de los métodos particionales, obteniendo 30 *clusters* (caso A).

	i1		i2		e1		g1		g1p		h1		h2	
	Size	**ISim**	**Size**	**ISim**	**Size**	**ISim**	**Size**	**ISim**	**Size**	**ISim**	**Size**	**ISim**	**Size**	**ISim**
rb	30	1,000	30	1,000	58	0,919	1	1,000	145	0,618	30	1,000	46	0,990
	56	0,981	46	0,990	182	0,758	1	1,000	213	0,598	56	0,981	131	0,980
	46	0,990	57	0,956	161	0,707	30	1,000	79	0,504	46	0,990	57	0,956
	131	0,970	58	0,919	138	0,646	15	1,000	194	0,441	131	0,980	58	0,919
	58	0,919	136	0,916	102	0,544	15	1,000	78	0,361	58	0,919	39	0,817
	18	0,956	39	0,817	175	0,492	10	1,000	114	0,367	38	0,855	195	0,701
	65	0,942	29	0,731	63	0,378	28	1,000	139	0,240	27	0,836	27	0,679
	38	0,855	27	0,679	40	0,408	46	0,990	163	0,223	178	0,774	139	0,642
	27	0,836	197	0,683	92	0,359	66	0,975	326	0,186	137	0,658	176	0,488
	69	0,851	137	0,658	102	0,390	18	0,929	75	0,209	31	0,607	49	0,397
	17	0,734	150	0,506	140	0,263	28	0,920	186	0,170	149	0,510	103	0,387
	110	0,787	31	0,435	141	0,262	29	0,916	153	0,155	33	0,455	156	0,243
	22	0,714	91	0,364	73	0,264	68	0,911	40	0,151	90	0,371	68	0,270
	119	0,676	103	0,387	54	0,184	20	0,812	81	0,138	103	0,387	151	0,235
	30	0,625	123	0,309	155	0,152	16	0,750	165	0,135	124	0,328	71	0,175
	31	0,615	137	0,290	65	0,115	194	0,702	55	0,129	119	0,331	155	0,134
	35	0,421	52	0,273	65	0,108	41	0,454	53	0,124	52	0,222	80	0,115
	85	0,395	84	0,228	105	0,100	176	0,488	33	0,117	56	0,194	78	0,112
	71	0,423	58	0,167	100	0,093	69	0,399	51	0,125	125	0,184	199	0,110
	24	0,366	57	0,160	144	0,089	47	0,388	107	0,108	94	0,214	92	0,106
	111	0,357	56	0,140	70	0,086	206	0,405	91	0,105	57	0,143	76	0,108
	124	0,328	86	0,135	93	0,088	41	0,293	54	0,106	80	0,123	77	0,104
	78	0,320	176	0,126	94	0,079	65	0,275	94	0,098	108	0,116	142	0,090
	40	0,218	74	0,105	141	0,070	149	0,297	69	0,091	56	0,111	90	0,085
	81	0,238	85	0,089	103	0,066	152	0,234	67	0,083	194	0,067	86	0,082
	118	0,197	92	0,082	106	0,067	109	0,106	37	0,083	218	0,052	68	0,075
	66	0,155	113	0,070	77	0,056	220	0,096	68	0,079	140	0,037	85	0,068
	55	0,139	278	0,048	73	0,057	235	0,059	85	0,076	115	0,036	104	0,068
	476	0,049	229	0,046	95	0,054	457	0,045	39	0,070	319	0,039	151	0,061
	858	0,020	258	0,028	82	0,027	537	0,028	35	0,064	125	0,026	140	0,036
	0,336	0,569	**0,324**	0,411	**0,286**	0,263	**0,301**	0,616	**0,241**	**0,198**	**0,327**	0,418	**0,309**	0,341

	i1		i2		e1		g1		g1p		h1		h2	
	30	1,000	**Size**	**ISim**	**Size**	**ISim**	**Size**	**ISim**	**Size**	**ISim**	**Size**	**ISim**	**Size**	**ISim**
	55	1,000	30	1,000	185	0,743	1	1,000	211	0,619	30	1,000	138	0,889
	46	0,990	57	0,956	137	0,658	1	1,000	174	0,508	129	1,000	189	0,722
	131	0,970	133	0,955	189	0,518	29	1,000	175	0,492	56	0,981	138	0,651
	58	0,919	47	0,951	37	0,492	15	1,000	119	0,346	46	0,990	36	0,505
	60	0,984	58	0,919	174	0,494	15	1,000	50	0,342	58	0,919	174	0,494
	38	0,855	39	0,817	99	0,395	28	1,000	253	0,300	38	0,855	77	0,382
	63	0,885	29	0,731	148	0,270	46	0,990	159	0,219	27	0,836	104	0,381
	27	0,836	195	0,690	145	0,251	66	0,975	111	0,203	180	0,762	100	0,325
	105	0,809	138	0,651	75	0,263	18	0,929	178	0,187	136	0,661	129	0,298
	17	0,734	33	0,502	159	0,218	67	0,937	45	0,182	33	0,571	152	0,252
	38	0,778	149	0,511	46	0,177	29	0,917	142	0,166	147	0,518	118	0,232
	22	0,714	92	0,359	148	0,155	29	0,916	139	0,165	90	0,371	119	0,221
	115	0,696	111	0,357	146	0,158	11	0,880	156	0,145	106	0,381	79	0,250
	30	0,625	136	0,293	150	0,144	15	0,834	185	0,139	132	0,304	46	0,178
	31	0,615	130	0,292	113	0,132	20	0,812	84	0,142	126	0,307	97	0,149
rbr	38	0,406	58	0,244	95	0,122	187	0,723	60	0,139	45	0,294	150	0,153
	85	0,395	52	0,230	74	0,112	23	0,653	59	0,133	88	0,227	87	0,142
	73	0,417	82	0,237	92	0,113	139	0,631	93	0,130	148	0,153	86	0,139
	24	0,366	60	0,157	77	0,115	40	0,466	67	0,131	92	0,117	79	0,118
	110	0,361	57	0,156	66	0,109	174	0,490	56	0,131	88	0,097	91	0,115
	121	0,337	147	0,153	68	0,112	60	0,399	82	0,127	101	0,090	87	0,113
	78	0,317	57	0,137	115	0,111	50	0,378	70	0,126	144	0,090	82	0,109
	40	0,218	108	0,109	79	0,105	49	0,299	50	0,123	93	0,073	119	0,106
	111	0,209	82	0,106	75	0,105	127	0,296	48	0,115	138	0,068	85	0,101
	94	0,222	84	0,095	56	0,095	76	0,268	58	0,114	135	0,066	88	0,094
	47	0,175	125	0,068	60	0,097	235	0,214	73	0,112	239	0,066	88	0,089
	104	0,126	150	0,065	70	0,094	136	0,165	32	0,107	127	0,062	75	0,088
	437	0,051	225	0,054	101	0,089	283	0,055	61	0,106	115	0,056	67	0,086
	861	0,021	225	0,046	62	0,069	444	0,046	62	0,106	81	0,043	128	0,076
	30	1,000	200	0,032	48	0,043	676	0,025	37	0,093	121	0,039	81	0,053
	0,336	0,582	**0,324**	0,396	**0,261**	0,219	**0,308**	0,643	**0,244**	0,198	**0,325**	0,4	**0,284**	0,25

	i1		i2		e1		g1		g1p		h1		h2	
	Size	**ISim**	**Size**	**ISim**	**Size**	**ISim**	**Size**	**ISim**	**Size**	**ISim**	**Size**	**ISim**	**Size**	**ISim**
direct	30	1,000	46	0,990	170	0,789	30	1,000	190	0,677	30	1,000	139	0,875
	55	1,000	57	0,956	46	0,674	10	1,000	142	0,555	129	1,000	180	0,738
	129	1,000	133	0,955	167	0,612	46	0,990	56	0,514	58	0,919	44	0,707
	58	0,919	180	0,734	124	0,650	57	0,956	162	0,504	38	0,855	123	0,651
	48	0,911	44	0,707	36	0,508	133	0,955	94	0,360	168	0,799	162	0,504
	38	0,855	40	0,625	161	0,507	58	0,919	60	0,354	44	0,707	119	0,407
	15	0,834	96	0,493	91	0,387	38	0,855	238	0,310	119	0,676	96	0,367
	19	0,822	163	0,501	60	0,354	15	0,834	159	0,216	102	0,544	137	0,275
	12	0,780	180	0,429	148	0,270	155	0,832	170	0,210	33	0,560	51	0,278
	177	0,776	92	0,359	145	0,252	20	0,756	61	0,228	162	0,504	154	0,244
	136	0,661	132	0,303	171	0,203	158	0,546	179	0,186	91	0,365	85	0,274
	20	0,624	132	0,289	145	0,164	174	0,490	150	0,151	91	0,384	122	0,211
	31	0,607	134	0,320	147	0,157	54	0,451	148	0,148	124	0,313	125	0,209
	175	0,492	60	0,227	148	0,149	81	0,387	148	0,156	151	0,250	94	0,162
	38	0,406	51	0,226	90	0,130	121	0,337	53	0,156	101	0,244	151	0,151
	85	0,395	59	0,192	83	0,131	45	0,323	77	0,144	151	0,152	85	0,141
	24	0,366	141	0,159	110	0,123	69	0,262	86	0,137	89	0,133	80	0,136
	110	0,361	63	0,134	66	0,128	35	0,238	107	0,134	87	0,124	96	0,118
	20	0,342	64	0,126	66	0,133	243	0,222	62	0,137	69	0,121	69	0,121
	121	0,337	63	0,121	102	0,117	38	0,189	93	0,130	115	0,101	105	0,112
	165	0,274	103	0,118	81	0,113	94	0,188	50	0,122	86	0,105	55	0,107
	45	0,224	78	0,103	86	0,104	45	0,176	75	0,123	95	0,092	84	0,108
	46	0,190	73	0,086	57	0,103	49	0,147	72	0,125	100	0,083	79	0,113
	40	0,152	96	0,082	62	0,103	88	0,137	76	0,121	71	0,068	89	0,100
	84	0,144	51	0,076	97	0,100	97	0,102	53	0,120	144	0,070	112	0,088
	36	0,119	122	0,073	111	0,090	151	0,078	98	0,113	102	0,060	78	0,086
	229	0,069	91	0,070	71	0,087	171	0,055	57	0,110	208	0,062	77	0,080
	173	0,067	212	0,068	127	0,083	271	0,048	49	0,102	121	0,061	64	0,070
	358	0,034	159	0,044	61	0,075	232	0,042	82	0,105	83	0,050	161	0,073
	572	0,027	174	0,034	60	0,072	311	0,030	42	0,097	127	0,040	73	0,064
	0,328	0,493	**0,304**	0,32	**0,267**	0,246	**0,32**	0,452	**0,25**	0,218	**0,31**	0,348	**0,28**	0,252

Tabla 16. Resultados de los métodos acumulativos, obteniendo 30 *clusters* (caso A).

	i1		i2		e1		g1		g1p		h1		h2		slink		clink		upgma	
otgga	**Size**	**ISim**	**Size**	**ISim**	**Size**	**ISim**	**Size**	**ISim**	**Size**	**ISim**	**Size**	**ISim**	**Size**	**ISim**	**Size**	**ISim**	**Size**	**ISim**	**Size**	**ISim**
	2	1,000	2	1,000	2	1,000	271	0,348	2	1,000	183	0,047	97	0,362	2	1,000	188	0,200	2	1,000
	100	0,368	102	0,738	86	0,079	1	1,000	48	0,158	129	1,000	100	0,117	30	1,000	58	0,919	96	0,493
	129	1,000	80	0,392	127	0,063	67	0,962	71	0,103	119	0,338	141	0,053	1	1,000	112	0,353	728	0,309
	1556	0,023	129	1,000	107	0,101	515	0,033	84	0,093	153	0,850	132	0,068	1	1,000	24	0,160	9	0,294
	55	1,000	38	0,855	102	0,079	73	0,347	49	0,116	101	0,745	129	1,000	1	1,000	8	0,386	267	0,191
	58	0,919	68	0,139	192	0,462	44	0,433	201	0,625	58	0,919	74	0,095	2	0,986	8	0,348	7	0,226
	46	0,990	137	0,278	71	0,078	22	0,840	190	0,450	55	1,000	168	0,755	1	1,000	5	0,292	5	0,357
	118	0,341	183	0,680	61	0,107	32	1,000	49	0,090	119	0,299	126	0,066	1	1,000	5	0,414	7	0,201
	68	0,837	58	0,919	83	0,330	16	1,000	179	0,116	188	0,063	139	0,260	1	1,000	28	0,368	77	0,120
	99	0,756	56	0,981	209	0,127	46	0,990	196	0,158	46	0,990	144	0,220	1	1,000	2	0,581	3	0,409
	38	0,855	125	0,285	67	0,091	138	0,255	253	0,275	97	0,271	119	0,066	1	1,000	694	0,325	77	0,128
	30	1,000	379	0,040	201	0,090	340	0,266	63	0,103	129	0,056	63	0,089	1	1,000	2	1,000	5	0,294
	80	0,392	46	0,990	159	0,488	39	0,229	127	0,159	166	0,418	121	0,223	1	1,000	65	0,780	5	0,442
	76	0,244	31	0,400	160	0,811	1	1,000	149	0,137	86	0,366	157	0,496	1	1,000	16	0,193	3	0,419
	38	0,774	173	0,459	158	0,125	26	0,963	167	0,170	200	0,044	82	0,468	1	1,000	35	0,229	396	0,187
	38	0,459	30	1,000	140	0,629	64	1,000	126	0,138	38	0,855	90	0,171	1	1,000	6	0,350	1	1,000
	96	0,852	161	0,050	96	0,270	132	0,885	122	0,225	30	1,000	156	0,103	1	1,000	6	0,328	3	0,641
	15	1,000	82	0,097	112	0,261	24	0,594	94	0,111	67	0,452	58	0,684	1	1,000	162	0,700	36	0,074
	26	0,688	97	0,188	96	0,109	63	0,200	100	0,146	127	0,073	74	0,075	1	1,000	4	0,648	101	0,104
	29	0,502	93	0,066	109	0,161	16	1,000	95	0,107	244	0,054	126	0,076	1	1,000	5	0,408	10	0,232
	24	0,895	67	0,448	63	0,294	25	0,903	68	0,096	110	0,164	99	0,280	1	1,000	53	0,114	9	0,258
	17	0,734	425	0,028	114	0,067	42	0,368	60	0,102	38	0,774	65	0,343	1	1,000	38	0,855	41	0,277
	37	0,503	29	0,502	74	0,163	16	0,891	147	0,522	26	0,688	122	0,061	1	1,000	12	0,335	2	0,538
	10	1,000	26	0,688	95	0,158	15	0,887	65	0,138	97	0,095	64	0,317	1	1,000	20	0,254	152	0,056
	29	0,624	36	0,277	113	0,327	83	0,195	83	0,148	70	0,177	60	0,234	1	1,000	7	0,253	3	0,560
	145	0,201	112	0,086	39	0,239	21	0,563	57	0,086	46	0,373	64	0,093	2	0,863	5	0,404	2	0,615
	12	0,780	157	0,059	65	0,462	425	0,032	43	0,146	24	0,895	101	0,071	1	1,000	7	0,206	2	0,933
	57	1,000	38	0,774	52	0,123	16	1,000	42	0,136	146	0,052	65	0,130	1	1,000	746	0,031	1012	0,035
	25	0,547	38	0,459	49	0,116	415	0,036	53	0,372	98	0,039	38	0,774	2	0,818	65	0,075	5	0,283
	36	0,578	91	0,282	87	0,120	101	0,130	106	0,346	99	0,048	115	0,637	3027	0,033	703	0,030	23	0,173
	0,333	0,695	**0,319**	0,472	**0,248**	0,251	**0,281**	0,612	**0,229**	0,219	**0,319**	0,438	**0,279**	0,28	**0,052**	0,957	**0,216**	0,385	**0,168**	0,362

B Resultados para el caso de estudio B con 20 y 30 *clusters*

B.1 Resultados métodos particionales para 20 *clusters*

Tabla 17. Resultados de los métodos particionales, obteniendo 20 *clusters* (caso B).

	i1		i2		e1		g1		g1p		h1		h2	
	Size	**ISim**	**Size**	**ISim**	**Size**	**ISim**	**Size**	**ISim**	**Size**	**ISim**	**Size**	**ISim**	**Size**	**ISim**
rb	574	1,000	3858	0,999	3861	0,998	95244	3,890	3940	0,963	994	1,000	3866	0,995
	752	0,989	3860	0,998	3856	1,000	4	1,000	3857	0,999	521	0,975	3856	1,000
	994	1,000	3870	0,994	3859	0,999	1	1,000	3857	1,000	644	0,877	3859	0,999
	521	0,975	3865	0,990	3844	0,999	1	1,000	3819	1,000	3857	1,000	3844	0,999
	702	0,884	3859	0,999	3857	0,999	1	1,000	3809	0,998	3856	1,000	3858	0,999
	3857	1,000	3860	0,998	3857	1,000	1	1,000	3874	0,994	3859	0,999	3857	1,000
	3856	1,000	3877	0,963	3820	0,999	1	1,000	4101	0,907	3844	0,999	3824	0,998
	3859	0,999	3879	0,988	3830	0,993	1	1,000	4394	0,777	3804	0,997	3906	0,980
	3843	1,000	3860	0,983	3873	0,994	1	1,000	32355	0,288	3858	0,999	3937	0,960
	3798	0,999	1346	0,574	4327	0,813	2	1,000	5040	0,605	3857	1,000	4487	0,752
	3857	0,999	2025	0,497	4439	0,209	2	1,000	5157	0,575	3820	0,999	1387	0,537
	3857	1,000	34767	0,262	23581	0,347	31	1,000	6584	0,088	3852	1,000	30217	0,317
	3819	1,000	977	0,419	2770	0,339	3	1,000	2720	0,080	1339	0,670	2362	0,396
	3852	1,000	5147	0,606	4367	0,203	1	1,000	6539	0,036	4001	0,938	2297	0,126
	3877	0,976	984	0,145	6028	0,045	152	0,989	2397	0,039	1269	0,450	3414	0,119
	581	0,599	3606	0,108	7405	0,023	75	0,961	2621	0,038	19754	0,419	7850	0,021
	14972	0,467	1417	0,073	2971	0,027	276	0,832	3623	0,032	11263	0,159	3097	0,025
	14021	0,217	2722	0,033	4553	0,040	5704	0,120	1901	0,097	4180	0,033	4733	0,041
	23415	0,007	10183	0,015	7160	0,005	5210	0,053	2610	0,026	19323	0,006	7516	0,005
	11748	0,041	8793	0,026	4497	0,044	44	0,044	3557	0,049	8860	0,040	4588	0,052
	0,495	0,808	**0,468**	0,584	**0,468**	0,554	3,484	0,994	**0,442**	0,48	**0,492**	0,728	**0,474**	0,566
	Size	**ISim**	**Size**	**ISim**	**Size**	**ISim**	**Size**	**ISim**	**Size**	**ISim**	**Size**	**ISim**	**Size**	**ISim**
rbr	574	1,000	3869	0,994	3893	0,986	97333	2,120	4518	0,764	994	1,000	3862	0,998
	748	0,995	3883	0,988	3922	0,974	4	1,000	4520	0,762	586	0,978	3882	0,991
	994	1,000	3864	0,990	3872	0,989	1	1,000	4552	0,760	521	0,975	3992	0,942
	521	0,975	3958	0,956	3867	0,975	1	1,000	4539	0,757	3857	1,000	3823	0,989
	700	0,887	3861	0,998	3859	0,999	1	1,000	4589	0,755	3860	0,999	3880	0,985
	3857	1,000	3861	0,998	3857	1,000	1	1,000	4800	0,672	3865	0,998	3857	1,000
	3857	1,000	3889	0,958	3852	1,000	1	1,000	4787	0,662	3843	1,000	3860	0,998

	i1		i2		e1		g1		g1p		h1		h2	
	3861	0,999	3861	0,982	3825	0,998	1	1,000	4960	0,640	3802	0,998	3853	0,999
	3843	1,000	3894	0,982	3828	0,993	1	1,000	4981	0,625	3859	0,999	3848	0,989
	3798	0,999	1384	0,547	4665	0,712	2	1,000	5442	0,523	3857	1,000	3934	0,959
	3857	0,999	2018	0,501	23507	0,349	2	1,000	26272	0,360	3820	0,999	27777	0,333
	3857	1,000	4586	0,742	2821	0,335	31	1,000	812	0,246	3853	0,999	2342	0,199
	3819	1,000	31684	0,299	3829	0,231	3	1,000	5817	0,107	3863	0,983	4260	0,126
	3853	0,999	1061	0,365	6449	0,095	1	1,000	2162	0,078	20253	0,412	2797	0,090
	3826	0,990	3503	0,113	6801	0,037	152	0,989	3938	0,043	2603	0,190	2858	0,199
	577	0,604	1194	0,113	3845	0,034	74	0,978	2426	0,040	3418	0,078	5170	0,054
	15252	0,464	1957	0,045	2572	0,059	271	0,855	2782	0,038	10353	0,167	5704	0,024
	14483	0,207	3008	0,028	5922	0,021	3163	0,118	3219	0,033	10167	0,018	4763	0,024
	23258	0,007	10636	0,015	3160	0,062	5676	0,121	3226	0,025	7217	0,049	5522	0,055
	11220	0,040	10784	0,032	8409	0,007	36	0,060	8413	0,023	12164	0,007	6771	0,007
	0,495	0,808	**0,474**	0,582	**0,462**	0,543	**1,948**	0,912	**0,412**	0,396	**0,486**	0,692	**0,47**	0,548
direct	**Size**	**ISim**	**Size**	**ISim**	**Size**	**ISim**	**Size**	**ISim**	**Size**	**ISim**	**Size**	**ISim**	**Size**	**ISim**
	574	1,000	3868	0,994	4359	0,814	170	0,990	29484	0,328	574	1,000	4066	0,913
	183	0,921	215	0,679	15431	0,719	334	0,692	19906	0,673	183	0,921	15431	0,719
	695	0,893	866	0,552	19149	0,712	836	0,586	16302	0,655	3857	1,000	19288	0,706
	3857	1,000	15445	0,718	29350	0,329	1949	0,530	6090	0,435	15429	0,719	30447	0,316
	356	0,712	31508	0,302	1585	0,165	169	0,471	5399	0,121	19251	0,708	4497	0,169
	15428	0,719	20064	0,662	5122	0,134	14307	0,694	2096	0,111	29807	0,324	1739	0,149
	19161	0,711	2968	0,378	681	0,135	39314	0,216	1342	0,092	2846	0,404	1291	0,104
	285	0,398	745	0,276	1110	0,100	11955	0,607	1645	0,070	1884	0,221	4092	0,083
	2842	0,404	900	0,152	1345	0,091	1748	0,373	831	0,064	854	0,144	1781	0,062
	605	0,380	3370	0,120	3160	0,089	15667	0,535	5293	0,052	1088	0,139	1181	0,072
	510	0,341	1096	0,123	638	0,080	565	0,303	1091	0,054	1376	0,072	1029	0,048
	209	0,309	930	0,050	1241	0,043	109	0,284	960	0,049	1924	0,052	1093	0,040
	26649	0,356	1138	0,043	2961	0,040	379	0,263	2484	0,047	1425	0,042	1318	0,039
	450	0,127	1007	0,040	2336	0,039	2192	0,201	1608	0,046	1684	0,040	1268	0,034
	556	0,100	1896	0,033	1588	0,036	238	0,191	1093	0,044	4807	0,028	3308	0,034
	678	0,091	2968	0,036	3299	0,070	454	0,156	2376	0,079	2991	0,023	2190	0,030
	933	0,071	1637	0,021	3503	0,029	666	0,135	1263	0,037	1118	0,064	2018	0,031
	3158	0,047	2348	0,015	2120	0,030	814	0,066	2303	0,027	3334	0,014	2263	0,025
	16815	0,005	4293	0,018	3492	0,013	693	0,018	2538	0,020	6568	0,051	5526	0,056
	12811	0,051	9493	0,038	4285	0,007	14196	0,011	2651	0,011	5755	0,005	2929	0,011
	0,398	0,432	**0,386**	0,263	**0,378**	0,184	**0,355**	0,366	**0,361**	0,151	**0,392**	0,299	**0,38**	0,182

B.2 Resultados métodos particionales para 30 *clusters*

Tabla 18. Resultados de los métodos particionales, obteniendo 30 *clusters* (caso B).

	i1		i2		e1		g1		g1p		h1		h2	
	Size	**ISim**	**Size**	**ISim**	**Size**	**ISim**	**Size**	**ISim**	**Size**	**ISim**	**Size**	**ISim**	**Size**	**ISim**
rb	574	1,000	523	0,968	3861	0,998	95244	3,890	3940	0,963	752	0,989	3866	0,995
	152	0,989	623	0,875	3856	1,000	1	1,000	3857	0,999	994	1,000	3856	1,000
	752	0,989	3858	0,999	3859	0,999	4	1,000	3857	1,000	521	0,975	3859	0,999
	994	1,000	3860	0,998	3844	0,999	1	1,000	3819	1,000	183	0,921	3844	0,999
	521	0,975	3870	0,994	3857	0,999	1	1,000	3809	0,998	644	0,877	3858	0,999
	183	0,921	3865	0,990	3857	1,000	1	1,000	3874	0,994	300	0,848	3857	1,000
	702	0,884	3859	0,999	3820	0,999	1	1,000	4101	0,907	3857	1,000	3824	0,998
	270	0,872	3860	0,998	3830	0,993	1	1,000	4394	0,777	3856	1,000	3906	0,980
	3857	1,000	3877	0,963	3873	0,994	2	1,000	4414	0,774	3859	0,999	3937	0,960
	3856	1,000	3879	0,988	4327	0,813	1	1,000	5040	0,605	3844	0,999	1454	0,590
	3859	0,999	3860	0,983	2317	0,408	1	1,000	3414	0,201	3804	0,997	889	0,528
	3843	1,000	370	0,665	2122	0,338	1	1,000	25860	0,307	3858	0,999	4487	0,752
	3798	0,999	1346	0,574	866	0,389	2	1,000	3105	0,276	3857	1,000	1387	0,537
	3857	0,999	1502	0,557	20863	0,367	1	1,000	3390	0,254	3820	0,999	908	0,449
	3857	1,000	29798	0,319	2770	0,339	1	1,000	3668	0,104	3852	1,000	2172	0,210
	3819	1,000	423	0,464	827	0,292	2	1,000	3170	0,110	4001	0,938	2534	0,378
	3852	1,000	5147	0,606	3006	0,257	2	1,000	366	0,106	354	0,640	24323	0,332
	361	0,700	481	0,360	1025	0,210	31	1,000	2720	0,080	345	0,587	3360	0,266
	3877	0,976	354	0,329	6028	0,045	4	1,000	927	0,159	587	0,592	1408	0,116
	351	0,647	604	0,281	1361	0,146	3	1,000	1290	0,071	1269	0,450	1242	0,103
	581	0,599	2416	0,175	3326	0,033	1	1,000	377	0,074	454	0,418	747	0,166
	411	0,467	614	0,124	4079	0,029	2	1,000	1980	0,048	1966	0,456	1316	0,052
	1438	0,505	1190	0,116	2634	0,056	2	1,000	1643	0,045	17788	0,426	3486	0,033
	13534	0,474	813	0,065	2971	0,027	152	0,989	2397	0,039	731	0,221	2668	0,027
	417	0,251	2397	0,038	1111	0,063	75	0,961	1320	0,032	2343	0,240	3048	0,031
	653	0,156	1818	0,023	7160	0,005	276	0,832	2621	0,038	8466	0,146	3097	0,025
	2909	0,247	4682	0,022	1919	0,032	5704	0,120	974	0,082	4180	0,033	2188	0,010
	11112	0,218	3104	0,019	1788	0,055	3677	0,101	1869	0,062	7214	0,015	4733	0,041
	20617	0,007	8793	0,026	1171	0,048	28	0,064	1688	0,048	10196	0,006	2660	0,005
	11748	0,041	4969	0,054	427	0,037	1533	0,021	2871	0,013	8860	0,040	3841	0,046
	0,511	0,731	**0,488**	0,519	**0,48**	0,432	**3,486**	0,966	**0,458**	**0,372**	**0,508**	0,66	**0,487**	0,454

	i1		i2		e1		g1		g1p		h1		h2	
	Size	**ISim**	**Size**	**ISim**	**Size**	**ISim**	**Size**	**ISim**	**Size**	**ISim**	**Size**	**ISim**	**Size**	**ISim**
rbr	574	1,000	522	0,972	3861	0,998	96985	2,295	4325	0,825	748	0,995	3921	0,972
	152	0,989	3866	0,995	3871	0,995	1	1,000	4323	0,825	994	1,000	3862	0,998
	748	0,995	663	0,787	3860	0,994	4	1,000	4340	0,823	521	0,975	3872	0,995
	994	1,000	3880	0,990	3833	0,987	1	1,000	4337	0,816	183	0,921	3816	0,992
	521	0,975	3864	0,990	3859	0,999	1	1,000	4385	0,814	644	0,877	3862	0,992
	655	0,938	3941	0,964	3857	1,000	1	1,000	4597	0,734	271	0,868	3857	1,000
	183	0,921	3860	0,998	3823	0,999	1	1,000	4553	0,730	298	0,856	3860	0,998
	268	0,879	3859	0,999	3852	1,000	1	1,000	4709	0,706	3857	1,000	3853	0,999
	3857	1,000	3883	0,961	3827	0,994	2	1,000	4726	0,692	3860	0,999	3838	0,993
	3857	1,000	3860	0,983	4256	0,845	1	1,000	5142	0,585	3865	0,998	3932	0,960
	3861	0,999	3898	0,980	20473	0,368	1	1,000	21027	0,378	3843	1,000	2010	0,263
	3843	1,000	1301	0,614	2565	0,374	1	1,000	2497	0,387	3802	0,998	2611	0,369
	3798	0,999	1481	0,572	1152	0,325	2	1,000	720	0,289	3858	0,999	23765	0,341
	3857	0,999	4532	0,758	2888	0,294	1	1,000	2732	0,313	3857	1,000	2649	0,185
	356	0,712	450	0,485	920	0,273	1	1,000	2621	0,165	3820	0,999	1675	0,163
	3857	1,000	30143	0,315	867	0,250	2	1,000	4261	0,122	3853	0,999	3171	0,277
	3819	1,000	441	0,432	3090	0,153	2	1,000	926	0,121	3851	0,987	862	0,237
	3853	0,999	482	0,364	4528	0,107	31	1,000	407	0,109	16303	0,437	1773	0,131
	3826	0,990	444	0,233	1230	0,198	4	1,000	791	0,107	2526	0,381	1460	0,113
	347	0,655	733	0,206	554	0,109	3	1,000	3095	0,068	1033	0,208	3127	0,104
	577	0,604	2366	0,182	554	0,096	1	1,000	1179	0,089	731	0,280	1370	0,066
	359	0,538	732	0,107	840	0,087	2	1,000	1012	0,062	2892	0,255	2009	0,047
	12893	0,481	1235	0,108	6454	0,041	2	1,000	1611	0,052	903	0,117	3236	0,034
	2215	0,420	1178	0,043	1557	0,075	152	0,989	5683	0,046	2925	0,103	3169	0,030
	886	0,144	2358	0,039	1549	0,071	74	0,978	1974	0,049	8448	0,171	2171	0,063
	1272	0,112	2814	0,023	884	0,062	271	0,855	1856	0,050	1721	0,052	2442	0,031
	3199	0,249	2349	0,016	2441	0,033	2858	0,143	1734	0,042	5548	0,025	2655	0,066
	11583	0,212	5212	0,021	3641	0,029	5674	0,121	1582	0,027	6017	0,017	3242	0,020
	19719	0,007	7845	0,033	3811	0,027	21	0,098	2325	0,019	6479	0,051	1954	0,012
	10826	0,040	4563	0,053	7858	0,007	654	0,081	3285	0,011	9104	0,006	2731	0,009
	0,512	0,729	**0,49**	0,507	**0,474**	0,426	**2,101**	0,919	**0,435**	0,335	**0,503**	0,619	**0,481**	0,415

	i1		i2		e1		g1		g1p		h1		h2	
	Size	**ISim**	**Size**	**ISim**	**Size**	**ISim**	**Size**	**ISim**	**Size**	**ISim**	**Size**	**ISim**	**Size**	**ISim**
direct	98	1,000	1070	0,874	3872	0,994	98	1,000	15526	0,712	574	1,000	3928	0,970
	574	1,000	3860	0,998	4147	0,889	170	0,990	28220	0,341	994	1,000	3905	0,981
	994	1,000	3885	0,986	15428	0,719	562	0,957	15558	0,703	3857	1,000	15433	0,719
	695	0,893	836	0,587	15277	0,719	267	0,880	4993	0,656	358	0,707	15384	0,714
	3857	1,000	15444	0,718	25230	0,355	301	0,841	5626	0,509	3853	0,999	27307	0,336
	122	0,771	31315	0,304	2237	0,229	662	0,838	2465	0,182	1854	0,571	1946	0,279
	356	0,712	16093	0,665	1473	0,184	1118	0,805	1756	0,146	389	0,552	1441	0,198
	3853	0,999	470	0,434	3847	0,144	395	0,532	3989	0,136	15430	0,719	3382	0,182
	347	0,655	2428	0,375	791	0,153	1968	0,521	869	0,131	15386	0,714	791	0,154
	1844	0,575	497	0,365	3563	0,251	14181	0,699	547	0,165	442	0,474	918	0,132
	15428	0,719	2325	0,187	1983	0,132	149	0,427	2847	0,109	21018	0,404	3031	0,112
	15304	0,718	767	0,186	1064	0,101	11752	0,628	485	0,105	1847	0,230	1028	0,107
	439	0,478	417	0,116	1116	0,098	14309	0,639	312	0,092	571	0,111	2941	0,225
	285	0,398	1211	0,114	841	0,095	191	0,394	1188	0,079	436	0,102	1352	0,080
	210	0,333	658	0,091	523	0,097	289	0,357	1234	0,070	842	0,092	1531	0,075
	207	0,311	540	0,092	317	0,087	523	0,330	1075	0,072	584	0,086	566	0,089
	16256	0,454	1020	0,090	703	0,081	499	0,209	1359	0,065	1646	0,064	1259	0,073
	275	0,269	428	0,075	1173	0,074	15689	0,359	1493	0,091	373	0,073	1387	0,063
	132	0,258	1183	0,077	1635	0,065	350	0,191	856	0,055	842	0,065	678	0,054
	469	0,249	691	0,059	1294	0,064	246	0,194	891	0,054	1348	0,061	507	0,058
	280	0,196	694	0,048	1297	0,043	267	0,147	2039	0,052	9138	0,168	799	0,048
	379	0,180	1246	0,039	2332	0,079	127	0,122	1195	0,052	876	0,048	938	0,048
	594	0,170	1442	0,035	1936	0,041	1084	0,123	855	0,050	1357	0,056	1297	0,047
	448	0,124	1725	0,039	1136	0,041	465	0,116	1497	0,049	928	0,090	1572	0,045
	655	0,102	1396	0,047	1434	0,040	1899	0,077	1753	0,041	2262	0,037	2069	0,039
	448	0,071	1894	0,037	1990	0,036	2101	0,036	731	0,042	2083	0,024	4516	0,061
	13519	0,200	1233	0,031	2679	0,027	24761	0,136	2288	0,032	2863	0,020	2182	0,020
	5519	0,023	1093	0,023	1294	0,026	680	0,018	2054	0,021	3315	0,013	1145	0,019
	12581	0,006	2525	0,014	2853	0,013	1402	0,018	1196	0,017	5590	0,054	1522	0,021
	10587	0,040	8369	0,041	3290	0,007	10250	0,013	1858	0,012	5699	0,008	2000	0,012
	0,426	0,463	**0,409**	0,258	**0,399**	0,196	**0,381**	0,42	**0,381**	0,161	**0,42**	0,318	**0,402**	0,199

C Características descriptivas (*features*) para el caso de estudio B

C.1 Características descriptivas para 10 *clusters*

Tabla 19. Estudio de características descriptivas para 10 *clusters*, caso B.

	i1	i2	h1
rb	Cluster 0, Size: 3857, ISim: 1.000, ESim: 0.212 100.00% maquinas equipamiento mejoras ambientales herramientas	Cluster 0, Size: 3858, ISim: 0.999, ESim: 0.212 99.98% maquinas equipamiento mejoras ambientales herramientas	Cluster 0, Size: 3857, ISim: 1.000, ESim: 0.212 100.00% maquinas equipamiento mejoras ambientales herramientas
	Cluster 1, Size: 3856, ISim: 1.000, ESim: 0.246 100.00% periodicas especificas publicaciones encontraron registros	Cluster 1, Size: 3860, ISim: 0.998, ESim: 0.246 99.92% periodicas especificas publicaciones encontraron registros	Cluster 1, Size: 3856, ISim: 1.000, ESim: 0.246 100.00% periodicas especificas publicaciones encontraron registros
	Cluster 2, Size: 3859, ISim: 0.999, ESim: 0.251 99.97% instrumentos resultados calidad encontraron registros	Cluster 2, Size: 3865, ISim: 0.990, ESim: 0.261 99.98% domesticas tareas pagina informacion sid	Cluster 2, Size: 3859, ISim: 0.999, ESim: 0.251 99.97% instrumentos resultados calidad encontraron registros
	Cluster 3, Size: 3843, ISim: 1.000, ESim: 0.262 100.00% domesticas tareas pagina informacion sid	Cluster 3, Size: 3877, ISim: 0.963, ESim: 0.257 99.50% diplomaturas asignaturas pagina informacion sid	Cluster 3, Size: 3844, ISim: 0.999, ESim: 0.262 100.00% domesticas tareas pagina informacion sid
	Cluster 4, Size: 3798, ISim: 0.999, ESim: 0.261 100.00% diplomaturas asignaturas pagina informacion sid	Cluster 4, Size: 11589, ISim: 0.775, ESim: 0.245 99.89% encontraron registros pagina informacion sid	Cluster 4, Size: 3804, ISim: 0.997, ESim: 0.261 100.00% diplomaturas asignaturas pagina informacion sid
	Cluster 5, Size: 3852, ISim: 1.000, ESim: 0.296 100.00% internacionales pagina informacion sid discapacidad	Cluster 5, Size: 34767, ISim: 0.262, ESim: -0.171 99.74% informacion sid revistas correo discapacidad	Cluster 5, Size: 3852, ISim: 1.000, ESim: 0.296 100.00% internacionales pagina informacion sid discapacidad
	Cluster 6, Size: 7714, ISim: 0.869, ESim: 0.272 87.49% encontraron registros pagina incentivos 87.49% encontraron registros pagina evaluadores	Cluster 6, Size: 12886, ISim: 0.667, ESim: 0.242 85.22% pagina informacion sid discapacidad alizacion	Cluster 6, Size: 7715, ISim: 0.869, ESim: 0.272 87.48% encontraron registros pagina incentivos 87.48% encontraron registros pagina evaluadores
	Cluster 7, Size: 7696, ISim: 0.846, ESim: 0.278 87.14% pagina convocatorias informacion sid 87.18% pagina alizacion informacion sid	Cluster 7, Size: 6749, ISim: 0.097, ESim: 0.005 32.06% found moved server document apache	Cluster 7, Size: 7821, ISim: 0.831, ESim: 0.276 87.05% pagina convocatorias informacion sid 87.08% pagina alizacion informacion sid
		Cluster 8, Size: 6328, ISim: 0.042, ESim: 0.001 5.58% explorador 2.40% lamin	Cluster 8, Size: 31017, ISim: 0.308, ESim: -0.158

	i1	i2	h1
	Cluster 8, Size: 28993, ISim: 0.333, ESim: -0.149 99.96% informacion sid revistas correo discapacidad Cluster 9, Size: 39287, ISim: 0.012, ESim: -0.040 22.08% personas 5.98% found moved server 8.76% slide	4.44% arriba 36.38% slide page Cluster 9, Size: 18976, ISim: 0.014, ESim: 0.013 21.85% personas quot espa social curso	99.53% informacion sid revistas correo discapacidad Cluster 9, Size: 37130, ISim: 0.012, ESim: -0.028 21.92% personas 6.33% found moved server 9.27% slide
rbr	Cluster 0, Size: 3857, ISim: 1.000, ESim: 0.212 100.00% maquinas equipamiento mejoras ambientales herramientas Cluster 1, Size: 3857, ISim: 1.000, ESim: 0.246 99.98% periodicas especificas publicaciones encontraron registros Cluster 2, Size: 3860, ISim: 0.999, ESim: 0.251 99.95% instrumentos resultados calidad encontraron registros Cluster 3, Size: 3843, ISim: 1.000, ESim: 0.262 100.00% domesticas tareas pagina informacion sid Cluster 4, Size: 3798, ISim: 0.999, ESim: 0.261 100.00% diplomaturas asignaturas pagina informacion sid Cluster 5, Size: 3853, ISim: 0.999, ESim: 0.296 99.98% internacionales pagina informacion sid discapacidad Cluster 6, Size: 7714, ISim: 0.869, ESim: 0.272 87.49% encontraron registros pagina incentivos 87.49% encontraron registros pagina evaluadores Cluster 7, Size: 7662, ISim: 0.850, ESim: 0.279 87.24% pagina convocatorias informacion sid 87.28% pagina alizacion informacion sid Cluster 8, Size: 29002, ISim: 0.333, ESim: -0.150 99.95% informacion sid revistas correo discapacidad	Cluster 0, Size: 3883, ISim: 0.987, ESim: 0.210 99.46% maquinas equipamiento mejoras ambientales herramientas Cluster 1, Size: 3891, ISim: 0.984, ESim: 0.244 99.30% periodicas especificas publicaciones encontraron registros Cluster 2, Size: 3864, ISim: 0.990, ESim: 0.261 99.96% domesticas tareas pagina informacion sid Cluster 3, Size: 3915, ISim: 0.947, ESim: 0.255 98.80% diplomaturas asignaturas pagina informacion sid Cluster 4, Size: 11613, ISim: 0.773, ESim: 0.244 99.70% encontraron registros pagina informacion sid Cluster 5, Size: 12547, ISim: 0.699, ESim: 0.246 84.99% pagina informacion sid discapacidad alizacion Cluster 6, Size: 32737, ISim: 0.286, ESim: -0.164 99.79% informacion sid revistas correo discapacidad Cluster 7, Size: 6675, ISim: 0.099, ESim: 0.005 32.38% found moved server document apache Cluster 8, Size: 6429, ISim: 0.041, ESim: 0.001 5.49% explorador 2.36% lamin 4.39% arriba 35.99% slide page	Cluster 0, Size: 3857, ISim: 1.000, ESim: 0.212 100.00% maquinas equipamiento mejoras ambientales herramientas Cluster 1, Size: 3860, ISim: 0.999, ESim: 0.246 99.92% periodicas especificas publicaciones encontraron registros Cluster 2, Size: 3861, ISim: 0.999, ESim: 0.251 99.93% instrumentos resultados calidad encontraron registros Cluster 3, Size: 3843, ISim: 1.000, ESim: 0.262 100.00% domesticas tareas pagina informacion sid Cluster 4, Size: 3804, ISim: 0.997, ESim: 0.261 100.00% diplomaturas asignaturas pagina informacion sid Cluster 5, Size: 3853, ISim: 0.999, ESim: 0.296 99.98% internacionales pagina informacion sid discapacidad Cluster 6, Size: 7715, ISim: 0.869, ESim: 0.272 87.48% encontraron registros pagina incentivos 87.49% encontraron registros pagina evaluadores Cluster 7, Size: 7728, ISim: 0.843, ESim: 0.278 87.25% pagina convocatorias informacion sid 87.28% pagina alizacion informacion sid Cluster 8, Size: 35298, ISim: 0.257, ESim: -0.174 99.16% informacion sid revistas correo discapacidad

	i1	i2	h1
	Cluster 9, Size: 39309, ISim: 0.012, ESim: -0.040 22.08% personas 5.98% found moved server 8.76% slide	Cluster 9, Size: 21201, ISim: 0.015, ESim: 0.018 20.26% quot espa social curso 24.89% personas quot espa social	Cluster 9, Size: 32936, ISim: 0.011, ESim: -0.014 6.95% found moved server document 10.45% slide

C.2 Características descriptivas para 20 *clusters*

Tabla 20. Estudio de características descriptivas para 20 *clusters*, caso B.

	i1	i2	h1
r b	Cluster 0, Size: 574, ISim: 1.000, ESim: 0.007 100.00% inexistente consulte solicitada encontrado servidor Cluster 1, Size: 752, ISim: 0.989, ESim: 0.008 99.79% found moved port apache document Cluster 2, Size: 994, ISim: 1.000, ESim: 0.024 100.00% haya encuentra ndash quitado escribio Cluster 3, Size: 521, ISim: 0.975, ESim: 0.005 99.69% moved permanently port apache document Cluster 4, Size: 702, ISim: 0.884, ESim: 0.004 29.72% slide ppt graphic view previous Cluster 5, Size: 3857, ISim: 1.000, ESim: 0.212 100.00% maquinas equipamiento mejoras ambientales herramientas Cluster 6, Size: 3856, ISim: 1.000, ESim: 0.246 100.00% periodicas especificas publicaciones encontraron registros Cluster 7, Size: 3859, ISim: 0.999, ESim: 0.251 99.97% instrumentos resultados calidad encontraron registros	Cluster 0, Size: 3858, ISim: 0.999, ESim: 0.212 99.98% maquinas equipamiento mejoras ambientales herramientas Cluster 1, Size: 3860, ISim: 0.998, ESim: 0.246 99.92% periodicas especificas publicaciones encontraron registros Cluster 2, Size: 3870, ISim: 0.994, ESim: 0.251 99.77% instrumentos resultados calidad encontraron registros Cluster 3, Size: 3865, ISim: 0.990, ESim: 0.261 99.98% domesticas tareas pagina informacion sid Cluster 4, Size: 3859, ISim: 0.999, ESim: 0.289 99.94% evaluadores encontraron registros pagina informacion Cluster 5, Size: 3860, ISim: 0.998, ESim: 0.290 99.94% incentivos encontraron registros pagina informacion Cluster 6, Size: 3877, ISim: 0.963, ESim: 0.257 99.50% diplomaturas asignaturas pagina informacion sid Cluster 7, Size: 3879, ISim: 0.988, ESim: 0.295	Cluster 0, Size: 994, ISim: 1.000, ESim: 0.024 100.00% haya encuentra ndash quitado escribio Cluster 1, Size: 521, ISim: 0.975, ESim: 0.005 99.69% moved permanently port apache document Cluster 2, Size: 644, ISim: 0.877, ESim: 0.007 93.48% encontrado consulte inexistente solicitada error Cluster 3, Size: 3857, ISim: 1.000, ESim: 0.212 100.00% maquinas equipamiento mejoras ambientales herramientas Cluster 4, Size: 3856, ISim: 1.000, ESim: 0.246 100.00% periodicas especificas publicaciones encontraron registros Cluster 5, Size: 3859, ISim: 0.999, ESim: 0.251 99.97% instrumentos resultados calidad encontraron registros Cluster 6, Size: 3844, ISim: 0.999, ESim: 0.262 100.00% domesticas tareas pagina informacion sid Cluster 7, Size: 3804, ISim: 0.997, ESim: 0.261

	i1	i2	h1
	Cluster 8, Size: 3843, ISim: 1.000, ESim: 0.262 100.00% domesticas tareas pagina informacion sid Cluster 9, Size: 3798, ISim: 0.999, ESim: 0.261 100.00% diplomaturas asignaturas pagina informacion sid Cluster 10, Size: 3857, ISim: 0.999, ESim: 0.289 99.98% evaluadores encontraron registros pagina informacion Cluster 11, Size: 3857, ISim: 1.000, ESim: 0.290 100.00% incentivos encontraron registros pagina informacion Cluster 12, Size: 3819, ISim: 1.000, ESim: 0.295 100.00% alizacion pagina informacion sid discapacidad Cluster 13, Size: 3852, ISim: 1.000, ESim: 0.296 100.00% internacionales pagina informacion sid discapacidad Cluster 14, Size: 3877, ISim: 0.976, ESim: 0.293 99.20% convocatorias pagina informacion sid discapacidad Cluster 15, Size: 581, ISim: 0.599, ESim: 0.006 99.28% found server requested url port Cluster 16, Size: 14972, ISim: 0.467, ESim: 0.184 99.97% informacion sid revistas correo discapacidad Cluster 17, Size: 14021, ISim: 0.217, ESim: 0.142 99.95% informacion sid correo revistas discapacidad Cluster 18, Size: 23415, ISim: 0.007, ESim: 0.005 5.56% page untitled 5.20% quot arriba 7.89% quot page	99.81% internacionales pagina informacion sid discapacidad Cluster 8, Size: 3860, ISim: 0.983, ESim: 0.293 99.96% alizacion pagina informacion sid discapacidad Cluster 9, Size: 1346, ISim: 0.574, ESim: 0.018 80.79% haya encuentra link visited file Cluster 10, Size: 2025, ISim: 0.497, ESim: 0.002 86.96% moved server apache port 88.36% found server apache port Cluster 11, Size: 34767, ISim: 0.262, ESim: -0.171 99.74% informacion sid revistas correo discapacidad Cluster 12, Size: 977, ISim: 0.419, ESim: 0.005 62.66% documento consulte inexistente solicitada encontrado Cluster 13, Size: 5147, ISim: 0.606, ESim: 0.237 93.13% convocatorias pagina informacion sid discapacidad Cluster 14, Size: 984, ISim: 0.145, ESim: 0.001 20.73% error 9.96% authorized 32.42% conceptos 33.38% archivo ausente Cluster 15, Size: 3606, ISim: 0.108, ESim: 0.001 59.37% slide graphic 46.90% slide page overview continue Cluster 16, Size: 1417, ISim: 0.073, ESim: 0.001 34.82% gtk widget void 41.18% untitled document Cluster 17, Size: 2722, ISim: 0.033, ESim: 0.002 13.10% explorador admite marcos 5.58% lamin 10.18% arriba	100.00% diplomaturas asignaturas pagina informacion sid Cluster 8, Size: 3858, ISim: 0.999, ESim: 0.289 99.97% evaluadores encontraron registros pagina informacion Cluster 9, Size: 3857, ISim: 1.000, ESim: 0.290 100.00% incentivos encontraron registros pagina informacion Cluster 10, Size: 3820, ISim: 0.999, ESim: 0.295 100.00% alizacion pagina informacion sid discapacidad Cluster 11, Size: 3852, ISim: 1.000, ESim: 0.296 100.00% internacionales pagina informacion sid discapacidad Cluster 12, Size: 1339, ISim: 0.670, ESim: 0.004 90.58% found server port apache moved Cluster 13, Size: 4001, ISim: 0.938, ESim: 0.289 98.85% convocatorias pagina informacion sid discapacidad Cluster 14, Size: 1269, ISim: 0.450, ESim: 0.002 56.74% slide graphic view previous version Cluster 15, Size: 19754, ISim: 0.419, ESim: 0.165 99.97% informacion sid revistas correo discapacidad Cluster 16, Size: 11263, ISim: 0.159, ESim: 0.127 98.75% informacion sid correo revistas discapacidad Cluster 17, Size: 4180, ISim: 0.033, ESim: 0.012 26.88% quot derecho trabajo 30.80% educacion quot alumnos trabajo Cluster 18, Size: 19323, ISim: 0.006, ESim: 0.004 4.51% arriba quot 4.25% untitled

	i1	i2	h1
	9.55% slide page Cluster 19, Size: 11748, ISim: 0.041, ESim: 0.052 78.55% personas informacion sid discapacidad social	Cluster 18, Size: 10183, ISim: 0.015, ESim: 0.005 17.17% quot curso facultad derecho ciencias Cluster 19, Size: 8793, ISim: 0.026, ESim: 0.022 38.38% personas social espa empleo trabajo	9.92% page slide 7.74% page quot Cluster 19, Size: 8860, ISim: 0.040, ESim: 0.044 82.23% personas informacion social sid discapacidad
r b r	Cluster 0, Size: 574, ISim: 1.000, ESim: 0.007 100.00% inexistente consulte solicitada encontrado servidor Cluster 1, Size: 748, ISim: 0.995, ESim: 0.008 99.57% found moved port apache www Cluster 2, Size: 994, ISim: 1.000, ESim: 0.024 100.00% haya encuentra ndash quitado escribio Cluster 3, Size: 521, ISim: 0.975, ESim: 0.005 99.69% moved permanently port apache document Cluster 4, Size: 700, ISim: 0.887, ESim: 0.004 29.57% slide ppt graphic view previous Cluster 5, Size: 3857, ISim: 1.000, ESim: 0.212 100.00% maquinas equipamiento mejoras ambientales herramientas Cluster 6, Size: 3857, ISim: 1.000, ESim: 0.246 99.98% periodicas especificas publicaciones encontraron registros Cluster 7, Size: 3861, ISim: 0.999, ESim: 0.251 99.93% instrumentos resultados calidad encontraron registros Cluster 8, Size: 3843, ISim: 1.000, ESim: 0.262 100.00% domesticas tareas pagina informacion sid Cluster 9, Size: 3798, ISim: 0.999, ESim: 0.261 100.00% diplomaturas asignaturas pagina informacion sid	Cluster 0, Size: 3869, ISim: 0.994, ESim: 0.211 99.75% maquinas equipamiento mejoras ambientales herramientas Cluster 1, Size: 3883, ISim: 0.988, ESim: 0.244 99.46% periodicas especificas publicaciones encontraron registros Cluster 2, Size: 3864, ISim: 0.990, ESim: 0.261 99.96% domesticas tareas pagina informacion sid Cluster 3, Size: 3958, ISim: 0.956, ESim: 0.246 98.02% instrumentos resultados calidad encontraron registros Cluster 4, Size: 3861, ISim: 0.998, ESim: 0.289 99.94% evaluadores encontraron registros pagina informacion Cluster 5, Size: 3861, ISim: 0.998, ESim: 0.290 99.93% incentivos encontraron registros pagina informacion Cluster 6, Size: 3889, ISim: 0.958, ESim: 0.256 99.26% diplomaturas asignaturas pagina informacion sid Cluster 7, Size: 3861, ISim: 0.982, ESim: 0.293 99.95% alizacion pagina informacion sid discapacidad Cluster 8, Size: 3894, ISim: 0.982, ESim: 0.294 99.47% internacionales pagina informacion sid discapacidad Cluster 9, Size: 1384, ISim: 0.547, ESim: 0.018	Cluster 0, Size: 994, ISim: 1.000, ESim: 0.024 100.00% haya encuentra ndash quitado escribio Cluster 1, Size: 586, ISim: 0.978, ESim: 0.007 98.36% consulte inexistente solicitada encontrado servidor Cluster 2, Size: 521, ISim: 0.975, ESim: 0.005 99.69% moved permanently port apache document Cluster 3, Size: 3857, ISim: 1.000, ESim: 0.212 100.00% maquinas equipamiento mejoras ambientales herramientas Cluster 4, Size: 3860, ISim: 0.999, ESim: 0.246 99.92% periodicas especificas publicaciones encontraron registros Cluster 5, Size: 3865, ISim: 0.998, ESim: 0.251 99.84% instrumentos resultados calidad encontraron registros Cluster 6, Size: 3843, ISim: 1.000, ESim: 0.262 100.00% domesticas tareas pagina informacion sid Cluster 7, Size: 3802, ISim: 0.998, ESim: 0.261 100.00% diplomaturas asignaturas pagina informacion sid Cluster 8, Size: 3859, ISim: 0.999, ESim: 0.289 99.96% evaluadores encontraron registros pagina informacion Cluster 9, Size: 3857, ISim: 1.000, ESim: 0.290

	i1	i2	h1
	Cluster 10, Size: 3857, ISim: 0.999, ESim: 0.289 99.98% evaluadores encontraron registros pagina informacion Cluster 11, Size: 3857, ISim: 1.000, ESim: 0.290 100.00% incentivos encontraron registros pagina informacion Cluster 12, Size: 3819, ISim: 1.000, ESim: 0.295 100.00% alizacion pagina informacion sid discapacidad Cluster 13, Size: 3853, ISim: 0.999, ESim: 0.296 99.98% internacionales pagina informacion sid discapacidad Cluster 14, Size: 3826, ISim: 0.990, ESim: 0.295 99.56% convocatorias pagina informacion sid discapacidad Cluster 15, Size: 577, ISim: 0.604, ESim: 0.006 99.38% found server requested url port Cluster 16, Size: 15252, ISim: 0.464, ESim: 0.183 99.97% informacion sid revistas correo discapacidad Cluster 17, Size: 14483, ISim: 0.207, ESim: 0.140 99.86% informacion sid correo revistas discapacidad Cluster 18, Size: 23258, ISim: 0.007, ESim: 0.005 3.72% untitled 5.14% quot arriba 7.85% quot page 9.62% slide page Cluster 19, Size: 11220, ISim: 0.040, ESim: 0.048 78.88% personas informacion social sid discapacidad	78.21% haya encuentra link visited ndash Cluster 10, Size: 2018, ISim: 0.501, ESim: 0.002 87.15% moved server apache port 88.48% found server apache port Cluster 11, Size: 4586, ISim: 0.742, ESim: 0.259 94.84% convocatorias pagina informacion sid discapacidad Cluster 12, Size: 31684, ISim: 0.299, ESim: -0.160 99.76% informacion sid revistas correo discapacidad Cluster 13, Size: 1061, ISim: 0.365, ESim: 0.005 58.70% documento consulte inexistente solicitada servidor Cluster 14, Size: 3503, ISim: 0.113, ESim: 0.001 60.62% slide graphic 46.92% slide page overview continue Cluster 15, Size: 1194, ISim: 0.113, ESim: 0.002 8.29% authorized 38.53% conceptos 25.75% archivo error 30.90% archivo ausente Cluster 16, Size: 1957, ISim: 0.045, ESim: 0.001 27.87% gtk widget void 34.75% untitled document Cluster 17, Size: 3008, ISim: 0.028, ESim: 0.002 11.90% explorador admite marcos 5.05% lamin 9.54% arriba Cluster 18, Size: 10636, ISim: 0.015, ESim: 0.005 17.17% quot curso facultad derecho ciencias Cluster 19, Size: 10784, ISim: 0.032, ESim: 0.034 50.73% personas social espa informacion empleo	100.00% incentivos encontraron registros pagina informacion Cluster 10, Size: 3820, ISim: 0.999, ESim: 0.295 100.00% alizacion pagina informacion sid discapacidad Cluster 11, Size: 3853, ISim: 0.999, ESim: 0.296 99.98% internacionales pagina informacion sid discapacidad Cluster 12, Size: 3863, ISim: 0.983, ESim: 0.294 99.66% convocatorias pagina informacion sid discapacidad Cluster 13, Size: 20253, ISim: 0.412, ESim: 0.163 99.94% informacion sid revistas correo discapacidad Cluster 14, Size: 2603, ISim: 0.190, ESim: 0.037 47.22% found server port apache moved Cluster 15, Size: 3418, ISim: 0.078, ESim: 0.021 29.02% slide graphic view previous version Cluster 16, Size: 10353, ISim: 0.167, ESim: 0.129 99.83% informacion sid correo revistas discapacidad Cluster 17, Size: 10167, ISim: 0.018, ESim: 0.006 18.49% quot curso facultad derecho ciencias Cluster 18, Size: 7217, ISim: 0.049, ESim: 0.043 61.36% personas social empleo informacion espa Cluster 19, Size: 12164, ISim: 0.007, ESim: 0.002 2.70% arriba 5.78% untitled document 8.70% page archivo

D *Scripts* para el procesamiento de los documentos y obtención del documento XML final.

D.1 *Script* para el filtrado de páginas previo.pl

```
#!/usr/bin/perl
$documento=$ARGV[0];
open(F,$documento) or die "No se pudo abrir $documento\n";
$tx="";
while($k=<F>)
        {
        $k=~s/if.*?{$/{/;           # Sustituye if condición y { por {
        $k=~s/if.*?$//;             # Elimina if y cualquier cosa sin llave al final
        $k=~s/\/\/.*?$//;           # Elimina cualquier comentario que empiece por //
        $k=~s/var .*?$//i;          # Elimina cualquier declaración de variable que empiece por var
        $k=~s/else.*?{$/{/i;        # Sustituye else cualquier cosa y { por {
        $k=~s/else.*?$//i;          # Elimina else seguido de cualquier cosa que no acabe en {
        $k=~s/function.*?{$/{/i;    # Sustituye function nombre y { por {
        $k=~s/function.*?$//i;      # Elimina function seguido de cualquier cosa que no acabe en {
        chomp($k);                  # Elimina el salto de línea
        $tx="$tx $k";}              # Concatena la línea procesada
close F;

$tx=~s/<script.*?\/script>//sgi;      # Elimina todo entre <script y /script>
$tx=~s/<DOCHDR.*?\/DOCHDR>//sgi;      # Elimina todo entre <DOCHDR y /DOCHDR>

$tx=~s/{.*?}//sg;                     # Elimina todo entre llave de apertura y de cierre
$tx=~s/<.*?>//sg;                     # Elimina etiquetas
$tx=~s/window\..*?=//sgi;

# normalizar caracteres
$tx=~s/ / /gs;
$tx=~s/&iexcl;/¡/gs;
$tx=~s/&cent;/¢/gs;
$tx=~s/&pound;/£/gs;
$tx=~s/&curren;/¤/gs;
$tx=~s/&yen;/¥/gs;
$tx=~s/&sect;/§/gs;
$tx=~s/&uml;/¨/gs;
$tx=~s/&copy;/©/gs;
$tx=~s/&ordf;/ª/gs;
$tx=~s/&laquo;/«/gs;
$tx=~s/&not;/¬/gs;
$tx=~s/&reg;/®/gs;
$tx=~s/&macr;/¯/gs;
$tx=~s/&deg;/°/gs;
$tx=~s/&plusmn;/±/gs;
$tx=~s/&acute;/´/gs;
$tx=~s/&micro;/µ/gs;
…
```

```
…
$tx=~s/&para;/¶/gs;
$tx=~s/&middot;/·/gs;
$tx=~s/&cedil;/¸/gs;
$tx=~s/&ordm;/º/gs;
$tx=~s/&raquo;/»/gs;
$tx=~s/&iquest;/¿/gs;
$tx=~s/&Agrave;/A/gs;
$tx=~s/&Aacute;/A/gs;
$tx=~s/&Acirc;/A/gs;
$tx=~s/&Atilde;/A/gs;
$tx=~s/&Auml;/A/gs;
$tx=~s/&Aring;/A/gs;
$tx=~s/&AElig;/A/gs;
$tx=~s/&Ccedil;/Ç/gs;
$tx=~s/&Egrave;/E/gs;
$tx=~s/&Eacute;/E/gs;
$tx=~s/&Ecirc;/E/gs;
$tx=~s/&Euml;/E/gs;
$tx=~s/&Igrave;/I/gs;
$tx=~s/&Iacute;/I/gs;
$tx=~s/&Icirc;/I/gs;
$tx=~s/&Iuml;/I/gs;
$tx=~s/&Ntilde;/Ñ/gs;
$tx=~s/&Ograve;/O/gs;
$tx=~s/&Oacute;/O/gs;
$tx=~s/&Ocirc;/O/gs;
$tx=~s/&Otilde;/O/gs;
$tx=~s/&Ouml;/O/gs;
$tx=~s/&Oslash;/O/gs;
$tx=~s/&Ugrave;/U/gs;
$tx=~s/&Uacute;/U/gs;
$tx=~s/&Ucirc;/U/gs;
$tx=~s/&Uuml;/U/gs;
$tx=~s/&szlig;/B/gs;
$tx=~s/&agrave;/a/gs;
$tx=~s/&aacute;/a/gs;
$tx=~s/&acirc;/a/gs;
$tx=~s/&atilde;/a/gs;
$tx=~s/&auml;/a/gs;
$tx=~s/&aring;/a/gs;
$tx=~s/&aelig;/a/gs;
$tx=~s/&ccedil;/ç/gs;
$tx=~s/&egrave;/e/gs;
$tx=~s/&eacute;/e/gs;
$tx=~s/&ecirc;/e/gs;
$tx=~s/&euml;/e/gs;
$tx=~s/&igrave;/i/gs;
$tx=~s/&iacute;/i/gs;
$tx=~s/&icirc;/i/gs;
...
```

```
…
$tx=~s/&iuml;/i/gs;
$tx=~s/&ntilde;/ñ/gs;
$tx=~s/&ograve;/o/gs;
$tx=~s/&oacute;/o/gs;
$tx=~s/&ocirc;/o/gs;
$tx=~s/&otilde;/o/gs;
$tx=~s/&ouml;/o/gs;
$tx=~s/&divide;/÷/gs;
$tx=~s/&oslash;/o/gs;
$tx=~s/&ugrave;/u/gs;
$tx=~s/&uacute;/u/gs;
$tx=~s/&ucirc;/u/gs;
$tx=~s/&uuml;/u/gs;
$tx=~s/&yuml;/y/gs;

# acentos que hubiera sin codificar
$tx=~y/áéíóúàèìòùâêîôûäëïöüÁÉÍÓÚÀÈÌÒÙÄËÏÖÜÂÊÎÔÛ/aeiouaeiouaeiouaeiouaeiouaeiouaeiouaei
ou/;

#pasar a minúsculas
$tx=lc($tx);
$tx=~s/Ñ/ñ/gs;
$tx=~s/Ç/ç/gs;

#sustituir separadores por blancos
$tx=~s/[^a-zñç]/ /gs;

#reducir los espacios en blanco
$tx=~s/ +/ /gs;

print $tx;
```

D.2 *Script* para la obtención de las etiquetas XML correspondientes a los *clusters*, para el tipo particional, plano (metercluster.pl).

```
#!/usr/bin/perl
$doc0=$ARGV[0];		# Fichero con las features de los documentos
$doc1=$ARGV[1];		# Fichero con la pertenencia de cada documento a su cluster
$ncl=$ARGV[2];		# Número de clusters

for ($i=0; $i<$ncl;$i++){
	$eltos[$i]="";
	$feat[$i]="";
	}

open(F0,$doc0) or die "No se pudo abrir $doc0\n";
$tx="<clusters>\n";
$cont=1;
$enc= 0;
$j= -1;

while($k=<F0>)  {		# Se procesan los datos para obtener las features de cada cluster
	if ($k =~ /^-/)
		{
		$enc=0;
		}
	if ($k =~ /^Cluster/)  {
		$enc=1;
		$j++;
		}
	elsif ($enc==1)    {
		chomp($k);
		$k=~s/.*%//;
		$k=~s/^ *//;
		$feat[$j]="$feat[$j] $k";
		}
	}			# Concatena la línea procesada
close F0;

				# Se obtienen los documentos de cada cluster
open(F1,$doc1) or die "No se pudo abrir $doc1\n";
while($k=<F1>)  {
	chomp($k);			# Elimina el salto de línea
	$eltos[$k]="$eltos[$k] $cont";
	$cont+=1;
	}			# Concatena la línea procesada
close F1;

				# Se añaden los datos obtenidos, generando las etiquetas XML, al fichero
de salida
for ($i=0; $i<$ncl; $i++)   {
	$tx= "$tx <cluster idcluster=\"c$i\">\n<features>$feat[$i]</features>\n<eltos>\n";
	$eltos[$i]=~s/^ //;
	@ident=split(/ /,$eltos[$i]);
	foreach$id(@ident) {
		$tx="$tx <elto idelto=\"d$id\"/> \n";
		}
	$tx= "$tx </eltos>\n";
	$tx= "$tx </cluster>\n";
	}
$tx="$tx \n</clusters>\n</IR>";
print $tx;
```

D.3 *Script* para la obtención de las etiquetas XML correspondientes a los *clusters*, para el tipo acumulativo, en árbol (metercluster2.pl).

```
#!/usr/bin/perl
$doc0=$ARGV[0];                 # Fichero con las features de los documentos
$doc1=$ARGV[1];                 # Fichero con la pertenencia de cada documento a su cluster
$doc2=$ARGV[2];                 # Fichero con la relación jerárquica entre clusters
$ncl=$ARGV[3];                  # Número de clusters

for ($i=0; $i<(2 * $ncl - 1);$i++)
        {
        $eltos[$i]="";                  # Cada posicion contendrá los documentos de un cluster
        $feat[$i]="";                   # Cada posicion contendrá las caracterísiticas de un
cluster
        $hijos[$i]="";                  # Cada posicion contendrá los hijos de un cluster
        $padre[$i]="";                  # Cada posicion contendrá el padre de un cluster
        }

open(F0,$doc0) or die "No se pudo abrir $doc0\n";
$tx="<clusters>\n";
$cont=1;
$enc= 0;
$j= -1;

                # Se procesan las características de cada cluster
while($k=<F0>)                  # Se recorre el fichero de features
        {

        if ($k =~ /^-/)
                {
                $enc=0;
                }
        if ($k =~ /^Cluster/)           # Localiza un cluster previo a su línea de features
                {
                $enc=1;
                $j++;
                }
        elsif ($enc==1)
                {
                chomp($k);
                $k=~s/.*%//;            # Elimina los valores numéricos de la línea de features
                $k=~s/^ *//;
                @ident=split(/ /,$k);
                foreach$id(@ident)
                        {
                        if ($feat[$j] !~ m/$id/) {    # Si un término aún no está en las características
                                $feat[$j]="$feat[$j] $id";
                                }
                        }
                }
        }                               # Concatena la línea procesada
close F0;
…
```

```
…
		# Se obtienen los documentos que contiene cada cluster
open(F1,$doc1) or die "No se pudo abrir $doc1\n";
while($k=<F1>)				# Recorre el fichero que contiene en cada línea
(documento) el número de su cluster
	{
	chomp($k);			# Elimina el salto de línea
	if ($k ne "-1") {
		$eltos[$k]="$eltos[$k] $cont";	# se mete en el cluster correspondiente un nuevo
elemento
	}
	$cont+=1;
	}				# Concatena la línea procesada
close F1;

		# Se obtiene el padre de cada cluster y sus hijos
$cont=0;
open(F2,$doc2) or die "No se pudo abrir $doc2\n";
while($k=<F2>)				# Recorre el fichero que contiene en cada línea (cluster)
el número de su padre
	{
	chomp($k);			# Elimina el salto de línea
	$k=~s/\s.*//;			# Elimina el resto de la línea a partir del primer blanco
	if ($k ne "-1") {
		$hijos[$k]="$hijos[$k] $cont";	# se mete en el cluster correspondiente un nuevo
hijo
	}
	$padre[$cont]=$k;		# se mete en el cluster correspondiente su padre
	$cont+=1;
	}				# Concatena la línea procesada
close F2;

		# Se da formato de salida a la información y se añaden las etiquetas XML
for ($i=0; $i<(2 * $ncl - 1); $i++)
	{
	$hijos[$i]=~s/^ //;
	if ($hijos[$i] ne "") {
		@ident=split(/ /,$hijos[$i]);
		foreach$id(@ident)
			{
			@identif=split(/ /,$feat[$id]);
			foreach$ide(@identif)
				{
				if ($feat[$i] !~ m/$ide/) {  # Si un término aún no está en las
características
					$feat[$i]="$feat[$i] $ide";
					}
				}
			}
	}

	$tx= "$tx <cluster idcluster=\"c$i\">\n<features>$feat[$i]</features>\n";

	$tx= "$tx <padre> $padre[$i] </padre>\n";

…
```

```
…


        $eltos[$i]=~s/^ //;
        if ($eltos[$i] ne "") {
                $tx= "$tx <eltos>\n";
                @ident=split(/ /,$eltos[$i]);
                foreach$id(@ident) {
                        $tx="$tx <elto idelto=\"d$id\"/> \n";
                        }
                $tx= "$tx </eltos>\n";
        }

        if ($hijos[$i] ne "") {
                $tx= "$tx <hijos>\n";
                @ident=split(/ /,$hijos[$i]);
                foreach$id(@ident)
                        {
                        $tx="$tx <hijo> $id </hijo> \n";
                        }
                $tx= "$tx </hijos>\n";
        }

        $tx= "$tx </cluster>\n";
        }


$tx="$tx \n</clusters>\n</IR>";

print $tx;
```

E Aplicaciones de transformación y presentación en JAVA.

E.1 Aplicación que transforma un formato XML representando la estructura jerárquica en un XML anidado.

```
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.parsers.ParserConfigurationException;

import javax.xml.parsers.DocumentBuilder;
import javax.xml.xpath.XPathConstants;
import javax.xml.xpath.XPathExpressionException;
import javax.xml.xpath.XPathFactory;

import org.w3c.dom.Document;
import org.w3c.dom.Element;
import org.w3c.dom.Node;
import org.w3c.dom.NodeList;

import com.sun.org.apache.xml.internal.serialize.LineSeparator;
import com.sun.org.apache.xml.internal.serialize.OutputFormat;
import com.sun.org.apache.xml.internal.serialize.XMLSerializer;

import java.io.BufferedWriter;
import java.io.CharArrayWriter;
import java.io.File;
import java.io.FileWriter;
import java.io.IOException;
import java.io.Writer;

/**
 * Aplicación que transforma el formato del fichero XML de entrada con
 * información plana sobre la herencia de los clusters, en otro
 * fichero XML con los clusters anidados<p>
 * Ejemplo de ejecución:<p>
 * java prujaxpir ir.xml salidaIr.xml 98
 *
 * @author Juan Carlos Álvarez
 *
 */
public class prujaxpir {
	/**
	 * Documento a procesar
	 */
	static Document doc;
	/**
	 * Documento que se va a generar
	 */
	static Document docnuevo;


...
```

```
...
	/**
	 * Toma un fichero de entrada (parámetro 1), y genera un fichero
	 * de salida (parámetro 2), el tercer parámetro es el número de
	 * clusters a procesar.
	 * @param argv Lista de parámetros de entrada
	 * @param argv[0] Fichero XML de entrada
	 * @param argv[1] Fichero XML de salida
	 * @param argv[2] Número de clusters a procesar
	 * @throws IOException Excepción de entrada salida
	 */
	public static void main(String argv[]) throws IOException {
		if (argv.length != 1) {
			System.err.println("Se ejecuta: java prujaxpir
fichEnt.xml fichSal.xml numClusters ");
			System.exit(1);
		}
		try {
			// Se parsea documento de entrada mediante el API DOM
			DocumentBuilder documentBuilder =
DocumentBuilderFactory
					.newInstance().newDocumentBuilder();
			doc = documentBuilder.parse(new File(argv[0]));
		} catch (IOException ioe) {
			ioe.printStackTrace();
		} catch (ParserConfigurationException e) {
			// TODO Auto-generated catch block
			e.printStackTrace();
		} catch (Exception e) {
			// TODO Auto-generated catch block
			e.printStackTrace();
		}

		// Se crea el nuevo documento XML
		DocumentBuilder docBuilder;

		try {
			docBuilder = DocumentBuilderFactory.newInstance()
					.newDocumentBuilder();
			docnuevo = docBuilder.newDocument();
		} catch (ParserConfigurationException e) {
			// TODO Auto-generated catch block
			e.printStackTrace();
		}
		insertarCluster(argv[2], (Element)docnuevo);
		// Procesamos el arbol DOM convirtiéndolo en un String
		// Definimos el formato de salida: encoding, identación,
		// separador de línea,...
		// Pasamos doc como argumento para tener un formato de partida
		OutputFormat format = new OutputFormat(doc);
		format.setLineSeparator(LineSeparator.Unix);
		format.setIndenting(true);
		format.setLineWidth(0);
		format.setPreserveSpace(false);
		// Definimos donde vamos a escribir. Puede ser cualquier
		// OutputStream o un Writer
		CharArrayWriter salidaXML = new CharArrayWriter();
		// Serializamos el árbol DOM
		XMLSerializer serializer = new XMLSerializer((Writer)
salidaXML, format);
		serializer.asDOMSerializer();
		serializer.serialize(docnuevo);
...
```

```
...
              // Ya tenemos el XML serializado en el objeto salidaXML
              System.out.println(salidaXML.toString());
              String sfichero = argv[1];
              FileWriter fw = new FileWriter(sfichero);
              BufferedWriter bw =new BufferedWriter(fw);
              bw.write(salidaXML.toString());
              bw.close();
       }
       /**
        * Añade nuevos clusters al árbol
        * @param idc Número de cluster a localizar e insertar
        * @param root Punto del árbol en el que se inserta un nuevo elto
        */
       public static void insertarCluster(String idc, Element root) {

              Node clusterBuscar = null;
              Node documBuscar = null;
              String st = "Cluster" + idc;
              String nst;
              String tipo;
              String cadEval = "//cluster[@idcluster='c" + idc + "']";
              try {
                     // XPath para desplazarnos descendientes de un cluster
                     clusterBuscar = (Node)
(XPathFactory.newInstance().newXPath()
                                   .evaluate(cadEval, doc, XPathConstants.NODE));
              } catch (XPathExpressionException e) {
                     // TODO Auto-generated catch block
                     e.printStackTrace();
              }
              if (clusterBuscar != null) {
                     // Añadimos etiquetas al documento
                     Element nuevoCluster = docnuevo.createElement(st);
                     NodeList hijos = clusterBuscar.getChildNodes();
                     Element newFeat = docnuevo.createElement("Features");
                     newFeat.setTextContent(hijos.item(1).getTextContent());
                     nuevoCluster.appendChild(newFeat);
                     tipo = (hijos.item(5)).getNodeName();
                     if (tipo == "hijos") {
                            NodeList nietos =
(hijos.item(5)).getChildNodes();
                            st = nietos.item(1).getTextContent();
                            nst = st.trim();
                            st = "Cluster" + nst;
                            insertarCluster(nst, nuevoCluster);
                            st = nietos.item(3).getTextContent();
                            nst = st.trim();
                            st = "Cluster" + nst;
                            insertarCluster(nst, nuevoCluster);
                     } else {
                            Element newDocum =
docnuevo.createElement("Documentos");
                            NodeList eltos = (hijos.item(5)).getChildNodes();
                            Element nuevoDocum= null;
                            for (int i = 0; i < eltos.getLength(); i++) {
                                   Node elem = eltos.item(i);
                                   if (elem instanceof Element){
                                          Element el = (Element) elem;
                                          st = el.getAttribute("idelto");
                                          nst = st.substring(1);
                                          nst = "Documento" + nst;
                                          String docEval =
"//elemento[@identif='" + st + "']";
...
```

```
...
            try {
                        documBuscar = (Node) (XPathFactory.newInstance()

      .newXPath().evaluate(docEval, doc, XPathConstants.NODE));
                  } catch (XPathExpressionException e) {
                        // TODO Auto-generated catch block
                        e.printStackTrace();
                        }
                  if (documBuscar!=null)
                  {
                        nuevoDocum = docnuevo.createElement(nst);
                        NodeList descend = documBuscar.getChildNodes();
                        Element nueTit =
                        docnuevo.createElement("Titulo");
                        nueTit.setTextContent(descend.item(1).getTextCont
                  ent());
                        nuevoDocum.appendChild(nueTit);
                        Element nueUrl = docnuevo.createElement("URL");
                        nueUrl.setTextContent(descend.item(3).getTextCont
                  ent());
                        nuevoDocum.appendChild(nueUrl);
                  }
            newDocum.appendChild(nuevoDocum);
            }
      }
      nuevoCluster.appendChild(newDocum);
      }
root.appendChild(nuevoCluster);
}
}
}
```

E.2 Visualizador de documentos XML.

```
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.parsers.ParserConfigurationException;

import javax.xml.parsers.DocumentBuilder;
import org.xml.sax.SAXException;
import org.xml.sax.SAXParseException;

import org.w3c.dom.Document;
import java.io.File;
import java.io.IOException;

import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.JScrollPane;
import javax.swing.JTree;

import javax.swing.JSplitPane;
import javax.swing.JEditorPane;

import java.awt.BorderLayout;
import java.awt.Dimension;
import java.awt.Toolkit;
import java.awt.event.WindowEvent;
import java.awt.event.WindowAdapter;

import javax.swing.border.EmptyBorder;
import javax.swing.border.BevelBorder;
import javax.swing.border.CompoundBorder;

import javax.swing.tree.*;
import javax.swing.event.*;
import java.util.*;

/**
 * Visualizador de árbol de contenidos navegable
 * @author Juan Carlos Álvarez
 */
public class DomyJaxp extends JPanel {
       /**
        * Documento a procesar
        */
       static Document document;

       boolean compress = true;
       static final int windowHeight = 460;
       static final int leftWidth = 300;
       static final int rightWidth = 340;
       static final int windowWidth = leftWidth + rightWidth;


...
```

```
…

public DomyJaxp() {
	// Se crea el borde
	EmptyBorder eb = new EmptyBorder(5, 5, 5, 5);
	BevelBorder bb = new BevelBorder(BevelBorder.LOWERED);
	CompoundBorder cb = new CompoundBorder(eb, bb);
	this.setBorder(new CompoundBorder(cb, eb));

	// Se crea el árbol
	JTree tree = new JTree(new DomToTreeModelAdapter());

	// Se crea el panel izquierdo para la estructura
	JScrollPane treeView = new JScrollPane(tree);
	treeView.setPreferredSize(new Dimension(leftWidth, windowHeight));

	// Se crea el panel derecho para los contenidos
	final JEditorPane htmlPane = new JEditorPane("text/html", "");
	htmlPane.setEditable(false);
	JScrollPane htmlView = new JScrollPane(htmlPane);
	htmlView.setPreferredSize(new Dimension(rightWidth, windowHeight));

	// Permite mostrar datos cuando hay cambios en la navegación
	tree.addTreeSelectionListener(new TreeSelectionListener() {
		public void valueChanged(TreeSelectionEvent e) {
			TreePath p = e.getNewLeadSelectionPath();
			if (p != null) {
				AdapterNode adpNode = (AdapterNode) p
						.getLastPathComponent();
				if (adpNode.domNode.getNodeName().equals("URL"))
{
					try {
						htmlPane.setPage(adpNode.content());
					} catch (IOException e1) {
						e1.printStackTrace();
					}
				}
				else {

	htmlPane.setText("<h3><b>"+adpNode.content()+"</b></h3>");
				}
			}
		}
	});

	// Se crea la vista del split-pane
	JSplitPane splitPane = new JSplitPane(JSplitPane.HORIZONTAL_SPLIT,
			treeView, htmlView);
	splitPane.setContinuousLayout(true);
	splitPane.setDividerLocation(leftWidth);
	splitPane.setPreferredSize(new Dimension(windowWidth + 10,
			windowHeight + 10));

	// Añade componentes GUI
	this.setLayout(new BorderLayout());
	this.add("Center", splitPane);
} // constructor

...
```

```
/**
 * Se visualiza en forma de árbol navegable, el XML pasado como argumento
 * @param argv Lista de parámetros de entrada
 * @param argv[0] Fichero XML de entrada
 */
public static void main(String argv[]) {
	if (argv.length != 1) {
		System.err.println("Se ejecuta: java DomyJaxp nombFichXML");
		System.exit(1);
	}
	// Se parsea el documento XML mediante el API DOM
	DocumentBuilderFactory factory =
DocumentBuilderFactory.newInstance();
	try {
		DocumentBuilder builder = factory.newDocumentBuilder();
		document = builder.parse(new File(argv[0]));
		makeFrame();

	} catch (SAXParseException spe) {
		// Error generado por el  parser
		System.out.println("\n** Error de Parsing" + ", linea "
			+ spe.getLineNumber() + ", uri " + spe.getSystemId());
		System.out.println("   " + spe.getMessage());
		Exception x = spe;
		if (spe.getException() != null)
			x = spe.getException();
		x.printStackTrace();
	} catch (SAXException sxe) {
		// Error generado por aplicación o en nicialización del parser
		Exception x = sxe;
		if (sxe.getException() != null)
			x = sxe.getException();
			x.printStackTrace();
	} catch (ParserConfigurationException pce) {
		// No se puede construir el Parser con las opciones dadas
		pce.printStackTrace();
	} catch (IOException ioe) {
		// Error de E/S
		ioe.printStackTrace();
	}

} // main
/**
 * Crea el visualizador con sus marcos
 */
public static void makeFrame() {
	JFrame frame = new JFrame("Recuperación Información");
	frame.addWindowListener(new WindowAdapter() {
		public void windowClosing(WindowEvent e) {
			System.exit(0);
		}
	});
	final DomyJaxp echoPanel = new DomyJaxp();
	frame.getContentPane().add("Center", echoPanel);
	frame.pack();
	Dimension screenSize = Toolkit.getDefaultToolkit().getScreenSize();
	int w = windowWidth + 10;
	int h = windowHeight + 10;
	frame.setLocation(screenSize.width / 3 - w / 2, screenSize.height /
2
			- h / 2);
	frame.setSize(w, h);
	frame.setVisible(true);
} // makeFrame
```

```
...
	/**
	 *  Posibles tipos de nodos DOM   */
		static final String[] typeName = { "none", "Element", "Attr",
	"Text", "CDATA", "EntityRef", "Entity", "ProcInstr", "Comment",
	"Document", "DocType", "DocFragment", "Notation", };
	static final int ELEMENT_TYPE = 1;
	static final int ATTR_TYPE = 2;
	static final int TEXT_TYPE = 3;
	static final int CDATA_TYPE = 4;
	static final int ENTITYREF_TYPE = 5;
	static final int ENTITY_TYPE = 6;
	static final int PROCINSTR_TYPE = 7;
	static final int COMMENT_TYPE = 8;
	static final int DOCUMENT_TYPE = 9;
	static final int DOCTYPE_TYPE = 10;
	static final int DOCFRAG_TYPE = 11;
	static final int NOTATION_TYPE = 12;

	/**
	* Clase que envuelve un nodo DOM, con métodos para devolver
	* el texto que aparece en el árbol, los nodos hijos,
	* los valores de los índices y el nº de hijos */
	public class AdapterNode {
		org.w3c.dom.Node domNode;

		public AdapterNode(org.w3c.dom.Node node) {
			domNode = node;
		}

		/**
		* Genera una cadena que identifica el nodo en el árbol
		* @return La cadena generada		*/
		public String toString() {
			String s = domNode.getNodeName();
			if (compress) {
				String t = content().trim();
				int x = t.indexOf("\n");
				if (x >= 0)
					t = t.substring(0, x);
				s += " " + t;
			}
			return s;
		}

		/**
		* Genera una cadena, con el valor del nodo,
		* cuando el tipo de nodo puede tener valor
		* @return La cadena generada	*/
		public String content() {
			String s = "";
			org.w3c.dom.NodeList nodeList =
domNode.getChildNodes();
			for (int i = 0; i < nodeList.getLength(); i++) {
				org.w3c.dom.Node node = nodeList.item(i);
				int type = node.getNodeType();
				if (type != ELEMENT_TYPE) {
					s += node.getNodeValue();
				}
			}
			return s;
		}
```

```
...

public int index(AdapterNode child) {
	int count = childCount();
	for (int i = 0; i < count; i++) {
		AdapterNode n = this.child(i);
		if (child.domNode == n.domNode)
			return i;
	}
	return -1;
}

public AdapterNode child(int searchIndex) {
	org.w3c.dom.Node node = domNode.getChildNodes().item(searchIndex);
	if (compress) {
		int elementNodeIndex = 0;
		for (int i = 0; i < domNode.getChildNodes().getLength(); i++)
{
			node = domNode.getChildNodes().item(i);
			if (node.getNodeType() == ELEMENT_TYPE
					&& elementNodeIndex++ == searchIndex) {
				break;
			}
		}
	}
	return new AdapterNode(node);
}

public int childCount() {
	if (!compress) {
		return domNode.getChildNodes().getLength();
	}
	int count = 0;
	for (int i = 0; i < domNode.getChildNodes().getLength(); i++) {
		org.w3c.dom.Node node = domNode.getChildNodes().item(i);
		if (node.getNodeType() == ELEMENT_TYPE) {
			++count;
		}
	}
	return count;
}
}

/**
 * Adaptador que convierte el documento DOM a un modelo árbol (JTree)
 */
public class DomToTreeModelAdapter implements javax.swing.tree.TreeModel {
	public Object getRoot() {
		return new AdapterNode(document);
	}
	public boolean isLeaf(Object aNode) {
		AdapterNode node = (AdapterNode) aNode;
		if (node.childCount() > 0)
			return false;
		return true;
	}

	public int getChildCount(Object parent) {
		AdapterNode node = (AdapterNode) parent;
		return node.childCount();
	}
```

```
...
public Object getChild(Object parent, int index) {
	AdapterNode node = (AdapterNode) parent;
	return node.child(index);
}

public int getIndexOfChild(Object parent, Object child) {
	AdapterNode node = (AdapterNode) parent;
	return node.index((AdapterNode) child);
}
public void valueForPathChanged(TreePath path, Object newValue) {
	// Se necesita para satisfacer la interface pero no se usa
}

private Vector listenerList = new Vector();

public void addTreeModelListener(TreeModelListener listener) {
	if (listener != null && !listenerList.contains(listener)) {
		listenerList.addElement(listener);
	}
}

public void removeTreeModelListener(TreeModelListener listener) {
	if (listener != null) {
		listenerList.removeElement(listener);
	}
}
public void fireTreeNodesChanged(TreeModelEvent e) {
	Enumeration listeners = listenerList.elements();
	while (listeners.hasMoreElements()) {
		TreeModelListener listener = (TreeModelListener) listeners
				.nextElement();
		listener.treeNodesChanged(e);
	}
}
public void fireTreeNodesInserted(TreeModelEvent e) {
	Enumeration listeners = listenerList.elements();
	while (listeners.hasMoreElements()) {
		TreeModelListener listener = (TreeModelListener) listeners
				.nextElement();
		listener.treeNodesInserted(e);
	}
}
public void fireTreeNodesRemoved(TreeModelEvent e) {
	Enumeration listeners = listenerList.elements();
	while (listeners.hasMoreElements()) {
		TreeModelListener listener = (TreeModelListener) listeners
				.nextElement();
		listener.treeNodesRemoved(e);
	}
}

public void fireTreeStructureChanged(TreeModelEvent e) {
	Enumeration listeners = listenerList.elements();
	while (listeners.hasMoreElements()) {
		TreeModelListener listener = (TreeModelListener) listeners
				.nextElement();
		listener.treeStructureChanged(e);
	}
}
}
}
```

VNiVERSiDAD
D SALAMANCA