Universidad Rey Juan Carlos

Representación Autocontenida de Documentos HTML: una propuesta basada en Combinaciones Heurísticas de Criterios

TESIS DOCTORAL

Víctor Diego Fresno Fernández

2006

Universidad Rey Juan Carlos

Escuela Superior de Ciencias Experimentales y Tecnología

Departamento de Ingeniería Telemática y Tecnología Electrónica

Representación Autocontenida de Documentos HTML: una propuesta basada en Combinaciones Heurísticas de Criterios

Tesis Doctoral

Directoras:
Dra. Dª. Raquel Martínez Unanue
Dra. Dª. Ángela Ribeiro Seijas
Tutor:
Dr. D. José María Cañas Plaza
Doctorando:
D. Víctor Diego Fresno Fernández

2006

Dra. D^{a}. Raquel Martínez Unanue, Profesora Titular de Universidad del Departamento de Lenguajes y Sistemas Informáticos de la Universidad Nacional de Educación a Distancia y Dra. D^{a}. Ángela Ribeiro Seijas, Científico Titular del Instituto de Automática Industrial, perteneciente al Consejo Superior de Investigaciones Científicas, codirectoras de la Tesis Doctoral "*Representación Autocontenida de Documentos HTML: una propuesta basada en Combinaciones Heurísticas de Criterios*" realizada por el doctorando D. Víctor Diego Fresno Fernández,

HACEN CONSTAR

que esta Tesis Doctoral reúne los requisitos necesarios para su defensa y aprobación.

En Móstoles, a ____ de _____ de 2006,

Dra. D^{a}. Raquel Martínez Unanue Dra. D^{a}. Ángela Ribeiro Seijas

Dr. D. José María Cañas Plaza
(Tutor)

"*Si no fuera físico, sería probablemente músico.*
Pienso a menudo en música.
Vivo mi sueño despierto en música.
Veo mi vida en términos de la música.
Consigo la mayoría de alegrías con la música"

Albert Einstein

Agradecimientos

Quiero dedicar esta tesis especialmente a Carol, con quien comparto mi vida, por todo el amor que me da y lo muchísimo que la quiero. ¡¡¡ Nikutemwa !!!

A mi directora Raquel por enseñarme tantas cosas y por todo lo que ha aportado a este trabajo. Gracias por ayudarme siempre sin pedir nada a cambio y por tu apoyo. Gracias compañera.

A mi otra directora, Ángela, por iniciarme en la investigación y aportarme ideas que han resultado fundamentales en este trabajo, gracias de verdad. Y gracias por todo.

A todos los investigadores con los que he trabajado y compartido inquietudes: a Jose María por su sentimiento "jondo", gracias por todo 'payo', por estar siempre ahí; a los becarios del IAI (os quiero compañeros); a Rodrigo, pues sí... espero verte pronto, amigo; a Luis Magdalena (gracias por todo), a Óscar Cordón, José Miguel Goñi y Félix Monasterio, de Teleko y Granada, gracias por vuestro cariño y enorme conocimiento; a Arantza Casillas, por todo lo que nos queda por hacer, aupa; y, por último, a mis compañeros de la Rey Juan Carlos... a Jaime, a Carlos (meto aquí también a Pedrito y El Salvador) y al resto de compañeros de la Escuela, en especial a los que estuvieron ahí en los momentos difíciles... gracias a Roberto, Vicente, José, Pedro, Sergio, Antonio, Juan Antonio y su gente, Ana, Holger, Juanma... gracias compañeros

Al DITTE, porque otro Departamento es posible... GRACIAS

Quiero destacar mi agradecimiento a mi actual grupo de investigación Gavab y, en especial, a Soto (eres la mejor, compañera !!), Abraham, Juanjo y Antonio (esos gambitos ricos... por toda la ciencia que nos queda por hacer y por todo lo que me habéis ayudado), a Patxi y a Mica (ánimo compañeros... y gracias por vuestra luz en mis eclipses). Gracias a Ángel Sánchez, Paco Nava, Belén Moreno, Alfonso, Maite...

A mis alumnos de PFC, David y Daniel, por todas las líneas de código que he usado en la experimentación de esta tesis. Gracias chavales ;)

A mi familia, gracias por TODO... a mi padre y a mi madre (¡¡ qué puedo decir, gracias por haber creído siempre en mí y haberme cuidado tanto, aunque a veces no haya sido fácil !!), a mis hermanas, Moni e Isa, os quiero muchísimo pequeñas... gracias por soportar mis malos humos y quererme tanto... y no digo más porque si no, lloro.

A mis cuñados: Santi (qué grande eres), Raúl, Patri y Susi. A mi suegri Carmen y a Julio. A mis tíos y tías pegados (porque son más que cercanos). A mis más primos: Luigi, Alberto y Penélope, y Maykol, impresionante lo vuestro... y a mis sobris, Mariola y Noa, un besazo muy

fuerte.

A mis amigos del Ramiro y de la UAM (de Física, Filosofía, Matemáticas, Historia, Psicología...), por ayudarme a ser como soy. A Gonzalo, Consuelo y Willy os pongo como representantes de los que no tengo ni espacio ni tiempo para nombrar...

A la música porque me remueve más que ninguna otra cosa. A los músicos que escucho y con los que toco, he tocado y espero seguir tocando. A Niko, gracias especialmente por este último año y pico de ensayos; no hubiera podido acabar esta tesis sin tu talento. A Ramón, Jai, David, Marcelo, Mónica, Arantxa,... ¡¡ Qué grandes sois !! ¡¡ Gracias artistas !!... sin vosotros me faltaría la armonía.

... y, por último, quiero agradecer y dedicar este trabajo a toda la gente que lucha por cambiar las cosas. A los que tienen principios aunque, a veces, estos supongan su final. Salud.

Sin vosotros el mundo sería muchísimo peor.

Índice general

Índice de figuras

Índice de tablas

Lista de acrónimos

ACC Representación de documentos HTML basada en una combinación analítica de criterios heurísticos, (*Analytical Combination of Criteria*)

ANTF Frecuencia aumentada y normalizada (*Augmented Normalized Term Frequency*)

ASP *Active Server Pages*

Bin Función de proyección binaria, (*Binary*)

BinIDF Función de proyección basada en la frecuencia inversa del documento, (*Binary-Inverse Frequency Document*)

DC Agrupación de documentos, (*Document Clustering*)

DOM Modelo de Objeto de Documento, (*Document Object Model*)

DTD Definición de Tipo de Documento, (*Document Type Definition*)

FCC Representación de documentos HTML basada en una combinación borrosa, o *fuzzy*, de criterios heurísticos, (*Fuzzy Combination of Criteria*)

GF-IDF Función de ponderación basada en la frecuencia global de un rasgo corregida con la frecuencia inversa del documento, (*Global Frequency - Inverse Document Frequency*)

HTML Lenguaje de marcado de hipertexto, (*HyperText Language Markup*)

IG Ganancia de Información, *Information Gain*

IR Recuperación de Información, (*Information Retrieval*)

JSP *Java Server Pages*

k-NN Algoritmo de los k vecinos más cercanos (*k-Nearest Neighbour*)

LSA Análisis de Semántica Latente, (*Latent Semantic Analysis*)

LSI Índice de Latencia Semántica, (*Latent Semantic Indexing*)

NB Clasificador *Naïve Bayes*

NCP Memoria a Corto Plazo

MI Información Mutua, *Mutual Information*

MinMax Reducción de rasgos *term-frequency/document-frequency*

NLP Memoria a Largo Plazo

NN Algoritmo de los vecinos más cercanos (*Nearest Neighbour*)

PF Reducción de rasgos con la propia función de ponderación *Proper Function*

PHP *Hypertext Preprocessor*

PIF Frecuencia Inversa Probabilística

SAX *Simple API for XML*

SGML *Standard Generalized Markup Language*

SVD *Singular Value Descomposition*

SVM *Support Vector Machine*

TC Clasificación automática de textos, (*Text Classification*)

TF Función de ponderación basada en frecuencias de aparición o bolsa de palabras, (*Term Frequency*)

TF-IDF Función ponderación basada en la frecuencia de un rasgo corregida con la frecuencia inversa del documento, (*Text Frequency - Inverse Document Frequency*)

URL Ubicador Uniforme de Recursos, (*Uniform Resource Locator*)

VSM Modelo de espacio vectorial, (*Vector Space Model*)

W3C *World Wide Web Consortium*

WIDF Función de ponderación basada en la frecuencia inversa ponderada, (*Weighted Inverse Document Frequency*)

WTF Función de ponderación basada en frecuencias normalizadas o Bolsa de palabras ponderada, (*Weighted Term Frequency*)

XHTML Reformulación de HTML en XML, (*eXtensible Hypertext Markup Language*)

XML Lenguaje de marcado extensible, (*eXtensible Markup Language*)

Resumen

En esta tesis doctoral se presenta una propuesta de representación autocontenida de páginas web basada en combinaciones heurísticas de criterios. Se proponen dos funciones de ponderación de rasgos como parte de la definición general de un modelo de representación de documentos. Con estas funciones se pretende determinar el peso que tiene un rasgo en el contenido de un documento HTML; para ello se establece un marco teórico general apoyado en una hipótesis fundamental: la lectura supone un proceso activo donde tanto el autor de un documento, como el lector del mismo, aportan su experiencia y conocimiento previo al proceso informativo documental.

Se parte con el objetivo principal de desarrollar representaciones basadas únicamente en el contenido textual de los documentos HTML. El ámbito de aplicación será la clasificación automática y el *clustering* de páginas web. Estos procesos pueden utilizarse en la creación de directorios web temáticos o aplicarse sobre los resultados devueltos tras una consulta a un motor de búsqueda. Una buena parte de las representaciones empleadas actualmente en estos contextos son de tipo mixto, es decir, basadas en un análisis de la estructura del hipergrafo que forma en sí mismo la Web, así como en un estudio del contenido de texto de la propia página web. Las funciones propuestas tratan de mejorar las representaciones basadas en contenido encontradas en la literatura, y podrán emplearse como representaciones autocontenidas o bien formando parte de representaciones de tipo mixto.

Una de las funciones propuestas en esta tesis, llamada ACC (*Analytical Combination of Criteria*), se basa en una combinación lineal de criterios heurísticos extraídos de los procesos de lectura y escritura de textos. La otra, FCC (*Fuzzy Combination of Criteria*), se construye a partir de una combinación borrosa, o *fuzzy*, de esos mismos criterios.

Una de las ventajas que ofrecen ACC y FCC es que permiten representar un documento HTML sin necesidad de analizar previamente ninguna colección de referencia. No será necesario extraer información relativa a las frecuencias de aparición de los diferentes rasgos dentro de la colección. Esta propiedad resulta interesante en el contexto de la Web, visto su tamaño actual y su tasa de crecimiento.

Además, en un contexto en el que la heterogeneidad de los contenidos es una de las características principales, las representaciones propuestas permiten la generación de representaciones independientes del tipo de página que se esté considerando, siempre que tengan contenido textual, de modo que no habrá que definir diferentes heurísticas para representar diferentes tipos

de documentos.

Para la evaluación de las representaciones propuestas se utiliza un algoritmo de clasificación automática *Naïve Bayes* y un algoritmo de *clustering* de partición. Se ha elegido un algoritmo *Naïve Bayes* por ser un clasificador sencillo que ha ofrecido muy buenos resultados en tareas de clasificación de documentos en numerosos contextos. Del mismo modo, se emplea el algoritmo de *clustering* de partición *k-way via Repeated Bisections*, perteneciente a la librería CLUTO, por haber sido aplicado en distintos trabajos de *clustering* de documentos con muy buen comportamiento.

Tras el análisis de los resultados obtenidos en la evaluación de las funciones propuestas, en comparación con funciones de ponderación clásicas –aplicadas tradicionalmente a la representación de textos– y otras funciones específicas para la representación de páginas web, se puede concluir que las representaciones generadas con ACC y FCC tienen un comportamiento destacable. En el caso de la clasificación *Naïve Bayes*, se consideraron diferentes funciones de probabilidad con las que se obtuvieron resultados muy diferentes según la representación estudiada. ACC y FCC presentan el comportamiento más estable en términos generales, de entre el conjunto de funciones evaluadas, destacando especialmente cuando se emplean dimensiones de representación pequeñas. Este hecho implica que se puede obtener una misma calidad de clasificación con vectores de representación muy pequeños. En el caso del *clustering* de páginas web, el comportamiento de las representaciones propuestas en esta tesis resulta aún mejor. Se obtienen, en general, los mejores resultados independientemente del número de grupos considerados. Además, al aumentar el número de *clusters*, el comportamiento relativo frente al resto de las funciones evaluadas mejora sustancialmente.

Abstract

In this dissertation, a new approach to self-content web page representation is proposed. Two term weighting fuctions are presented as a part of a document representation model definition. This approach is built on a main hypothesis: reading is an active process where author and reader contribute their experience and knowledge to the communication.

Internet can be seen as a huge amount of online unstructured information. Due to this inherent chaos, the necessity of developing systems based on Information Technologies emerges, being necessary to create systems that aid us in the processes of searching and efficient accessing to information. The main aim of this research is the development of web page representations only based in text content, and the field of application is automatic web page classification and document clustering. These tasks are applied on the creation of web directories and to obtain a clustering of the documents retrieved by a search engine. In these contexts, repesentations use to be mixed; they are based on an analysis of the hypergraph structure and on the page content. The proposed approach can be complementary, exploring the text content analysis.

One function, called ACC (*Analytical Combination of Criteria*), is based on a linear combination of heuristical criteria extracted from the text reading and writing processes. The other one, FCC (*Fuzzy Combination of Criteria*), is build on a fuzzy engine that combine the same criteria. ACC and FCC allow us to represent HTML documents without any analysis of a reference document collection. It is not needed to count the term frequencies in different documents into a collection; representations are generated without need to download any web page. Furthermore, the ACC and FCC design is independent from the document type; the same heuristics are applied for any web page.

The evaluation is carried out in web page classification and clustering processes. A Naïve Bayes classifier is selected for the supervised machine learning process and a partition algortihm is chosen for the clustering process. Naïve Bayes algorithm is very simple and has previously obtained good results in many researches. The selected clustering algorithm, belonging to CLUTO toolbox, has been applied in many different document clustering tasks obtaining very good results too.

After an experimental analysis, ACC and FCC showed the best general behaviour. In Naïve Bayes classification, four prior probabilities functions were analyzed. The F-measure results showed different behaviours depending on the selected term weighting functions. In general, ACC and FCC showed the most stable F-measure results and one of the best when the di-

mensions of the representations were minimum. Therefore, similar rates can be obtained using smaller dimensions of the representations in the classification tasks. In general, in the partitional clustering problems the results obtained by ACC and FCC functions were the best ones, and these were better when the number of clusters was increased.

Capítulo 1

Introducción

"La Era de la Información es nuestra era. Es un periodo histórico caracterizado por una revolución tecnológica centrada en las tecnologías digitales de información y comunicación, concomitante, pero no causante, con la emergencia de una estructura social en red, en todos los ámbitos de la actividad humana, y con la interdependencia global de dicha actividad. Es un proceso de transformación multidimensional que es a la vez incluyente y excluyente en función de los valores e intereses dominantes en cada proceso, en cada país y en cada organización social..."

Manuel Castells, en la conferencia
"*Innovación, Libertad y Poder en la Era de la Información*",
dentro del Foro Social Mundial de Porto Alegre, Brasil, 2005.

1.1. Motivación

Dado el enorme tamaño y crecimiento que experimenta hoy en día la Web, la aplicación de técnicas de clasificación automática y *clustering* de documentos a páginas web puede resultar muy útil para facilitar y mejorar las tareas de extracción y búsqueda de información en Internet.

La representación de documentos HTML (*HyperText Markup Language*) supone el primer paso en la aplicación de cualquiera de estas técnicas; es necesario transformar un documento desde su formato inicial –texto etiquetado con un marcado HTML– a una forma que resulte adecuada a las entradas de los algoritmos que se vayan a emplear a continuación, y que permita realizar con éxito las tareas en cuestión.

Los métodos de representación empleados por la mayor parte de los motores de búsqueda web se basan en una exploración intensiva de la estructura de la Web. Estas representaciones permiten recuperar, de entre todo el contenido web accesible, un conjunto de documentos en respuesta a una consulta de usuario. Sobre estos documentos recuperados se podrían aplicar procesos de clasificación y *clustering* basados únicamente en el contenido de texto del documento, empleando representaciones que no necesitaran de ninguna información externa a la página, representaciones que resultaran *autocontenidas*. La organización de documentos almacenados en servidores web

o la agrupación de resultados devueltos tras una consulta a un motor de búsqueda pueden dar ejemplo de las posibilidades que ofrecen estas técnicas aplicadas en este contexto.

1.1.1. La *World Wide Web*

La *World Wide Web* es un sistema de información global que ha supuesto un nuevo modelo de colaboración e interacción entre individuos (Berners-Lee et al., 1992). En la actualidad, el problema ya no es tanto el acceso a la información en sí misma, sino conseguir un acceso realmente acorde a nuestras necesidades, de entre toda la cantidad de información accesible. De poco sirve disponer de una inmensa cantidad de datos si no se es capaz de acceder a ellos de un modo realmente provechoso.

Medir el tamaño completo del contenido de la Web es una tarea complicada, habida cuenta de que su naturaleza es dinámica. Sin embargo, es posible realizar estimaciones acerca del tamaño de la "web visible" (*visible/surface web*), es decir, el conjunto de páginas accesibles desde los diferentes motores de búsqueda. Hay otro tipo de contenido, conocido como "web invisible" (*deep web*), formado por información almacenada en bases de datos, y accesible por medio de códigos de lenguajes interpretados (ASP, PHP, JSP...) embebidos en el código HTML de las propias páginas web. Tras una petición a un servidor web, éste genera un documento a partir de ciertos contenidos almacenados en una base de datos, que luego será devuelto al cliente como un documento HTML estático (Berners-Lee et al., 1992).

En (Bharat y Broder, 1998), los autores estimaron el tamaño de la web visible para *Hotbot*[1], *Altavista*[2], *Excite*[3] e *Infoseek*[4] - los mayores motores de búsqueda del momento– en 200 millones de páginas. Además, la intersección entre los documentos indexados por cada uno de estos buscadores era de menos de un 4 %, alrededor de 2.2 millones de páginas. En (Lawrence y Giles, 1999), un año más tarde, se hablaba de 800 millones de páginas. A día de hoy, estas cifras resultan ridículas. Según estudios del ISC[5](Caceres, 2004) de Enero de 2004, el número de documentos accesibles vía web se situaba alrededor de unos 550.000 millones, de acceso libre en un 95 % de los casos. Otros estudios del OCLC[6] que no consideraban los *hosting* virtuales –sitios web que comparten direcciones IP– hablaban de un 71 % de contenidos sin restricción de acceso. En (O'Neill et al., 2002) puede encontrarse un estudio detallado de la evolución de los contenidos públicos en Internet en los últimos años. El último informe encontrado, de Enero de 2005 (Gulli y Signorini, 2005), basado en datos del *Search Engine Watch*[7], muestra que el número de páginas web indexadas en ese momento era de 11.5 billones de documentos. De todas ellas, más de 8 billones eran accesibles con el motor de búsqueda *Google*[8]. En el caso del

[1] http://www.hotbot.com/
[2] http://www.altavista.com/
[3] http://www.excite.com/
[4] http://www.infoseek.com
[5] Internet Software Consortium, Inc. (http://www.isc.org/)
[6] Online Computer Library Center. (http://www.oclc.org/)
[7] http://searchenginewatch.com/
[8] http:///www.google.com

buscador MSN[9], los documentos indexados eran 5 billones, y el total de documentos accesibles por Yahoo![10] y ASK/Teoma[11], era de 4 y 2 billones respectivamente.

Pero además de su tamaño, una de las características inherentes a la Web es la capacidad que ofrece de publicar contenidos libremente. Esto ha supuesto un cambio muy importante en la forma que se tenía hasta ahora de acceder al conocimiento, y constituye una seria dificultad a la hora de diseñar sistemas que asistan en el acceso a la información contenida en la Web, ya que se trata de una información mínimamente estructurada. Además, el contenido de la Web abarca todos los temas imaginables, desde los más generales a los más específicos. En este contexto, donde los documentos resultan ser completamente heterogéneos, ya que han sido creados por personas de todo tipo y condición, los sistemas de acceso a la información deben ir más allá del empleo de dominios semánticos restringidos.

La Web supone, por tanto, un escenario idóneo para la convergencia de comunidades científicas que centran sus investigaciones en distintos campos como, por ejemplo, la Clasificación Automática de Textos (*Text Classification*, TC), el Clustering de Documentos (*Document Clustering*, DC), la Recuperación de Información (*Information Retrieval*, IR), la Minería de Datos, la Minería Web y el Procesamiento de Lenguaje Natural, entre otros. Sin embargo, su propia naturaleza provoca que modelos aplicados con éxito en otros contextos puedan ser poco efectivos en el tratamiento de la información contenida en la Web.

La Web Semántica nace con la intención de resolver el problema que supone la heterogeneidad de contenidos en la Web. Su objetivo es crear un medio universal para el intercambio de información basado en representaciones del significado de los recursos de la Web, de manera que resulten inteligibles para las máquinas. Se pretende tener los documentos etiquetados con información semántica para que su contenido pueda ser procesado, con la ayuda de vocabularios comunes (ontologías) y con las relaciones entre ellos. Con ello se pretende ampliar la interoperabilidad entre los sistemas informáticos y reducir la mediación de operadores humanos en los procesos inteligentes de flujo de información (Wikipedia, 2006).

1.1.2. Acceso a la información web

La búsqueda de información en Internet se realiza principalmente utilizando dos estrategias. En primer lugar, se puede ir explorando y "navegando" por la red, siguiendo los hiperenlaces y accediendo así de una página a otra enlazada. Una segunda estrategia, más usada, está orientada a la "búsqueda de información" con la ayuda, tanto de buscadores, como de la exploración directa de directorios web temáticos (Chen y Dumais, 2000). Ambas pueden combinarse cuando, a partir de los resultados de un motor de búsqueda o de una página clasificada en un directorio, se exploran algunos de los hiperenlaces que se van encontrando.

[9] http://www.msn.com
[10] http://www.yahoo.com
[11] http://www.ask.com

Esta tesis doctoral se centra en la representación de páginas web para su aplicación en sistemas de clasificación y *clustering*, en el contexto de la agrupación de contenidos devueltos por un motor de búsqueda o en creación y mantenimiento de directorios web temáticos.

Motores de búsqueda

En el caso de emplear un motor de búsqueda para acceder a la información web, el usuario debe introducir en el sistema un elemento de búsqueda, lo que se conoce como término de consulta o *query*. Los buscadores web se construyen sobre la creación y mantenimiento de ficheros invertidos, es decir, ficheros de índices que apuntan al conjunto de páginas web accesibles desde el propio motor de búsqueda y susceptibles de ser recuperadas tras una consulta. El buscador devuelve una lista, que puede estar ordenada, con los documentos más relevantes en relación a dicha consulta. Recientemente, algunos sistemas consideran que un usuario se encuentra más cómodo cuando la consulta se puede expresar en términos de lenguaje natural [(Yang et al., 2000), (Crestani, 2002), (Crestani y Pasi, 2003), (Bordogna y Pasi, 2004) y (Cigarrán et al., 2005)].

El número de consultas que manejan a diario estos sistemas es enorme y su crecimiento espectacular. Entre Marzo y Abril de 1994, se realizaron una media de 1500 consultas diarias. En Noviembre de 1997, *Altavista* declaró haber gestionado diariamente alrededor de 20 millones de consultas (Barfourosh et al., 2002). En 1998, con el aumento del número de usuarios, los motores más utilizados trataron del orden de cientos de millones de consultas diarias (Brin y Page, 1998). En la actualidad, estas cifras son ridículas. En el momento en que se escribe esta memoria, y según datos proporcionados por la propia empresa, el número de consultas diarias procesadas por *Google* es del orden de billones, y el número de direcciones URL accesibles para este motor de búsqueda es de más de 8000 millones.

En (Barfourosh et al., 2002) se realiza un estudio sobre el funcionamiento y comportamiento de los motores de búsqueda web. Este trabajo concluye que los buscadores web funcionan bien cuando se emplea un término de búsqueda muy general. Por ejemplo, si queremos recuperar información relativa a "Internet", el sistema nos devolverá miles de documentos que contienen el término "Internet" o términos relativos a este tema, pero encontrar una página adecuada, entre todas las devueltas, se hace muy difícil. Para este tipo de información, algunos trabajos apuntan que el uso de los directorios temáticos que ofrecen muchos de los portales web puede dar mejores resultados.

En (Gonzalo, 2004) se estudia el comportamiento del motor de búsqueda *Google* y se concluye que éste es muy bueno cuando se busca un determinado tipo de información, como pueden ser: páginas personales o sitios correspondientes a instituciones, corporaciones o eventos... y no tan bueno cuando se pretende realizar una búsqueda muy específica como puede ser encontrar un formulario dentro de un determinado portal web. Además, se destaca el hecho de que este tipo de buscadores web dirigen al usuario hasta una determinada página donde son abandonados a su

suerte con la única asistencia de buscadores locales que, por otra parte, actualmente no tienen un funcionamiento muy destacable.

En la mayoría de los casos, la lista ordenada que nos devuelven estos motores de búsqueda no suele estar conceptualmente organizada, ni suele relacionar información extraída de diferentes páginas. En este sentido, hay buscadores como *Vivisimo*[12] que, además de la lista de documentos relevantes, presenta una jerarquía de *clusters* asociada; los documentos recuperados se agrupan por similitud para ayudar al usuario a discriminar la información que le interese, de entre el total de páginas respuesta a su consulta.

Dado el crecimiento y tamaño de la Web, ya no es posible pensar en una indexación total de las páginas contenidas en Internet por ningún sistema de búsqueda. Los índices deben actualizarse constantemente, del mismo modo que lo hacen muchas de las páginas que tratan de indexar. Por otro lado, el tamaño efectivo de la Web hace que una respuesta típica de estos motores de búsqueda pueda ser de miles de documentos, que el usuario tendrá que interpretar, para filtrar aquella información que necesite. En pocas palabras, la explosión informativa en la que vivimos hace cada día más necesario el empleo de sistemas que nos asistan en la búsqueda de información relevante.

Directorios web temáticos

En el caso de que el acceso se realice por medio de directorios web temáticos, los documentos HTML se presentan clasificados en taxonomías –ordenación jerárquica y sistemática de contenidos–, de modo que la búsqueda de información se sustenta en dicha categorización/jerarquía. El usuario, en este caso, deberá desplazarse explorando la estructura que le ofrece el directorio, en busca de un documento que responda a sus necesidades de información. Bastará, por tanto, con encontrar dentro de esta jerarquía, la categoría más cercana o afín a la información que se esté buscando.

El problema esencial de estos directorios preclasificados es el esfuerzo humano que requiere su creación y mantenimiento. Las taxonomías suelen construirse manualmente y, con frecuencia, la clasificación también es manual. En muchos casos, se pide al autor de la página que se quiere hacer accesible desde un directorio determinado, que realice él mismo la clasificación en base a la taxonomía disponible en el directorio. De esta forma, se reduce el tiempo de clasificación, ya que nadie mejor que el creador de una página podrá encontrar su ubicación más adecuada dentro de una jerarquía de clases. Aún así, una clasificación manual de los contenidos de estos directorios web, en poco tiempo resultaría inabarcable (Off, 2000).

Esta jerarquía suele mantenerse manualmente, sin asistencia automática, añadiendo nuevos documentos y actualizando la información de las páginas ya clasificadas. Los portales web más utilizados cuentan con este tipo de directorios: *Google*, *Yahoo*, *LookSmart*[13], *Open Directory*

[12] http://www.vivisimo.com
[13] http://www.looksmart.com

Project[14], etc. La diferencia entre ellos radica en la cantidad de información que almacenan, además de en el tipo de información que extraen de cada página.

1.1.3. Clasificación y *clustering* de páginas web en Internet

La clasificación automática de páginas web emerge como una de las soluciones con mayor potencial para mejorar el acceso a información web. El objetivo de un sistemas de clasificación automática de documentos es encontrar la categoría más cercana para un determinado documento a partir de una jerarquía de clases, previamente creada y etiquetada, compuesta por una colección de documentos de referencia. El desarrollo de las taxonomías propias de los directorios web puede estar asistido por estas técnicas, tanto en su creación, como en su mantenimiento. Asimismo, la clasificación automática de las páginas devueltas tras una consulta a un motor de búsqueda supone otra posible aplicación de estos sistemas. De este modo, se ofrecería al usuario una información adicional sobre las páginas recuperadas, relativa a una jerarquía de clases fijada a priori. Esta información podría aliviar, en parte, la saturación de información que sufre un usuario de un motor de búsqueda.

El *clustering* de documentos ofrece una solución complementaria a la clasificación automática. El objetivo de estos sistemas es el de dividir un conjunto de documentos en grupos disjuntos, o *clusters*, de modo que la similitud sea máxima entre elementos de un mismo grupo y mínima entre los elementos de conjuntos distintos (Everitt et al., 2001). En primer lugar, puede emplearse para asistir en las primeras fases de creación de jerarquías temáticas, permitiendo una clasificación de los resultados sin el elevado coste y complejidad que supone la construcción de taxonomías, y sin la necesidad de contar con grandes colecciones de páginas web pre-etiquetadas. De este modo, una vez realizado el *clustering* sobre un conjunto de documentos, y obtenido un conjunto de clases, éstas pueden ser asociadas con cada una de las categorías presentes en una determinada taxonomía.

En general, el *clustering* de documentos puede aplicarse a tareas como: la agrupación automática previa y posterior a la búsqueda –lo que facilitaría enormemente un acceso más eficiente a la información–, búsquedas por similitud entre rasgos y documentos o, como en el caso antes mencionado del motor de búsqueda *Vivisimo*, a la visualización de los resultados de una búsqueda de una manera estructurada [(Moore et al., 1997), (Han et al., 1998a), (Zamir y Etzioni, 1999) y (Leuski y Allan, 2000)].

La aplicación de cualquier técnica de clasificación automática o *clustering* de documentos requiere siempre un paso previo de representación o transformación de la información, desde su formato inicial a otro más adecuado a las tareas que se vayan a realizar.

[14]http://www.dmoz.org

1.1.4. Representación de páginas web

Para representar una página web es necesario, en primer lugar, realizar un análisis de la misma. Este análisis puede verse desde varias perspectivas, dependiendo de qué características de la página web se quieran explotar. Desde el punto de vista de la Minería Web –definida como el uso de técnicas propias de *Data Mining* para el descubrimiento y extracción automática de información, tanto de documentos como de servicios web–, este análisis se ha dividido tradicionalmente en tres categorías: *Web Structure Mining*, *Web Usage Mining* y *Web Content Mining* (Kosala y Blockeel, 2000).

La *minería web de estructura* (*Web Structure Mining*), también conocida como *link analysis* o *link mining*, se interesa por la estructura inter-documento que forman las páginas contenidas en la Web, lo que requiere una exploración en profundidad de toda ella. Estas técnicas están inspiradas en el estudio de las redes sociales [(Wasserman, 1994), (Getoor, 2003)] y en el análisis de co-referencias (Chakrabarti, 2003), y son conocidas como *representaciones por contexto*. Dentro de este enfoque se encuentran trabajos basados en el análisis de los textos presentes en los hiperenlaces que apuntan a una determinada página (*Anchortexts* y *Extended Anchortexts*). Basta nombrar el algoritmo PageRank (Brin y Page, 1998), y el éxito del buscador *Google* que lo emplea, para comprender por qué son las representaciones más utilizadas en el ámbito de la IR. Otros trabajos destacados en este ámbito son (Glover et al., 2002) y (Richardson y Domingos, 2002).

En cuanto a la *minería web de uso* (*Web Usage Mining*), centra su atención en la interacción del usuario con los sistemas web, desarrollando técnicas que tratan de predecir su comportamiento. Un primer enfoque mapea datos dentro de bases de datos relacionales para posteriormente analizarlos (Mobasher et al., 2000), mientras que otro enfoque analiza directamente la información guardada en los archivos *logs* de los servidores web (Borges y Levene, 2004). Un *log* es un archivo que registra toda la actividad e interacción con los usuarios de un servidor web.

Por último, la *minería web de contenido* (*Web Content Mining*) se dedica a los procesos de extracción y descubrimiento de información a partir del contenido de una página web. Puede incluir el análisis de texto plano, imágenes o información multimedia. Algunos trabajos dentro del campo de la recuperación de información multimedia (*Multimedia IR*) se enmarcarían dentro de este área, y suelen explorar la información contenida en los archivos de imagen, audio y vídeo presentes en las páginas web (Feng et al., 2003). Aunque el número de investigaciones en este campo está creciendo en los últimos años, la mayor parte de los sistemas de representación cuentan principalmente con el texto de los documentos como único medio para representar un documento HTML.

En particular, si un método de representación tiene en cuenta simplemente la información presente en una propia página para representarla, podría considerarse como una "representación autocontenida". No utilizan ninguna información externa a la página, de modo que se evita tener

que disponer, por ejemplo, de información acerca de las frecuencias de los diferentes rasgos de un vocabulario en los documentos de una colección, como suele ser necesario en el caso de la mayoría de las representaciones utilizadas en el ámbito de la IR. En el caso de páginas web, y en el límite, esta dependencia externa implicaría considerar el total de los documentos contenidos en la Web o, al menos, un subconjunto suficientemente significativo del ámbito de aplicación de la representación. De este modo, cualquier representación autocontenida resultará completamente independiente del tamaño actual y futuro de la Web, así como de su estructura, ya que no se tendrá que analizar ningún otro documento para representar uno dado. Además, podría aplicarse en sistemas sin necesidad de contar con enormes medios de almacenamiento ni de procesamiento, ni tampoco requerirían una exploración intensiva de colecciones de documentos correlacionados. Esta tesis doctoral se centra en este tipo específico de métodos de representación de documentos HTML.

Mientras que la mayoría de los trabajos enmarcados en la *Web Usage Mining* provienen del campo de la DM y la investigación en bases de datos, los enfoques *Web Content Mining* y *Web Structure Mining* se encuentran mucho más mezclados; numerosos trabajos de investigación combinan información basada en *links* con información de contenido, específicamente dentro del campo de la IR [(Kaindl et al., 1998), (Lawrence y Giles, 1998), (Embley et al., 1999) y (Chakrabarti et al., 2001)], la TC [(Lewis, 1992), (Chek, 1997), (Mase, 1998), (Attardi et al., 1999), (Asirvatham y Ravi, 2001) y (Lu y Getoor, 2003)], la DC [(Moore et al., 1997) y (Yi y Liu, 2003)] o la Extracción de Información (*Information Extraction*, IE) (Muslea et al., 1998).

Históricamente, los primeros pasos que se dieron en representación de páginas web trasladaron directamente técnicas que se habían aplicado hasta ese momento en la representación de textos, empleándose, por tanto, representaciones basadas en contenido. De este modo, las frecuencias de aparición de los rasgos en la propia página, así como las frecuencias inversas de documento (el número de documentos dentro de la colección en los que aparece un determinado rasgo), constituían la base de estas primeras representaciones. También se aplicaron algunas técnicas basadas en el análisis de algunas etiquetas HTML, pero pronto se introdujeron en las representaciones elementos propios de los documentos web (información de hiperenlaces inter-documento), aplicándose técnicas basadas principalmente en un análisis de correlaciones y desarrollándose, de este modo, representaciones basadas en estructura. Este tipo de representaciones, fundamentalmente aplicadas en el campo de la IR, pasaron a ser predominantes frente a las representaciones por contenido no sólo en IR, sino también en problemas de TC o DC.

Antes de la existencia de Internet, la información de contexto se había aplicado en la representación de colecciones de documentos estructurados y semi estructurados dentro de dominios semánticos controlados, tras un análisis de las co-referencias, o co-citaciones, entre los documentos de la colección. Como ya se ha dicho, es difícil encontrar patrones que describan la forma o la estructura en la que se presentan los contenidos en Internet. La propia naturaleza

de las *representaciones por contexto*, que toman información externa a la página, hace que se puedan presentar algunos problemas si la finalidad es clasificar o agrupar documentos web en base a su contenido y a lo que el autor de la página quiere expresar. Éste sería el caso de querer realizar agrupamientos anteriores y posteriores a una consulta a un buscador web, o si se desea clasificar documentos para crear jerarquías de clase basadas en el contenido.

Cuando se hace una consulta a un buscador que emplea una representación de este tipo, como *Google*, usando como término de consulta, y como ejemplo de caso extremo, una palabra como "miserable", la primera de las páginas devueltas es la página personal del señor "George W. Bush" en la Casa Blanca[15], dentro de la web de la Casa Blanca. Esto es debido a que, posiblemente, existen multitud de páginas en la Web que contienen enlaces con textos donde aparece el término "miserable", o en un texto cercano al enlace, y que apuntan a la página personal del presidente de los EE.UU. Por supuesto, además de esta página, también serán recuperadas muchas otras donde el término "miserable" sí se encuentre en el propio contenido de la página. Además, es posible encontrar páginas relativas a la obra de Víctor Hugo, aunque antes de cualquiera de ellas, en la ordenación que ofrece *Google*, encontramos las páginas personales de Michael Moore, Jimmy Carter, Ángel Acebes y Alfredo Pérez Rubalcaba.

Este tipo de comportamiento puede resultar muy útil en algunos problemas de IR, pero no en todos. Si lo que se desea es acceder a información en base al contenido de un documento, sería bueno contar con métodos de representación complementarios a las representaciones *por contexto* existentes. Las representaciones buscadas deberán tratar de capturar la esencia del contenido de las páginas web, con el fin de ser empleadas en sistemas que ayuden al usuario, mediante clasificación y *clustering* de documentos, a superar la saturación de información que sufre al analizar el contenido de los documentos recuperados.

Además, si se tiene en cuenta el tamaño actual y crecimiento de Internet, el coste de introducir información de contexto, frente a considerar únicamente información contenida en la página web, puede resultar muy elevado. Todo ello motiva el interés por desarrollar métodos de representación autocontenidos de documentos HTML que mejoren las representaciones existentes en el contexto de tareas de clasificación y *clustering* de páginas web.

1.2. Hipótesis

La hipótesis general que se establece como base de la investigación desarrollada en esta tesis doctoral se puede formular en los siguientes términos:

La representación autocontenida de documentos HTML, para su posterior utilización en problemas de clustering y clasificación automática de documentos, podría mejorarse aplicando

[15] A fecha de 27/10/2005, la primera página devuelta por *Google* tras una consulta con el término "miserable" era la página http://www.whitehouse.gov/president/gwbbio.html, correspondiente a la página personal de George W. Bush dentro de la Casa Blanca.

heurísticas acerca de cómo se realiza la lectura de textos e hipertextos y cómo se ojean en busca de información. La aplicación de estas heurísticas requiere de una información que podrá extraerse del vocabulario HTML, de modo que se puedan definir funciones de cálculo de relevancia –funciones de ponderación de rasgos– para obtener representaciones autocontenidas de documentos HTML.

Esta hipótesis general se puede formular, a su vez, en base a las siguientes subhipótesis:

1. Una representación autocontenida de documentos HTML es, por definición, independiente de la estructura y tamaño actual y futuro de la Web, en el sentido de que no se deberá analizar ningún otro documento para obtenerla. No requiere tampoco la existencia de ninguna colección de referencia.

2. Se considera la comunicación por medio de páginas web como un ejemplo de proceso informativo-documental. En este contexto, Internet representa el medio y el HTML es el código por el cual un emisor (el autor de una página web) codifica un mensaje (el contenido de la propia página) que posteriormente será visualizado por un receptor.

   a) El HTML es un lenguaje de marcado híbrido cuyos elementos tienen la función principal de indicar a los navegadores web el modo en que deben mostrar el contenido al usuario. Si bien dispone de algunas etiquetas capaces de indicar la estructura del documento y etiquetas de tipo META (conocidas como metaetiquetas), que contienen información acerca de la propia página, ésta no es la función principal del lenguaje.

   b) Se puede considerar que, así como se espera que el título de un documento aporte información sobre su contenido, las partes enfatizadas de un documento HTML responden a la intención del autor por destacar esas zonas del resto del contenido del documento.

3. Limitando el estudio a las representaciones autocontenidas, y sin considerar los elementos multimedia que pudieran encontrarse en una página web, la representación de un documento HTML será similar a la de cualquier documento de texto en formato electrónico. De este modo, estará formado por el propio texto contenido en el documento, así como por las anotaciones propias de cualquier lenguaje de marcado.

4. La lectura es un proceso activo donde la información que se transmite a través del texto no sólo depende de la intención del autor, sino que también depende del conocimiento previo del lector (Valle et al., 1990). Si leer es la habilidad de extraer información visual de una página y comprender el significado de un texto (Rayner y Pollatsek, 1998), cada vez que leemos un libro, un periódico o una página web, acumulamos, de un modo casi inconsciente, un conocimiento que nos acompañará más adelante en la lectura del siguiente texto, para intentar encontrar el sentido correcto a la información que allí se nos presenta. Pero no sólo

las habilidades del lector y las características estáticas del texto influyen en la comprensión lectora, también determinadas señales estructurales, sintácticas y semánticas afectan al proceso de comprensión. Estas señales muestran evidencia acerca de la relevancia de las diferentes partes de un texto. Diversos estudios han examinado estas señales, incluyendo cómo influyen los títulos (Bransford y Johnson, 1972), la frecuencia de aparición (Perfetti y Goldman, 1974), las enumeraciones (Lorch y Chen, 1986) o las primeras frases (Kieras, 1981) en la comprensión de un texto.

5. Cuando a cualquiera de nosotros se nos pide que, en un tiempo reducido, leamos un texto –y más aún una página web– para extraer el conjunto de rasgos más representativos de su contenido, y poder así determinar el tema de que trata, un mecanismo típico es combinar información acerca de si un rasgo aparece o no en el título, si se encuentra destacado de algún modo en el texto, en qué partes aparece, si resulta especialmente frecuente dentro del documento, etc. Toda la información necesaria para llevar a cabo estos procesos heurísticos resulta accesible desde el código HTML; basta con realizar una selección de algunas etiquetas HTML de carácter tipográfico y estructural, recoger información acerca de la posición en la que aparece cada rasgo dentro del contenido del documento, y después fusionar toda la información con el fin de obtener un valor de relevancia para cada rasgo que nos permita extraer los rasgos más representativos de una página web.

1.3. Objetivos

El objetivo principal de esta tesis se deriva directamente de la hipótesis general y las subhipótesis planteadas, y es:

El desarrollo de un modelo de representación autocontenida de documentos HTML, aplicable a problemas de clustering y clasificación automática de páginas web. Se tratará de trasladar heurísticas aprendidas en los procesos de la lectura y escritura de textos a una función de ponderación dentro de una representación de páginas web "por contenido" dentro del modelo de espacio vectorial. Las heurísticas se tratarán de aplicar por medio de una combinación lineal de factores, así como por medio de una fusión de criterios en la que se aprovechen las características propias de la lógica borrosa, o fuzzy logic. A su vez, se pretende que la representación resultante pueda formar parte de representaciones mixtas, combinación de representaciones basadas en contexto y contenido.

En este trabajo se intentarán trasladar los pasos que se siguen a la hora de leer un texto –o más específicamente, de "ojearlo"– para determinar si, utilizando la semántica del vocabulario HTML en unión con heurísticas extraídas de los procesos de lectura y creación de textos e hipertextos [(Gould et al., 1987), (Foltz, 1996a) y (Potelle y Rouet, 2003)], se

pueden obtener representaciones de páginas web que mejoren los resultados de la clasificación automática y *clustering* de documentos. Se trata de obtener representaciones *autocontenidas* de páginas web. A partir de información de carácter tipográfico y estructural extraída del propio vocabulario HTML, se intentará extrapolar información relativa a la intención del autor al crear el documento, tratando de reproducir el mecanismo natural que sigue una persona al ojear un texto cuando trata de extraer información acerca de su contenido.

Este objetivo principal puede articularse en los siguientes subobjetivos específicos:

1. Realizar un estudio bibliográfico de los diferentes métodos empleados para representar documentos HTML, centrándonos principalmente en las representaciones que puedan considerarse autocontenidas. Se estudiarán también en profundidad las distintas alternativas que existen en la representación de textos.

2. Establecer un marco teórico para el desarrollo de un modelo de representación de documentos HTML. Se revisará un modelo de procesamiento de la información aplicado al procesamiento de la lectura y enmarcado en el campo de la psicología cognitiva y de la arquitectura funcional de la mente.

3. Proponer criterios heurísticos utilizados en el proceso de lectura y escritura de textos e hipertextos. Una vez establecidos los criterios a considerar, se definirán las funciones de captura para cada uno de ellos.

4. Definir métodos de combinación de criterios para el cálculo de la relevancia total de un rasgo en el contenido de un documento.

5. Buscar colecciones estándar de páginas web preclasificadas manualmente, con el fin de tener distintos bancos de pruebas sobre los que evaluar la calidad de las representaciones. El carácter de las colecciones será:

   a) De temática restringida.

   b) De contenido heterogéneo.

6. Desarrollar un modelo de representación a partir de una combinación lineal analítica de criterios (*Analytical Combination of Criteria*, ACC). Realizar un estudio estadístico para establecer un conjunto de coeficientes para la combinación lineal de los criterios seleccionados, evaluándose la calidad de los coeficientes por medio de un algoritmo de clasificación automática *Naïve Bayes*. Los coeficientes deberán ser independientes de la colección con la que se realice la experimentación.

7. Desarrollar un modelo de representación a partir de una combinación borrosa de criterios (*Fuzzy Combination of Criteria*, FCC).

   a) Establecer el conjunto de "variables lingüísticas" asociadas a cada criterio considerado

   b) Definir los procesos de borrosificación y desborrosificación, así como la base de conocimiento –o conjunto de reglas– del sistema borroso.

8. Seleccionar un conjunto de representaciones aplicadas tradicionalmente a textos y que puedan aplicarse a la representación de páginas web, para poder así comparar con los modelos propuestos. Éstas serán:

   a) Funciones basadas en la frecuencia de aparición de un rasgo en un documento.

   b) Funciones enriquecidas con información de la colección, basadas en la frecuencia inversa de documento y frecuencias ponderadas.

9. Seleccionar un conjunto de representaciones autocontenidas de documentos HTML de entre las más relevantes encontradas en la literatura. Estas representaciones servirán para evaluar los modelos propuestos.

10. Aplicar los modelos propuestos en 6 (ACC) y 7 (FCC), así como los seleccionados en 8 y 9, a problemas de clasificación *Naïve Bayes*.

    a) Considerar diferentes aprendizajes y funciones de clasificación dentro de un clasificador de páginas web *Naïve Bayes*:

       1) Modelos basados en eventos.

       2) Funciones gaussianas.

    b) Evaluar todas las representaciones seleccionadas con cada una de estas funciones y para diferentes problemas de clasificación:

       1) Clasificación binaria entre clases semánticamente cercanas.

       2) Clasificación binaria entre clases semánticamente lejanas.

       3) Clasificación multiclase entre clases semánticamente cercanas.

       4) Clasificación multiclase entre clases semánticamente lejanas.

11. Aplicar los modelos de representación propuestos en 6 y 7 a problemas de *clustering* de páginas web, con un algoritmo de partición y su comparación con las representaciones seleccionadas en 8 y 9.

    a) Evaluar todas las representaciones (6, 7, 8 y 9) con cada una de estas funciones y para diferentes problemas de *clustering*.

       1) *Clustering* binario entre clases semánticamente cercanas.

       2) *Clustering* binario entre clases semánticamente lejanas.

       3) *Clustering* multiclase entre clases semánticamente cercanas.

       4) *Clustering* multiclase entre clases semánticamente lejanas.

1.4. Organización de la memoria

En esta sección se describe brevemente la estructura del presente documento, resumiéndose el contenido de cada capítulo.

En el capítulo 2 se establece un marco teórico sobre el que poder establecer definiciones formales para, a continuación, presentar aspectos relacionados con el análisis y tratamiento automático de documentos. Se presentan en detalle modelos vectoriales como el modelo de espacio vectorial (*Vector Space Model*) y el índice de latencia semántica (*Latent Semantic Indexing*), muy utilizados en procesos de IR, TC y DC. Por último, se muestran funciones de ponderación y reducción de rasgos, que pueden forman parte de este tipo de representaciones.

En el capítulo 3 se realiza una revisión de las técnicas aplicadas al análisis del contenido web, así como a los diferentes métodos de representación de documentos HTML encontrados en la literatura. Se presta especial atención a aquellos modelos aplicados a problemas de clasificación y clustering de páginas web, así como a los métodos de representación autocontenidos.

En el capítulo 4 se propone una metodología general para la representación autocontenida de documentos HTML que se encuadra dentro de las representaciones basadas en contenido y cuya idea fundamental reside en la selección y aplicación de heurísticas extraídas de la lectura y escritura de textos. En primer lugar, se realiza una asignación semántica a conjuntos de elementos del vocabulario HTML para, a continuación, combinarlos. A partir de una información de carácter tipográfico, se intenta extrapolar información relativa a qué partes del contenido del documento han sido destacadas intencionalmente por parte del autor en el momento de su creación, tratando posteriormente de reproducir el mecanismo natural que sigue una persona al ojear un texto. En este capítulo se establece el marco teórico general sobre el que se apoyan las representaciones propuestas en la tesis.

En los capítulos 5 y 6, una vez establecido el modelo teórico para la construcción de representaciones autocontenidas de páginas web, se presentan las características específicas de la representación basada en una combinación analítica de criterios (ACC) y en una combinación de criterios heurísticos por medio de un sistema de reglas borrosas (FCC). En el capítulo 5, en primer lugar, se definen las funciones de captura de cada uno de los criterios considerados para, posteriormente, establecer los coeficientes de la combinación lineal. Los valores de estos coeficientes establecen la importancia que se da a cada criterio. En el capítulo 6, inicialmente se definen los conjuntos borrosos del sistema y se relacionan con cada uno de los criterios considerados en la representación. Seguidamente se establecen las funciones de captura para cada criterio y las funciones de pertenencia para cada conjunto borroso. A continuación, se define el sistema de inferencia borroso y la base de conocimiento del sistema, formada por el conjunto de reglas IF-THEN que guardan las heurísticas extraídas de los procesos de lectura y escritura de textos e hipertextos.

En el capítulo 7 se presentan aquellos aspectos relacionados con el diseño experimental comunes a la experimentación en clasificación automática y en *clustering* de páginas web.

En el capítulo 8 se evalúa la calidad de las funciones de ponderación propuestas en esta tesis doctoral mediante la aplicación de un algoritmo de clasificación automática de documentos Naïve Bayes, comparando las representaciones ACC y FCC con funciones de ponderación clásicas en textos (BinIDF, TF, TF-IDF y WIDF) y funciones aplicadas en el ámbito de la representación de páginas web que utilizan el etiquetado HTML (Title y la combinación de categorías propuesta en (Molinari et al., 2003)). En el capítulo 9 se evalúa la calidad de las representaciones propuestas mediante la aplicación de un algoritmo de *clustering* de documentos, comparando las representaciones ACC y FCC con funciones de ponderación clásicas en textos (BinIDF, TF, TF-IDF y WIDF) y otras funciones que se han aplicado en el ámbito de la representación de páginas web y que utilizan el etiquetado HTML (Title y la combinación de categorías propuesta en (Molinari et al., 2003)).

Finalmente, el capítulo 10 está dedicado a las conclusiones y aportaciones generales derivadas de la investigación desarrollada en esta tesis doctoral. Además, se enumeran algunas de las líneas de investigación que quedan abiertas y que se pretenden cubrir en trabajos futuros.

Capítulo 2

Representación automática de documentos

"Tengo que utilizar palabras cuando hablo contigo"
T. S. Eliot

En este capítulo se presenta una breve revisión de los principales modelos de representación de documentos. Se fija el marco teórico sobre el que poder establecer definiciones formales para, a continuación, presentar aspectos relacionados con el análisis y tratamiento automático de documentos. Se analizan en detalle modelos vectoriales como el Vector Space Model –que será el empleado en las representaciones propuestas– y el Latent Semantic Indexing, muy utilizados ambos en procesos de TC, DC e IR. Por último, se mostrará un conjunto de funciones de ponderación y reducción de rasgos, que pueden formar parte de un modelo de representación de documentos aplicado a páginas web.

2.1. Introducción

Representar de un modo adecuado un documento resulta una tarea fundamental para ciertos tipos de procesamiento automático y deberá ser, por tanto, la primera acción a realizar. En líneas generales, la representación deberá ser fiel, en primer lugar, al contenido del documento, incluyendo la información necesaria para poder extraer el conocimiento útil que se espera obtener y, a la vez, deberá ser adecuada a las especificaciones de los algoritmos que se empleen a continuación.

Cuando se trata de representar un documento HTML, el problema puede enfocarse desde diversos ángulos, dependiendo de los elementos que se quieran considerar. Así, una página web puede verse, fundamentalmente, como la suma de:

- *Texto enriquecido*; es decir, combinación entre el contenido de texto de una página y una información específica, en forma de anotaciones y capaz de aportar información tipográfica sobre cómo debe mostrarse el contenido.
- *Metainformación*; metaetiquetas en las que se puede incluir información relativa a la propia página web, como puede ser: autores, palabras claves (*keywords*) que describan el contenido

y ayuden en el proceso de exploración automática y descarga de las páginas web por medio de *web crawlers*, e incluso que contengan un resumen del contenido del documento.

- *Estructura de hiperenlaces* o *hyperlinks*; lo más característico de un documento web. Los hiperenlaces de una página web son referencias a documentos, o determinadas partes de documentos, en una relación 1 a 1 unidireccional. De este modo, la Web puede verse como un grafo de hipertextos enlazados de un modo lógico por medio de estas referencias.

- *Conjunto de parámetros estadísticos*. Por un lado, pueden considerarse aquellos datos de una página web que, almacenados en los ficheros *logs* de los servidores, revelan el número de visitas que ha recibido dicha página, los enlaces seguidos a partir de ella, el tiempo empleado de cada una de sus visitas, etc. Por otro lado, una página web también puede considerarse como el conjunto de elementos que la constituyen y verlo desde un punto de vista estadístico. Dentro de estas variables se pueden incluir: el tamaño, el número de imágenes o archivos que contiene, la cantidad de hiperenlaces externos e internos a la página, etc.

En función de cualquiera de estos elementos, un modelo de representación de documentos deberá definir el espacio matemático de representación de la página web. Algunos pueden considerar estos aspectos aisladamente, mientras que otros definen el espacio de representación en función de una combinación de ellos.

A la hora de representar un documento HTML, en primer lugar, y como aproximación más sencilla, éste podría considerarse como un texto. Debido a que el objetivo general de esta tesis doctoral es hallar una representación autocontenida, que por hipótesis no podrá contener información sobre la estructura de la Web, aunque sí podrá considerar la información de marcado, las representaciones propuestas tendrán en cuenta sólo el contenido textual de cada página, dejando al margen los elementos multimedia. Por esta razón, tanto los modelos desarrollados dentro del campo de la representación de textos, como los modelos desarrollados específicamente para páginas web van a ser tenidos en cuenta de un modo especial.

En este capítulo se parte de una serie de definiciones que permiten formalizar lo que es un modelo de representación de textos, así como describir los principales modelos empleados en el contexto de la clasificación y el *clustering* de documentos: el modelo de espacio vectorial y de modelo de índice de latencia semántica. Finalmente se muestra que el modelo de espacio vectorial resulta más adecuado para la creación de representaciones autocontenidas. Se presentan diferentes funciones de ponderación, tanto locales como globales, que pueden ser empleadas en los modelos de representación considerados. Se muestran también funciones de reducción de rasgos que, aún no habiendo sido aplicadas algunas de ellas directamente a documentos HTML, podrían ser usadas como funciones de asignación de pesos. Por último, se ven aspectos relacionados con las fases de preproceso de los documentos HTML que permiten reducir el conjunto de rasgos con los que se generarán las representaciones.

2.2. Caracterización formal de la representación de documentos

Un *modelo formal* abstrae las características generales y las comunes de un conjunto de sistemas desarrollados para problemas similares, explicando sus estructuras y procesos. En esta sección se revisan brevemente los fundamentos matemáticos requeridos para realizar una discusión acerca de la definición de un modelo de representación de documentos. Estos conceptos incluyen: "conjuntos", "relación", "función", "secuencia", "tupla", "cadena" (o *string*), "grafo", "álgebra", "espacio medible" y "espacio de medida", "espacio de probabilidad", "espacio vectorial" y "espacio topológico". A partir de ellos se podrá definir formalmente un modelo de representación de documentos.

Un **conjunto** es una colección no ordenada de entidades indistinguibles, llamadas "elementos". Que x sea un elemento de S se denota como $x \in S$, y el conjunto "vacío" es ($\emptyset$). Formalmente, tanto *conjunto* como $\in$ ("elemento de") se toman como términos no definidos dentro de los axiomas de teoría de conjuntos. En nuestro caso, este matiz carece de importancia para nuestro desarrollo. Un conjunto no puede contenerse a sí mismo y el conjunto de "conjunto de conjuntos" no existe.

La notación $S = \{x \mid P(x)\}$ define el conjunto S de elementos x que cumplen la proposición lógica $P(x)$. Las operaciones estándar entre dos conjuntos A y B incluyen unión: $A \cup B = \{x \mid x \in A \text{ o } x \in B\}$; intersección: $A \cap B = \{x \mid x \in A \text{ y } x \in B\}$; y producto cartesiano: $A \times B = \{(a, b) \mid a \in A \text{ y } b \in B\}$, donde (a, b) es un par ordenado. A es subconjunto de B, y se denota como $A \subset B$, si $x \in A$ implica que $x \in B$; y el conjunto de todos los subconjuntos de S (incluyendo el $\emptyset$) se llama *conjunto potencia* de S, y se denota S^2.

Definición 1 *Una **relación** binaria R entre dos conjuntos A y B es un subconjunto de $A \times B$. Una relación n-aria R (generalización de la relación binaria) entre conjuntos $A_1, A_2, \ldots, A_n$ es un subconjunto del producto cartesiano $A_1 \times A_2 \times \ldots \times A_n$.*

Definición 2 *Dados dos conjuntos A y B, una **función** es una relación binaria de $A \times B$ tal que, para cada $a \in A$ existe un $b \in B$ tal que $(a, b) \in f$, y si $(a, b) \in f$ y $(a, c) \in f$, entonces $b = c$. Al conjunto A se le conoce como dominio y a B como dominio de imágenes, o co-dominio, de f. Escribimos $f : A \to B$ y $b = f(a)$ como notación para (a, b). Al conjunto $\{f(a) \mid a \in A\}$ se le conoce como recorrido de f.*

Definición 3 *Una **secuencia** es una función f cuyo dominio es el conjunto de números naturales, o algún subconjunto inicial $\{1, 2, \ldots, n\}$ de éste cuyo dominio de imágenes sea cualquier conjunto.*

Definición 4 *Una **tupla** es una secuencia finita, a menudo denotada por una lista de valores de la función* $\langle f(1), f(2), \ldots, f(n)\rangle$.

Definición 5 *Una **cadena**, o string, es una secuencia finita de caracteres o símbolos pertenecientes a un conjunto finito de al menos dos elementos, llamado **alfabeto**. Una cadena se forma concatenando elementos del alfabeto sin signos de puntuación.*

Sea $\sum$ un alfabeto, $\sum^*$ es el conjunto de todas las cadenas de $\sum$, incluyendo la cadena vacía $\emptyset$. De este modo, un lenguaje es un subconjunto de $\sum^*$.

Definición 6 *Una **gramática libre de contexto** es una cuádrupla* $\langle V, \sum, R, s_0\rangle$*, donde* V *es un conjunto finito de variables llamadas "No Terminales",* $\sum$ *es un alfabeto de símbolos Terminales,* R *es un conjunto finito de reglas y* s_0 *es un elemento distinguible de* V *llamado símbolo inicial.*

Una *regla*, también llamada regla de producción, es un elemento de un conjunto $V \times (V \cup \sum)^*$. Cada producción es de la forma $A \rightarrow \alpha$, donde A es un No Terminal y α es una cadena de símbolos (Terminales y No Terminales).

Por otro lado, un **espacio** es un conjunto de objetos junto con una serie de operaciones sobre ellos y que obedecen ciertas reglas. A pesar de la generalidad de esta definición, los espacios suponen un instrumento extremadamente importante en construcciones matemáticas; los espacios se pueden generalizar dentro de "espacios de características" y suponen conceptos clave. Por ejemplo, los espacios afines, lineales, métricos y topológicos definen las bases del análisis y el álgebra. En el campo de la IR, Salton y Lesk formularon una teoría algebraica basada en espacios vectoriales y la implementaron en su sistema SMART (Salton y Buckley, 1965). Los espacios se distinguen fundamentalmente por las operaciones que se definen para sus elementos. Así, pueden emplearse diferentes espacios en problemas de TC, DC o IR.

Definición 7 *Sea* X *un conjunto, una* σ***-álgebra** es una colección* $\mathbb{B}$ *de subconjuntos de* X *que satisface las siguientes condiciones:*

1. *El conjunto vacío pertenece a* $\mathbb{B}$, $\emptyset \in \mathbb{B}$.

2. *La unión de una colección numerable de conjuntos en* $\mathbb{B}$ *pertenece a* $\mathbb{B}$*. Es decir, si* $A_i \in \mathbb{B}(i = 1, 2, \ldots, n)$*, entonces* $\bigcup_{i=1}^{n} A_i \in \mathbb{B}$.

3. *Si* $A \in \mathbb{B}$*, entonces* $\bar{A} \in \mathbb{B}$*, siendo* $\bar{A}$ *el complementario de* A *respecto a* X.

Una consecuencia de las condiciones 2 y 3 de la definición anterior es que la intersección de una colección numerable de conjuntos en $\mathbb{B}$ pertenece también a $\mathbb{B}$.

Definición 8 *Un* ***espacio medible*** *es una tupla* $(X, \mathbb{B})$ *formado por un conjunto* X *y una* $\sigma-$*álgebra* $\mathbb{B}$ *de subconjuntos de* X*.*

Un subconjunto A de X se considera medible (o medible respecto a $\mathbb{B}$) si $A \in \mathbb{B}$. Una función de medida μ en un espacio medible $(X, \mathbb{B})$ es una función de valor real definida por todos los conjuntos de $\mathbb{B}$ que satisfacen las siguientes condiciones:

1. $\mu(\emptyset) = 0$, donde $\emptyset$ es el conjunto vacío, y
2. $\mu(\bigcup_{i=1}^{n} A_i) = \sum_{i=1}^{n} \mu(A_i)$ para una secuencia de conjuntos medibles disjuntos dos a dos.

Definición 9 *Un* ***espacio de medida*** $(X, \mathbb{B}, \mu)$ *es un espacio medible* $(X, \mathbb{B})$*, con una función de medida o métrica* μ *definida sobre* $\mathbb{B}$*.*

Definición 10 *Un* ***espacio de probabilidad*** *es un espacio de medida* $(X, \mathbb{B}, \mu)$ *con una función de medida* $\mu(X) = 1$*.*

Definición 11 *Un* ***espacio vectorial*** *es un conjunto* V *de objetos (vectores) junto con un campo* S *de escalares y con las operaciones de la suma* $(+ : V \times V \rightarrow V)$ *y la multiplicación* $(* : S \times V \rightarrow V)$ *definidas, tal que si* $\vec{x}, \vec{y}, \vec{z} \in V$*, y* $\alpha, \beta \in S$*, entonces:*

1. *Existe un vector único* $\vec{0} \in V$ *tal que* $\vec{x} + \vec{0} = \vec{x}$ *para todo* $\vec{x} \in V$ *(identidad aditiva);*
2. *Para cada vector* $\vec{x} \in V$ *existe un vector* $-\vec{x} \in V$ *tal que* $\vec{x} + (-\vec{x}) = \vec{0}$ *(inverso aditivo);*
3. $(\vec{x} + \vec{y}) + \vec{z} = \vec{x} + (\vec{y} + \vec{z})$ *(propiedad asociativa de la suma);*
4. $\vec{x} + \vec{y} = \vec{y} + \vec{x}$ *(propiedad conmutativa de la suma);*
5. $1 * \vec{x} = \vec{x}$ *(identidad);*
6. $(\alpha * \beta) * \vec{x} = \alpha * (\beta * \vec{x})$ *(propiedad asociativa del producto);*
7. $(\alpha + \beta) * \vec{x} = \alpha * \vec{x} + \beta * \vec{x}$*, y* $\alpha * (\vec{x} + \vec{y}) = \alpha * \vec{x} + \alpha * \vec{y}$ *(propiedad distributiva del producto respecto a la suma).*

Definición 12 *Un* ***espacio topológico*** *es un par* (X, T)*, donde* X *es un conjunto y* $T \subset 2^X$ *una familia de subconjuntos de* X *tales que:*

1. $\emptyset \in T$ *y* $X \in T$*;*

2. *para cualquier colección de conjuntos en* T*,* $\{A_i \in T \mid i \in I\}$*,* $\cap_{i \in I} A_i$ *está contenido en* T*, y si* I *es finito,* $\cup_{i \in I} A_i$ *está también contenido en* T*.*

Se dice que T es una topología para X y los elementos de T son **conjuntos abiertos**. El complemento de un conjunto abierto es un **conjunto cerrado**.

A menudo, los espacios vectoriales y los espacios de medida se construyen sobre espacios topológicos. El uso de cualquier concepto de distancia o, como veremos más adelante, similitud entre elementos dentro de un espacio, implica un espacio métrico subyacente, un espacio topológico cuyos conjuntos abiertos se definen por $\{y \mid d(x, y) < r\}$, donde $d(x, y)$ representa la distancia entre x e y. A partir de las definiciones anteriores se puede decir que un **espacio** es un espacio medible, un espacio de medida, un espacio de probabilidad, un espacio vectorial o un espacio topológico.

Llegados a este punto ya se está en condiciones de definir formalmente un modelo de representación de documentos.

Definición 13 *Un* ***modelo de representación de documentos*** *se puede definir como una cuádrupla* $\langle X, \mathbb{B}, \mu, F \rangle$*, donde* $\langle X, \mathbb{B}, \mu \rangle$ *es un espacio de medida y* F *es una función sobre el conjunto de objetos* X*, de forma que* $F : f(X) \to \mathbb{R}$*.*

Por tanto, todo modelo de representación de documentos se debe definir a partir de los siguientes elementos:

1. Un *conjunto de objetos* X, formado por un subconjunto de $\sum^*$, conjunto total de cadenas de caracteres (o conjuntos de cadenas) sobre de un alfabeto $\sum$. Este conjunto de objetos formará el vocabulario empleado en la representación y serán denominados como "rasgos".

2. Un *álgebra* $\mathbb{B}$, que generalice las relaciones entre los objetos.

3. Una *función de medida* μ, con la que establecer la distancia (o similitud) entre objetos dentro del espacio medible $(X, \mathbb{B})$;

4. Una *función de proyección*, o función de ponderación F, aplicable sobre el conjunto de objetos X, tal que $F = f(X)$, y que establezca la proyección de cada objeto del espacio sobre la recta real, indicando la "relevancia" de dicho objeto en la representación.

Sea cual sea el modelo que se quiera emplear, casi todos ellos coinciden en considerar la palabra como elemento fundamental. Así, en última instancia, una representación será un

conjunto de cadenas (en este caso palabras) que, de una u otra forma, representen el contenido del documento a representar.

Uno de los métodos empleados para tratar de mejorar los modelos de representación de textos es el empleo de procesos de análisis sintácticos para identificar sintagmas (*phrases*) y usarlos como objetos dentro del modelo de representación. Los sintagmas son conjuntos de cadenas que funcionan como una única unidad lingüística, pudiendo constituir por tanto un objeto dentro de la representación. Sin embargo, algunos trabajos han mostrado que, en el campo de la IR, las mejoras obtenidas por el uso de sintagmas frente a cadenas aisladas son muy pequeñas (Lewis, 1990a). Las ventajas que ofrece el uso de sintagmas es que suelen resultar menos ambiguos que las palabras aisladas. Por el contrario, sus propiedades estadísticas son menores; en particular, el gran número de sintagmas diferentes que se pueden crear a partir de un Corpus de referencia, y su baja frecuencia de aparición complica la estimación de frecuencias relativas, necesarias en muchos modelos basados en métodos estadísticos.

Otros trabajos han experimentado con diferentes modelos de representación, incluyendo la variación de cadenas aparecidas en el texto, asignando manualmente "palabras clave", citaciones o información acerca de la publicación, y diferentes estructuras producidas tras diversos análisis lingüísticos (Lewis, 1990a). El empleo de un modelo u otro dependerá del fin último que se le quiera dar a la representación y así, se pueden encontrar modelos diferentes en problemas de IE o IR, así como en trabajos más centrados en TC o DC.

En esta tesis se consideran modelos de representación con un conjunto X formado por cadenas aisladas. A partir de este momento, estas cadenas serán consideradas como rasgos. De este modo, un rasgo supondría un objeto de X dentro de la definición formal de modelo de representación.

2.3. Modelos vectoriales

Los modelos de representación vectoriales son un tipo dentro del conjunto de técnicas de representación de documentos que han sido muy empleadas en sistemas de IR, TC y DC en los últimos años. Las representaciones vectoriales resultan muy sencillas y descansan sobre la premisa de que el significado de un documento puede derivarse del conjunto de rasgos presentes en el mismo. Representan modelos formales y pueden considerarse "basados en rasgos" –o características–; estos rasgos serán, de un modo u otro, los vectores generadores de un espacio vectorial. Los documentos se modelan como conjuntos de rasgos que pueden ser individualmente tratados y pesados. De este modo, en tareas de TC y DC, los documentos pasan a ser representados como vectores dentro de un espacio euclídeo, de forma que midiendo la distancia entre dos vectores se trata de estimar su similitud como indicador de cercanía semántica. En el caso de la IR basta con representar las consultas (*queries*) del mismo modo que se representaría cualquier documento que contuviera los rasgos presentes en dicha *query*. Después se calcula la distancia entre el vector de consulta y el conjunto de documentos presentes en la colección que

se esté considerando. En este contexto, los modelos vectoriales permiten un emparejamiento parcial entre los documentos y el vector de consulta.

En la mayoría de los casos, estos modelos no tratan de reducir las dimensiones del espacio, colapsándolas en un subconjunto más reducido, y consideran cada rasgo como un objeto independiente. A pesar de esto, no son simples ficheros invertidos que guardan información de relación entre rasgo y documentos, sino que representan modelos más flexibles, al permitir realizar el pesado de cada rasgo individualmente (por medio de la función de proyección F), de forma que éste pueda considerarse más o menos importante dentro de un documento o del global de la colección. Aplicando diferentes métricas para el cálculo de la distancia entre documentos con la función de medida μ (véase definición 13) se puede intensificar o no la importancia de unos rasgos frente a otros. Por ejemplo, el producto escalar entre dos vectores mide la similitud entre dos documentos como la distancia euclídea entre los vectores de representación. En algunos problemas que tratan de encontrar la similitud semántica entre documentos, las direcciones de los vectores dentro de un espacio son indicadores más eficaces que la distancia euclídea entre ambos (Baeza-Yates y Ribeiro-Neto, 1999) y así, por ejemplo, la distancia coseno mediría el ángulo que forman dichos vectores, con lo que se quita importancia al módulo de los mismos.

2.3.1. Antecedentes

Los primeros trabajos en representación vectorial surgieron en el ámbito de la clasificación de documentos. Ésta se realizaba de forma manual, asumiendo que un documento pertenecía a una determinada clase sólo si contenía determinados rasgos que previamente se habían etiquetado como pertenecientes a la clase.

Posteriormente, H. P. Luhn consideró que los vocabularios controlados y los esquemas de clasificación que se empleaban en esta indexación manual podrían cambiar en el tiempo y así, en sus trabajos (Luhn, 1953) y (Luhn, 1957) propuso el uso de un "espacio de conceptos" para atenuar los problemas de clasificación derivados de suponer uniformidad en el vocabulario. Cada documento podría representarse, dentro de una colección, como una lista de elementos correspondientes a ideas independientes del lenguaje, lo que consideró como conceptos. Así, dos documentos que compartieran conceptos similares tendrían, presumiblemente, una similitud semántica y estarían representados en posiciones cercanas dentro del espacio de conceptos. En el trabajo de (Luhn, 1957), mostraba que realizar un análisis estadístico sobre las palabras presentes en un documento ofrecía buenas pistas para encontrar similitudes semánticas. Documentos que contenían, por ejemplo, las palabras "chicos", "chicas", "profesor", "clase" o "aritmética" con gran probabilidad podrían pertenecer a la clase "educación". Consideró el proceso de comunicar nociones por el significado de las palabras y, así, sugirió que las dos representaciones más parecidas, de acuerdo al hecho de tener unos mismos elementos y distribuciones, tendrían la mayor probabilidad de representar la misma información.

Luhn parte de la asunción de que la frecuencia puede ser usada para la extracción de rasgos

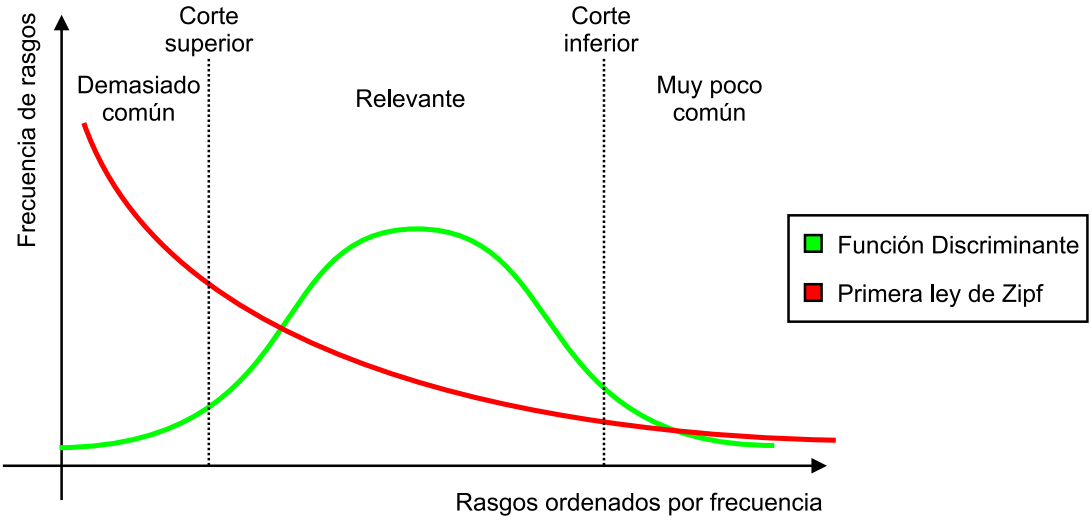

Figura 2.1: Relación entre la frecuencia de aparición de un rasgo y su ordenación en función de esa frecuencia, según Luhn.

y sintagmas que representen el contenido de un documento. Considerando las frecuencias de los rasgos en un documento y su ordenación en función de esta frecuencia, mostró que los rasgos siguen una curva hiperbólica como se muestra en la figura 2.1. Esta curva demostraba la ley de Zipf, que establece que si se cuenta el número de veces que se usa cada palabra en distintos textos en inglés, y se ordenan de mayor a menor frecuencia, se cumple que la frecuencia de la palabra *i-ésima*, multiplicada por i, es igual a una constante C que depende del texto escogido. Actualmente, es necesario elevar i a un exponente t mayor que 1 y cercano a 2 para muchos de los textos existentes, en particular páginas web (Baeza-Yates, 2004). George Kipling Zipf verificó su ley en el *Law of American Newspaper English* y propuso una segunda (figura 2.2). Lo que hizo Luhn fue establecer dos umbrales, *upper cut-off* y *lower cut-off* (denotados como corte superior y corte inferior en la figura 2.1), tratando de excluir así a las palabras no significativas. Los rasgos que excedían en frecuencia el corte superior eran considerados palabras de uso común, mientras que las que no llegaban al corte inferior se consideraban poco comunes, y no contribuirían significativamente a describir el contenido del artículo.

Realizó también estudios sobre las frecuencias de los rasgos para encontrar la relevancia de un rasgo en el contenido de un texto. Consistente con estos estudios, asumió una función discriminaste para la relevancia de un rasgo, con la que medía su capacidad de discriminación dentro del contenido en un texto, alcanzando el máximo en la mitad aproximada de las posiciones entre los dos cortes (figura 2.1). Esta función decae a ambos lados del máximo hasta un valor casi nulo en los cortes. El establecimiento de estos cortes no fue arbitraria, sino que se establecieron por prueba de ensayo y error. De este modo, la superficie que queda debajo de la función de Zipf y entre los dos umbrales, representa las palabras con mayor significación en el contenido.

Luhn empleó estas ideas en el desarrollo de sistemas de extracción de resúmenes. Estableció una medida para la relevancia de sintagmas basada en la frecuencia de rasgos

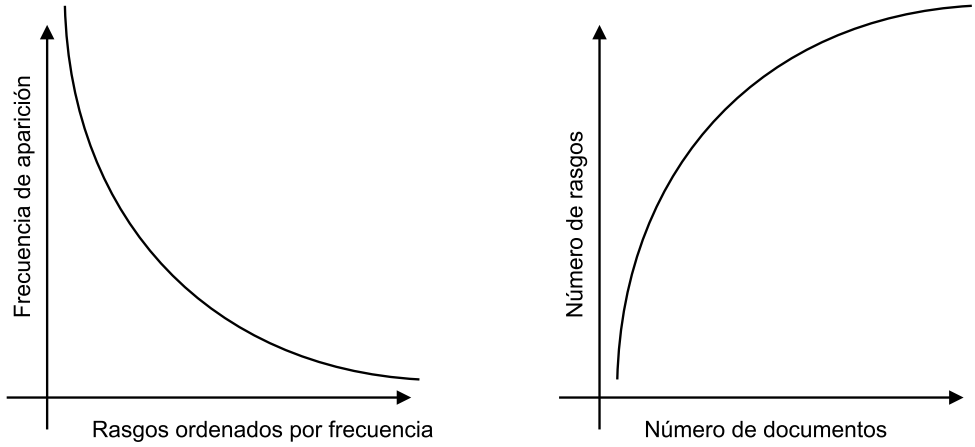

Figura 2.2: (Izq.) Ley de Zipf, relación entre la frecuencia y la ordenación en base a dicha frecuencia en un documento. (dcha.) Ley de Heaps, relación entre el tamaño del corpus y el vocabulario que genera.

relevantes y no relevantes en cada parte del sintagma. Así, estos eran ordenados de acuerdo a esa medida, de modo que los sintagmas más puntuados formarían parte del resumen. Edmondson y Wyllys generalizaron el trabajo de Luhn normalizando sus medidas con respecto a la frecuencia de los rasgos en otros tipos de textos de temática general (Edmonson y Wyllys, 1961).

Poco después, H. Borko y M. Bernick consideraron que dos documentos pertenecientes a una misma clase contendrían rasgos similares (Borko y Bernick, 1963). A partir de esta idea desarrollaron un método de clasificación basado en la presencia de determinados rasgos en documentos pertenecientes a una clase dada. De un modo similar, Maron (Maron, 1961) había aplicado un formalismo estadístico al problema de la indexación de documentos que implicaba, primero, la determinación de ciertas probabilidades relacionales entre rasgos y categorías; y segundo, el uso de estas probabilidades para establecer la clase a la que un documento con ciertas palabras pertenecía. Usando un conjunto de palabras seleccionadas, el sistema era capaz de predecir la clase a la que con mayor probabilidad pertenecía un determinado documento. Este fue el punto de partida del desarrollo de clasificadores basados en la aparición de palabras clave. La evaluación de estos sistemas la realizaba manualmente un experto que determinaba si el sistema funcionaba bien, es decir, si las decisiones tomadas eran correctas o no.

En 1979, M. Koll amplió las ideas de Borko y Bernick, relativas a la reducción del espacio vectorial rasgo-documento, creando un sistema de IR (Koll, 1979) en el que se consideraba, como método de recuperación, en lugar del emparejamiento con un rasgo, la similitud entre documentos. Este sistema trataba de capturar la similitud a partir del uso común de ciertos rasgos en documentos y, para ello, representaba los rasgos y los documentos como vectores dentro de un espacio vectorial. Los documentos se consideraban como "contextos". Los rasgos eran considerados "conceptos" y la media entre las posiciones de los vectores de los documentos en los que un rasgo aparecía determinaba la localización del rasgo. Los contextos se formaban aplicando la media entre las posiciones de los vectores rasgo que constituían el documento.

A lo largo de los años, se han propuesto diferentes modificaciones a estas ideas iniciales, lo que ha hecho que estos modelos sean cada vez más complejos y sofisticados. Dentro de los modelos vectoriales, cabe destacar dos modelos fundamentales: el modelo de espacio vectorial (Raghavan y Wong, 1986) y el índice de latencia semántica (Landauer et al., 1998). Aunque éste último pueda incluirse dentro del *Vector Space Model*, es un modelo lo suficientemente importante y característico como para que se le dedique un apartado propio en esta memoria.

2.3.2. Modelo de espacio vectorial (*Vector Space Model*, VSM)

El modelo de espacio vectorial es un modelo matemático que ha sido empleado para la representación de textos en multitud de sistemas dentro del campo de la IR, DC y TC (Salton, 1983).

Se caracteriza, fundamentalmente, porque asume el "*principio de independencia*", por el que se considera que las cadenas aparecidas en un mismo texto no tienen relación entre sí, pudiendo ser cuantificadas individualmente; además, no tiene en cuenta el orden en el que aparecen en el texto. De este modo, la semántica de un documento queda reducida a la suma de los significados de los rasgos que contiene.

Estas suposiciones, aunque incorrectas, reducen drásticamente la complejidad computacional del problema, ya que permiten representar el documento simplemente como un vector. Además, en muchos casos, esta aproximación tan simple no empeora sustancialmente los resultados obtenidos en problemas de IR, TC y DC (Sebastiani, 2002). Dentro de este modelo se pueden encontrar excepciones a estas asunciones, como (Cohen y Singer, 1999) y (Billhardt, 2002) que, en el ámbito de la IR, utilizan información de coaparición entre rasgos para, sin salirse del VSM ni llegar a colapsar las dimensiones del espacio, establecer una función F que permite evaluar la relevancia de cada rasgo en el documento y así poder recuperar el documento más cercano a un término de consulta. La idea que reside en estas representaciones, que consideran más información aparte de la simple frecuencia de aparición de un rasgo en un documento y en una colección, es la de enriquecer las funciones de ponderación dentro del modelo en lugar de aumentar la complejidad del propio modelo. La propuesta de representación presentada en esta memoria comparte también esta idea fundamental.

Definición 14 *El **modelo de espacio vectorial**, como cualquier modelo de representación de documentos, se puede definir como una cuádrupla* $\langle X, \mathbb{B}, \mu, F \rangle$*, donde*

1. *X es el conjunto de elementos base del espacio vectorial. La definición de estos vectores como objetos requiere de un proceso de asignación léxica A.*

2. *$\mathbb{B}$ es el $\sigma-$álgebra del espacio consistente con la definición 11.*

3. *μ es una función de medida de similitud, métrica del espacio (coseno, distancia euclídea,...)*

4. *F es una función de proyección, que puede verse como una función de ponderación (secciones 2.4) aplicada sobre cada uno de los vectores base del espacio.*

Por tanto, la representación de un documento dentro del VSM se genera a partir de un conjunto de rasgos definidos a priori, que forman un vocabulario y que constituyen los vectores base del espacio vectorial. Con la excepción de algunos sistemas de IR, las representaciones suelen ser vectores con componentes booleanas o numéricas (Lewis, 1990a).

Definición 15 *Se define un **corpus** C como un conjunto de N documentos d_j de modo que,*

$$C = \{d_1, \ldots, d_j, \ldots, d_N\} \tag{2.1}$$

Definición 16 *Dado un corpus C, un **Vocabulario** V se crea con la unión de todos los rasgos seleccionados de los documentos,*

$$V = \{t_1, \ldots t_i, \ldots, t_n \mid t_i \in d_j, \quad y \quad d_j \in C \ , \forall j\} \tag{2.2}$$

siendo n el número total de rasgos encontrados en C y donde $t_i \in d_j$ representa que un rasgo t_i está contenido en el documento d_j.

De este modo, V forma la base del espacio vectorial. Así, todo documento dentro del VSM quedará representado como una combinación lineal de vectores base, donde cada coeficiente de la combinación representará la relevancia de cada rasgo en el contenido del documento, calculada con una función F. Si ahora se quiere construir el vector $\vec{d_j}$ –representación del documento d_j– a partir de este conjunto, cada rasgo t_i deberá considerarse como un vector ortonormal $\vec{t_i}$, de forma que:

$$\begin{aligned} \vec{t_1} &= (1, \ldots\ldots\ldots, 0) \\ \vec{t_i} &= (0, \ldots, 1, \ldots, 0) \\ \vec{t_n} &= (0, \ldots\ldots\ldots, 1) \end{aligned} \tag{2.3}$$

y de modo que cada documento d_j quedaría representado con un vector:

$$\vec{d_j} = t_{1j}\ \vec{t_1} + t_{2j}\ \vec{t_2} + \ldots + t_{nj}\ \vec{t_n} \tag{2.4}$$

$$\vec{d_j} = (t_{1j}, t_{2j}, \ldots, t_{nj}) \tag{2.5}$$

donde t_{ij} representa la relevancia del rasgo t_i dentro del contenido del documento d_j.

Dentro del VSM, dos representaciones de un mismo documento d_i ($\vec{d_i}$ y $\vec{d_i'}$) serán diferentes siempre que el conjunto de valores t_{ij} que tomen las componentes de los vectores $\vec{d_i}$ y $\vec{d_i'}$ sean diferentes; es decir,

$$\vec{d_j} \neq \vec{d_j'} \text{ si y sólo si } \{i|t_{ij} \neq t_{ij}', \forall i\} \neq \emptyset \tag{2.6}$$

En consecuencia, definir una representación dentro del VSM se reduce a encontrar una función de asignación de pesos $f(\vec{t_i}) = t_{ij}$ para el cálculo del valor de cada componente del vector $\vec{d_j}$.

Gerard Salton, junto a investigadores de la *Harvard University* primero y, posteriormente, en la *Cornell University*, desarrollaron y utilizaron un sistema de recuperación, SMART (Salton y Buckley, 1965), con el que estudiaron la ponderación de rasgos dentro del modelo de espacio vectorial. A su vez, estudiaron el *clustering* de documentos, el proceso de truncamiento, el estudio de sinónimos y el uso de sintagmas, el empleo de diccionarios y tesauros, etc. Este sistema les permitió comparar su comportamiento aplicando diferentes técnicas, realizando diferentes análisis como la eliminación de palabras comunes, etc. Varias décadas después, estos estudios siguen siendo referencia para investigadores en los campos de la IR, TC y DC, y sus ideas están detrás de muchas de las ideas propuestas en esta tesis.

2.3.3. Índice de latencia semántica (*Latent Semantic Indexing*, LSI)

El índice de latencia semántica se plateó como una variante al modelo de espacio vectorial. Se basa en el análisis de latencia semántica (*Latent Semantic Analysis*, LSA), que permite comparaciones de similitudes semánticas entre textos (Landauer et al., 1998). La característica fundamental de esta representación es, a su vez, la principal diferencia con el modelo de espacio vectorial: la dependencia semántica entre rasgos.

Aquí, un texto se representa en un espacio de coordenadas donde los documentos y los rasgos se expresan como una combinación de factores semánticos subyacentes. El LSI formaliza esta similitud a partir del álgebra lineal. Cuando dos palabras están relacionadas, como por ejemplo "coche" y "automóvil", cabría esperar que aparecieran en conjuntos similares de documentos. El LSI puede verse como un método de análisis de coaparición entre rasgos. En lugar de usar una función de medida que considere la similitud entre rasgos como una métrica vectorial, como puede ser la función coseno, en este enfoque se usa una función más sofisticada, la descomposición de valores singulares (*Singular Value Descomposition*, SVD). Con ella se pretende medir la similitud entre diferentes rasgos de un vocabulario en base a coaparición entre rasgos.

La SVD se puede considerar también como un método de reducción de rasgos, ya que permite reducir la dimensión de las representaciones. Los conjuntos de rasgos coocurrentes (presentes en varios documentos) son llevados a una misma dimensión en un espacio vectorial de

dimensiones reducidas. De este modo, se pretende incrementar la semejanza en la representación entre documentos cercanos semánticamente. El análisis de coapariciones y la reducción de la dimensionalidad son, por tanto, dos formas de ver el LSI.

Formalmente, el LSI es la aplicación de la función SVD a las matrices rasgo-documentos.

Definición 17 *Una **matriz rasgo-documento** es aquella matriz $M_{i\times j}$ cuyas componentes m_{ij} almacenan información relativa a las frecuencias de aparición de los rasgos t_i en los documentos d_j,*

$$M_{i\times j} = \{\vec{t_i} \mid t_i \in d_j\} \times \{\vec{d_j} \mid d_j \in C\}, \quad con\ dim(\vec{t_i}) = n, \quad y \quad dim(C) = N \tag{2.7}$$

$$M_{i\times j} = \begin{pmatrix} \dots & \dots & \dots & t_{1j} & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \\ t_{i1} & \dots & \dots & t_{ij} & \dots & t_{iN} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & t_{nj} & \dots & \dots \end{pmatrix} \tag{2.8}$$

siendo $\vec{d_j}$ el vector de representación de un documento d_j.

$$\vec{d_j} = \begin{pmatrix} t_{1j} \\ \dots \\ t_{ij} \\ \dots \\ t_{nj} \end{pmatrix}$$

Definición 18 *Una **matriz de distancias** es aquella matriz A_{ij} tal que su componente a_{ij} representa la distancia, en función de una métrica μ, entre un documento $\vec{d_i}$ y otro $\vec{d_j}$, donde $d_i, d_j \in C$.*

La matriz de distancias es, por tanto, una matriz cuadrada de dimensión $N \times N$.

Definición 19 *La **Descomposición de Valores Singulares** (Singular Value Descomposition) toma una matriz de distancias $A_{i\times j}$ y la representa como otra matriz $\hat{A}_{i\times j}$ con un número menor de autovectores, empleando la 2-norma como función de medida μ:*

$$\triangle = || A - \hat{A} ||_2 \tag{2.9}$$

La 2-norma es un métrica entre matrices equivalente a la distancia euclídea aplicada sobre vectores. Respecto a la aplicación del cálculo de la 2-norma, se debe aclarar que las matrices A y $\hat{A}$ tienen el mismo número de filas y columnas. Esto se explica con la siguiente analogía: una recta es un objeto de dos dimensiones, pero puede también representarse en un espacio de 3 dimensiones. La matriz $\hat{A}$ puede representarse en un espacio de dimensión menor, previa transformación de los ejes del espacio vectorial. Así, SVD proyecta un espacio m-dimensional en un espacio k-dimensional, donde $k \ll m$.

En el contexto de la representación de documentos, la matriz rasgo-documento $M_{i\times j}$ tiene una dimensión $n \times N$ y la dimensión del espacio reducido es $n \times d$, con $d < N$, siendo N la dimensión del espacio sin reducir. Los valores de d (dimensión reducida) se toman entre 100-150. Por tanto, la proyección transforma un vector, representación de un documento, de un espacio vectorial N-dimensional a un espacio reducido d-dimensional, con $d < N$.

Definición 20 *La proyección SVD se calcula descomponiendo la matriz rasgo-documento $M_{i\times j}$ en un producto de tres matrices*[1]*:*

$$M_{i\times j} = T_{i\times n'} S_{n'\times n'} (D_{j\times n'})^T \tag{2.10}$$

donde n' toma valores entre 1 y $min(dim(\vec{t_i}), d)$, es decir, $n' = 1, \ldots, min(n, d)$. La matriz $(D_{j\times n'})^T$ representa la traspuesta de una matriz $D_{j\times n'}$ con N filas y $min(n, d)$ columnas.

La matriz $M_{i\times j}$ contiene en cada columna un vector $\vec{d_j}$ que representa a un documento de la colección, de modo que la componente m_{ij} representa la frecuencia de aparición del rasgo t_i en el documento d_j.

Además de la frecuencia de aparición, también pueden usarse otras funciones de ponderación (que se verán es detalle en la sección 2.4) para el cálculo de m_{ij}, componentes de la matriz rasgo-documento. Típicamente, las funciones de ponderación empleadas suelen combinar componentes "locales" y "globales" con la intención de aumentar o reducir la relevancia relativa de cada rasgo en un documento con la frecuencia en el rasgo en la colección. Así, el valor de cada componente m_{ij} será:

$$m_{ij} = L(\vec{t_i}, \vec{d_j}).G(\vec{t_i}) \tag{2.11}$$

donde $L(\vec{t_i}, \vec{d_j})$ es la función de ponderación local aplicada sobre un rasgo $\vec{t_i}$ de un documento $\vec{d_j}$ y $G(\vec{t_i})$ es la función de ponderación global sobre el mismo rasgo. Algunas de las funciones locales y globales más populares son recogidas en la sección (2.4). Por ejemplo, en (Manning y Schutze, 1999) se propone una matriz de pesos estadísticos que incluye $log(t_{ij} + 1)$ como peso

[1] Técnicamente, ésta es la definición de la llamada "SVD reducida". La SVD toma la forma $M_{i\times j} = T_{i\times i} S_{i\times j} (D_{j\times j})^T$, donde las filas y columnas extra de $S_{i\times j}$ respecto a $S_{n'\times n'}$ son vectores cero, y T y D son matrices cuadradas ortogonales (Manning y Schutze, 1999)

local para reducir las diferencias en frecuencias, disminuyendo así el peso de las palabras con una frecuencia alta. La probabilidad global suele reducir el peso de las palabras que están igualmente distribuidas a lo largo de una colección de documentos y aumenta el peso de las palabras menos frecuentes; por tanto, aquellas que pueden resultar más específicas. Otra función muy empleada es la TF-IDF (ecuación 2.24).

Por simplicidad en la exposición, se supondrá en lo que sigue que el valor de a_{ij} –componente de la matriz A_{ij}– es binario; es decir, que toma el valor 1 si el rasgo aparece en el documento y 0 si no aparece. Asumiendo esta función de ponderación binaria, las matrices T y D están formadas por vectores ortonormales (el módulo de cada vector columna es 1 y todos ellos son linealmente independientes). De este modo,

$$T^T T = D^T D = I \tag{2.12}$$

donde I es una matriz diagonal con valores '1' en la diagonal y '0' en el resto de posiciones.

En resumen, SVD encuentra la proyección óptima a un espacio de dimensión reducida explotando patrones de coaparición entre rasgos. En este proceso, aquellos rasgos que tienen similares patrones de coaparición son proyectados (o colapsados) a una misma dirección. Estos patrones tratan de inferir similitud semántica entre rasgos, lo que en ocasiones no es acertado (Manning y Schutze, 1999). Sin embargo, en muchos casos la coaparición ha demostrado ser un indicador válido de relaciones temáticas. En (Foltz, 1996b), la representación del documento se reduce a un conjunto de 20 palabras y se consideran conjuntos de sintagmas, tratados como rasgos, para crear el vector de características, usando la similitud normalizada entre fragmentos de textos.

Como se ha visto, el LSI requiere un análisis en profundidad del corpus de referencia, considerando las frecuencias de coaparición entre rasgos o, en general, el cálculo de cualquier función empleada para encontrar el valor de las componentes m_{ij} de la matriz rasgo-documento $M_{i\times j}$. Esta característica es contraria a la definición de representación autocontenida. Por este motivo, los modelos de representación presentados en esta tesis se desarrollarán dentro del modelo de espacio vectorial y no con el índice de latencia semántica.

A continuación, se presentan las principales funciones de ponderación, aplicadas tanto como funciones de proyección en un espacio de representación, como en tareas de reducción de rasgos para disminuir la dimensión de un vocabulario.

2.4. Funciones de ponderación

En la literatura pueden encontrarse multitud de funciones de ponderación empleadas para calcular la importancia, o relevancia, de un rasgo en el contenido de un texto. Estas funciones constituyen funciones de proyección F dentro de una definición de modelo de representación

de documentos. Son de carácter variado, dependiendo del uso posterior que se vaya a dar a la representación. Estas funciones pueden emplear parámetros diferentes según los casos; desde la frecuencia de aparición de un rasgo en el documento o en la colección, hasta probabilidades condicionadas de un rasgo a una clase en problemas de TC.

Definición 21 *Se define* ***función de ponderación*** *como una función* $f(\vec{t_i}, \vec{d_j})$*, aplicada sobre el conjunto de rasgos* $\{t_i \mid t_i \in d_j \text{ y } d_j \in C\}$*. Estas funciones tratan de establecer la relevancia de un rasgo dentro del contenido de un documento, pudiéndose corresponder con una función de proyección* F *dentro de una definición de modelo de representación de documentos.*

Muchas de estas funciones permiten, a su vez, ordenar un conjunto de rasgos para posteriormente reducir un vocabulario con el que representar un documento. Basta con seleccionar el subconjunto de rasgos con mayores valores del total de rasgos de cada documento, o el subconjunto con mayores valores en el conjunto de la colección. Pero no toda función de ponderación puede ser empleada como función de reducción de rasgos. En el caso de funciones cuyo recorrido sea un conjunto pequeño de valores, su aplicación puede llevar a selecciones arbitrarias. Si existe un subconjunto de rasgos con un mismo valor y éste es mayor que el tamaño del subconjunto de rasgos a reducir, la función dejaría de ser discriminante, y sería necesaria más información para poder completar el proceso de reducción. Este problema está siempre presente en todas las funciones de reducción, pero se puede suponer que usando funciones de recorrido continuo y, si el tamaño de la colección es suficientemente grande, el efecto de esta arbitrariedad tiende a minimizarse.

Las funciones de ponderación se basan fundamentalmente en un "conteo" de frecuencias, ya sea dentro del documento a representar, o en el conjunto de documentos de la colección. Del total de funciones que pueden encontrarse en la literatura, se presentan aquí algunas de las más utilizadas. En primer lugar, pueden distinguirse funciones de carácter "local" y "global". En los siguientes apartados se presentan las características principales de cada tipo, así como un conjunto de ejemplos característicos.

2.4.1. Funciones locales

Se consideran funciones de ponderación "local" aquellas que toman únicamente información del propio documento para obtener una representación, sin necesidad de ninguna información externa. Es importante resaltar que como consecuencia de esta definición, cuando una función local se aplica a la representación de un documento HTML se obtiene, necesariamente, una representación autocontenida.

Definición 22 *Se define* ***función de ponderación local*** *como una función de ponderación* $f(\vec{t_i}, \vec{d_j})$ *con dominio* $D = \{\vec{t_i} \mid t_i \in d_j\}$

A continuación, se presentan las funciones de ponderación local más usadas:

- **Función binaria** (*Binary*, Bin). El método de representación más sencillo, dentro de los modelos de representación vectorial, es el conocido como *conjunto de palabras* o *espacio vectorial binario*. En él, la función de ponderación F dentro de la definición de un modelo de representación es una función binaria, que considera únicamente la presencia o ausencia de un rasgo en un documento para calcular su relevancia dentro del mismo. La función de relevancia es un valor {0,1} y se puede expresar como:

$$F: \quad Bin(\vec{t_i}, \vec{d_j}) = \begin{cases} 1, & \text{si el rasgo } t_i \text{ aparece en } d_j \\ 0, & \text{si no aparece} \end{cases} \tag{2.13}$$

  Con esta función, dos documentos tendrán diferente representación si contienen diferentes rasgos. Así:

$$\vec{d_k} \neq \vec{d_l}, \text{ si y sólo si } \{t_p \mid t_p \in d_k\} \neq \{t_q \mid t_q \in d_l\} \tag{2.14}$$

- **Frecuencia de aparición** (*Term Frequency*, TF). La representación más sencilla dentro de los modelos no binarios es la generada con la función TF y conocida como Bolsa de palabras (*bag of words*). La relevancia se representa por la frecuencia de aparición del rasgo en el documento y puede representarse como:

$$F: \quad TF(\vec{t_i}, \vec{d_j}) = f_{ij}, \ frecuencia\ del\ rasgo\ t_i\ en\ d_j \tag{2.15}$$

  En este caso, la representación de dos documentos distintos es diferente ($\vec{d_k} \neq \vec{d_l}$) si y sólo si:

$$\{t_p | f_{pk} \neq f_{pl}, \forall p\} \neq \emptyset \tag{2.16}$$

  donde $f_{p,k} = TF(\vec{t_p}, \vec{d_k})$ y $f_{p,l} = TF(\vec{t_p}, \vec{d_l})$. Es decir, que dos representaciones son diferentes si contienen rasgos diferentes o si las frecuencias de los rasgos no son todas iguales. El principal problema que presenta la representación TF es que sobrevalora los rasgos muy frecuentes (véase la ley de Luhn, figura 2.1), cuando es sabido que estos rasgos suelen ser palabras de uso común y no resultan ser muy característicos dentro del contenido de un documento. Para corregir esta sobrevaloración, se plantean funciones que corrigen este comportamiento:

- **Frecuencia normalizada** (*Weighted Term Frequency*, WTF). Con esta función se genera una representación conocida como bolsa de palabras normalizada, donde la relevancia se calcula como la frecuencia de aparición normalizada del rasgo en el documento, y puede representarse como:

$$F: \quad WTF(\vec{t_i}, \vec{d_j}) = \frac{f_{ij}}{\sum_{t_p \in d_j} f_{pj}} \tag{2.17}$$

donde f_{ij} es la frecuencia de un rasgo t_i en d_j. Por tanto, esta función supone una normalización de la frecuencia de un rasgo en un documento con la suma total de frecuencias del conjunto de rasgos presentes en el mismo. En este caso, la condición que se debe cumplir para que $\vec{d_k} \neq \vec{d_l}$ es la misma que en el caso de la bolsa de palabras (2.16).

- **Frecuencia aumentada y normalizada** (*Augmented Normalized Term Frequency*, ANTF); esta función representa una frecuencia normalizada de un rasgo en un documento. La normalización se realiza con la mayor de las frecuencias presentes en el documento, y puede representarse como:

$$F: \quad ANTF(\vec{t_i}, \vec{d_j}) = 0,5 + 0,5\frac{f_{ij}}{max(\{f_{pj} \mid t_p \in d_j\})} \tag{2.18}$$

siendo f_{ij} la frecuencia de t_i en d_j. Utilizando esta función, dos documentos tendrán diferente representación en las mismas condiciones de la función anterior, ya que las condiciones en (2.16) implican el cumplimiento de la condición:

$$max(\{f_{pk} \mid t_p \in d_k\}) \neq max(\{f_{ql} \mid t_q \in d_l\}) \tag{2.19}$$

siendo $f_{pk} = TF(\vec{t_p}, \vec{d_k})$ y $f_{ql} = TF(\vec{t_q}, \vec{d_l})$.

- **Pesado logarítmico**. Esta función de ponderación, junto con la binaria (2.13) y la TF (2.15), se ha utilizado en muchos trabajos como función local para el cálculo de la matriz rasgo-documento $M_{t\times D}$ dentro del LSI, y tiene la siguiente expresión:

$$F: \quad L(\vec{t_i}, \vec{d_j}) = log_2(f_{ij} + 1) \tag{2.20}$$

donde f_{ij} representa la frecuencia de t_i en d_j. En este caso, el cumplimiento de las condiciones (2.16) implica que dos documentos diferentes tendrán siempre representaciones diferentes. Es decir, que si $d_k \neq d_l$, entonces sus representaciones serán $\vec{d_k} \neq \vec{d_l}$.

Además de estas funciones, se han propuesto otras muchas que tratan de mejorar la representación binaria y la bolsa de palabras, y que, en muchos casos, son funciones de carácter *global*.

2.4.2. Funciones globales

Las funciones de ponderación "global" son aquellas que toman información de la colección para generar las representaciones. Los coeficientes t_{ij} de los vectores de representación $\vec{d_j}$ son calculados a partir de información externa al propio documento.

Definición 23 *Se define* ***función de ponderación global*** *como una función de ponderación*

$f(\vec{t_i}, \vec{d_j})$ *con un dominio* $D = \{\vec{t_i} \mid t_i \in d_j, \forall d_j \in C\}$.

De este modo, con una función de ponderación global se puede tener que $f(\vec{t_i}, \vec{d_j}) = f(\vec{t_i}, \vec{d_k}),\ \forall\ \vec{t_i}$ si $d_j \neq d_k$, con lo que $f(\vec{t_i}, \vec{d_j}) = f(\vec{t_i})$. Sin embargo, se pueden considerar también como funciones de ponderación global aquellas que posean una parte local y otra global, por lo que esta condición, en esos casos, no se cumpliría.

En 1972, Spark Jones examinó el uso de esquemas de pesado global para mejorar los sistemas de IR (Jones, 1972). Consideró que los rasgos que aparecían frecuentemente en una colección podían considerarse importantes en tareas de recuperación de información; sin embargo, si lo que se pretendía era encontrar las diferencias entre los documentos, entonces los rasgos poco frecuentes en la colección deberían ser tenidos muy en cuenta, y pesados en mayor grado que los rasgos más frecuentes. En ese trabajo se empleó una función F para el pesado global de tres colecciones de referencia,

$$F: \quad f(n) = m, \text{ tal que } 2^{m-1} < n \leqslant 2^m \tag{2.21}$$

En una colección de N documentos, si un rasgo ocurría n veces, obtenía un peso $f(N) - f(n) + 1$. En las tres colecciones evaluadas el sistema mejoraba cuando se empleaban pesos globales. Estos estudios buscaban mejorar los sistemas de IR, pero sus ideas han sido luego usadas en muchas de las funciones de ponderación utilizadas en TC y DC.

A continuación, se presentan las funciones de ponderación globales más conocidas:

- **Frecuencia Inversa del Documento** (*Inverse Document Frequency*, BinIDF). Esta función trata de enriquecer la representación binaria suponiendo que los rasgos que aparecen en muchos documentos de la colección no son tan descriptivos como aquellos que aparecen en unos pocos, y se puede expresar como:

$$F: \quad BinIDF(\vec{t_i}, \vec{d_j}) = \begin{cases} 1 + \log(\frac{N}{df(\vec{t_i})}), & \text{si } f_{ij} \neq 0 \\ 0, & \text{si } f_{ij} = 0 \end{cases} \tag{2.22}$$

  donde $df(t_i)$ es la frecuencia de documentos, el número de documentos de la colección en los que aparece el rasgo t_i, f_{ij} la frecuencia de t_i en d_j y N la dimensión del corpus. Con esta representación, un rasgo t_i tomará el mismo valor t_{ij} en cualquier documento d_j de la colección. De este modo,

$$BinIDF(\vec{t_i}, \vec{d_j}) = K \quad \forall d_j \in C \tag{2.23}$$

  siendo K una constante. Sin embargo, el hecho de que dos documentos tengan conjuntos de rasgos diferentes hará que sus representaciones sean también diferentes, como se expresa en la condición (2.14).

- **Frecuencia del Término × Frecuencia Inversa del Documento** (*Term Frequency - Inverse Document Frequency*, TF-IDF). Para evitar que el valor t_{ij} sea constante $\forall d_j \in C$, Gerard Salton en (Salton, 1988) propuso combinar la función $TF(\vec{t_i}, \vec{d_j})$ con el factor $IDF(\vec{t_i}, \vec{d_j})$:

$$F: \quad TF - IDF(\vec{t_i}, \vec{d_j}) = f_{ij} \times \log(\frac{N}{df(\vec{t_i})}) \tag{2.24}$$

  En este caso, f_{ij}, frecuencia del rasgo t_i en d_j, corrige el factor $IDF(\vec{t_i}, \vec{d_j})$ de forma que el valor que toma un mismo rasgo en dos documentos es diferente siempre que la frecuencia de dicho rasgo en cada documento sea también diferente. Así:

$$TF - IDF(\vec{t_i}, \vec{d_j}) \neq TF - IDF(\vec{t_i}, \vec{d_k}), \quad \forall f_{ij} \neq f_{ik} \quad \text{con} \quad d_j, d_k \in C \tag{2.25}$$

  Empleando esta función de ponderación, las representaciones de dos documentos son diferentes ($\vec{d_k} \neq \vec{d_l}$) si y solo si cumplen las condiciones (2.16), es decir, si las frecuencias del conjunto de rasgos que contienen son diferentes.

- **Frecuencia inversa ponderada** (*Weighted Inverse Document Frequency*, WIDF). Esta función normaliza las frecuencias f_{ij} de un rasgo t_i en un documento d_j con la frecuencia de dicho rasgo en la colección. Por tanto, esta función tiene la forma:

$$F: \quad WIDF(\vec{t_i}, \vec{d_j}) = \frac{f_{ij}}{\sum_{d_k \in C} f_{ik}} \tag{2.26}$$

  donde el sumatorio en el denominador recorre el conjunto de valores de 1 a N. Esta función supone una corrección a la sobreponderación que realiza la función TF con los rasgos frecuentes. Distingue, de un modo diferente a la función IDF, entre los rasgos que resultan frecuentes en un documento y los que son frecuentes en el conjunto de la colección, penalizando estos últimos.

- **Frecuencia inversa probabilística** (*Probabilistic Inverse Frequency*, PIF). Esta función establece una corrección a la sobreponderación de la frecuencia de documento $df(\vec{t_i})$, penalizando a los rasgos que aparecen en un mayor número de documentos, y se expresa como:

$$F: \quad PIF(\vec{t_i}, \vec{d_j}) = log(\frac{N - df(\vec{t_i})}{df(\vec{t_i})}) \tag{2.27}$$

  donde N es la dimensión del corpus y $df(\vec{t_i})$ es la frecuencia de documentos de t_i. Esta función se emplea en el ámbito del LSI como parte global dentro de la función de asignación de pesos a las componentes m_{ij} de la matriz rasgo-documento M. Con esta función basta con cumplir que el conjunto de rasgos en dos documentos sea distinto (condición 2.14) para asegurar representaciones diferentes en documentos diferentes.

- Función **Normal** (N). Esta función corrige los rasgos con una frecuencia alta en el conjunto de rasgos de la colección, y se expresa como:

$$F: \quad N(\vec{t_i}, \vec{d_j}) = \sqrt{\frac{1}{\sum_{d_k \in C} f_{ij}^2}} \tag{2.28}$$

siendo $\sum_{j=1...N} f_{ij}$ una suma sobre el total de documentos d_j en el corpus C.

- La función **Frecuencia Global x Frecuencia Inversa de Documento** (*Global Frequency - Inverse Document Frequency*, GF-IDF) calcula la relevancia de un rasgo mediante una relación entre la frecuencia global en la colección $gf(\vec{t_i}) = \sum_{d_j \in C} f_{ij}$ y la frecuencia de documento $df(\vec{t_i})$, medidas ambas de carácter global.

$$F: \quad GF - IDF(\vec{t_i}, \vec{d_j}) = \frac{gf(\vec{t_i})}{df(\vec{t_i})} \tag{2.29}$$

Esta función suele emplearse en el cálculo de la proyección SVD del LSI. Si se emplea esta función como función de proyección F en la representación de un documento, dos documentos diferentes ($d_i \neq d_j$) tendrán diferente vector de representación ($\vec{d_i} \neq \vec{d_j}$) siempre que se cumplan las condiciones de que los documentos no tengan el mismo conjunto de palabras (condición 2.14), porque en ese caso, para todos los rasgos del documento, $gf(\vec{t_i})$ y $df(\vec{t_i})$ serían constantes.

- La función **Entropía** (H). Esta función suele emplearse también en el cálculo de las componentes de la matriz rasgo-documento, dentro del modelo de representación LSI.

$$F: \quad H(\vec{t_i}, \vec{d_j}) = 1 - \sum \frac{p_{ij} log_2(p_{ij})}{log_2(N)} \text{ , con } \; p_{ij} = \frac{f_{ij}}{gf(\vec{t_i})} \tag{2.30}$$

con $f_{ij} = TF(\vec{t_i}, \vec{d_j})$ y $gf(\vec{t_i}) = \sum_{j=1...N} f_{ij}$, frecuencia global de t_i en el corpus C. Como en el caso de la función GF-IDF, cumpliéndose la condición 2.14 se asegura que dos documentos diferentes tengan diferente representación.

En ciertos casos, empleando representaciones generadas con funciones de carácter global se obtienen mejores resultados que cuando se usan funciones de ponderación local. Por otro lado, se observa que, en el caso de las funciones globales, las condiciones necesarias para asegurar vectores de representación diferentes, cuando se tienen documentos diferentes, son menos restrictivas que en el caso de las funciones locales.

En general, la función TF suele mejorar la representación binaria en problemas TC y DC; a su vez, las representaciones con factor IDF (la función TF-IDF es la de uso más extendido) suelen ofrecer mejores resultados que la representación TF. La función WIDF se ha empleado en algunos trabajos, aunque los resultados no son concluyentes. Éstas serán algunas de las

funciones con las que se van a evaluar las representaciones propuestas en esta tesis. La selección se ha realizado por haberse encontrado como las funciones más usadas, tanto en trabajos de TC como de DC.

El resto de funciones presentadas en este apartado se han aplicado fundamentalmente como parte de funciones de ponderación utilizadas en LSI, ya sea como componente local o como función global. Otro aspecto a destacar en todas ellas es que no son "orientadas a tarea", es decir, la representación $\vec{d_j}$ no dependerá del uso que se le vaya a dar a continuación y así, una representación generada con cualquiera de las funciones anteriores será la misma independientemente de su aplicación posterior, ya sea en tareas de *clustering* o de clasificación automática de documentos y, en este caso, independiente del posible método de aprendizaje utilizado.

2.4.3. Funciones de reducción de rasgos en TC

Muchas de las funciones definidas en problemas de TC para tareas de reducción de rasgos (*Feature Selection*) pueden emplearse como funciones de ponderación dentro de modelos de representación vectoriales.

Las funciones de reducción permiten realizar una ponderación en base a la cual se ordenan todos los rasgos contenidos en un vocabulario para, a continuación, seleccionar un subconjunto de ellos. La selección se puede hacer estableciendo un umbral de ponderación mínima o prefijando una dimensión reducida, generando así un vocabulario que resultará ser un subconjunto del vocabulario inicial. Por este motivo, ya que se asigna un valor a cada rasgo t_i del vocabulario, la representación de un documento se podría realizar tomando estas funciones de reducción de rasgos como funciones de ponderación.

Definición 24 *Sea* $R = \{t_i \mid t_i \in V\}$ *un subconjunto de rasgos del vocabulario* V *y* $\bar{R} = \{t_i' \mid t_i' \in V\}$ *su complementario, de modo que* $R \cup \bar{R} = V$*; se define* ***función de reducción de rasgos*** *como una función* $f(\vec{t_i})$ *que aplicada sobre un conjunto de rasgos cumple que* $f(\{t_i \mid t_i \in V\}) > f(\{t_i' \mid t_i' \in V\})$*, de modo que* $f : V \rightarrow R$*.*

En este caso se estaría suponiendo que los rasgos más relevantes en la colección son los más representativos en un documento. La diferencia fundamental entre una función de ponderación local y una función de reducción de rasgos es que en las funciones de reducción el pesado se realiza en términos de una información extraída de la colección y no del documento. Esto mismo sucedía con las funciones de ponderación global que no tenían componente local.

D.C. John, en un trabajo centrado en la búsqueda de rasgos irrelevantes en un corpus de referencia (John et al., 1994), y como parte de un proceso general de selección de características dentro de un problema de TC con aprendizaje automático, apuntó las dos grandes familias en las que se pueden ubicar la mayor parte de las funciones de reducción de rasgos. Por un lado, se

tiene el enfoque de "filtrado", donde las funciones son independientes del método de aprendizaje que se vaya a aplicar y, por tanto, resultan "no orientadas a tarea". La mayoría de las funciones aplicadas en problemas de IR se encuentran dentro de este enfoque, así como las funciones de ponderación descritas en la sección 2.5. Por otro lado, en el enfoque de "envoltura", la selección de rasgos se realiza con el mismo método de aprendizaje que será empleado posteriormente, y sobre una colección representada con el propio resultado del proceso de reducción. Estos métodos son "orientados a tarea" y se encuentran fundamentalmente en trabajos relacionados con el aprendizaje automático, dentro de las "técnicas de reducción de dimensionalidad supervisadas" (Karypis y Han, 2000): aquellas que consideran la información de pertenencia a una clase para calcular la ponderación de un rasgo en un vocabulario.

A continuación, se muestran algunas de las funciones de reducción de rasgos más utilizadas en la literatura. Como se ha indicado anteriormente, estas funciones podrían formar parte de un modelo de representación como función de proyección F dentro de la cuádrupla $\langle X, \mathbb{B}, \mu, F\rangle$ que define el modelo.

- **Ganancia de información** (*Information Gain*, IG). Esta medida se usa para establecer la calidad de un determinado rasgo en una tarea de aprendizaje automático. Mide la cantidad de información que guarda dicho rasgo calculando la cantidad de información obtenida para la predicción de una determinada categoría en ausencia y presencia del rasgo.

  Sea c_j $\forall j = 1 \dots m$ un conjunto de categorías prefijadas, $P(c_j)$ representa la probabilidad a priori de una determinada clase c_j (en caso de que no sean equiprobables); $P(\vec{t_i})$ es la probabilidad a priori del rasgo t_i y $P(\bar{\vec{t_i}})$ la probabilidad a priori de cualquier rasgo $\{t_j | t_j \neq t_i, \forall i, j\}$. Así, la IG puede definirse como la siguiente función:

$$\begin{aligned} F: \quad IG(\vec{t_i}) = &- \textstyle\sum_{j=1}^{m} P(c_j) log P(c_j) + P(\vec{t}) \sum_{j=1}^{m} P(c_j|\vec{t_i}) log P(c_j|\vec{t_i}) + \\ &+ P(\bar{\vec{t_i}}) \textstyle\sum_{j=1}^{m} P(c_j|\bar{\vec{t_i}}) log P(c_j|\bar{\vec{t_i}}) \end{aligned} \tag{2.31}$$

  Como puede verse, el cálculo de esta función requiere la estimación de probabilidades condicionadas $P(c_j|\vec{t_i})$, siendo ésta la probabilidad a posteriori de cada clase c_j dado un rasgo t_i; así como el cómputo de la entropía (ecuación 2.30). El cálculo de estas probabilidades tiene una complejidad en tiempo de $O(N)$, y en espacio de $O(Nn)$, donde n es la dimensión del conjunto de entrenamiento. Dado un corpus de entrenamiento, para cada rasgo $\vec{t_i}$ se computa su $IG(\vec{t_i})$ y se eliminan aquellos rasgos cuyo valor de IG no supere un umbral mínimo predeterminado.

  Esta función es de carácter global y deberá cumplir la condición (2.16) para que, teniendo dos documentos diferentes ($d_k \neq d_l$), se puedan obtener representaciones diferentes ($\vec{d_k} \neq \vec{d_l}$).

- **Información mutua** (*Mutual Information*, MI). Esta función se ha empleado

habitualmente en el contexto del modelado estadístico del lenguaje, fundamentalmente para encontrar relaciones entre rasgos. Debido a que toma un valor particular para cada clase, el $MI(\vec{t_i})$ se puede estimar de dos formas: como el valor medio sobre el conjunto total de clases ($MI_{adv}(\vec{t_i})$, ecuación 2.32), o como el valor máximo sobre el total de clases ($MI_{max}(\vec{t_i})$, ecuación 2.33).

Si se considera A como el número de documentos de una clase c_j en los que un rasgo t_i aparece; B como el número de documentos en los que no aparece; y C el número de documentos totales que pertenecen a la clase c_j; entonces, las funciones que expresan la MI "global" y "máxima" para un rasgo t_i son:

$$F: \quad MI_{adv}(\vec{t_i}) = \sum_{j=1...m} P(c_j) I_{adv}(\vec{t_i}, c_j) \approx \sum_{j=1...m} P(c_j) log \frac{A \times N}{(A+C) \times (A+B)} \tag{2.32}$$

$$F: \quad MI_{max}(\vec{t_i}) = max_{j=1...m}\{I_{max}(\vec{t_i}, c_j)\} \approx max_{j=1...m}\{log \frac{A \times N}{(A+C) \times (A+B)}\} \tag{2.33}$$

La complejidad de esta medida es $O(Nm)$, donde m es el número de clases consideradas. La debilidad de esta función es que su pesado está fuertemente influenciado por las probabilidades marginales de los rasgos, ya que $I(\vec{t_i}, c_j)$ puede expresarse también como:

$$I(\vec{t_i}, c_j) = logP(\vec{t_i}|c_j) - logP(\vec{t_i}) \tag{2.34}$$

De este modo, $I_{adv}(\vec{t_i}, c_j)$ es el valor medio de los valores de $I(\vec{t_i}, c_j)$ sobre todas las clases consideradas. En el caso de contar con rasgos con la misma probabilidad condicionada a una clase $P(\vec{t_i}|c_j)$, los rasgos menos comunes obtendrán mayores valores. Además, los pesos estimados por esta función no son comparables en el caso de rasgos con frecuencias muy diferentes.

Esta medida no depende de d_j, es decir, que el valor que toma cada rasgo es el mismo independientemente del documento, por lo que se tendrá que cumplir la condición (2.14) para tener diferentes vectores de representación para diferentes documentos.

- **Chi-square** (χ^2). La función $\chi^2(\vec{t_i}, c_j)$ mide la falta de independencia entre un rasgo t_i y un documento d_j. Esta función puede ser comparada con la distribución χ^2 con un grado de libertad. Si tomamos las mismas definiciones para A, B, C y N que en (2.32) y (2.33), la medida χ^2 de un rasgo tiene la siguiente expresión:

$$F: \quad \chi^2(\vec{t_i}, c_j) = \frac{N \times (AD - CB)^2}{(A+C) \times (B+D) \times (A+B) \times (C+D)} \tag{2.35}$$

Esta medida toma un valor nulo en el caso de que haya total independencia entre el documento y la clase. Al igual que en el caso de la MI, la medida $\chi^2(\vec{t_i}, c_j)$ es particular

de cada clase, de modo que se puede estimar de dos formas: como el valor medio sobre el conjunto total de clases o como el valor máximo sobre el total de clases. Así, las $\chi^2(\vec{t_i}, c_j)$ "media" y "máxima" se expresan como:

$$\chi^2(\vec{t_i}, c_j)_{adv} = \sum_{j}^{m} P(c_j)\chi^2(\vec{t_i}, c_j) \tag{2.36}$$

$$\chi^2(\vec{t_i}, c_j) = max_{j=1...m}\{\chi^2(\vec{t_i}, c_j)\} \tag{2.37}$$

Esta función χ^2 no tiene dependencia de d_j, es decir, que si se emplea como una función F dentro de un modelo de representación y queremos que dos documentos diferentes tengan diferente representación, deberá imponerse la condición (2.14).

- **Odds Ratio**. En esta función, la probabilidad $P(\vec{t_i}|c_j)$ de que un rasgo t_i sea característico de una determinada clase c_j suele calcularse considerando la ocurrencia de los diferentes rasgos. Así, una expresión general para esta función es:

$$F: \quad OddsRatio(\vec{t_i}) = \frac{odds(\vec{t_i}|c_1)}{odds(\vec{t_i}|c_2)} = \frac{P(\vec{t_i}|c_1)(1 - P(\vec{t_i}|c_2))}{(1 - P(\vec{t_i}|c_1))P(\vec{t_i}|c_2)} \tag{2.38}$$

Cabe destacar que este valor $OddsRatio(\vec{t_i})$ no se mide sobre todo el conjunto de clases, como en el caso de la IG, sino tomando clases dos a dos. Esta función no depende de $\vec{d_j}$, es decir, que el valor que toma cada rasgo en el documento es independiente del mismo. Existen variantes de esta función que permiten introducir la frecuencia del rasgo en una página en el cálculo de la relevancia. Tres variantes planteadas por Dunja Mladenic (Mladenic, 1998) son:

$$FreqOddsRatio(\vec{t_i}) = f_{ij} \times OddsRatio(\vec{t_i}) \tag{2.39}$$

$$FreqLogP(\vec{t_i}) = f_{ij} \times log\ \frac{odds(\vec{t_i}|c_1)}{odds(\vec{t_i}|c_2)} \tag{2.40}$$

$$ExpP(\vec{t_i}) = e^{P(\vec{t_i}|c_1) - P(\vec{t_i}|c_2)} \tag{2.41}$$

donde f_{ij} es el valor de la función $TF(\vec{t_i}, \vec{d_j})$.

Las expresiones de las funciones anteriores se pueden encontrar en [(Salton y Yang, 1973), (Sparck, 1972) y (Mladenic, 1998)], y estudios comparativos en [(Caropreso et al., 2001) y (Yang y Pedersen, 1997)]. En (Kosala y Blockeel, 2000) se pueden encontrar otros métodos menos tradicionales.

Hay que destacar que las funciones de reducción de rasgos requieren un tratamiento previo,

y en profundidad, de la colección de documentos que se esté considerando. No basta con realizar tareas de conteo de frecuencias, como en las funciones de ponderación globales, sino que en muchos casos será necesario realizar cálculos complejos para determinar probabilidades condicionadas, etc. En el caso de la representación de páginas web, como ya se ha dicho, sería deseable no necesitar esta información externa.

2.5. Selección del vocabulario

En esta sección se introducen algunos aspectos relacionados con la selección de rasgos como elementos de transformación de una información que inicialmente es de carácter cualitativo y que debe ser transformada a un conjunto de objetos X dentro de un espacio medible $\langle X, \mathbb{B} \rangle$, de carácter cuantitativo.

Como selección de vocabulario pueden entenderse aquellas fases que transforman un texto en el conjunto de rasgos que lo podrán representar; las más comunes cuando la representación se basa en las palabras individuales son las siguientes.

2.5.1. Análisis léxico

El *análisis léxico* es la fase en la que se analiza un texto para distinguir las cadenas que formarán parte de la representación. Para encontrar las cadenas presentes en un texto escrito en lenguas de nuestro continente, se deberán tener en cuenta, por ejemplo, separadores como espacios en blanco, signos de puntuación, guiones y otros signos especiales.

2.5.2. Lematización y truncado (*stemming*)

El proceso de lematización es aquel en el que a cada forma flexiva se le asigna su lema. De este modo, formas diferentes como "tendrán" o "tuvieran", compartirían el lema "tener" y serían considerados como un mismo rasgo dentro del vocabulario. Este proceso requiere recursos lingüísticos adecuados como pueden ser diccionarios electrónicos y un software específico. Ejemplos de herramientas que realizan la lematización son FreeLing[2] o TreeTagger[3].

El truncamiento, o truncado (*stemming*), tiene el mismo objetivo: reducir el número de rasgos del vocabulario. En este caso, a cada rasgo encontrado en un documento se le eliminan caracteres de los prefijos o sufijos para lograr así agrupar diferentes palabras con una misma forma. De este modo, formas diferentes como "comunicación", "comunicante" o "comunicado", compartirían el *stem* "comunic-" y serían considerados como un mismo rasgo dentro del vocabulario. Con el truncamiento se pretende aproximar un proceso de lematización de un modo sencillo y sin necesidad de contar con un analizador morfológico ni recursos lingüísticos. La idea en la que

[2] http://garraf.epsevg.upc.es/freeling/
[3] http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/DecisionTreeTagger.html

reposa este proceso es la misma que en el proceso de lematización: diferentes palabras construidas sobre un mismo lexema suelen representar un mismo concepto.

Existen diferentes algoritmos de truncado (*stemmers*) para diferentes idiomas. Para el inglés el más utilizado es el de Porter (Porter, 1997), aunque existen otros [(Lovins, 1968), (Dawson, 1974)]. También existen *stemmers* para otros idiomas como francés (Savoy, 1999), castellano (Figuerola et al., 2002), árabe (Abu-Salem et al., 1999), holandés (Kraaij, 2002), etc. Una alternativa a la eliminación de sufijos es el aprendizaje de reglas de truncamiento a partir de grandes colecciones de documentos. Analizando un conjunto de palabras que forman parte de un idioma, se detectan los prefijos y sufijos que las forman, seleccionando después como raíz de cada palabra el prefijo más probable (López-Ostenero et al., 2004).

2.5.3. Eliminación de *stop-words* (o "palabras vacías")

En este proceso se eliminan aquellas palabras que se emplean para articular el discurso, tales como artículos, preposiciones, conjunciones,..., pero que no tienen por sí solas una semántica relevante en el contenido de un texto. Se considera que este tipo de palabras no tienen capacidad discriminante. La comunidad científica dispone de listas de *stop-words* para numerosos idiomas, entre las que se incluyen también algunos verbos, adverbios o adjetivos de uso frecuente.

La idea de eliminar estas palabras surgió de un trabajo de G. Salton, A. Wong y C. Yang (Salton et al., 1975), en el que se constató que se obtenían mejores resultados en tareas de IR cuando los documentos eran menos similares entre sí, es decir, si se les quitaban muchas de las palabras que compartían, logrando reducir así la densidad del espacio de los documentos.

2.5.4. Utilización de información sobre las categorías gramaticales

Si se dispone de información sobre las categorías gramaticales, ésta se puede utilizar para seleccionar rasgos, o incluso para ponderarlos. En (Friburger et al., 2002) y (Casillas et al., 2004b) exploraban la incidencia de la selección de entidades nombradas en el cálculo de la similitud entre textos y en el DC, respectivamente.

2.6. Conclusiones

El objetivo fundamental de esta tesis, el desarrollo de un modelo de representación autocontenida de documentos HTML, implica estudiar los modelos de representación de textos. En este capítulo se ha establecido una definición formal para un modelo general de representación de textos y se han presentado las características fundamentales de los modelos vectoriales más usados en tareas de TC, DC y IR. Se ha presentado una definición formal del modelo de espacio vectorial y del índice de latencia semántica. En ambos casos, la función de ponderación F resulta ser una pieza clave en la definición de dichos modelos ya que, junto con un espacio

de medida $\langle X, \mathbb{B}, \mu \rangle$, define el modelo de representación. Se han revisado estas funciones F, distinguiendo entre las que presentan carácter local o global. Asimismo, se han presentado funciones de reducción de rasgos que podrían emplearse, en determinadas condiciones, como funciones de ponderación.

En los modelos vectoriales cada componente se trata de modo independiente, lo que permite evaluar el impacto particular de cada rasgo en una representación y, por ende, el comportamiento de un sistema en función de la selección o ponderación de rasgos. Estos modelos representan los documentos dentro de un espacio vectorial en el que se podrán establecer diferentes medidas de similitud (métricas del espacio μ) y de ponderación (funciones de proyección F) para cada una de las componentes X de los vectores de representación. Uno de los objetivos de esta tesis es la aplicación de heurísticas aprendidas en los procesos de lectura y escritura de textos en un modelo de representación de documentos HTML. La primera aproximación a la hora de aplicar estas heurísticas sería hacerlo directamente sobre rasgos individuales, de modo que se pueda comprobar si aportan o no información relevante a la representación.

La diferencia fundamental entre el modelo de espacio vectorial y el índice de latencia semántica es el enriquecimiento que este último método incorpora por medio de un análisis de coapariciones. La consecuencia directa es que requiere información de colección, lo que supone un alto coste computacional (Karypis y Han, 2000), ya que necesita calcular tanto una matriz rasgo-documento como una matriz de distancias. Este método ha venido aplicándose a colecciones controladas, realizando un análisis cruzado de los contenidos de todos los documentos para generar las representaciones. En el caso del modelo de espacio vectorial, es posible desarrollar funciones de asignación de peso completamente independientes de cualquier información de colección, por lo que resulta más adecuado para el desarrollo de los objetivos de esta tesis.

Respecto a las funciones de ponderación de rasgos, el coste computacional necesario para encontrar el valor que toman las funciones globales sobre el conjunto de rasgos de un vocabulario aumenta con el tamaño de la colección de referencia. La aplicación de este tipo de funciones en el contexto de la Web pasaría por calcular estos valores sobre colecciones de páginas web más o menos pequeñas. A continuación, se debería asumir que la relación de valores de frecuencia y coaparición de los rasgos de un vocabulario en dicha colección se mantendrá en el conjunto de páginas web contenidas en la Web, y susceptibles de ser representadas. Esta asunción puede resultar muy arriesgada si se acepta la heterogeneidad propia de los contenidos de Internet. Usar funciones locales, por el contrario, no requiere de información previa sobre ningún vocabulario. Dentro de este enfoque se enmarcarán los métodos de representación propuestos en esta tesis.

Capítulo 3

Análisis y representación de documentos HTML

"Nuestra sociedad es una sociedad donde el poder es, sobre todo,
el poder del lenguaje, el poder de la comunicación."
Wu Ming

En este capítulo se realiza una revisión de los diferentes métodos de representación de documentos HTML encontrados en la literatura. Se prestará especial atención a aquellos modelos que hayan sido aplicados a problemas de clasificación y clustering de páginas web, así como a los métodos de representación autocontenida, es decir, aquellos que no requieren para la representación más información que la presente en la propia página.

3.1. Introducción

El modelo de datos de la Web se basa en los paradigmas del enlace de hipertexto y la búsqueda de información en textos (Berners-Lee et al., 1992). Este modelo se caracteriza porque:

- La información sólo debe estar representada una vez y mediante una referencia se podrá se replica una copia de ella.
- Los hiperenlaces permiten que la topología de la información evolucione, modelando el conocimiento humano en cualquier momento y sin restricción.
- Los documentos en la Web no tienen por qué existir físicamente como archivo; pueden ser documentos "virtuales" generados por un servidor en respuesta a una consulta.

La información contenida en la Web se presenta y relaciona principalmente por medio de documentos HTML, tanto estáticos como dinámicos. De este modo, la *World Wide Web* puede verse como un conjunto de documentos HTML relacionados por medio de hiperlenlaces.

Definición 25 *Un **grafo** G es un par (V, E), donde V es un conjunto no vacío de vértices y E es un conjunto de pares no ordenados de vértices, $\{u, v\}$, con $u, v \in V$ llamados aristas. Un*

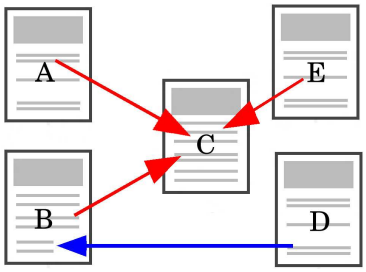

Figura 3.1: A, B y E son *inlinks* de C que, a su vez, es un *outlink* de A, B y E. Del mismo modo, B es un *outlink* de D, que supone un *inlink* de B.

grafo dirigido *G es un par (V, A), donde V es un conjunto no vacío de vértices y E es un conjunto de aristas dirigidas, o arcos, donde cada arista es un par ordenado de vértices (v_i, v_j) con $v_i, v_j \in V$. Se dice que un grafo es ponderado si, o bien un nodo, o bien un arco, o ambos, tienen asociado un peso.*

Teniendo en cuenta la definición anterior, la estructura de la Web es la de un grafo donde los documentos HTML representarían vértices –o nodos– y los hiperenlaces serían aristas –o arcos–. A su vez, un documento HTML puede tener hiperenlaces a otros documentos, así como cualquier otro documento HTML puede tener hiperenlaces hacia él.

Definición 26 *Si se considera la Web como un grafo (V, A) y un nodo $h_k \in V$ es un documento HTML contenido en ella, se definen sus **in-links** como el conjunto de aristas dirigidas $\{(v_i, v_j) \mid v_j = h_k\}$, siendo $v_i, v_j \in V$ y V un conjunto no vacío de vértices.*

Definición 27 *Si se considera la Web como un grafo (V, A) y un nodo $h_k \in V$ es un documento HTML contenido en ella, se definen sus **out-links** como el conjunto de aristas dirigidas $\{(v_i, v_j) \mid v_i = h_k\}$, siendo $v_i, v_j \in V$ y V un conjunto no vacío de vértices.*

Dicho de otro modo, los *in-links* de una página web sería el conjunto de hiperenlaces que apuntan hacia dicha página, mientras que sus *out-links* serían el conjunto de hiperenlaces que salen de la misma (figura 3.1).

Desde la creación del lenguaje HTML se han propuesto diferentes modelos para la representación de páginas web. Estos modelos se centraron inicialmente en el análisis del contenido textual de la propia página para, posteriormente, pasar a analizar en profundidad la estructura de hiperenlaces que forma el conjunto de hipertextos contenidos en la Web. Este segundo enfoque se mantiene como el más usado en la actualidad en tareas de IR, TC y DC. Los documentos HTML también han sido representado en función del uso que se hace de ellos, es decir, en función del orden en el que se siguen los hiperenlaces de una página, la secuencia de páginas que se siguen dentro de un mismo dominio web, etc.

Como ya se ha avanzado, estos aspectos sueden emplearse conjuntamente en representaciones de tipo "mixto". Un ejemplo típico puede ser analizar el contenido textual de la página y complementar la representación con información relativa a las relaciones existentes entre páginas de una determinada colección. En este sentido, según (Barfourosh et al., 2002), la combinación de estos factores mejorará la funcionalidad de los sistemas de acceso a la información web si se consiguen mejorar los métodos de representación de cada uno de los enfoques por separado.

3.2. Modelos de representación de documentos HTML

Como se ha visto en el capítulo 2, gran parte de los modelos de representación de documentos coinciden en el uso de la *palabra* como elemento fundamental en la representación de la información textual. De este modo, una representación, en última instancia, deberá ser un conjunto de rasgos que, de una manera u otra, representen el contenido del documento. Este conjunto de rasgos formará el vocabulario V, conjunto de objetos X dentro del modelo de representación, y tendrá asociado un valor de relevancia t_{ij} para cada rasgo t_i dentro del contenido de un documento d_j.

El uso de un modelo de representación u otro dependerá de la tarea que se quiera realizar a continuación, es decir, la representación no tiene por qué ser igual si se tiene un problema de IE o IR, o si el problema está más centrado en tareas de TC o DC.

La combinación de Internet como base de conocimiento con técnicas de minería de datos se conoce como Minería Web (*Web Mining*). R. Kosala y H. Blockeel la definen como "un área de investigación donde convergen diferentes comunidades científicas, tales como las Bases de Datos, la Recuperación de Información, la Inteligencia Artificial y, especialmente, el Aprendizaje Automático y el Procesamiento de Lenguaje Natural". Partiendo de esta idea, los modelos de representación de páginas web se han planteado principalmente bajo tres enfoques, correspondientes a cada una de las partes en las que puede dividirse la Minería Web (Kosala y Blockeel, 2000):

- Minería de contenido o representaciones **por contenido**. En este caso, el objetivo es la recogida de información e identificación de patrones relativos a los contenidos de las páginas web y a las búsquedas que se realizan sobre las mismas. Su carácter es esencialmente local y pueden distinguirse dos estrategias principales:
  - *Minería de páginas web*, donde se analizan directamente los contenidos presentes en las páginas web. Los datos a analizar pueden ser:
    - Texto libre (considerando contenido estático y dinámico).
    - Documentos escritos en HTML.
    - Documentos escritos en XHTML.
    - Elementos multimedia.

- Cualquier otro tipo de contenido presente en la página web.

En el caso de que no se necesitara ningún tipo de información externa a la página para su representación, se estaría hablando de modelos de representación autocontenida.

- *Minería sobre los resultados* devueltos por un motor de búsqueda, intentando identificar patrones entre las páginas web recuperadas en una tarea de IR.

- Minería de estructura o representaciones **por contexto**. En este caso se analiza la estructura de un sitio web o de una colección de páginas web a traves de los datos relativos a su conectividad. Se analiza, por tanto, la topología del grafo de hipertexto. El análisis es de carácter global y considera tanto los *in-links* como los *out-links* presentes en un documento HTML.

- Minería de uso o representaciones **por uso**. En esta categoría se analizan principalmente los registros de acceso almacenados en los servidores web, conocidos como *web logs*, o las *cookies* aceptadas por un usuario. Una *cookie* es información que se guarda en el disco duro del cliente, a petición del servidor de la página que se esté visitando, para luego poder ser recuperada por el propio servidor en posteriores visitas. Se emplea para intentar conocer las preferencias de los usuarios o para facilitar el acceso a servidores que requieren autenticación. El ámbito de este tipo de representaciones puede ser local o global, y su fin último es encontrar patrones sobre el uso que se le da a un determinado sitio web, analizando las páginas más visitadas, los recorridos más habituales, etc. Dentro de este enfoque se pueden distinguir dos tipos de análisis:

  - Análisis de *patrones generales* de acceso. No interesan tanto los patrones de un visitante concreto como su integración en tendencias generales. Estos patrones se obtienen a partir del análisis de los datos que quedan registrados en los *web logs*. Con estos datos, el responsable de un sitio web (*webmaster*) puede reestructurarlo para facilitar el acceso a nuevos usuarios potenciales.
  - Análisis *personalizado de patrones* de acceso. En este caso lo que interesa es obtener datos sobre el comportamiento e interacción de cada usuario en particular, a fin de establecer perfiles de acceso, de forma que pueda ofrecerse un tratamiento personalizado a cada usuario. De un modo similar al anterior, se analizan los registros de acceso de los servidores web.

En la figura 3.2 se muestra gráficamente esta clasificación; además, se destaca la rama o categoría en la que puede centrarse la investigación presentada en esta tesis doctoral.

En el resto de este capítulo se realiza una revisión de diferentes modelos de representación de páginas web que pueden encontrarse dentro de cada una de las categorías en las que se divide la minería web, haciendo especial hincapié en aquellas representaciones que han sido utilizadas en sistemas de *clustering* y clasificación automática de páginas web. En este punto hay que destacar

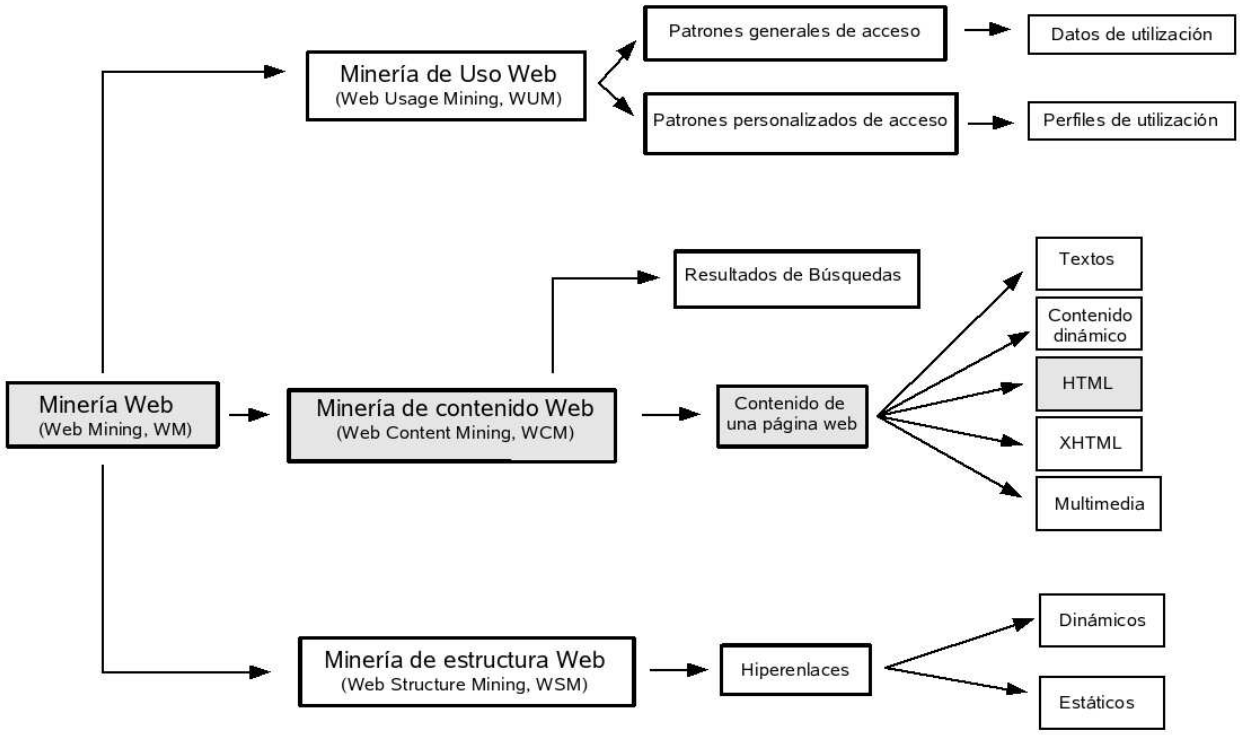

Figura 3.2: Mapa conceptual de la minería web.

la dificultad que conlleva la realización de una revisión sobre modelos de representación, ya que esta tarea por sí misma no tiene mucho sentido; la representación toma sentido cuando se aplica a un proceso de TC, DC, IR, etc. De hecho, en muchos trabajos, la representación no es tratada como el tema central de la investigación, sino como el paso previo a una tarea de IR, TC o DC.

Se revisan especialmente las representaciones *por contenido* y dentro de ellas, las representaciones autocontenidas, dentro de las cuales se enmarcan los modelos propuestos en esta tesis. Las representaciones *por contexto* se ven con cierto detalle debido a que en la mayoría de los casos se emplean en representaciones *mixtas* que combinan información de estructura con información extraída del análisis del contenido textual. Además, resultan los modelos de representación más utilizados. En el caso de las representaciones *por uso*, y al tratarse de representaciones más alejadas de los objetivos de esta tesis doctoral, se presentan unos pocos trabajos a modo de ejemplos ilustrativos.

La clasificación que aquí se realiza no puede ni debe considerarse de un modo estricto, ya que la mayor parte de los trabajos encontrados en la literatura tienen un caráter más bien transversal, de modo que podrían encajarse simultáneamente en varias categorías. De este modo, cuando se presenten representaciones *mixtas*, podrá hacerse allí donde uno de los enfoques que se esté combinando destaque de un modo especial frente al otro u otros.

Antes de entrar de lleno en la clasificación de los modelos de representación propiamente dicha, cabría destacar dos trabajos de investigación, (Ghani et al., 2001) y (Yang et al., 2002),

donde se presentan determinados aspectos que los autores definen como posibles "regularidades" a encontrar en los documentos HTML. El objetivo de estos trabajos era la exploración de diferentes hipótesis sobre la estructura de los hipertextos para intentar mejorar la clasificación automática de páginas web. Las conclusiones que pueden extraerse son también válidas para trabajos de DC, incluso, de IR.

Las "regularidades" que pueden encontrarse en un hipertexto y que deberían de ayudar a diseñar cualquier sistema de clasificación son:

- *Regularidad "No hipertexto"*. Cuando la información para determinar la categoría a la que pertenece un documento web está en la propia página. Ésta es la base de las representaciones autocontenidas. En este caso no se pretende sacar beneficio de la existencia de los hiperenlaces, por lo que la representación se puede usar con sistemas clasificadores de textos estándar. Además, en muchos casos, buscar información fuera del propio documento puede empeorar el comportamiento del sistema de clasificación (Yang et al., 2002).

- *Regularidad "Enciclopedia"*. Aparece cuando se supone que un documento perteneciente a una determinada categoría tiene enlaces sólo a documentos de esa misma categoría. Esta imposición resulta muy restrictiva y posiblemente no se encuentre en la mayoría de las colecciones web existentes; mucho menos en el caso de una selección aleatoria de páginas web. Sin embargo, refleja en cierto modo el concepto de páginas "hubs" definido por John M. Kleimberg en (Kleimberg, 1998), y que se verá en detalle más adelante.

- *Regularidad de "Co-referencias"*. Similar a la regularidad "*Enciclopedia*" pero, en este caso, la regularidad aparecerá sólo cuando los documentos de una determinada categoría compartan los destinos en sus enlaces. Por definición implica un preproceso de todos los documentos de una colección.

- *Regularidad "Preclasificada"*. Aparece cuando existe una precategorización como la de los directorios temáticos que ofrecen la mayor parte de los portales web y motores de búsqueda actuales. En estos casos, existen ciertas páginas web que contienen hiperenlaces a todos aquellos documentos que comparten una determinada categoría. Las categorías de Yahoo son un buen ejemplo de esta regularidad y bastaría con encontrar estas páginas que apuntan a los documentos de una misma categoría, en lugar de analizar los contenidos de cada uno de ellos. Estas páginas, caracterizadas por contar con un gran número de *out-links*, pueden encontrarse fácilmente si se representa cada documento HTML únicamente con los nombres de las páginas web a la que apunta.

- *Regularidad "MetaData"*. A menudo, la información de metadatos contenida en el elemento META (<META name="..." content="...") puede ser muy importante. El atributo "name" indica el tipo de información que se va a guardar y "content" almacena el contenido

de dicha información. En muchos casos, esta metainformación se genera desde fuentes externas y puede ser empleada como información adicional por sistemas capaces de explotar esta característica. El problema es el pequeño porcentaje de páginas que incluyen este tipo de elementos.

Estas regularidades han ido apareciendo posteriormente en numerosos trabajos de investigación, de ahí que se hayan destacado especialmente en este punto.

3.2.1. Representaciones por contenido

El análisis básico que se hace del contenido de un página web por parte, tanto de los motores de búsqueda en tareas típicas de IR, como de los sistemas de clasificación y *clustering* de páginas web, es el cálculo de la frecuencia de aparición de un rasgo en el contenido de texto de una página.

Con esta frecuencia se trata de establecer la importancia de un rasgo dentro del contenido de un documento y supone una primera aproximación al cálculo de su relevancia.

La ponderación de un rasgo se realiza por medio de funciones de proyección F y la similitud entre documentos –calculada por medio una función de distancia μ– suele estar parametrizada en base a estas ponderaciones. Sin embargo, como ya estableció Luhn, la frecuencia de aparición como función de proyección debe tratarse con sumo cuidado y corregirse para poder representar una función que asigne los mayores valores de relevancia a los rasgos más importantes dentro del contenido de un texto.

El hecho de que una función de ponderación tenga un carácter local o global trae consigo ciertas implicaciones cuando se aplican dentro de modelos de representación de páginas web *por contenido*. En este sentido, pueden destacarse dos aproximaciones fundamentales:

- En primer lugar, existen modelos de representación que, aún considerando únicamente el contenido presente en la propia página, requieren de cierta información adicional extraída de un análisis previo de una colección de referencia. Este es el caso de las representaciones que emplean funciones globales de ponderación; por ejemplo, una función TF-IDF. En este caso, sería necesaria información sobre la frecuencia inversa de documento en la colección para encontrar la ponderación de cada rasgo en la representación del documento y, por tanto, estas representaciones no podrían ser nunca representaciones autocontenidas. Si esta información se toma de alguna colección analizada previamente y se heredan las frecuencias inversas de cada rasgo en dicha colección, las representaciones generadas para cualquier otra colección podrían considerarse autocontenidas. En la revisión presentada en este capítulo se asumirá este hecho, aunque se presentarán como representaciones *por contenido*.

- Por otro lado, se tienen las representaciones verdaderamente *autocontenidas*; aquellas que no toman ninguna información externa a la página para generar una representación. Este

sería el caso de representaciones que empleen cualquier función de ponderación local como función de proyección dentro del modelo de representación.

Lo común en ambas aproximaciones es el hecho de que no se explora ni la estructura que forman los hiperenlaces, ni se buscan patrones de acceso al contenido web. De este modo, una página web no sería más que un documento electrónico con anotaciones relacionadas con aspectos tipográficos y estructurales –como puedan ser otros formatos documentales como el RTF– o algunos lenguajes de marcado creados con SGML o XML. Por tanto, en el enfoque "por contenido", la representación de páginas web se reduce básicamente a un problema de representación automática de textos.

Los primeros trabajos que presentaron métodos de clasificación automática o *clustering* de páginas web utilizaron representaciones basadas en contenido, construídas sobre un modelo de espacio vectorial con funciones de proyección clásicas, heredadas directamente de la representación automática de textos. Las funciones de proyección más empleadas en estos primeros trabajos fueron las funciones binaria, TF y TF-IDF. Como se indicó anteriormente, si la función de proyección es de carácter local la representación resultante será autocontenida, mientras que si la función es TF-IDF, deberá considerarse únicamente como una representación *basada en contenido*.

Uno de los primeros trabajos en el que se propuso una representación de páginas web basada en contenido, y no heredada directamente de una representación clásica de documentos de texto, fue (Molinari y Pasi, 1996). En este trabajo se parte de la suposición de que el etiquetado HTML puede aportar una información muy valiosa en tareas de IR y así, se considera que los elementos HTML añaden información al texto en diferentes niveles de importancia con respecto al contenido. La importancia de un rasgo en el contenido de una página web se evalúa en función de la frecuencia con la que dicho rasgo aparece en cada uno de los niveles considerados.

En este trabajó se planteó un modelo de representación que empleaba la lógica borrosa para mejorar el modelo de IR booleano (Baeza-Yates y Ribeiro-Neto, 1999) y obtener valores capaces de expresar diferentes grados de pertenencia en la relación rasgo-documento. En contextos diferentes al de las páginas web, la lógica borrosa ya había sido empleada con la misma finalidad [(Bookstein, 1981), (Cater y Kraft, 1989) y (Bordogna y Pasi, 1995)]. La asunción principal que realizaban los autores de (Molinari y Pasi, 1996) era que el etiquetado HTML suponía una indicación del autor acerca de la relevancia de los diferentes rasgos aparecidos en el documento, algo que, en palabras de los propios autores, se había estado buscando durante mucho tiempo en el seno de la comunidad IR.

Se seleccionó un conjunto de elementos HTML y se ordenó por importancia en las 12 categorías que se muestran en la tabla 3.1. Para cada término se evaluaba el grado de pertenecia a cada categoría por medio de una función TF-IDF donde el factor TF de la ecuación 2.24 estaba normalizado a la frecuencia máxima de un rasgo en la categoría. A continuación, se asignaba un

Ranking	**Criterio**	**Elementos HTML**
1	Title	TITLE
2	Header1	H1
3	Header2	H2
4	Header3	H3
5	Enphasized 1	TT, B, I, EM, STRONG, CITE, BLINK, BLOCKQUOTE, CREDIT, HP
6	Header4	H4, CAPTION, CENTER
7	Header5	H5
8	Header6	H6
9	Hypertext tags	HREF, IMG, LINK
10	lists	OL, UL, DL, MENU, DIR, TH, TR
11	Enphasized 2	SAMP, KBD, XMP, PRE, LISTING
12	Delimeters	P, TD, PLAINTEXT, texto no incluído en ninguna etiqueta

Tabla 3.1: Criterios y elementos HTML considerados en la representación (Molinari y Pasi, 1996).

peso $w \in [0, 1]$ diferente a cada categoría y, para ello, se empleaba la siguiente función general:

$$w_i = \frac{n - i + 1}{\sum_{i=1}^{n} i} \tag{3.1}$$

donde i representa el valor de '*ranking*' de cada categoría (véase la figura 3.1) y n es el número total de categorías consideradas.

Finalmente, se aplicaba una función borrosa $A : [0, 1]^m \rightarrow [0, 1]$ que daba como salida el grado de "cercanía" total de un término de consulta a un documento. La base de conocimiento del sistema borroso –el conjunto de reglas IF-THEN– era definida por el usuario a través de la interfaz gráfica.

En (Dumais y Chen, 2000) se exploraba la estructura jerárquica de una colección de páginas web para realizar tareas de clasificación automática de contenido web. Antes, cada documento HTML era preprocesado y se representaba considerando el texto visible que contenía, añadiéndo información extraída de las etiquetas META, el elemento IMG y su atributo ALT. Este atributo permite describir la imagen para los navegadores de sólo texto, así como etiquetar una imagen antes de que se cargue una página. En este caso, la función de proyección empleada fue la función binaria. En el caso de que la página web contara con varios frames, se extraía el texto de cada uno de ellos. Como este trabajo estaba orientado a TC se aplicaba posteriormente una función de reducción de rasgos MI para reducir el vocabulario, lo que hacía que, en ese momento, la representación dejara de ser una representación autocontenida.

La clasificación se realizaba por medio de un algoritmo de máquinas de soporte vectorial (*Support Vector Machines*, SVM). El objetivo era la clasificación de los documentos devueltos tras una consulta a un motor de búsqueda y, para ello, generaban automáticamente resúmenes y realizaban la clasificación a partir de ellos. La conclusiones de este trabajo se centraron fundamentalmente en las diferentes formas de emplear la estructura de la colección para el

aprendizaje y clasificación mediante SVM, siendo la representación de páginas web un problema no tratado en detalle.

En (Larsen et al., 2001) se realizaba un *clustering* de páginas web y la representación utilizada puede enmarcarse también dentro del modelo LSI con una función de proyección TF. Aunque esta función de ponderación sea de carácter local, el modelo LSI hace que la representación final generada nunca sea autocontenida. En (Sheikholeslami y Zhang, 1998) se presentaba un algoritmo para la generación automática de resúmenes de páginas web, donde se empleaba también una representación basada en un modelo LSI con una función TF-IDF como función de proyección dentro del espacio de representación.

En (Sebastiani, 2002) se revisan diferentes técnicas de aprendizaje automático que pueden aplicarse a tareas de TC. En el caso específico de la representación de documentos HTML, se plantea la posibilidad de extraer información sobre la estructura del conjunto de páginas web que forman una determinada colección, de un modo similar a (Dumais y Chen, 2000). La función de ponderación planteada para la representación de cada página fue la TF-IDF. Otro trabajo donde se empleaba un modelo de representación basado en contenido con función de proyección TF-IDF fue (Yu et al., 2004).

En (Yang et al., 2002) se realizaron estudios complementarios a los realizados en (Ghani et al., 2001), descritos en el apartado de representaciones autocontenidas, con tres colecciones de datos diferentes, tres algoritmos de clasificación y varias representaciones posibles de hipertexto. La intención era tratar de extrapolar conclusiones generales acerca de las regularidades propuestas inicialmente en (Ghani et al., 2001). En este trabajo, los autores probaron con dos representaciones por contenido y tres por contexto, generadas de diferentes modos:

1. Empleando únicamente el contenido textual de la página (*No Regularity*). En este caso se trataría de una representación por contenido que puede resultar autocontenida si se emplean funciones de ponderación locales.

2. Tomando el contenido de las etiquetas TITLE y META, y empleando una función de proyección TF. En este caso, se trataría de una representación autocontenida.

3. Tomando rasgos de los documentos enlazados (*Encyclopedia Regularity*). En este caso, se trataría de una representación por contexto.

4. Representando cada documento individualmente y usando una relación binaria para indicar los enlaces (*Co-referencing Regularity*). Al igual que en el caso anterior, se trataría de una representación por contexto.

5. Considerando los nombres de los documentos enlazados y desechando el contenido local de la página (*Partial Co-referencing*); de nuevo, se tendría una representación por contexto.

En (Glover et al., 2002) se realizaba una clasificación con SVM y para la fase de representación se tomaron los textos presentes en los hiperenlaces (*anchortext*) y en sus cercanías (*extended-*

anchortext). La representaciones se enmarcaron dentro del VSM y con funciones de proyección TF. Este trabajo concluía que, considerando únicamente los *anchortexts*, no se obtiene una mejora significativa frente al uso del texto presente en el documento como texto plano. Posteriormente, realizaron una fase de reducción de rasgos por medio de una función de entropía (véase ecuación 2.30).

Otros modelos de representación se apoyan en la suposición de que determinadas partes de un documento web pueden parecer más interesantes desde el punto de vista de las necesidades de un usuario que busca información, además de contar con el hecho de que determinados conjuntos de hiperenlaces suelen aparecer agrupados o en regiones específicas de las páginas web. Entre los estudios que analizan estos aspectos de diseño y navegación web cabe destacar el realizado por el SURL[1] y los del proyecto *Stanford Poynter Project*, dirigido por Marion Lewenstein.

Dentro de este enfoque se han propuesto métodos de segmentación de páginas web basado en visión (*Vision-based Page Segmentation*), donde un algoritmo detecta diferentes partes dentro de una página web, considerando que tienen diferente semántica. En (Yu et al., 2003) se emplearon separadores visuales como líneas en blanco que dividían la imagen, diferencias de color, o el texto y el tamaño de los diferentes nodos hijos. Este trabajo analizaba documentos XHTML, basándose en el modelo DOM (*Document Object Model*[2]) y empleaba como función de ponderación de un rasgo una variación de la función *m-estimation* que requiere de información relativa a la frecuencia de cada rasgo en cada documento del corpus de referencia. Como se ha indicado al inicio de esta sección, se puede considerar que esta información se hereda de colecciones previas, con lo que sería posible generar representaciones autocontenidas. En resumen, construye la estructura de los contenidos de la página en función de su distribución visual. De este modo, la recuperación de información podría mejorarse al permitir realizar consultas diferentes para cada una de las partes en las que se estructura el documento. En (Shih y Karger, 2004) se presentó un trabajo similar, en el que se proponía un método de clasificación automática donde, además de la localización visual de los contenidos textuales, se empleaban las cadenas de texto que forman las URLs.

En otros casos, los documentos web se representan a partir de un análisis de las cadenas que conforman sus URLs en tareas de clasificación automática [(Svatek y Berka, 2000), (Svatek et al., 2004) y (Kan y Thi, 2005)]. Estos trabajos están basados fundamentalmente en la información que ofrece la URL sobre la estructura de directorios existente en un determinado servidor web más que en el propio contenido de la página. Para ello, suele emplearse un conjunto de palabras clave que se buscan en las cadenas que forman las URLs.

Como se muestra en la figura 3.2, dentro de la Minería de contenido web cabe también el análisis del material multimedia presente en la propia página web. Algunos trabajos desarrollados en esta dirección son [(Srihari, 1995), (Ortega-Binderberger et al., 2000) y (Feng et al., 2003)], centrados fundamentalmente en el análisis de imágenes, audio y videos para tareas de

[1] Software Usability Research Laboratory
[2] htt://www.w3c.org/DOM/

recuperación de información multimedia. Estas aproximaciones se alejan del alcance de esta tesis doctoral, por lo que en esta memoria no se entra a describirlas.

Representaciones autocontenidas

La primera aproximación que puede hacerse a un modelo de representación de páginas web autocontenido sería cualquier representación que empleara funciones de proyección locales, como la función Binaria o la TF, dentro del VSM. Esta representación podría considerar el contenido de la página web como texto plano, o usar de alguna manera la información que aporta el etiquetado HTML.

En (Zamir et al., 1997) se realizaba un *clustering* de páginas web basándose en una definición de *cluster* como el conjunto de rasgos compartidos por todos los documentos contenidos en el propio *cluster*. Para ello, cada documento se representaba dentro de un modelo de representación con función de ponderación binaria. Los sistemas de *clutering* estaban pensados para ser aplicados sobre documentos de texto y no a páginas web, aunque Internet, con su heterogeneidad y su tamaño, ya ofrecía en ese momento un escenario idoneo para la experimentación en tareas de extracción de información y *clustering* de documentos.

En (Moore et al., 1997) se empleaba una representación "mixta" para clasificación automática y *clustering* de páginas web, donde destacaba la parte basada en contenido. Se consideraba como criterio de selección de texto el enfatizado, extraído de un conjunto de etiquetas HTML: TITLE, H1-H3, I, BIG, STRONG y EMPHASIZE. A continuación, consideraban únicamente el conjunto de rasgos que más veces había aparecido enfatizado. Los autores concluyeron que el enfatizado era un buen criterio de selección en conjunto, así como la selección de las 20 palabras más frecuentes en el documento.

Dentro de las representaciones autocontenidas cabe destacar el trabajo (Muslea et al., 1998), donde los autores argumentaban que las páginas web, al crearse para ser leídas, presentan determinadas convenciones estructurales que pueden resultar fácilmente identificables. De este modo, presuponían que la información de una página web se presentaría principalmente en una estructura jerárquica, así como en listas. En este trabajo se aplicaron heurísticas sobre algunos esquemas de anidamiento de elementos HTML que, suponían, solían repetirse en determinados tipos de documentos web.

Este enfoque estaba orientado a la extracción de información, de forma que se planteaban reglas sobre el contenido de determinadas etiquetas. De un modo similar, en (Cutler et al., 1997) y (Embley et al., 1999) las páginas web se trataban como árboles de elementos HTML anidados y, por medio diferentes heurísticas, se localizaban subárboles que contenían información relevante en un problema clásico de IR.

En (Merkl, 1998) se estudiaron los anidamientos que forman las etiquetas HTML en tareas de clasificación automática. La característica fundamental de todos estos trabajos es que se aplicaban sobre documentos con una estructura más o menos conocida, como pueden ser

páginas web dentro del ámbito universitatio: páginas personales, páginas web correspondientes a asignaturas, páginas de grupos de investigación, etc. Si se pretende encontrar un modelo de representación autocontenido capaz de cubrir la heterogenedidad presente en la web, estas heurísticas de carácter estructural pueden alejarnos del objetivo, por lo que habría que tratar de buscar heurísticas generales a todas las páginas web y no específicas de un determinado tipo.

En (Arimura et al., 2000) se pretendían extraer sintagmas a partir de patrones combinatorios; presentando una clase de patrones basados en técnicas de geometría computacional y emparejamiento de cadenas (*string matching*). El fin de estos métodos estadísticos era la extracción de palabras clave y, para ello, el documento se representaba por medio de una función de ponderación binaria. En este trabajo no se consideraba el etiquetado HTML como forma de enriquecimiento de la representación, pero en las conclusiones ya se apuntaba a que con el empleo de esta información añadida era de esperar que los resultados mejorasen.

En (Pierre, 2000) y (Pierre, 2001) se pretendía mejorar la clasificación automática de sitios web y, para ello, el autor proponía un modelo de representación basado en más componentes que el propio texto de la página web, enriqueciendo así la representacion con el etiquetado HTML. Se realizaba un estudio de algunos elementos como TITLE y META (<META name="description" content="..." y <META name="keywords" content="..."), así como del texto visible de la página, analizando en qué proporción estaban presentes.

De este estudio cabe destacar las siguientes conclusiones:

1. El elemento TITLE apareció en más del 95 % de las páginas consideradas y en casi un 90 % tenía una longitud entre 1 y 10 palabras. Trabajos anteriores indicaban que esta etiqueta sólo estaba presente en un 20 % de las páginas visitadas por el *WebCrawler* (Pinkerton, 1994).

2. Respecto al tamaño de los contenidos del elemento TITLE, el estudio de J. M. Pierre mostró que el elemento TITLE sólo aparecía en un 1 % de entre las páginas consideradas con más de 51 palabras.

3. Los contenidos dentro de elementos META (<META name="description" content="..." y <META name="keywords" content="...") aparecieron sólo en el 30 % de las páginas analizadas, dato que coincide con (Glover, 2001). Ambos resultados parecen indicar que no es conveniente basar un método de representación únicamente en información almacenada en este tipo de elementos.

4. Respecto al contenido de texto que aparece dentro de la etiqueta BODY, en casi el 60 % de los casos analizados se tenían más de 51 palabras y sólo el 17 % de las páginas web carecían de texto visible (contenido en la etiqueta BODY).

En este trabajo se compararon tres representaciones:

1. Representaciones generadas únicamente con el texto presente en el documento.

2. Representaciones que se pueden obtener a partir de los metacontenidos.

3. Representaciones que surgen de combinar el texto visible en la página con metacontenidos.

En este trabajo se empleó una representación por LSI con una función de ponderación TF-IDF para cada una de las representaciones anteriores. El hecho de considerar como espacio vectorial el LSI hace que, por definición, este modelo de representación nunca pueda generar una representación autocontenida. Sin embargo, dado que representa uno de los primeros análisis de las propiedades del etiquetado HTML, se incluye en este punto debido a la importancia que tienen estos análisis dentro de las representaciones autocontenidas. Además, este trabajo partía con la hipótesis fundamental de que el marcado HTML podía ayudar a mejorar la representación de páginas web en lugar de tener que buscar información externa a la página, idea sobre la que descansa toda representación autocontenida de páginas web.

Por último, la evaluación de sus resultados experimentales puso de manifiesto que el uso de metainformación podía mejorar la representación de documentos HTML frente a considerar únicamente el contenido del texto visible.

Los resultados obtenidos en este trabajo acerca de la naturaleza del contenido web, y relativos al elemento TITLE, coinciden en parte con estudios previos desarrollados en el marco de esta tesis doctoral. En la colección de páginas web utilizada en (Fresno y Ribeiro, 2001a) y (Ribeiro y Fresno, 2001) no se encontró ninguna página con un contenido en el elemento TITLE mayor de 11 rasgos (véase figura 5.1).

En (Asirvatham y Ravi, 2001) se analizaron diferentes tipos de representaciones de páginas web para tareas de clasificación automática. En este trabajo se apuntaba a que la representación basada en metaetiquetas (<META name="keywords" content="..."> y <META name="description" content="...">) presentaba un problema importante, pues en muchos casos, estos metacontenidos eran introducidos por el autor de la página web con la intención de mejorar la posición de la página dentro del conjunto total de documentos devueltos tras una consulta a un motor de búsqueda, en lugar de tratar de representar fielmente el contenido real del documento.

Respecto a los modelos de representación basados en el contenido textual de la página, los autores afirmaban, apoyándose en (Wong y Fu, 2000), que el 94,65 % de las páginas web contenían menos de 500 rasgos y que raramente estos presentaban una frecuencia mayor de 2 apariciones. Por este motivo, consideraban que los métodos basados en frecuencias no debían ser empleados en la representación de páginas web, proponiendo un método de representación y posterior clasificación que consideraba únicamente la estructura del etiquetado HTML.

Los autores de dicho trabajo asumían que las páginas pertenecientes a una categoría particular compartirían la estructura de los elementos HTML presentes en el documento. Para ello, fijaron una serie de estructuras típicas como son: páginas de "carácter informativo", páginas "personales" y de carácter "científico". A partir de una primera caracterización de cada una de estas tipologías se recogían determinadas variables estadísticas (por ejemplo, la relación entre

el tamaño del texto y el número de enlaces), lo que les permitía posteriormente realizar la clasificación. Aplicaron redes neuronales, y en la fase de entrenamiento se fijaban manualmente los pesos relativos a la ponderación de cada una de las variables estadísticas consideradas. En este trabajo se analizaron también las imágenes presentes en el documento, de forma que las páginas que presentaban muchos colores eran consideradas principalmente como páginas de tipo "informativo" o páginas personales (*homepages*). Consideraron, del mismo modo que en (Dumais y Chen, 2000), el texto alternativo que puede aparecer en una imagen (el atributo ALT del elemento IMG).

Si bien en (Asirvatham y Ravi, 2001) se obtuvieron buenas tasas de clasificacion, hay que destacar que la tipología de las páginas consideradas era muy reducida, y no podía llegar a representar la heterogeneidad presente en los contenidos web. No tiene la misma complejidad una clasificación de documentos HTML entre páginas del tipo "personales" y de carácter "científico", que entre páginas con contenidos relativos a "entidades bancarias" y "agencias de valores", por poner un ejemplo. En este último caso el problema es bien diferente, y la suposición acerca de que el contenido textual de un documento HTML no debería usarse en una representación del mismo, podría dejar de ser válida.

En (Ghani et al., 2001) se definieron las regularidades descritas en el punto 3.2 y se plantearon diferentes representaciones autocontenidas para su evaluación en una tarea de clasificación automática:

1. Representación "*Words on Page Only*"; basada únicamente en el texto presente en la propia página web.

2. Representación "*HTML Title*"; creadas empleando el contenido del elemento TITLE.

3. Representación "*HTML Meta Tags*"; considerando únicamente la metainformación presente en la página.

4. Representación "*Competitor Names*". Esta representación se generaba empleando determinadas palabras clave, relativas a nombres de empresas de un determinado sector.

5. Representaciones "*All Words from Linked Pages*" y "*All Words from Linked Pages Tagged*"; generadas considerando el texto de los documentos enlazados por la página a representar. En este caso, la representación era una representación por contexto.

Entre las conclusiones extraídas en (Ghani et al., 2001), así como en el trabajo posterior (Yang et al., 2002), se pueden destacar las siguientes:

- El hecho de identificar regularidades en el hipertexto y seleccionar una representación apropiada puede resultar crucial para el diseño de cualquier sistema de clasificacion automática.

- Añadir a los propios rasgos presentes en un documento los rasgos aparecidos en documentos enlazados no siempre mejora la clasificación. En sus experimentos, para una de las colecciones evaluadas, las tasas de clasificación empeoraron notablemente. Esta conclusión refuerza el hecho de buscar representaciones autocontenidas.

- El empleo de metadatos mejora enormemente los resultados de clasificación y así, habría que explorar técnicas para extraer y/o generar este tipo de información.

- Reconocer determinados campos HTML y usarlos para el diseño de sistemas de clasificación puede mejorar el comportamiento de los sistemas de clasificación.

Esta última conclusión se encuentra detrás de la hipótesis principal de esta tesis doctoral.

En (Riboni, 2002) se propuso una técnica para la representación de páginas web donde se extraía el texto contenido en las etiquetas BODY, META y TITLE para generar diferentes representaciones que, a continuación, fueron evaluadas por medio de un clasificador *Naïve Bayes* y un *Kernel Perceptron*. Como modelo de representación se empleó el LSI y como función de ponderación se empleó la función TF. Se definió otra función de proyección basada en TF que sobrevaloraba aquellos rasgos que aparecían en los elementos TITLE o META, utilizando un factor α que tomaba valores $\alpha = 2, 3, 4$ y 6 para recalcular el valor que asignaba la función de proyección F a cada rasgo. Así, se definía una función de ponderación w:

$$F: \ w = (\vec{t_i}, \vec{d_j}) = \alpha \ TF(\vec{t_i}, \vec{d_j}) \tag{3.2}$$

Los resultados mostraban que al representar un documento con el contenido de <META name="description"...> y TITLE se obtenían mejores valores de clasificación que cuando se empleaba únicamente el texto contenido en el elemento BODY y la función de proyección w. En este caso, el valor óptimo de α estaba en 3 para el *Kernel Perceptron* y en 6 para el clasificador *Naïve Bayes*. Al igual que en el caso de las representaciones utilizadas en (Pierre, 2000) y (Pierre, 2001), el estudio que se realizó del etiquetado HTML hace que en esta memoria se presente en este punto, aunque por deficición no pueda generar nunca representaciones autocontenidas.

Respecto a los resultados experimentales hay que remarcar dos hechos importantes. En primer lugar, en esta experimentación sólo se consideraron aquellas páginas que tenían representación con cada uno de los métodos, por lo que las conclusiones extraídas deben analizarse con sumo cuidado. Los valores de clasificación presentados en este trabajo no pueden extrapolarse a representaciones generales; como se ha dicho, la mayoría de las páginas web no tienen metadescripción y los contenidos del TITLE pueden ser, en muchos casos, generados automáticamente y no describir fielmente el contenido del documento. En segundo lugar, el hecho de que los mejores resultados en clasificación se obtengan con valores diferentes del factor α, para cada caso, indica que una representación de este tipo está muy condicionada por la colección de referencia que se esté considerando, ya que la función de ponderación dentro del

modelo es variable y debe ajustarse dependiendo de la colección. Una conclusión que sí parece clara es que cuando se disponga de metainformación, representaciones generadas con ella serán de mayor calidad que las que se pudieran obtener utilizando únicamente el contenido de texto visible de la página.

En (Liu et al., 2002) se realizaba una tarea de *clustering* jerárquico para visualización de la información. La representación de los documentos HTML se realizó dentro del modelo de espacio vectorial y considerando únicamente el texto presente en el documento. En este trabajo, los autores supusieron que la similitud entre documentos venía dada por un conjunto común de palabras clave (*keywords*) presentes en los documentos. La función de proyección empleada en este caso fue la función TF.

En (Molinari et al., 2003) se retomó el trabajo realizado en (Molinari y Pasi, 1996) y se propuso una representación de páginas web basada en una combinación lineal de categorías de elementos HTML, donde cada categoría representaba un nivel de importancia basado en conocimiento heurístico. Esta representación es una de las seleccionadas para la evaluación de las representaciones propuestas en esta tesis doctoral.

En ese trabajo de A. Molinari, G. Pasi y M.A. Marques Pereira se vuelven a plantear doce categorías ordenadas por importancia y se seleccionan, asociados a ellas, determinados subconjuntos de elementos y atributos HTML. El conjunto de etiquetas asignadas a cada clase difiere ligeramente del propuesto en (Molinari y Pasi, 1996) (tabla 3.1), fundamentalmente en el hecho de que inicialmente no se consideraban atributos, mientras que en este trabajo se toman tanto elementos como atributos. A continuación, se aplican funciones de normalización para encontrar el peso de cada rasgo en cada una de las categorías seleccionadas y estas contribuciones se combinan linealmente. En este caso, el modelo de representación no está tan unido a la tarea de IR como en (Molinari y Pasi, 1996) ya que, tras la asignación de los pesos por categoría a cada uno de los rasgos, se realiza siempre la misma combinación lineal no dependiente del usuario que realiza la búsqueda de información. Las 12 clases, así como sus elementos y atributos asociados se muestran en la tabla 3.2.

En (Molinari et al., 2003) se plantean dos funciones diferentes para encontrar el peso de un rasgo en cada categoría. La primera supone una frecuencia normalizada a cada categoría WTF_{tag} (la frecuencia de un rasgo en una categoría dividido entre la frecuencia máxima en dicha categoría), por lo que las representaciones generadas serán representaciones autocontenidas. La segunda es una función basada en la TF-IDF, lo que hace que la representación deje de ser autocontenida. De este modo, en el primer caso, la función que debe aplicarse para encontrar el peso de un rasgo en una página es:

$$F : F(\vec{t}, \vec{d}) = \sum_{i=1...n} w_i \ . \ WTF_{tag(i)}(\vec{t}, \vec{d}) \tag{3.3}$$

donde i representa cada una de las categorías consideradas y w_i se calcula según la ecuación 3.1.

Criterio	**Elementos HTML**
Title	TITLE, META keywords
Header1	H1, FONT size="7"
Header2	H2, FONT size="6"
Header3	H3, FONT size="5"
Linking	HREF
Enphasized	EM, STRONG, B, I, U, STRIKE, S, BLINK, ALT
Lists	UL, OL, DL, MENU, DIR
Enphasized2	BLOCKQUOTE, CITE, BIG, PRE, CENTER, FONT size="4"
Header4	H4, CAPTION, CENTER, FONT size="4"
Header5	H5, FONT size="3"
Header6	H6, FONT size="2"
Delimiters	P, TD, texto no incluído en ninguna etiqueta, FONT size="1"

Tabla 3.2: Criterios y elementos HTML considerados en la representación (Molinari et al., 2003).

Esta función, que da lugar a una representación autocontenida, se emplea en la experimentación presentada en esta tesis doctoral, comparándola con las representaciones propuestas.

En el otro caso, que llamaremos representación por contenido de Molinari, la expresión correspondiente a la relevancia de un rasgo en un documento viene dada por:

$$F : F(\vec{t}, \vec{d}) = \sum_{i=1...n} w_i \; . \; WTF_{tag(i)}(\vec{t}, \vec{d}) \; . \; IDF(\vec{t}, \vec{d}) \tag{3.4}$$

Otros trabajos como (Wang et al., 2003) han propuesto representaciones de páginas web autocontenidas basadas en el VSM y con funciones de ponderación binaria y TF.

En (Fathi et al., 2004) se expone un método de representación basado en la combinación de la información textual contenida en la página, el texto presente en los enlaces, así como en las etiquetas TITLE y META. Se consideró una frecuencia normalizada WTF como función de proyección que luego se combinaba con una función MI, de modo que obtenía un nuevo método de reducción de rasgos basado en el análisis del contenido textual de una página web. A partir de la representación obtenida, se evaluaron diferentes algoritmos de clasificación automática, en concreto: 2 clasificadores bayesianos, un clasificador basado en los k vecinos más cercanos y el clasificador ARC, propuesto por los propios autores.

Además de la representación propuesta en (Molinari et al., 2003), en esta tesis se evalúan otras representaciones autocontenidas de páginas web: una representación creada a partir de la función de proyección TF y otra basada en información del elemento TITLE. Las características de cada una de ellas, así como las del resto de funciones evaluadas, aplicadas tradicionalmente a la representación de textos, se describen en el capítulo 7 de esta memoria.

3.2.2. Representaciones por contexto

El análisis de la estructura del grafo de hipertexto que forma la Web ha sido muy utilizado en tareas de representación de páginas web, principalmente en el ámbito de la IR (Mehler et al., 2004). Por este motivo, los ejemplos más conocidos de explotación de esta información de estructura son, probablemente, los aplicados en la mejora del comportamiento de los sistemas de IR (Getoor, 2003).

Trabajos centrados en clasificación automática y *clustering* de documentos HTML han empleado también este tipo de representaciones, casi siempre heredadas de trabajos anteriores desarrollados en el campo de la IR (Getoor, 2003). Sin embargo, las características propias de los problemas de TC y DC hacen que pueda resultar más adecuada una representación basada en contenido.

El análisis de la estructura del hipertexto se ha empleado en la representación de páginas web en tareas de clasificación automática porque se considera que una colección de páginas web posee una estructura de hiperenlaces muy rica, de la que es posible extraer información suficiente para poder realizar una buena clasificación. Esta hipótesis está muy condicionada a la existencia de colecciones de referencia donde poder probarla. Algunos de los primeros trabajos donde se realizaba un análisis de la estructura jeráquica de colecciones web fueron [(Koller y Sahami, 1997), (Weigend et al., 1999) y (Ruiz y Srinivasan, 1999)].

Otro trabajo inicial que analizaba la estructura del grafo de hipertexto para la representación de páginas web fue (Botafogo et al., 1992). En él se establecía una matriz de distancias entre nodos dentro del grafo de hipertexto que formaban los documentos HTML dentro de una parte de la web de la Universidad de Mariland. Este trabajo representa un estudio muy serio sobre la aplicación de diferentes métricas para la identificación de tipos de hipertexto y la identificación de jerarquías dentro de una estructura de grafo. Siendo un estudio más cualitativo que cuantitativo, concluye que el hecho de analizar la estructura de una colección de páginas web puede ser un buen punto de partida para la generación automática de mapas y para la identificación de los nodos más importantes dentro de un grafo de hipertexto.

En trabajos posteriores se han presentado diferentes representaciones por contexto que exploran la información subyacente en los hiperenlaces. La hipótesis principal en la que se apoyan estas representaciones es que el contenido de los enlaces, y a veces incluso el contenido de la propia página que los contiene, puede ser muy representativo del contenido de la página web a la que se esté apuntando.

El análisis del texto que aparece en los enlaces, *anchortext*, el texto presente en la cercanía de los mismos, *extended-anchortext*, o los conceptos de *authorities* y *hubs* presentados por John M. Kleimberg en (Kleimberg, 1998), y que se verán más adelante, constituyen la base de gran parte de las representaciones por contexto encontradas en la literatura. Asimismo, la publicación del algoritmo PageRank (Brin y Page, 1998) por parte de Sergey Brin y Lawrence Page, y su eficacia probada como base del algoritmo que emplea *Google*, supuso un espaldarazo fundamental a las

representaciones basadas en el análisis de relaciones entre hiperenlaces.

En la mayoría de los trabajos encontrados en la literatura esta información de contexto se combina con información basada en contenido, formando representaciones mixtas. Los trabajos que se presentan en este apartado no están centrados en el análisis del contenido del documento, para lo que emplean generalmente funciones de ponderación clásicas aplicadas a textos (binaria, TF o TF-IDF), sino en el análisis de la estructura de hiperenlaces que forma la Web. En (Getoor, 2003) puede encontrase una revisión de diferentes métodos de representación de páginas web *por contexto* (también conocido como *Link Mining*).

Representaciones basadas en Anchortexts y Extended-Anchortexts

En (Chek, 1997) se describía una representación de páginas web mixta, donde destacaba una parte importante de representación basada en contenido. Con el fin de representar y luego clasificar documentos web desde el campo del aprendizaje automático, en este trabajo se empleaba información extraída de los hiperenlaces, de las etiquetas del elemento TITLE y de los encabezados HTML, así como de las cadenas de texto de las propias URLs.

A partir de estos elementos HTML se trataba de extraer información sobre la estructura de una página web para enriquecer su representación y mejorar así la clasificación por medio de un algoritmo *Naïve Bayes*. Se consideraba primero una representación basada en contenido, donde se toma únicamente el título y los encabezados. En este caso se suponía que las páginas web estarían tituladas con palabras que describirían el contenido de las mismas. Los resultados experimentales concluyeron que usando estos elementos se podría capturar información relevante acerca del contenido de la página web. A continuación, combinaron elementos de estructura tomando las palabras que aparecían subrayadas en todos los enlaces que apuntan a una determinada página web, los *anchortexts* de los *outlinks* de una página web. Se basaban en la siguiente hipótesis: los creadores de las páginas web subrayan con etiquetas de hiperenlaces aquellas palabras que describen el contenido de la página apuntada por el hiperenlace. La principal conclusión que se extrae de este estudio es el hecho de que la combinación de representaciones por contenido y contexto podía mejorar la tarea de clasificación bayesiana de páginas web.

En (Graham, 1997) se consideraba que los hiperenlaces aportaban una información importante relativa a la relación entre documentos, pero suponían que estos, por sí sólos, no explicaban las razones de dicha relación. Por ello, analizaron los atributos REL y REV del elemento <A>. Este elemento permite definir un hiperenlace y sus atributos tratan de establecer las relaciones entre los documentos enlazados, de forma que REL describe la relación entre el documento origen y el destino, y REV describe un vínculo inverso desde el origen. A partir de esta información se intentan establecer jerarquías entre páginas. La idea de emplear estos *anchortexts* se repetirá más adelante en numerosas representaciones como [(Brin y Page, 1998), (Kleimberg, 1998), (Chakrabarti et al., 1998b), (Glover et al., 2002) y (Richardson y Domingos,

2002)].

En (Kaindl et al., 1998) y (Lawrence y Giles, 1998) se planteó explícitamente como un problema el hecho de que los buscadores emplearan únicamente representaciones por contenido y no consideraran la estructura que forma en sí misma la Web. Así, combinaron estructura y contenido, con funciones de ponderación binaria y TF dentro de la parte de representación basada en contenido. La representación se enmarcaba dentro del modelo de espacio vectorial.

En (Slattery y Craven, 1998) y (Craven et al., 1998b) se clasificaban documentos HTML con un algoritmo *Naïve Bayes*. Como probabilidad a priori se tomaba la función *m-estimation*, basada en la combinación de la frecuencia de un término en una página con su frecuencia inversa en la colección, y normalizada al tamaño del vocabulario. En esta representación se consideraba la aparición de una palabra en el propio texto de la página, los *anchortext*, los *extended-anchortext*, así como el hecho de que la palabra comenzara en mayúsculas o tuviera caracteres alfanuméricos. En estos experimentos se realizaba una clasificación automática buscando relaciones entre documentos, mostrando que, en algunos casos, sus algoritmos de aprendizaje de primer orden empleados funcionaban mejor cuando se usaba esta información frente a los clasificadores que usaban representaciones de textos basadas únicamente en frecuencias de aparición.

En (Bharat y Henzinger, 1998) se presentó una mejora sobre un algoritmo de selección de rasgos presentado en (Chakrabarti et al., 1998b), que empleaba también los *anchortexts*, reforzando heurísticamente el peso que daba a aquellas páginas que se encontraban en diferentes dominios pero con contenidos similares. En este trabajo, orientado a tareas de IR, se utilizó el VSM y la función de ponderación TF-IDF.

En (Attardi et al., 1999) se describe un sistema de clasificación de páginas web basado en representaciones mixtas y se compara con diferentes sistemas. En la parte correspondiente a la representación por contexto se consideraron los *anchortexts*, mientras que en el análisis del contenido textual de la página se consideró la estructura de los elementos HTML, a la vez que se guardaba información relativa al título, los encabezados y las listas. En este trabajo se descartó explícitamente el uso de elementos relacionados con el enfatizado (<EM> y <B>) y relativos al color de la fuente (<FONT>) y el centrado (<CENTER>).

En (Furnkranz, 1999) se estudiaron también los *anchortext* y *extended-anchortext* para predecir la categoría del documento al que se estaba enlazando. En este trabajo se empleó, además, el texto de los encabezados de las secciones donde aparecieron los enlaces y los resultados de su clasificador basado en reglas mejoraron un 20 % respecto de los obtenidos cuando consideró únicamente el texto presente en la propia página.

Otros trabajos donde pueden encontrase representaciones basadas en *anchortext* y *extended-anchortext* son [(Vlajic y Card, 1998), (Getoor et al., 2001), (Jin y Dumais, 2001), (Thelwall, 2001), (Halkidi et al., 2003), (Chakrabarti et al., 2003) y (Song et al., 2004)].

Representaciones basadas en el análisis de co-referencias

En (Armstrong et al., 1995) y (R. Armstrong y Mitchell, 1995) se describía el sistema WebWatcher, un agente que asiste al usuario en la localización de información en páginas web. En este caso, el método de representación empleado por el sistema ignoraba el contenido de texto y se centraba únicamente en el conjunto de páginas web que apuntaba a un documento dado; se consideraba que una página quedaba representada por los documentos con hiperenlaces apuntándola. Se asumía, por tanto, que dos páginas serían similares entre sí siempre que existiera una tercera con enlaces hacia ambas.

La hipótesis que reside tras esta representación comparte muchas de las ideas que se encuentran en las representaciones posteriores de Brin y Page, *PageRank* (Brin y Page, 1998), y Kleimberg, *authorities & hubs* (Kleimberg, 1998), por lo que puede considerarse como un trabajo pionero en el campo de las representaciones por contexto.

Dentro de este tipo de representaciones, cabe destacar también el trabajo de (Moore et al., 1997), donde se aplicaban reglas de asociación para tareas de clasificación y *clustering* de páginas web. Para la representación de los documentos se consideraba la estructura del hipergrafo que formaba la colección de páginas web que se estaba considerando, además de un análisis del contenido textual. Los pesos entre los nodos se establecían en función de la confianza media entre la reglas de asociación basadas en la coaparición entre conjuntos de palabras.

De un modo parecido, en (Zamir et al., 1997) se realizaba un *clustering* de páginas web tomando métricas basadas en coapariciones entre los *clusters*, considerando tanto rasgos aislados (*word-intersection*) como sintagmas (*phrase-intersection*). Un análisis similar se realizó en (Pitkow y Pirolli, 1997). En el caso de los rasgos monograma se empleaba una representación TF, mientras que cuando usaban sintagmas, la cohesión se tomaba como la longitud del sintagma común más largo. Los resultados de este estudio no fueron concluyentes.

En (Chen y Dumais, 2000) se empleó una representación similar, pero la función de proyección empleada para el análisis del contenido fue la función binaria, aplicada en algoritmos de aprendizaje sobre representaciones basadas en el VSM. Este trabajo estaba orientado a la mejora de la clasificación automática mediante el desarrollo de interfaces gráficas.

Soumen Chakrabarti y su equipo estudiaron el uso de co-citaciones en una colección de patentes pertenecientes a IBM (Chakrabarti et al., 1998c) en tareas de IR. Supusieron que estas citaciones eran similares a los hiperenlaces presentes en la Web y así definieron diferentes categorías dentro de una jerarquía temática. Sus resultados mostraron una reducción del error del 31 % comparado con el caso en el que se consideraba únicamente el texto presente en el documento. Consideraron también el texto de un documento enlazado como texto propio de la página web a representar; en este caso, bajaron la tasa de error un 6 % en un problema de IR. El tratamiento que se dió al texto fue de texto plano, es decir, no consideraron la información que aporta el etiquetado HTML, usando una función de ponderación TF dentro de un VSM.

En otro trabajo más centrado en tareas de TC (Oh et al., 2000a), se tomaron los *anchortexts*

como si fuera texto perteneciente al propio documento y, en este caso, el comportamiento de un clasificador *Naïve Bayes* empeoró en un 24 % frente a cuando consideraban únicamente el texto presente en el propio documento. Después, en lugar de tomar todos los hiperenlaces que apuntaban al documento, se seleccionaron únicamente los presentes en un subconjunto de los documentos enlazados. Para la selección de este subconjunto, se empleó como métrica de selección la función coseno. En este caso los resultados mejoraron en un 7 % respecto del caso en el que se consideraban todos los enlaces.

Como ya se ha visto, estos métodos requieren información acerca de los *in-links* y los *out-links* de un documento, lo que supone un análisis previo e intensivo del corpus considerado. Este hecho hace que consuman mucho más tiempo y recursos que otras técnicas de representación más clásicas (Riboni, 2002).

En (Ester et al., 2002) se consideró el conjunto de páginas contenidas en un sitio web como un gran documento HTML y se utilizó en tareas de clasificación automática aplicando modelos ocultos de Markov. Se seleccionaron determinados tipos de páginas web, relativos al ámbito empresarial, y se analizó su estructura de hiperenlaces estableciendo el número mínimo de enlaces como métrica de distancia entre dos páginas dentro de un mismo sitio web. En (Sugiyama et al., 2003) se propuso una redefinición de la función de ponderación TF-IDF que tenía en cuenta los contenidos de los documentos HTML enlazados a uno dado.

Authorities & Hubs

En (Kleimberg, 1998) los resultados obtenidos tras una consulta a un motor de búsqueda se organizaron seleccionando un pequeño subconjunto de páginas a las que se denotó como "*authoritatives*" y "*hubs*". Las primeras estaban caraterizadas por tener muchos *in-links*, considerándolas por ello buenas fuentes de información. Las segundas eran consideradas buenas fuentes de hiperlenlaces y se definieron bajo la hipótesis de que muchas páginas web contienen enlaces que apuntan a otras páginas del mismo tipo. Se trataría, por tanto, de páginas web con muchos *out-links*.

Por lo tanto, desde el punto de vista de la estructura de hiperenlaces que forma la Web, pueden distinguirse páginas consideradas como muy "importantes" en un determinado ámbito y páginas con un gran número de enlaces a estas páginas "importantes". John M. Kleimberg propuso el algoritmo HITS para identificar estos dos tipos de documentos en la Web.

Dado un término de consulta, el algoritmo HITS trata de encontrar *authorities* y *hubs*. HITS asigna un peso no negativo del carácter *authoritative* ($x^{<h_k>}$) y un peso, también no negativo, del carácter *hub* ($y^{<h_k>}$). Los pesos de cada tipo son normalizados de forma que la suma de sus cuadrados sea igual a 1.

Siendo $h_1^{'}, \ldots, h_n^{'}$ el conjunto de páginas que apuntan a h_k, y $h_1^{''}, \ldots, h_m^{''}$ el conjunto de

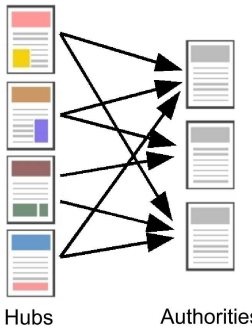

Figura 3.3: *Authorities* y *Hubs*.

páginas a las que apunta h_k, entonces:

$$x^{<h_k>} = \sum_{\{(v_i,v_j)|v_i=h_k \forall j=1...n\}} y^{<h_k^{'}>} \tag{3.5}$$

$$y^{<h_k>} = \sum_{\{(v_i,v_j)|v_j=h_k \forall j=1...m\}} x^{<h_k^{''}>} \tag{3.6}$$

Krishna Bharat y Monika R. Henzinger señalaron que el algoritmo HITS no funcionaba bien en todos los casos por tres razones fundamentales (Bharat y Henzinger, 1998):

- *Relación de refuerzo mutuo entre servidores web*. En algunas ocasiones, un conjunto de documentos de un servidor apunta a un único documento de un segundo servidor. Esto aumenta los valores *hubs* de los documentos del primer servidor y los valores *authoritative* del segundo. En el caso contrario, cuando un único documento de un servidor apunta a muchas páginas del otro, se tiene el mismo problema.

- *Hiperenlaces generados automáticamente*. En ocasiones, cuando una página web es generada automáticamente por una herramienta, ésta introduce hiperenlaces automáticos.

- *Nodos no relevantes*. La vecindad del grafo a menudo contiene documentos no relevantes para una consulta. Si estos nodos están bien conectados, surge el problema: las *authorities* y las *hubs* más puntuadas no tienden a indicar el tema original. Por ejemplo, si se buscan páginas por medio de una consulta con los términos "música de cámara", el algoritmo llevará al tema general "música".

La mejora que plantearon Bharat y Henzinger se basó en la suma del contenido de texto de la página web a la estructura de grafo. Para cada nodo se consideró la relevancia de la consulta al tema calculando un peso de relevancia al tema, $W(h_k)$. De este modo, los pesos corregidos

"authoritative" $-A(h_k)-$ y "hubs" $-H(h_k)-$ de un nodo h_k se expresan como:

$$H(h_k) = \sum_{\{(v_i,v_j)|v_i=h_k \forall j=1...n\}} A(h_k^{'}).W_{hub}(h_k^{'}) \tag{3.7}$$

$$A(h_k) = \sum_{\{(v_i,v_j)|v_j=h_k \forall j=1...n\}} H(h_k^{''}).W_{hub}(h_k^{'}) \tag{3.8}$$

El motor de búsqueda Clever[3] de IBM implementó el algoritmo de Kleinberg, aunque con el tiempo éste fue mejorado con las ideas planteadas en (Chakrabarti et al., 1999a). La mejora propuesta por Bharat y Henzinger fue utilizada en el Focused Crawler (Chakrabarti et al., 1999b), también de IBM.

En (Bharat y Henzinger, 1998) se empleó una función de ponderación TF-IDF en el análisis de *authorities* y *hubs*. También Mark Craven y Jean Slattery encontraron que combinando dos algoritmos que exploraban la relación estructural de las etiquetas HTML y la topología de los hiperenlaces, con el programa FOIL –*First Order Inductive Learning* (Quilan, 1990)– y el *Hubs & Authorities style algorithm*, la clasificación automática mejoraba (Craven et al., 1998b).

Otras representaciones como (Chakrabarti et al., 2001) consideraron como información el texto presente, tanto en los árboles, como en los subárboles que forman las etiquetas de marcado HTML, así como el texto presente en los hiperelenlaces entre páginas. El análisis estructural se llevó a cabo con el algoritmo HITS. En este trabajo, esta información se empleaba en tareas de "destilación de tema" (*topic distillation*); es decir, dado un tema, encontrar un número pequeño de páginas que sean representativas de dicho tema. Un trabajo que tomó como punto de partida estas ideas para realizar *clustering* de páginas web fue (Hou y Zhang, 2003).

El siguiente paso fue considerar la propagación en esta característica *authoritative* asignada a las páginas web, lo que supone la idea fundamental del algoritmo PageRank (Brin y Page, 1998). Como ya se ha apuntado, este algoritmo resultó revolucionario y ha sido empleado posteriormente en buscadores como *Google*, NEC NetPlaza[4] y en el IBM Clever.

PageRank

Toda página web tiene un número de enlaces de salida, *out-links*, y otro número de enlaces de entrada, *in-links*. Accediendo a una página no es posible saber el número exacto de *in-links* que tiene, pero si se descarga, en ese mismo instante se conoce su conjunto total de *out-links*. A partir de esta idea, Sergey Brin y Lawrence Page (cofundadores de *Google Inc.*) establecieron la bases del algoritmo PageRank. En realidad no supone más que un análisis de co-referencias en el contexto web, un tipo de análisis que ya había sido muy estudiado en el mundo de la documentación electrónica y la gestión documental en dominios semánticos restringidos.

[3]http://www.almaden.ibm.com/cs/k53/clever.html
[4]http://netplaza.biglobe.ne.jp/

Las páginas web difieren mucho unas de otras en el número de *in-links* que tienen; por ejemplo, sitios web como *Google* o *NSN*[5] tienen muchos *in-links*, pero la página personal, por ejemplo, de un profesor ayudante en una universidad tendrá muy pocos. El número de referencias, o citaciones, que tiene un objeto puede ser una evidencia clara de su importancia. Considerando esto, se puede asumir que una página muy citada será más "importante" que una página con pocos *in-links* (Brin et al., 1998).

Sin embargo, el algoritmo PageRank [(Brin y Page, 1998), (Brin et al., 1998)], en palabras de sus creadores, es mucho más que contar el número de *in-links* de una página. La razón es que hay muchos casos donde el simple conteo de referencias no se corresponde con el sentido de "importancia" que guía nuestro sentido común. Por ejemplo, no es lo mismo que un documento HTML tenga un *in-link* desde la página principal de *MSN*, que desde la página personal de un individuo anónimo. Si un documento es referenciado desde una página "importante", deberá ser considerado como más importante que otra página web que, aún teniendo un número mayor de *in-links*, estos provengan de páginas "poco importantes".

Este algoritmo analiza el grafo que constituye la Web del siguiente modo. Se supone que una página h_k tiene un conjunto de páginas que la apuntan $h_i^{'}, \ldots, h_n^{'}$. El parámetro d es un factor de "amortiguación" que toma valores en el intervalo $[0, 1]$. En (Brin y Page, 1998) se supone un valor $d = 0,85$. Por otro lado, $C(h_k)$ es el número de páginas a las que apuntan los hiperenlaces de h_k. Así, el valor que asigna el algoritmo PageRank a una página h_k viene dado por la expresión:

$$PageRank(h_k) = (1 - d) + d.(\frac{PageRank(h_1^{'})}{C(h_1^{'})}) + \ldots + PageRank(\frac{PageRank(h_n^{'})}{C(h_n^{'})}) \qquad (3.9)$$

Esta expresión crea una distribución de probabilidad sobre el conjunto de páginas web que considera, de forma que la suma de todos los valores PageRank de todas las páginas es 1.

El *ranking* de una página se divide de modo uniforme sobre sus *out-links* para contribuir a los *rankings* de las páginas a las que apuntan. El algoritmo es recursivo, pero si se empieza por cualquier conjunto de *rankings* y se itera, el valor llega a converger (Brin y Page, 1998). Este algoritmo requiere de unas pocas horas para calcular el *ranking* de millones de páginas. El buscador *Google* combina este algoritmo con otros parámetros, como los *anchortexts* de sus *in-links* y el texto presente en la propia página. En la actualidad, el algoritmo exacto que aplica *Google* no es público, dado que la empresa cotiza en bolsa y basa gran parte de su éxito en la eficacia de su PageRank.

Algunos trabajos han tratado de complementar este algoritmo mejorando el análisis de los contenidos de la páginas web. En (Richardson y Domingos, 2002) se empleó una función de ponderación TF-IDF, previa eliminación de todo el etiquetado HTML. Las representaciones propuestas en esta tesis podrían seguir esta línea y aplicarse en el análisis del contenido de las

[5] http://www.msn.com

páginas web.

En (Lu y Getoor, 2003) se empleó una representación mixta para clasificación automática. Se compararon representaciones basadas únicamente en el contenido textual de la página con diferentes representaciones obtenidas a partir del análisis de los *in-links* y *out-links*, obteniendo mejores resultados en este último caso. La representación por contenido se realizó mediante representación por bolsa de palabras, con una función de ponderación TF.

Otros trabajos donde se han empleado *authorities & hubs* y co-referencias son (Weiss et al., 1996), (Piroli et al., 1996), (Mukherjea y Hara, 1997), (Dean y Henzinger, 1999), (Lempel y Moran, 2000), (Wang y Kitsuregawa, 2001), (Wang y Kitsuregawa, 2002), (Wang y Su, 2002), (Chirita et al., 2003) y (Ding et al., 2004).

3.2.3. Representaciones por uso

En (Cheung et al., 1998) los autores desarrollaron un agente basado en aprendizaje para descubrir los temas que interesaban a los usuarios web. Conociendo el perfil de un usuario, el sistema podía llevar a cabo una búsqueda o un acceso inteligente a la información. Definieron un perfil de usuario como un conjunto de vectores de temas y cada uno de estos vectores se construía con una colección de palabras clave con unos determinados pesos. Esencialmente, cada vector representaba un tema de interés para el usuario y los pesos asociados a las palabras clave en un vector de temas medía el interés relativo de dichas palabras. De este modo, si se aplicaran técnicas similares para capturar perfiles en sitios web, sería posible emparejar el perfil de un usuario con el de un sitio web y crear sistemas capaces de recomendar al usuario.

En este trabajo, a cada usuario se le asociaba un conjunto de registros dentro del *web log* que guardaban las rutas que había seguido dentro del sitio web. Cada una de las páginas visitadas pasaba a ser representada como un vector de documento. Al igual que en el caso de los vectores de temas, estos vectores de documento estaban formados por conjuntos de pares rasgo-peso. Un rasgo era incluido en un vector de documentos si aparecía en alguna de las páginas visitadas por el usuario. Estos pesos podían calcularse con cualquier función de ponderación F. Después de procesar todas las páginas visitadas, se tenía un conjunto de vectores de documento para cada usuario. El siguiente paso para construir el perfil de usuario era agrupar los vectores de documento. El algoritmo Leader (Hartigan, 1975) fue el que se empleó para esta fase de *clustering*. La similitud entre documentos se medía con el producto escalar. Cuando se quería introducir un nuevo documento en la pila de páginas agrupadas, las distancias a los centroides eran recalculadas y el nuevo documento se añadía al *cluster* con el centroide más cercano. Tras esta fase de *clustering* se extraía el vector de temas.

En (Mobasher et al., 2000) se emplea una representación mixta para realizar tareas de personalización de contenidos web. En ella se integra la minería web por contenido, empleando una función de proyección TF-IDF, con la minería web por uso. En este trabajo se trataba de identificar perfiles de uso y de contenido.

En (Smith y Ng, 2003) se realiza una tarea de *clustering* de páginas web por medio de mapas autoorganizativos (*Self-Organized Maps*, SOM) para la asociación de patrones de navegación de usuarios. Para ello, se analizan los registros de acceso de los servidores web identificando diferentes usuarios y sesiones, y representándolos como vectores que luego servirán como datos de entrada al SOM.

Otros trabajos donde se realiza un análisis de patrones personalizado son (Bezdek y Pal, 1992), (Cooley et al., 1997) y (Widyantoro, 1999).

En el caso de [(Cooley et al., 1997), (Srikant, 1998), (Borges y Levene, 1999) y (Felzenszwalb et al., 2003)], por el contrario, el análisis del uso del contenido web se realiza para la búsqueda de patrones generales.

3.3. Conclusiones

Los modelos de representación de páginas web encontrados en la literatura difieren principalmente en el enfoque en el que se encuentran dentro de la Minería Web. Aunque muchos de ellos poseen un carácter mixto, un análisis detallado de cada enfoque por separado permitirá combinarlos mejor y encontrar nuevos modelos con los que se puedan mejorar procesos posteriores de TC y DC.

La obtención de una buena representación de páginas web es una tarea que resulta muy dependiente de los procesos que se quieran aplicar a continuación. De este modo, una representación para tareas de IR no debería considerar los mismos elementos que en el caso de que querer aplicarse a problemas de TC o DC, en las que el contenido del documento puede resultar más relevante.

En el caso de la *web usage mining*, los diferentes modelos realizan un análisis del uso que se hace de la propia Web –qué enlaces se siguen dentro de un sitio web y en qué orden– para mejorar el diseño y la estructura con la que deben presentarse los contenidos en un portal web.

Dentro de la *web structure mining*, un documento HTML se considera como un nodo dentro del grafo de hipertexto que forma la Web y se emplea la propia estructura del hipertexto para enriquecer las representaciones. Según los creadores del algoritmo PageRank, usado por *Google*, el tiempo empleado en analizar millones de páginas es de unas pocas horas, lo que implica que deberá tardar del orden de cientos o miles de horas en analizar los billones de páginas que, a día de hoy, dicen tener indexadas. Si bien el éxito de este buscador certifica la calidad de sus algoritmos de recuperación de información y, por tanto, de sus modelos de representanción, cuando lo que se desea es aplicar procesos sobre el contenido textual de un documento, el comportamiento ya no es tan brillante. Basta con usar un término de búsqueda como “ladrones” para comprobar que algunas de las suposiciones que realizan las representaciones por estructura no siempre son válidas. El análisis de los hiperenlaces ayuda a identificar las páginas más populares, pero esta popularidad no siempre está relacionada con la calidad en el acceso a la información web. Vistas

la peculiaridades de la Web, puede pensarse que las representaciones por estructura son más adecuadas para su aplicación en ciertos procesos de IR, y considerarse que las representaciones basadas en contenido lo serían para problemas de TC y DC.

De los tres enfoques en los que se divide la Minería Web, el único que permite generar modelos de representación autocontenidos para la representación de documentos HTML es la *web content mining*. Un documento HTML se analiza aisladamente como pudiera hacerse con cualquier otro documento electrónico. En este caso, además, deberían tenerse en cuenta las características propias del lenguaje de marcado. Enric J. Glover, en su tesis doctoral (Glover, 2001), afirma que las página web contienen características propias que no se presentan el texto plano y así, las funciones de ponderación como TF-IDF pueden ser mejoradas teniendo en cuenta la información relacionada con estas características. De un modo parecido, Michael Maudlin, describiendo el funcionamiento del portal Lycos (Mauldin, 1997), afirma que aspectos como la posición de los términos en un documento, en el título o en la parte alta del mismo, permitirían mejorar la calidad de la representación de un documento HTML.

Sin embargo, muchos de los trabajos encontrados en la literatura dentro de este enfoque no pasan de un análisis del contenido textual del documento, empleando funciones de ponderación clásicas como son la función binaria, TF, TF-IDF y WIDF, y aplicándolas directamente al contenido de texto presente en el documento.

Aparte de estas funciones clásicas, algunos modelos han empleado los contenidos de algunos elementos HTML, principalmente el META y el TITLE. En el caso de la metainformación, el porcentaje de documentos en los que está presente suele ser muy bajo y en el caso del título, el tamaño de sus contenidos es muy pequeño. Esto impide la utilización de estos elementos por separado y aisladamente, si lo que se quiere desarrollar son modelos de representación generales, aplicables a cualquier tipo de página web. Otros modelos se basan en el análisis, por medio de heurísticas, de determinadas estructuras de anidamiento en los elementos HTML. Estas representaciones son muy dependientes del tipo de página web que se esté considerando, por lo que tampoco podrían emplearse en modelos de representación de ámbito general.

Un modelo de representación autocontenido que destaca por ser diferente al resto de los encontrados es el presentado en (Molinari et al., 2003), cuando se emplea su función de ponderación local. Aunque su ámbito de aplicación ha sido la IR, este modelo puede servir para evaluar las representaciones autocontenidas presentadas en esta tesis, junto con un conjunto de funciones de ponderación heredadas directamente de la representación de textos.

Un último aspecto a destacar es el hecho de que los resultados experimentales presentados en la mayoría de los trabajos descritos dependen mucho de las colecciones empleadas, aunque ciertos aspectos pueden considerarse como generales.

Capítulo 4

Marco teórico general para representaciones autocontenidas de documentos HTML

"La palabra es mitad de quien la pronuncia,
mitad de quien la escucha"
Michel Eyquem de la Montaigne

En esta tesis se propone el desarrollo de un modelo para la representación autocontenida de documentos HTML. Esta propuesta se encuadra dentro de las representaciones "por contenido" y se basa esencialmente en la selección y aplicación de heurísticas extraídas de la lectura de textos. Primero, se realiza una asignación semántica a conjuntos determinados de elementos del vocabulario HTML para, a continuación, combinarlos. En este capítulo se establece el marco teórico sobre el que se apoyarán las representaciones propuestas en esta tesis.

4.1. Introducción

El *HyperText Markup Language*[1] es un lenguaje de marcado diseñado para estructurar documentos y presentarlos en forma de hipertexto. Este término apareció por primera vez en (Nelson, 1965) para describir un documento con estructura no lineal, donde la información se concibe como una red de nodos enlazados y donde cada nodo almacena un fragmento de información.

Ted Nelson definió el hipertexto como: "un cuerpo de material escrito o pictórico, interconectado en una forma compleja que no puede ser representado de forma conveniente haciendo uso de papel". En (Díaz-Pérez et al., 1996), el hipertexto se define como "una tecnología que organiza una base de información en bloques discretos de contenido llamados nodos, conectados a través de una serie de enlaces cuya selección provoca la inmediata recuperación de información destino".

A partir estas definiciones se desprende que la lectura de un hipertexto es no lineal. En un hipertexto el usuario accede selectivamente a diferentes bloques de información y decide

[1] http://www.w3c.org/HTML

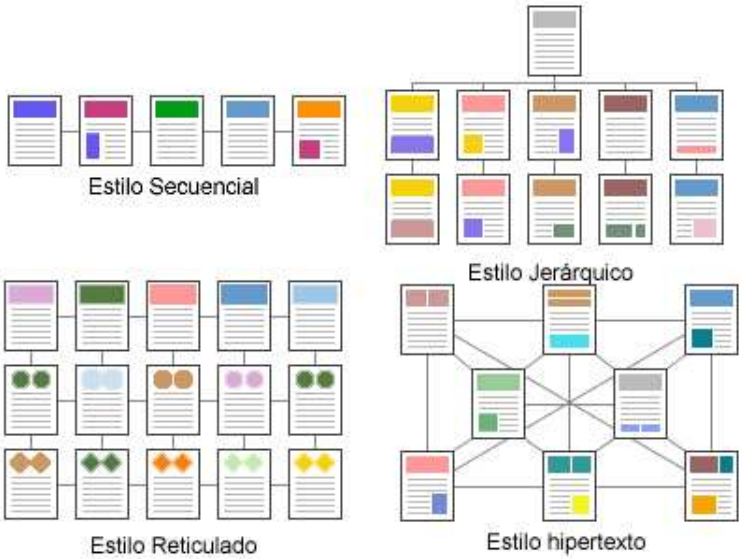

Figura 4.1: Estilos de estructuración del contenido de un documento.

qué camino debe seguir su lectura dentro de la red de nodos. Se tiene, por tanto, una nueva manera de organizar y transmitir el conocimiento, antitética a las formas clásicas de estructuración y narración.

Un hipertexto, en términos ideales, deberá permitir organizar y presentar información poco o nada estructurada. Podrán utilizarse esquemas jerárquicos para mantener la estructura de los sistemas de documentación de texto tradicionales o crear nuevas estructuras en red. El lenguaje HTML aporta la información específica sobre cómo debe mostrarse el contenido de cada fragmento de información y permite, además, establecer relaciones unidireccionales entre documentos. En la figura 4.1 se representan diferentes formas de estructurar el contenido de un documento.

La estructura de hipertexto hace que los procesos de escritura y lectura puedan variar respecto a estructuras/lecturas más tradicionales. Pero no todos los textos tradicionales (textos impresos) son de lectura lineal, ya que pueden contener elementos que introducen no linealidad y que, sin embargo, resultan de uso común. Entre estos elementos que introducen no linealidad están las notas al pie de página y al margen, referencias, paréntesis o guiones aclaratorios, anotaciones, tabla de contenidos, apéndices, etc. A pesar de estar acostumbrado a una lectura con este tipo de elementos, la no linealidad del HTML es mucho más evidente y puede ocasionar problemas de desorientación al lector [(Conklin, 1987), (Wright, 1991)].

Dado que el propósito de esta tesis es la búsqueda de una representación autocontenida de documentos HTML, a partir de la aplicación de heurísticas extraídas de los procesos de escritura y lectura, no se considerará información externa a cada nodo y no se estudiarán tampoco los procesos de lectura no lineal. Sólo serán considerados aquellos procesos aplicables a la lectura y escritura de cada documento por separado, considerando así cada página web como un documento aislado.

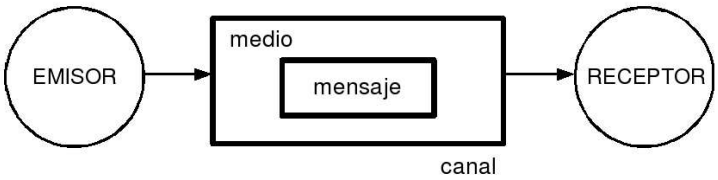

Figura 4.2: Fases del proceso informativo-documental.

La comunicación por medio de páginas web se puede considerar como un proceso informativo-documental (figura 4.2). Un *emisor* codifica un *mensaje* en lenguaje HTML y lo transmite por un *medio* hacia un *receptor* que lo deberá decodificar. Este proceso es un proceso activo. Por un lado, el *emisor* utiliza las características del lenguaje para hacer llegar su mensaje al *receptor*, el cual deberá interpretarlo con ayuda del conocimiento que tenga del propio lenguaje. En este proceso, tanto el emisor como el receptor tienen una doble vertiente. Una primera relativa a su bagaje y experiencia personal, y otra que hace referencia al entorno cultural en el que se ha desarrollado. Ambas se encuentran en una relación dinámica y recíproca. Cuando dos personas leen un mismo texto, cada una de ellas hará su propia interpretación del contenido del mismo. Estas interpretaciones se construyen en base a procesos comunes de lectura presentes entre los miembros de una misma cultura, a los que cada lector aporta su componente personal. De los procesos comunes en la creación y lectura de un documento (la vertiente cultural) se tratará de extraer un conjunto de heurísticas que sean aplicables en representaciones autocontenidas de páginas web.

El resto de este capítulo se organiza como sigue. En primer lugar, se presentan brevemente las características principales del HTML como lenguaje de marcado. Se verá que se trata de un lenguaje orientado fundamentalmente a la especificación de información tipográfica. A continuación, se introduce un modelo de lectura considerado en el campo de la psicología cognitiva y, seguidamente, se muestran algunos criterios relacionados con el conocimiento heurístico empleado tanto en la escritura, como en la lectura de textos. Por último, se propone un modelo general de combinación de criterios heurísticos para la representación autocontenida de páginas web, objetivo fundamental de esta tesis doctoral.

4.2. Lenguajes de marcado

Un "lenguaje de marcado", también llamado "lenguaje de anotaciones" o "de etiquetas", se define como un conjunto de reglas que permiten estructurar y dar formato a un documento. Dentro de los lenguajes de marcado se pueden distinguir dos tipos de estructura (Díaz-Pérez et al., 1996):

- **Estructura lógica**, formada por las diferentes partes que componen el documento y

sus relaciones. Por ejemplo, un documento puede estar formado por una introducción, capítulos, secciones, subsecciones, etc.

- **Estructura física**, relativa a la forma en que se presenta el documento, incluyendo la tipografía, ordenación espacial de los componentes, etc.

En un documento impreso, ambas estructuras resultan inseparables, no así en el caso de documentos electrónicos, donde la información relativa a dichas estructuras puede almacenarse independientemente. Esto es posible gracias a los lenguajes de marcado "genéricos", que permiten mantener separados los componentes lógicos de las instrucciones de procesamiento que ejecutan los programas encargados de dar formato al documento (Bryan, 1988). Este hecho supone que un mismo documento podrá presentarse de muy diferentes maneras de acuerdo a las distintas normas de estilo que puedan aplicarse. Existen otros tipos de marcado que no mantienen esta separación entre los dos tipos de estructura, los lenguajes "específicos", que sólo contienen información relacionada con la apariencia que debe tener el documento (Díaz-Pérez et al., 1996).

SGML (*Standard Generalized Markup Language*) supone un buen ejemplo de un lenguaje de marcado genérico; permite hacer explícita la estructura de un documento, es decir, su contenido semántico o cualquier otra información que se quiera hacer patente, ya sea lingüística o extralingüística. Se considera estándar desde 1986, cuando apareció con el identificador 8879 como norma ISO[2], y su objetivo fundamental es el de definir lenguajes de marcado que permitan la representación y transmisión de documentos en un formato adecuado para posteriores procesos de edición e impresión. Por esta razón, se considera como metalenguaje (McLeod et al., 1991). Los lenguajes de marcado XML y HTML, basados en la especificación SGML, se han convertido en un estándar de facto para la presentación e intercambio de información en Internet.

Una anotación (o marcado) supone añadir metainformación a un documento, es decir, información que no forma parte propiamente del contenido del documento. Existen tres tipos generales de lenguajes de marcado en función del significado general de su marcado:

- **Procedimental**, si el las anotaciones describen simplemente operaciones tipográficas.
- **Estructural**, si describen la estructura lógica de un documento pero no su tipografía (mediante hojas de estilo se permitiría la traducción de anotaciones estructurales a tipográficas).
- **Híbrido**, si resulta ser una combinación de lenguaje procedimental y estructural.

Mediante una definición de tipo de documento (*Document Type Definition*, DTD) se puede especificar la estructura lógica de los documentos SGML; qué elementos pueden incluirse en el contenido de un documento y en qué orden, definiendo con precisión aquellos elementos que son necesarios en la elaboración de un documento o un grupo de documentos estructurados de

[2]International Organization for Standarization

manera similar (Abaitua, 2005). Cada uno de estos elementos se marca con una etiqueta de comienzo y otra de final, que vienen especificadas mediante un identificador que define el tipo de elemento, junto con unas características –atributos– que lo cualifican (Díaz-Pérez et al., 1996).

Alrededor de los lenguajes de marcado se han desarrollado numerosas tecnologías, y en especial en torno a XML: lenguajes para crear hojas de estilo (XSL[3]) o transformaciones de formato (XSLT[4]); lenguajes para seleccionar una determinada parte de un documento (XPath[5]) o para enlazar y apuntar a diferentes partes de un documento (XLink[6], XPointer[7]). Por otro lado, la estandarización ha permitido el desarrollo de diferentes modelos de acceso a esta información semiestructurada (DOM[8] y SAX[9]).

En Enero de 2000 se estandarizó la especificación XHTML (eXtensible HyperText Markup Language[10]). Se trata de una reformulación de HTML en XML, y combina las características del HTML con la potencia de XML. Con la Web Semántica, la Red se encamina hacia XML y XHTML permitirá reciclar toda la documentación existente en la Web escrita ya en HTML a un formato compatible con XML con un mínimo esfuerzo.

4.3. El vocabulario HTML

El HTML es un lenguaje creado en 1989 a partir de SGML. En el ánimo de su creador, Tim Berners-Lee, estaba únicamente visualizar e interconectar el contenido de documentos electrónicos y así, consideró un conjunto pequeño de etiquetas que marcaran párrafos, títulos, hipervínculos y poco más. A continuación, se asociaron comportamientos concretos a dichas etiquetas.

Con el tiempo, la ventaja que representaba la simplicidad de HTML se convirtió en un inconveniente, ya que su marcado no siempre cubría todos los aspectos de presentación que los usuarios requerían. La solución adoptada fue el desarrollo de extensiones del lenguaje privadas, lo que complicó la estandarización. Las luchas comerciales entre las principales empresas de desarrollo de navegadores web durante los primeros años de Internet condujeron a un lenguaje HTML que, aunque universalmente utilizado e interpretado, carece de una estandarización real.

HTML cumple con los dos objetivos esenciales para el diseño y visualización de un documento digital:

- Estructura un documento en elementos lógicos, como por ejemplo: encabezado, párrafo, etc.

[3] http://www.w3.org/Style/XSL/
[4] http://www.w3.org/TR/xslt
[5] http://www.w3.org/TR/xpath
[6] http://www.w3.org/TR/xlink/
[7] http://www.w3.org/TR/WD-xptr
[8] http://www.w3.org/DOM/
[9] http://www.saxproject.org/
[10] http://www.w3.org/TR/xhtml1/

- Especifica las operaciones tipográficas y funciones que debe ejecutar un programa visualizador sobre dichos elementos.

Aunque deba considerarse como un lenguaje de marcado híbrido, su uso está orientado principalmente a la descripción de operaciones tipográficas (Musciano y Kennedy, 1997); por tanto, se trata de un lenguaje con un carácter esencialmente procedimental.

Desde finales de 1993 existen comités de estandarización, impulsados por el World Wide Web Consortium[11] (W3C), que tienden a un modelo único de HTML. El HTML4.0 (Musciano y Kennedy, 2000), sirva como ejemplo, tiene tres DTDs: *loose.dtd*[12], *frameset.dtd*[13] y *strict.dtd*[14]. La *loose.dtd* incluye elementos de versiones anteriores no estandarizadas. La *frameset.dtd* permite dividir la pantalla en varias zonas, conocidas como marcos (*frames*), de forma que cada marco lleva asociado una página web. Estos marcos se comportan de forma independiente y se necesita una página contenedora que especifique cuántos *frames* tiene el diseño, su tamaño y su distribución. Por último, se tiene la *strict.dtd*, o DTD pura.

En general, un documento HTML sigue la sintaxis de cualquier lenguaje de marcado y su estructura global es la siguiente. A partir de un elemento raíz <html> se pueden anidar otros dos elementos: <head> y <body>, correspondientes a la cabecera y cuerpo del documento. Un ejemplo sencillo de documento HTML podría ser el que sigue:

```
< html >
    < head >
        < title >
            título de la pagina
        < /title >
    < /head >
    < body >
        < h1 >  Titulo del contenido visible < /h1 >

        texto visible
        < font  color = "#000080" > texto  en diferente color < /font >
        texto visible
    < /body >
< /html >
```

[11] http://www.w3c.org
[12] http://www.w3c.org/TR/REC-html40/loose.dtd
[13] http://www.w3c.org/TR/REC-html40/frameset.dtd
[14] http://www.w3c.org/TR/REC-html40/strict.dtd

Cabecera (<head>)

En la cabecera se incluyen definiciones generales a todo el documento; se puede agregar un fondo de pantalla, definir los colores del texto, etc. El texto contenido en este elemento <head> no resultará visible en un navegador web. Estas definiciones pueden estar relacionadas con el formato global del documento, para lo que se emplea la etiqueta <style>, o tratarse de características más cercanas a la visualización que el autor desea dar a cada elemento, y que podrían diferir de las que establezca por defecto el navegador.

El elemento <title> debe ir también en la cabecera, especifica el título del documento y se muestra en la barra de título del navegador. El contenido de este elemento suele usarse como el texto con el que se guarda una página en los marcadores (*bookmarks*). También es el texto que muestra un motor de búsqueda en los enlaces devueltos tras una consulta. Este elemento es opcional, aunque sería muy recomendable que todo documento HTML tuviera un título.

En la cabecera también pueden incluirse códigos escritos en diferentes lenguajes interpretados (JavaScript, PHP, ASP, ...), contenidos dentro del elemento <script>. Con estos códigos se consigue implementar el acceso y recuperación de contenidos almacenados en una base de datos o simplemente aportar dinamismo al documento.

Con la etiqueta <meta> se permite introducir información para la que no se definió ningún elemento del lenguaje. La información almacenada en este elemento tiene gran importancia porque permite transmitir datos etiquetados semánticamente a una aplicación que posteriormente procese el documento. Un ejemplo de uso de este elemento es el siguiente:

```
<meta name="Tipo de Página" content="Personal">
<meta name="Nombre" content="Víctor Fresno Fernández">
<meta name="Puesto" content="Profesor Ayudante">
```

De este modo, el programador pasa una metainformación al navegador con ayuda de los atributos "name" y "content" de este elemento <meta>. Esta característica podría ser muy importante en tareas de acceso a la información web. Sin embargo, diversos estudios [(Pierre, 2001), (Riboni, 2002)] han mostrado que este tipo de elementos se encuentran en menos de un 30 % de las páginas analizadas.

Cuerpo (<body>)

El cuerpo de un documento HTML está formado por elementos relativos a la estructura y a cómo debe visualizarse la información contenida en el documento HTML. Dentro de esta etiqueta se incluye el texto que se desea hacer visible en la página web.

Dentro del <body> pueden utilizarse diferentes encabezados (<h1> ... <h6>) que permiten realizar una ordenación jerárquica de los apartados en los que se quiera estructurar un

documento. Un ejemplo podría ser utilizar <h1> para el título del contenido del documento (diferente del título de la página), <h2> para los posibles apartados y <h3> para los subapartados, aunque no es necesario mantener exactamente esta jerarquía.

En general, el vocabulario HTML tiene dos tipos de estilos: físicos y lógicos. Los estilos físicos son aquellos que siempre implican un mismo efecto tipográfico, mientras que los lógicos marcan un texto que por sus características debe tener un modo de mostrarse propio.

Por ejemplo, son estilos lógicos: el elemento <address>, que codifica direcciones de correo electrónico o direcciones personales; o <blockquote>, que permite marcar citas textuales, mostrando el texto resaltado y separándolo del texto que lo circunda. El elemento <dfn> especifica una definición y con se indica que el autor quiere destacar el contenido de ese elemento con énfasis. Con el elemento <code> se puede introducir como texto un fragmento de código fuente sin que llegue a ser interpretado por el navegador. Con <kbd> se pueden marcar textos tecleados por el usuario. Con <strike> se presenta un texto tachado, mientras que con la etiqueta se resalta el contenido. Con <var> se especifica una tipografía diferente para marcar que se trata de una variable, en el caso de que en el contenido del documento se quiera mostrar un código fuente.

Como ejemplo de estilos físicos se pueden destacar: el elemento <b>, que destaca una porción de texto en negrita; <i>, que hace lo propio, pero en cursiva; <sub> y <sup>, que permiten formatear un texto como subíndice o superíndice; los elementos <big> y <small>, que se emplean si se quiere mostrar una porción de texto en mayor o menor tamaño; o el <tt>, que muestra su contenido a modo de máquina de escribir.

HTML no posee gran capacidad para establecer la estructura lógica de un documento. Uno de los aspectos primordiales de este lenguaje es el formateo de la propia fuente. En la práctica, resulta muy común presentar texto resaltado en negrita, itálica, con otros efectos tipográficos.

Se puede obtener un mismo resultado empleando estilos físicos y lógicos. Aunque la tendencia actual es utilizar estilos físicos, las dos formas son adecuadas. El lenguaje HTML es interpretado por los navegadores según su criterio, por lo que una misma página web puede ser mostrada de distinto modo según el navegador. Mientras que <b> significa simplemente negrita y todos los navegadores la interpretarán como negrita, es una etiqueta que indica que su contenido debe resaltarse y cada navegador será responsable de hacerlo como estime oportuno. En la práctica, muestra el texto en negrita, pero podría ser que un navegador decidiese resaltarlo con negrita, subrayado y en color rojo. En el caso de querer aplicar un estilo de fuente itálica también existirían dos posibilidades: <i>, que sería interpretado como itálica; y , que se interpretaría como el estilo lógico de enfatizar, aunque igualmente se suela mostrar como un texto en itálica.

Un aspecto importante es que a partir de esta información de carácter tipográfico se puede extrapolar información relativa a la intención del autor en el momento de crear el documento, intuyendo qué partes quiso destacar frente a otras o con qué elementos del discurso quiso llamar

la atención del lector.

En esta tesis se considera el carácter híbrido del lenguaje HTML y se emplea información extraída, tanto de etiquetas de carácter tipográfico, como de elementos de carácter estructural. La idea fundamental reside en una asignación semántica a determinados elementos del vocabulario HTML, de forma que se puedan relacionar con heurísticas aplicadas a la lectura.

4.4. Procesamiento de la información escrita

Keith Rayner y Alexander Pollatsek, en su libro "The psychology of reading", afirman que "leer es la habilidad de extraer información visual de una página y comprender el significado del texto"(Rayner y Pollatsek, 1998).

En toda comunicación se pueden distinguir dos niveles de información: el conjunto de datos que componen lo que tradicionalmente entendemos por contenido, y una información más sutil que se superpone a ese contenido. Esta información adicional puede ser: la negrita en un libro, el subrayado en un manuscrito, un tono de voz alto en una conversación, ...y se denomina etiquetado, o *markup* (Valle et al., 1990). En el caso de la comunicación por medio de textos, la función de este etiquetado es la de aportar información que, por lo general, refleja la estructura de un documento en un sentido amplio, es decir, destaca unas partes de otras y establece distintas relevancias para diferentes partes del texto o de su contenido, de forma que ayuda al lector a procesar la información que contiene. De este modo, elementos de enfatizado como la 'cursiva', la 'negrita' o el 'subrayado' pueden usarse como indicadores de relevancia dentro del contenido del texto.

Si se asume que la lectura es un proceso de adquisición de información, la lectura de un hipertexto podría expresarse siguiendo modelos psicológicos generales, que al aplicarse a la lectura pasarían a considerarse procesos psicolingüísticos. La psicolingüística analiza los procesos relacionados con la comunicación mediante el uso del lenguaje (sea oral, escrito, etc.).

Los procesos psicolingüísticos pueden dividirse en dos categorías, de *codificación* (producción del lenguaje), y de *decodificación* (o comprensión del lenguaje). Los primeros analizan los procesos que hacen posible que seamos capaces de crear oraciones gramaticalmente correctas partiendo de un vocabulario y de unas estructuras gramaticales prefijadas, mientras que los segundos estudian las estructuras psicológicas que nos capacitan para entender expresiones, palabras, oraciones, textos, etc. La comunicación entre dos personas puede considerarse un continuo:

Percepción-Comprensión-Producción

La riqueza del lenguaje hace que esta secuencia se desarrolle de varias formas; dependiendo del estímulo externo, las etapas sensoriales en percepción serán diferentes.

4.4.1. Modelos de lectura

Los *modelos de lectura* describen las dificultades a las que un lector se tiene que enfrentar en el proceso de comprensión de un texto. Se pueden distinguir dos modelos fundamentales: el "dirigido por texto" (*text-driven*) o "de abajo a arriba" (*botton-up*), y el "dirigido por contexto" (*context-driven*) o "de arriba a abajo" (*top-down*) (Spirydakis, 2000).

El modelo de lectura dirigido por el texto sostiene que la lectura comienza con la percepción de características textuales (signos) de los que se van identificando letras, palabras, unidades sintácticas y, finalmente, el conjunto de ideas que son extraídas y almacenadas (Gough, 1972). En este caso, el lector estaría aplicando heurísticas relativas a cómo suele presentarse la información en un texto.

Por el contrario, en el modelo dirigido por contexto se entiende que el conocimiento a priori que posee el lector sobre un determinado tema es lo que dirige la lectura. Este conocimiento, junto con la experiencia, indica al lector qué información debe tomar de una página. De esta forma, el lector formula hipótesis sobre lo que puede ir encontrándose en el texto, y entonces va probando si lo que lee confirma, o no, esas hipótesis iniciales (Goodman, 1967).

Existe un modelo que hibrida los dos anteriores: el modelo *interactive-compensatory*, que permite a los lectores analizar características del texto (*botton-up*) partiendo de su conocimiento a priori (*top-down*) (Stanovich, 1980). De este modo, un lector con un conocimiento amplio sobre un tema no tendrá que descomponer el texto en elementos básicos porque podrá contar con la información que tenga guardada en sus estructuras de memoria. Por otro lado, un lector acostumbrado a un lenguaje o sintaxis particular podrá confiar en su habilidad para procesar el texto de un modo sencillo, enlazando la nueva información que encuentre con el conocimiento que ya tenía adquirido.

Una vez el lector comienza la decodificación de las palabras y la identificación de relaciones sintácticas en el texto, comenzará a construir una jerarquía mental –o modelo de situación– del contenido del texto [(Kintsch y van Dijk, 1978), (Kintsch, 1992)]. Para construir esta representación se toma información del texto y se relaciona con la información almacenada en la estructuras de memoria que posee el lector. Cuando las relaciones entre la información extraída del texto y el conocimiento a priori que posee el lector no son claras, se realizan inferencias para ir desarrollando una red de información en memoria (Kintsch y van Dijk, 1978).

La Psicolingüística se enmarca dentro de la psicología cognitiva, el campo de la psicología que estudia los procesos mentales que, supuestamente, están detrás del comportamiento. Cubre la memoria, percepción, atención, razonamiento y otros procesos cognitivos superiores como es el caso de la lectura. Dentro de esta disciplina destacan dos corrientes principales: la *arquitectura funcional de la mente* y el *conexionismo*. Desde la arquitectura funcional de la mente se estudian las estructuras (elementos físicos) y los procesos o estrategias cognitivas del procesamiento, mientras que desde el conexionismo se estudia el procesamiento en paralelo de las neuronas, asumiendo la "metáfora cerebral" (Iglesias y Veiga, 2004). Las hipótesis de esta tesis respecto

a los modelos de lectura como procesamiento de información se inspiran en la corriente de la arquitectura funcional de la mente.

4.4.2. Proceso de lectura

En los últimos 40 años, una de las mayores contribuciones a la psicología cognitiva fue el desarrollo de modelos rigurosos de exploración de los procesos cognitivos, estableciendo teorías sobre ellos. Este enfoque se conoce como enfoque de "procesamiento de la información" (*information processing approach*) [(Klahr, 1989), (Klahr, 1992)] y, dentro de él, se pueden encontrar los sistemas de producciones [(Newell y Simon, 1963), (Ernst y Newell, 1967)].

El objetivo de la perspectiva del procesamiento de la información es el estudio de los procesos que ocurren en la mente y que permiten a un hombre realizar tareas complejas. Estas tareas pueden resumirse en un procesamiento de entradas y posterior incorporación en un almacén de memoria permanente (Iglesias y Veiga, 2004). De este modo, el sistema está organizado como una unión de procesos (fases de tratamiento de la información) y estructuras (almacenamiento de información). Este enfoque propone que el procesamiento de la información, en su conjunto, tiene una estructura modular; supone, por tanto, un análisis abstracto en serie de procesos cognitivos (Best, 2001).

Investigaciones en el campo de la psicología cognitiva han encontrado que los modelos basados en sistemas de producciones resultan una de las formas más directas de modelar la complejidad de la inteligencia humana (Klahr et al., 1987) y por ello, se han realizado numerosos estudios sobre estos sistemas desde que fueron planteados por Allen Newell en 1967.

Un sistema de producción esta formado por una base de hechos que contiene el estado del problema, una base de reglas y un motor de inferencia. La base de reglas es un conjunto de reglas de producción. Los sistemas de producciones pueden verse como sistemas generales de procesamiento de la información capaces de simular en ordenador algunas tareas o rango de tareas cognitivas (Young, 2001). Un sistema de producción opera como un ciclo de dos fases:

- Fase de **Reconocimiento**
- Fase de **Actuación**

Si la parte IF de una regla de producción se cumple (reconocimiento), entonces se ejecuta su consecuente THEN (actuación). Los sistemas de producción tienen ciertas propiedades que los hacen especialmente adecuados para el modelado de procesos cognitivos, incluyendo la combinación de procesamiento en serie y en paralelo, la independencia entre reglas y la flexibilidad que permiten a los mecanismos de control. Desde que en sus orígenes fueran planteados como modelos de resolución de problemas, los sistemas de producciones han llegado a convertirse en un formalismo muy utilizado para el modelado cognitivo y del aprendizaje, en áreas tales como la lectura, el diagnóstico médico o en partidas de ajedrez por ordenador. Desde

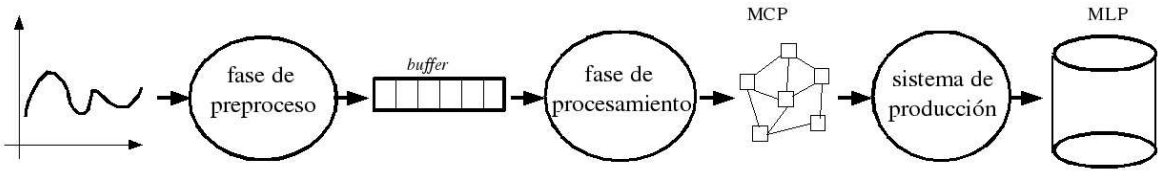

Figura 4.3: Representación de un sistema de procesamiento de información planteado por David Klahr.

1990, estos sistemas han formado parte de la mayoría de arquitecturas cognitivas integradas [(Rosenbloom et al., 1993), (Anderson, 1997), (Young, 2001)].

David Klahr, tratando de describir la estructura básica del sistema de procesamiento de la información del ser humano mediante sistemas de producciones, plantea que ésta puede realizarse en una secuencia de fases, comenzando con la impresión en los sentidos de los estímulos ambientales, y continuando con procesos más profundos, o centrales [(Klahr y Wallace, 1976), (Klahr et al., 1987)].

Asociada con cada fase, existe una capacidad de almacenamiento (un "retén" o *buffer*) que mantiene la información mientras se procesa posteriormente en los estadios ulteriores. Los procesos sensoriales, que reciben y almacenan toda la información sensorial durante fracciones de segundo, se encuentran en la fase exterior. Hasta ese momento, el sistema parece operar de forma paralela y no selectiva; se recoge toda la información posible para que pueda ser procesada a continuación. En el siguiente nivel, la información seleccionada y codificada parcialmente se conserva para un procesamiento posterior por medio de *buffers* de modalidad específica (por ejemplo: visual o auditiva), durante periodos un tanto más largos. Después, la información pasa por un retén de capacidad limitada, identificado habitualmente como memoria a corto plazo (MCP).

La información que se encuentra en la memoria a corto plazo debe ser examinada y conservada durante algún tiempo antes de que pueda transferirse a la memoria a largo plazo (MLP). Para que una información pase de la memoria a corto plazo a la memoria a largo plazo, la regla de producción comprueba el estado de conocimiento actual del sistema (por ejemplo, los contenidos de sus retenes en ese momento). Si se satisface la condición, entonces se ejecutan sus acciones, cambiando el estado de conocimiento y pasando la información a la memoria a largo plazo. El conjunto de reglas puede estar organizado y coordinado de distintas maneras para producir un procesamiento de la información con un objetivo determinado. A un proceso complejo como es la lectura se le podría asociar un sistema de producción.

La figura 4.3 muestra gráficamente un posible flujo de información dentro del sistema de procesamiento de la información planteado por David Klahr y basado en sistemas de

producciones. La primera fase, que llamaremos *etapa de preproceso*, sería la encargada de decodificar un conjunto de señales captadas por los sentidos, discriminando cierta información del total de datos captados por los sentidos, y guardándola en una estructura de almacenamiento temporal (que llamaremos *buffer*). A continuación, se aplica una *etapa de procesamiento* que transforma la información almacenada en el *buffer* y la prepara como entrada al sistema de reglas de producción. Esta información es la que se almacena temporalmente en la memoria a corto plazo. Este sistema de procesamiento finalizaría con la aplicación de un conjunto de reglas de producción que deben cumplirse para que cierta información pase de la memoria a corto plazo a la memoria a largo plazo, considerándose entonces que se produce una adquisición de conocimiento.

Aunque no es el propósito de esta tesis la aplicación de ningún sistema de producción, esta arquitectura funcional planteada por David Klahr es lo suficientemente general como para servir de modelo en la creación de un sistema de representación autocontenida de páginas web basado en combinaciones heurísticas de criterios. Este sistema se construiría como una secuencia de fases. En primer lugar, en la fase de preproceso, se seleccionaría el vocabulario con el que representar un documento. Posteriormente, en la etapa de procesamiento, se capturaría la información asociada a los criterios heurísticos que se quisieran combinar y, por último, se realizaría la combinación de criterios heurísticos por medio de una función de combinación lineal o de un sistema de combinación borroso, que sustituirían en el esquema de la figura 4.3 al sistema de producción.

4.5. Heurísticas aplicadas a los procesos de escritura/lectura

Tanto para escribir páginas web que resulten comprensibles, como para evaluar lo comprensible que resulta una determinada página web, los autores suelen confiar en directrices –o pautas– establecidas para la escritura de textos impresos. Esta idea se basa en la suposición de que los lectores "comprenden" la información del mismo modo en un texto escrito que en un documento electrónico (Spirydakis, 2000). Un usuario de la Web localiza un contenido que está buscando en una página web y, al igual que cuando se lee un texto escrito, trata de dar sentido a palabras, oraciones, párrafos, etc. Por tanto, una buena parte de los procesos implicados en la lectura de una página web deberán ser los mismos que se darían en la lectura de un texto escrito.

Un estudio desarrollado por la Universidad de Standford y el Instituto Poynter[15] que utilizaba cámaras para registrar los movimientos de los ojos de los usuarios de Internet, mostró los nuevos hábitos de lectura en Internet. Al contrario que los lectores de textos impresos, los "lectores" de páginas web suelen enfocar su vista en primer lugar sobre el texto, ignorando inicialmente los gráficos e imágenes, a los que regresan una vez se ha completado la lectura. Además, el porcentaje de texto leído en las páginas web resultó ser de un 75 % frente a un 30 % en el caso

[15] http://www.poynter.org

de texto impreso.

La comprensión de un texto no es un registro pasivo de información; supone un proceso reconstructivo en el que se despliega una gran actividad intelectual. Durante el proceso de comprensión el lector aplica un potencial de significados al texto que está percibiendo, tratando de reconocer en él algunas de sus concepciones previas (aquellas que su experiencia le ha hecho conocer y su memoria ha retenido) en cuanto al contenido y a la forma de los textos. En este proceso hay un componente personal que filtra y modela la información de una manera original o propia, pero también hay parámetros estándar o patrones sociales comunes a una misma cultura que también son activados o neutralizados para dar vida al texto (Coscollola, 2001).

Cuando leemos convertimos signos (o marcas) en significados que codificamos como parte del proceso de entendimiento; la interpretación que se haga de estos signos dependerá mucho del conocimiento previo adquirido. Parte de este conocimiento lo tenemos en forma de esquemas, o *schematas*, que podrían considerarse como conocimiento heurístico. El adjetivo "heurístico" significa "medio para descubrir" y está relacionado con el término griego *heuriskein* que significa hallar, inventar. Por heurístico se entiende un criterio, estrategia o método empleado para simplificar la resolución de problemas complejos. Las heurísticas suponen, por tanto, un conocimiento que se obtiene a través de la experiencia.

Según Calsamiglia y Tusón en su libro "Las cosas del decir. Manual del análisis del discurso" (Calsamiglia y Tusón, 1999), la enunciación escrita se caracteriza básicamente por:

- Una actuación independiente entre los emisores y receptores que se comunican a través de un texto (escritores y lectores).
- Los protagonistas no comparten ni el tiempo ni el espacio durante la comunicación.
- El texto contiene las instrucciones necesarias para ser interpretado de forma diferida.

Desde el punto de vista psicológico, el texto escrito supone dos procesos cognitivos relacionados con la expresión lingüística: el proceso de producción (escritura) y el proceso de interpretación (lectura). Algunos psicólogos cognitivos suponen que el proceso de interpretación requiere ir rellenando algunas representaciones internas (*schematas*) construidas previamente con un conocimiento adquirido, o heurísticas.

En un proceso de lectura se deben integrar multitud de procesos cognitivos como son la atención, la comprensión y la memoria. La comprensión del texto, como ya se ha dicho, dependerá enormemente del modelo mental del lector (Mandler y Nancy S. Johnson, 1977). Respecto al proceso de decodificación de un texto por un lector, según van Dijk y Kintsch (van Dijk, 1997) pueden existir cuatro tipos de representaciones mentales usadas en la comprensión de un texto:

1. **Microestructura**, relativa a la literalidad; supone la representación literal de lo que dice el texto.

2. **Macroestructura**, o el significado del texto; por tanto, derivada de la microestructura.

3. **Superestructura**, forma retórica o estilo del texto; que podría influir en cómo se interpreta el texto por parte del lector.

4. **Modelo de situación**, estructuras de conocimiento que representan acerca de qué habla el texto.

Un texto escrito se despliega de forma lineal en el espacio de una página. Esto requiere una configuración externa que arme los contenidos, su ordenación y su organización (Coscollola, 2001). Como se ha visto, la estructuración de los contenidos en un hipertexto no tiene por qué seguir una linealidad en el discurso, pero el texto de cada página web (de cada nodo dentro del grafo) sí que se suele cumplir esta condición. La linealidad o coherencia del texto convencional es más importante de lo que parece a simple vista. Gracias a los flujos de información lineal y continuos, los lectores pueden crear una interpretación interna del contenido en la que detectan y extraen la información de alto nivel, las ideas fundamentales (van Dijk, 1997).

La distribución de los enunciados que forman un texto está en relación con la distribución de los temas, los subtemas y los cambios de temas que se presentan en el texto. La unidad básica suele ser el párrafo, unidad significativa supraoracional, constituido por un conjunto de enunciados relacionados entre sí por el contenido. Las fronteras de cada párrafo son definidas por el autor, proporcionando una presentación temático/visual que orienta la lectura y proporciona un grado de legibilidad aceptable [(Serafini, 1992), (Cassany, 1995)]. Con la separación entre párrafos se dosifica la información, de forma que la segmentación quede al servicio de la comunicación del contenido (Coscollola, 2001).

Los autores de un documento incluyen señales en el texto que marcan o acentúan las ideas importantes (Cerezo Arriaza, 1994):

- Tamaños de los tipos de letra, uso de itálicas, subrayados.

- Orden de las palabras; las ideas más importantes suelen estar al comienzo de la frase, párrafo o texto.

- Los títulos de la obra, del capítulo o del apartado ayudan a resumir el contenido del texto o ponen de manifiesto la intención del autor.

De este modo, el efecto del texto sobre el lector dependerá enormemente del modo en el que se le presente la información. Esta es una de las ideas fundamentales sobre las que se apoya esta tesis doctoral.

Desde el punto de vista de la presentación de los contenidos, una de las primeras consideraciones realizadas en el desarrollo de páginas web es que, aún siendo igualmente texto, su contenido puede ser distinto al que se encuentra en un texto impreso. Pero no sólo porque el hipertexto permita una lectura no lineal y un desplazamiento entre contenidos en diferentes

páginas, sino por el hecho de que las personas se comportan de un modo diferente ante una pantalla que frente a una página de papel.

En un estudio realizado en 1997 por John Morkes y Jacob Nielsen se descubrió que la lectura de textos en pantallas de ordenador es diferente que si se lee en un texto en papel. Sólo un 16 % de los usuarios de prueba leyó las páginas web mostradas de modo secuencial, frente a un 79 % que, al leer un documento HTML, realizaron su lectura saltando entre los temas más importantes, fijando su atención en diferentes partes de la páginas, y no palabra por palabra como ocurre en los textos impresos. A partir de esta conclusión principal, surgen recomendaciones clave a la hora de escribir un documento HTML (Morkes y Nielsen, 1997); entre las más importantes se encuentran las siguientes:

- *Ser sucinto*: los usuarios no leen de la misma manera que en el texto impreso, por lo que deberían de expresarse las mismas ideas con la mitad de palabras.
- *Escribir pensando en la comprensión*: ofrecer sólo una idea por párrafo y en el caso del primer párrafo de un texto largo, debería de ser un resumen del tema que se esté desarrollando.
- Usar verbos directos. Emplear verbos en forma directa y menos palabras para expresar la acción que se está indicando.
- Evitar explicaciones negativas para mejorar la comprensión.
- *Establecer jerarquías de Información*, utilizando la llamada Pirámide invertida, ofreciendo lo más importante al principio.
- Preferir los hechos a los discursos: como norma general un usuario accederá a un sitio web para buscar elementos informativos que le permitan realizar acciones.
- Crear subtítulos destacados. Se deben dividir los textos en zonas que ayuden a la comprensión; en este sentido, es ideal que los subtítulos sean un resumen de los párrafos. De esta manera el usuario sabrá si en dicho apartado está lo que busca.
- *Enfatizar palabras significativas*: es conveniente destacar las palabras que sean más importantes de cada párrafo, para que fijen la atención de los ojos del usuario en el recorrido visual de la página. No obstante, una saturación de palabras enfatizadas en el texto hará perder el efecto que se buscaba inicialmente.
- *Crear listas* ofreciendo información estructurada antes que párrafos largos.
- *Ofrecer enlaces hacia otras páginas*, dividiendo el contenido de texto en diferentes páginas.
- *Evitar el uso de abreviaturas, siglas o acrónimos*, mejorando así la comprensión de los contenidos por parte de un mayor número de usuarios.

Es importante remarcar que este estudio no estaba basado en una tarea de búsqueda de información, sino que se pedía al usuario que leyera la página en busca de las tres ideas principales. A continuación, se les hacía una serie de preguntas sobre el contenido, a las que debían contestar. Se mostraron tres versiones diferentes de un mismo sitio web, incluyendo una versión que contenía listas, palabras clave destacadas en negrita, secciones de texto cortas, encabezados, etc. Una de las conclusiones de dicho estudio fue que, para generar documentos "comprensibles", deben usarse los tipos de elementos descritos, además de una estructuración clara en párrafos y un lenguaje sencillo.

Jan H. Spyridakis ha realizado varios estudios dirigidos a definir las líneas maestras que deben aplicarse en la creación de documentos HTML para mejorar la comprensión de su contenido. Las líneas más destacables son (Spirydakis, 2000):

- Respecto a la selección y presentación del contenido:
  - *Presentar el contenido de modo que el lector pueda orientarse.* Para ello la página debería contener un título en la parte superior, una introducción u oración introductoria que anuncie el tema que se va a tratar en el documento, y la repetición de términos importantes, siglas, acrónimos, ...
  - *Seleccionar el contenido relevante y destacarlo.* De este modo, los lectores retendrán mejor estos contenidos que aquellos que no consideren relevantes (a menudo los lectores no están interesados en elementos comunes a todas las páginas web)
  - *Minimizar el contenido de cada pagina y emplear resúmenes.* Crear hiperenlaces a otras páginas donde se desarrolle el tema.
- Respecto a la organización del contenido del documento:
  - *Agrupar la información para ayudar al lector a crear las estructuras jerárquicas en memoria.* Considerar que las personas son capaces de manejar 5 ± 2 conceptos a la vez, por lo que las agrupaciones deberían de tener un tamaño menor o igual que 5 elementos.
  - *Ordenar la información.* Utilizar una organización deductiva y situar la información importante al principio de los párrafos y al inicio del documento.
- Respecto al estilo:
  - *Emplear encabezados a varios niveles*: introducciones, índices y tablas, etc.
  - *Emplear términos sencillos*, cuya comprensión no requiera mucho esfuerzo, palabras concretas en lugar de usar términos más abstractos, palabras cortas, etc.

Aún asumiendo que el contenido de la web es muy heterogéneo, estas recomendaciones pueden resultar muy útiles si se está buscando un modelo de representación de documentos HTML basado en el conocimiento heurístico común a la mayoría de lectores de páginas web.

Cuando se ojea el contenido de una página web y se salta de una parte a otra en busca de información relevante, uno de los procesos que se ponen de manifiesto más activamente es la atención, ya que el autor quiere transmitir una información y el usuario debe buscar aquellas partes del contenido donde crea que pueda encontrar la información que precisa, sin necesidad de realizar una lectura lineal completa.

En términos generales, la atención se puede definir como un mecanismo que pone en marcha una serie de procesos u operaciones, gracias a los cuales, se es más receptivo a los sucesos del ambiente y se llevan a cabo tareas de una forma más eficaz. Julia García Sevilla, en su libro "Psicología de la atención", define la atención como "el mecanismo implicado activamente en la activación y el funcionamiento de los procesos y/u operaciones de selección, distribución y mantenimiento de la actividad psicológica".

Para desarrollar adecuadamente estos mecanismos de atención es necesario utilizar determinados procedimientos, o pasos, que reciben el nombre de *estrategias atencionales*. Estas estrategias son una habilidad que cada persona desarrolla dentro de sus capacidades, existiendo, por tanto, diferencias individuales. Una de las características más importantes de estas estrategias es que no son innatas, sino aprendidas. Esto es importante no sólo porque se pueden modificar y mejorar con la práctica, sino que se hace posible el desarrollo de estrategias encaminadas a mejorar los distintos mecanismo de atención, los factores que la mediatizan, así como la forma de controlarla.

Pero por otra parte, la atención no funciona de una manera aislada, sino que se relaciona directamente con los restantes procesos psicológicos. Con respecto a los procesos cognitivos, el que más estrechamente se ha vinculado a la atención ha sido la percepción. La atención se ha concebido en muchas ocasiones como una propiedad o atributo de la percepción gracias a la cual se selecciona más eficazmente la información que es más relevante en cada momento, y dependiendo de la pragmática de la acción que se esté realizando.

Por último, bajo la acepción de factores determinantes de la atención se incluyen todas las variables o situaciones que influyen directamente sobre el funcionamiento de los mecanismos atencionales. En algunas ocasiones porque hacen que tenga lugar de forma involuntaria y en otras, porque favorecen o entorpecen el funcionamiento de la atención en términos generales.

A partir de numerosos estudios realizados sobre la fase de captación y mantenimiento de la atención desde finales del siglo XIX, se puede concluir que las dimensiones físicas de los objetos que mejor captan y mantienen nuestra atención son (García Sevilla, 1997):

1. *El tamaño*. Normalmente los objetos de mayor tamaño llaman más la atención. En concreto, al doblar el tamaño aumenta el valor de la atención en un 42-60 %.

2. *La posición*. La parte superior atrae más; la mitad izquierda más que la mitad derecha. Por tanto, la mitad superior izquierda de nuestro campo visual es la que capta antes nuestra atención. Esto concuerda con los estudios generados y descritos en *Web Style Guide.com, 2nd Edition*.

3. *El color*. Los estímulos en color suelen llamar más la atención del sujeto que los que poseen tonos en blanco y negro.

4. *La intensidad del estímulo*. Cuando los estímulos son muy intensos tienen mayores probabilidades de llamar la atención

5. *El movimiento*. Los estímulos en movimiento captan antes y mejor la atención que los estímulos inmóviles.

6. *La complejidad del estímulo*. En términos generales los estímulos complejos, con un grado de información mayor, captan antes la atención que los no complejos. Ahora bien, los objetos que son muy complejos no captan tanto la atención como aquellos que presentan ciertas modificaciones con respecto a otros objetos que resultan familiares.

7. *La relevancia del estímulo*. Un estímulo puede adquirir un poder significativo a través de varios medios, como puede ser el proceso de pensamiento o la propia historia del sujeto.

8. *La novedad del estímulo*, o cambio de uno o varios de los atributos que componen el estímulo.

A continuación, se detalla cada uno de los criterios considerados en los modelos de representación propuestos en esta tesis. Dichos criterios tratan de reflejar parte del conocimiento heurístico presentado en esta sección y asociado a los procesos de lectura/escritura.

4.5.1. Frecuencia

Para determinar el tema de un texto se puede estudiar el concepto de isotopía (repetición a lo largo del discurso de una serie de elementos de significado y de construcción gramatical que permiten una continuidad temática). La isotopía se establece mediante redundancias y repeticiones de elementos similares o compatibles (Coscollola, 2001).

Por tanto, la frecuencia con la que aparece un término en un documento debe ser un factor determinante a la hora de establecer su relevancia. Ya se ha visto que resulta el parámetro más utilizado por la mayoría de las funciones de proyección dentro de los diferentes modelos de representación de documentos encontrados en la literatura.

En el estudio (Spirydakis, 2000) se afirmaba que el autor podía ayudar a orientar al lector por medio de la repetición de términos significativos. Los estudios clásicos de Zipf, realizados acerca de la frecuencia de las palabras en textos, determinaron que este criterio ayudaba a la comprensión del contenido de un texto (Zipf, 1949).

Sin embargo, no debe considerarse aisladamente, ya que esto podría potenciar palabras de uso común, palabras muy utilizadas en el discurso pero que no permiten distinguir claramente contenidos de documentos con temática diferente. Esta es la idea que reside tras los factores de corrección a la función TF. La ley de Luhn establece que las palabras más significativas, en

función de su frecuencia de aparición, son aquellas que tienen un valor de frecuencia medio. Ni las palabras con alta frecuencia en la colección, ni las palabras con frecuencia muy baja resultan ser realmente significativas.

La corrección a la frecuencia, como se ha visto, suele hacerse por medio de la frecuencia global del rasgo en la colección o la frecuencia de documento (el número de documentos en los que aparece un rasgo). El hecho de que el objetivo de las representaciones propuestas en esta tesis sea una representación autocontenida hace que, en este caso, la corrección a la frecuencia deba realizarse de otro modo. En el caso de representaciones autocontenidas lo que puede hacerse es considerar la frecuencia relativa de un rasgo en un documento o realizar la corrección por medio de la combinación de criterios.

4.5.2. Título

Parece obvio que si determinados rasgos se encuentran situados en el título de un documento entre las etiqueta <TITLE> <TITLE> deberían considerarse con una relevancia elevada dentro de la página web, ya que cabe esperar que resuman el contenido del documento. Recordemos que un título informativo y concreto ayuda al lector a orientarse (Spirydakis, 2000).

Sin embargo, el hecho de que el contenido de este elemento no sea visible en el cuerpo del documento HTML hace que este elemento no se encuentre en la mayor parte de las páginas web y, en muchos casos, sea resultado de una generación automática que no refleja el contenido real de la página.

Así pues, el título deberá ser un factor a tener en cuenta a la hora de calcular la relevancia de un rasgo en el contenido de un documento, pero no el único ni el más importante. Por otro lado, el contenido del elemento *title* no suele tener más de diez u once palabras. En las representaciones propuestas, al igual que en el caso de la frecuencia global, se considerará una frecuencia relativa dentro del título, es decir, la frecuencia con la que ha aparecido un rasgo en este elemento en relación a la frecuencia del rasgo más frecuente en el título de la página.

4.5.3. Posición

Una persona se puede orientar durante la lectura de un texto si encuentra una introducción que especifique el tema que se desarrolla en el texto que está leyendo (Spirydakis, 2000). Por tanto, la posición de un rasgo dentro de un documento puede resultar muy útil para encontrar su relevancia dentro del documento.

La elección del VSM como marco general en el que se encuadran las representaciones propuestas en esta tesis implica que los rasgos del vocabulario sean considerados de modo independiente. Además, la posición en la que aparecen dentro del contenido no se considera, aunque esta información podría asistir al lector en el proceso de verificación de hipótesis planteado en un modelo de lectura dirigido por contexto. Obviando por definición la posición

relativa entre rasgos, la posición absoluta toma especial importancia en determinados tipos de documentos como pueden ser los artículos periodísticos o científicos, donde la información suele estructurarse de un modo estándar.

Para Aebli (Aebli, 1988), el lenguaje posee una función expositiva, expresiva y de llamada. Los tipos de textos, según Aznar, Cros y Quintana (Aznar et al., 1991), y basándose en los presupuestos de Van Dijk (van Dijk, 1997), son tres:

- Narrativos; definidos como una concurrencia de sucesos y personas en el tiempo.
- Descriptivos; cuyo objetivo es la ordenación de objetos en el espacio, donde se distingue el tema (explícito o situacional) y la expansión (expresión de las propiedades o cualidades).
- Expositivos, que se dividen, a su vez, en:
  - Expositivos propiamente dichos; donde predominan las progresiones de tema constante y de hipertema.
  - Instructivos; con una presentación de la información lineal y no jerarquizada. No suele haber una información principal y otras secundarias, sino una serie de informaciones que poseen la misma relevancia y que están ordenadas temporalmente.
  - Argumentativos; donde se observa una estructura jerarquizada entre unas tesis y otras.

En esta tesis, las páginas web se consideran como nodos dentro de un grafo de hipertexto. Como hipótesis, se va a asumir que las páginas web tienen un carácter más bien expositivo. La posición puede dar pistas y resultar útil en la búsqueda de la relevancia de un rasgo en el contenido de un documento.

Por ejemplo, en la estructuración de los textos expositivos propiamente dichos suelen distinguirse las siguientes partes:

Introducción - Desarrollo - Conclusión

En los textos expositivos, la función lingüística predominante es la representativa. La información se suele construir sobre la base del siguiente esquema:

Planteamiento - Observaciones - Explicación - Solución

Respecto a la progresión informativa, en los textos argumentativos tiende a ser lineal, con el fin de destacar su lógica interna. Las tesis se van presentando como una sucesión de partes en las que parece que se pueden destacar:

Premisas - Argumentación - Conclusión

Es difícil interpretar si estas estructuras que aparecen en determinados tipos de textos impresos podrían ser aplicables a páginas web que, como se demostró en el estudio de Morkes,

no se leen de un modo secuencial. Sin embargo, como afirma Spyridakis cuando plantea aspectos relativos a cómo deben crearse los documentos HTML para que sean comprensibles, no está de más situar la información más importante en el primer párrafo del documento, o en el último [(Spirydakis, 2000), (Isakson y Spyridakis, 2003)]. El creador de un texto debe dominar las reglas gramaticales, así como poseer una serie de recursos relativos a la estructura interna (coherencia en las ideas) y estructura externa (o estilo), si quiere que su expresión escrita sea un buen vehículo de comunicación.

A partir de estas ideas, una primera aproximación podría ser dividir el texto de una página en cuatro partes, de forma que se considerase más representativo un rasgo que aparece en la primera y última parte, frente a otro que aparece en las partes centrales del discurso. Aunque gran parte de los documentos HTML contenidos en Internet no siguen ninguna estructura definida, existe una tendencia a emplear este esquema de creación en documentos.

4.5.4. Enfatizado

El hecho de que uno o más rasgos aparezcan frecuentemente enfatizados hace pensar que el autor ha querido destacarlos de entre el resto. Como se ha visto, en la creación de un documento es importante ir guiando la comprensión del lector, y una de las estrategias puede ser la ordenación jerárquica, el hecho de destacar algunos rasgos frente a otros, etc.

Así, los autores destacan la información que consideran importante para sus lectores. Esta tarea es difícil, en el sentido de que no siempre saben cuál va a ser la audiencia que van a tener, o si ésta será muy variada. Por su lado, el lector focalizará su atención y pondrá más mecanismos de comprensión en aquellas partes que más llamen su atención (Celsi y Olson, 1989).

Como se ha visto, el lenguaje HTML tiene etiquetas cuya función es la de destacar determinadas partes de un texto frente a otras (..., <u>...</u>, ..., ... o ...). El texto marcado con estas etiquetas llama la atención del usuario y, en muchos casos, basta con tomar estos fragmentos enfatizados para hacernos una idea sobre el contenido de un documento, aunque en otros casos esta derivada no sea tan directa.

Otra característica del texto escrito es la presencia de títulos en los encabezados (<h1> - <h6>). Suelen tratarse de forma especial desde el punto de vista tipográfico y, como hemos visto, tienen como función principal adelantar el contenido del texto. Pueden también aparecer en el índice, si lo hubiera, para que el lector pueda conocer de forma sencilla el contenido del documento. Los encabezados ayudan al lector a construir un marco conceptual para la decodificación de un texto [(Lorch et al., 1993), (Sanchez et al., 2001)]. En el lenguaje HTML, los títulos de secciones se realizan por medio de enfatizados, ya que el título de la página no es visible más que en la barra de título.

El uso de este tipo de anotaciones hace el texto más accesible visualmente (Spirydakis, 2000). Facilitan una lectura rápida, así como tareas de búsqueda de información, comprensión y reclamo, permitiendo establecer relaciones entre diferentes partes de un texto y ayudando

a percibir la información relevante del mismo (Spirydakis, 2000). Nielsen [(Nielsen, 1997b), (Nielsen, 1997a)] afirmaba que sería bueno emplear palabras enfatizadas y una organización clara en párrafos, además de emplear un lenguaje conciso.

En las representaciones propuestas no se consideran los colores de las fuentes como elementos de enfatizado, ya que lo que llama la atención de un usuario es el contraste más que el color. Como una página web puede tener una imagen de fondo, un cambio de color en la fuente puede ser simplemente para establecer un contraste alto entre el fondo y el texto.

Como consecuencia de lo anterior, el conjunto de elementos HTML que se van a considerar asociados al criterio enfatizado son:

- <b>, , <u>, , <big>, <i>
- <h1>, <h2>, <h3>, <h4>, <h5, <h6>
- <cite>, <dfn>, <blockquot>

4.6. Representación autocontenida basada en combinaciones heurísticas de criterios

A partir de las ideas expuestas en las secciones precedentes se puede pensar en desarrollar un modelo de representación autocontenido de páginas web que combine el conocimiento heurístico que el lector almacena con la experiencia, en su vertiente social, con la información de carácter tipográfico que ofrece el vocabulario HTML; se estaría asumiendo, por tanto, un modelo de lectura dirigido por el texto. El modelo de representación que se propone tendrá definida una función de proyección F basada en una selección de información y posterior combinación de la misma.

El hecho de que el carácter de Internet sea universal y que el acceso a sus contenidos se realice desde diversos ámbitos sociales y culturales, hace que la vertiente social presente en el proceso de lectura y escritura se homogeneice. Las recomendaciones acerca de cómo se debe presentar una información en un documento HTML, o el comportamiento de un lector que busca información en una página web, pueden considerarse comunes o de ámbito más o menos general. En cualquier caso, esta tesis se centra en una representación basada en heurísticas extraídas de la cultura occidental y la evaluación de las representaciones propuestas se hará con colecciones de páginas web escritas en inglés. Sin embargo, la idea subyacente en esta propuesta resulta totalmente generalizable.

El modelo de representación propuesto en esta tesis se basa en la siguiente idea: cuando a cualquiera de nosotros se nos pide que, en un tiempo reducido, leamos un texto para extraer el conjunto de palabras más representativos de su contenido, un mecanismo típico sería el de combinar información acerca de si la palabra aparece en el título, en la introducción o conclusión,

si aparece destacado de algún modo en el texto, si resulta frecuente dentro del documento, etc. De un modo similar será el comportamiento si se nos pide que leamos una página web. La decisión de un lector de entrar en un sitio web, seguir un hiperenlace o detenerse en una página dependerá del contenido de los títulos, encabezados y los enlaces (Spirydakis, 2000).

En los siguientes apartados se describen en detalle los aspectos fundamentales de las representaciones propuestas, así como cada una de las fases en las que se puede estructurar el sistema de representación aquí presentado.

4.6.1. Modelo de representación

Las representaciones propuestas en esta tesis se enmarcan dentro del VSM que se puede definir mediante una cuaterna $\langle X, \mathbb{B}, \mu, F \rangle$ (véase definición 13).

Como se vió en el capítulo 2, cualquier modelo de representación de documentos deberá contar, en primer lugar, con un vocabulario X cuya generación dependerá de la tarea que se vaya a realizar a continuación. En el caso de DC, este vocabulario X se crea con el conjunto de rasgos totales seleccionados en el corpus que se esté considerando. Sin embargo, en problemas de TC el corpus se divide en dos subconjuntos de documentos: uno de ellos se empleará para llevar a cabo las tareas de aprendizaje (con el que se creará X) y el otro (que suele ser de menor tamaño) se utilizará para evaluar posteriormente la calidad del sistema de clasificación. De este modo, cuando se quiere representar un documento perteneciente al subconjunto de prueba, podría ocurrir que dicho documento contuviera rasgos no presentes en X; esta situación se dará siempre que un rasgo no haya aparecido previamente en ningún documento del subconjunto de entrenamiento con el que se creó X. Si ocurre esto, la información relativa a este rasgo podría perderse. En el caso de DC esta situación no puede darse, ya que no se realiza un entrenamiento.

En los modelos de representación propuestos, el σ-álgebra $\mathbb{B}$ es el definido en un espacio vectorial (definición 11). La métrica μ –función de distancia entre objetos dentro del espacio de representación– puede ser cualquier función capaz de establecer la similitud entre dos objetos dentro de un espacio vectorial. Esta μ, normalmente, se establece en función del uso posterior que se quiera dar a las representaciones. En problemas de TC vendrá definida por la fase de aprendizaje y en nuestro caso, se considera la distancia euclídea. En problemas de DC son típicas la función coseno, la distancia euclídea o la función de correlación, métrica utilizada en las representaciones propuestas en esta tesis.

La función de proyección F –encargada de ponderar cada uno de los rasgos t_i dentro del vector de representación $\vec{d_j}$– es el elemento fundamental que permite distinguir dos métodos diferentes dentro de un mismo modelo vectorial. En los capítulos 2 y 3 se vieron algunas de las funciones empleadas tanto en el ámbito de la representación de textos, como de páginas web.

Las representaciones propuestas en esta tesis se basan, fundamentalmente, en la definición de estas funciones de proyección F. A partir del vocabulario HTML se extrae información relativa

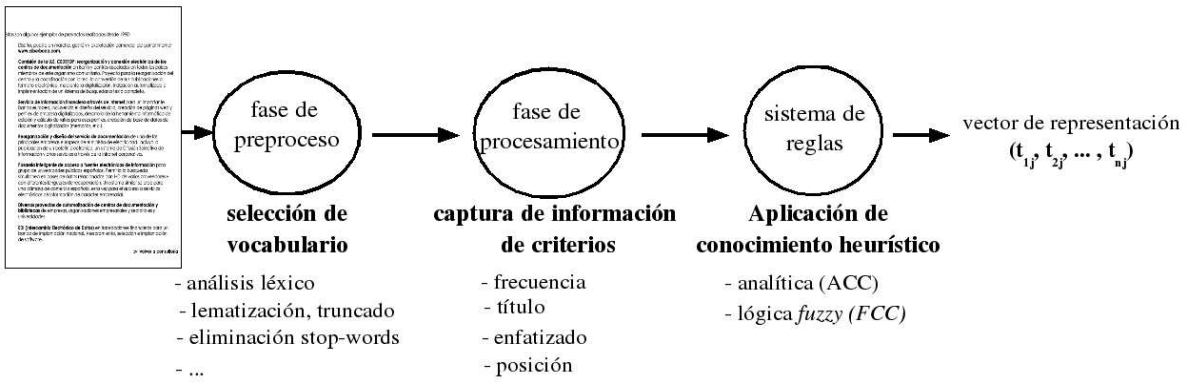

Figura 4.4: Arquitectura funcional del sistema de representación propuesto.

a los criterios heurísticos seleccionados en la sección 4.5 y que serán posteriormente combinados para encontrar la relevancia de un rasgo en el contenido de una páginas web.

La arquitectura del sistema de representación propuesto será secuencial, inspirándose en el sistema de procesamiento presentado en la sección 4.4.2. Dentro de esta arquitectura se pueden distinguir las siguientes fases, que se corresponderán directamente con las fases *preproceso*, *procesamiento* y *sistema de reglas*:

- *Selección de vocabulario*
- *Captura de información de criterios*
- *Aplicación de conocimiento heurístico*

La figura 4.4 muestra la arquitectura funcional del sistema de representación propuesto. El fin último de este proceso será la representación una página web y se correspondería con la fase de adquisición de conocimiento por medio de la lectura. A continuación, se presentan cada una de las etapas de forma detallada.

4.6.2. Selección del vocabulario

La primera fase en un sistema de procesamiento de la información como el planteado por David Klahr sería la correspondiente a una fase de *preproceso*. En ella, se filtra información de entre el total de estímulos que una persona percibe por cualquiera de sus sentidos. En el caso de un procesamiento visual de un texto –y simplificando–, se filtrarían de entre todo el conjunto de signos visibles presentes en un texto, aquellos elementos que puedan tener un sentido para el lector, así como aquellas partes que hayan sido destacadas intencionadamente por el autor mediante el uso de enfatizados. Se realizaría la identificación de caracteres, palabras, oraciones, etc.

Esta fase se corresponde con una fase de *selección de vocabulario*, en la que se transforma una información externa –el texto que se está leyendo como un conjunto de signos o caracteres– con el objetivo de poder tratarla en un proceso posterior. Esta primera fase es necesaria para crear una base de hechos que contenga el estado inicial y global del problema.

En esta fase de *selección de vocabulario* se realizan los procesos descritos en la sección 2.5 (análisis léxico, lematización, truncado, eliminación de *stop-words*, ...) u otros con la misma finalidad.

4.6.3. Captura de información de los criterios

Dentro del sistema de procesamiento de información que estamos considerando, la *fase de procesamiento* sería la encargada de transformar la información filtrada en la fase de *preproceso*, de modo que pueda aplicarse posteriormente un sistema de reglas. Esta fase se corresponde con una etapa de *captura de información de criterios*.

Es importante destacar que el objetivo de esta tesis no es tanto ver qué criterios habría que combinar, sino saber si combinando heurísticas familiares se podrían mejorar las representaciones autocontenidas encontradas en la literatura. Además, considerando pocos criterios es posible establecer más fácilmente modos de combinación entre ellos. Una vez validado el modelo se podrá explorar la definición de otros criterios, tanto para páginas web como para otro tipo de documentos escritos con lenguajes de marcado con vocabularios diferentes al HTML.

Los criterios que se han seleccionado para probar la validez de este modelo son los descritos en la sección 4.5:

- La **frecuencia** de aparición de un rasgo en el texto.
- La frecuencia de aparición de un rasgo en el **título** del documento.
- La **posición** en la que aparece un rasgo dentro del texto.
- La frecuencia de un rasgo en las partes **enfatizadas** del documento.

La información relacionada con cada uno de estos criterios resulta accesible desde el código HTML.

4.6.4. Aplicación de conocimiento heurístico

Una vez seleccionado el conjunto de criterios a combinar, el siguiente paso sería establecer cómo se va a realizar la combinación de la información suministrada por ellos. En esta tesis se presentan dos representaciones cuya diferencia principal radica en el modo en que se realiza dicha combinación: como una función de combinación lineal de criterios, que se verá en detalle en el capítulo 5, y como una combinación borrosa, desarrollada en el capítulo 6.

Ambas tratan de contener implícitamente el conocimiento heurístico que se aplica en el proceso de lectura y escritura de un página web y que se ha descrito en detalle en la sección 4.5.

4.7. Conclusiones

En este capítulo se ha presentado un modelo general para la obtención de representaciones autocontenidas de páginas web. Este modelo será la base de las representaciones propuestas en esta tesis, obtenidas tras la aplicación de una serie de fases de procesado y almacenamiento de información, que finalizan con la aplicación de un conocimiento heurístico que determina si la información procesada es relevante o no.

La primera fase de *preproceso* se correspondería con las etapas de análisis léxico, eliminación de *stop-words* y lematización o truncamiento. La segunda sería una fase de *procesamiento*, encargada de la captura de los criterios heurísticos a combinar. Para ello, se tendría en cuenta el vocabulario HTML. A partir de esta información es posible relacionar determinados contenidos de una página web con los criterios seleccionados. La última fase se correspondería con la aplicación del conocimiento heurístico disponible para realizar la representación.

Se han propuesto dos formas de combinación heurística de criterios: una basada en una combinación lineal y otra basada en un sistema de reglas borrosas. Ambas se desarrollarán en detalle en los capítulos 5 y 6, y pueden entenderse como una función de ponderación (función de proyección F) aplicada sobre los rasgos de un documento o de una colección, dentro de un modelo general de representación de documentos.

De este modo, estas representaciones se basarían en la identificación de criterios usados en la extracción de contenido relevante de un texto, su asociación con un subconjunto de elementos HTML que puedan cubrirlos para, a continuación, realizar una fusión de la información asociada a ellos mediante una combinación lineal o un sistema de reglas borrosas.

Una de las ideas fundamentales que se encuentra tras este modelo general es que es mejor combinar diferentes fuentes de conocimiento que considerar cada una de ellas por separado. Posiblemente, ningún criterio sea lo suficientemente bueno por sí sólo y según las situaciones, unos funcionarán mejor que otros.

Capítulo 5

Representación autocontenida basada en combinaciones analíticas de criterios

"La formulación de un problema es más importante que su solución."
Albert Einstein

Una vez establecido en el capítulo 4 el modelo teórico general para la creación de las representaciones autocontenidas de páginas web presentadas en esta tesis, en el presente capítulo se establecen las características específicas de un modelo de representación con una función de proyección basada en una combinación analítica de criterios heurísticos. En primer lugar se definen las funciones de captura de cada uno de los criterios considerados para, a continuación, establecer los coeficientes de la combinación analítica. Los valores de los coeficientes fijan la importancia que se le quiera dar a cada criterio.

5.1. Introducción

En este capítulo se presenta un método para establecer la relevancia de un rasgo en el contenido de una página web en función de una combinación lineal de criterios heurísticos (ACC). La forma más sencilla y directa de realizar una combinación es por medio de una función de combinación lineal. Bastaría con definir funciones de captura para cada uno de los criterios a combinar y, a continuación, ponderar cada uno de ellos con un determinado peso. De este modo, es posible establecer un peso diferente para cada criterio, de modo que unos podrían aportar más que otros en la combinación final.

Una vez que se definan las funciones de captura para cada criterio, $f_{criterio}$, se puede establecer una función de ponderación F basada en una combinación analítica de criterios como:

$$F(\vec{t_i}, \vec{d_j}) = C_1 f_{criterio_1}(\vec{t_i}, \vec{d_j}) + \ldots + C_n f_{criterio_n}(\vec{t_i}, \vec{d_j}) \tag{5.1}$$

Donde $F(\vec{t_i}, \vec{d_j})$ representa la relevancia del rasgo t_i en el contenido del documento d_j y

supone una función de proyección F dentro de la definición de un modelo de representación. De este modo, $C_k f_{criterio_i}(\vec{t_i}, \vec{d_j})$ representa la aportación del criterio k-ésimo a la relevancia final del rasgo t_i en el documento d_j.

Normalmente se considerará la siguiente condición de normalización para el conjunto de coeficientes C_k:

$$\sum_{k=1...n} C_k = 1 \tag{5.2}$$

Con esta combinación se busca recoger las relaciones implícitas existentes, si las hubiera, entre los diferentes criterios considerados, y se podrían ordenar los rasgos presentes en un documento HTML en base a su importancia dentro del contenido del documento.

5.2. Definición de las funciones de captura para los criterios

En esta sección se presentan las diferentes funciones de captura para cada uno de los cuatro criterios considerados en la sección 4.5. Una vez definidas estas funciones, ya se podrá establecer una representación autocontenida basada en una combinación analítica de criterios heurísticos.

5.2.1. Frecuencia

La función de ponderación de la frecuencia de un determinado rasgo t_i en el contenido de un documento d_j se expresa como:

$$f_{frec}(\vec{t_i}, \vec{d_j}) = \frac{f_{ij}}{N_j} \tag{5.3}$$

siendo f_{ij} la frecuencia del rasgo t_i en d_j y N_j la suma de las frecuencias del total de rasgos presentes en el documento d_j. Esta definición asegura valores normalizados para la función, de forma que $\sum_{1...k} f_{frec}(\vec{t_i}, \vec{d_j}) = 1$, donde k es el número de rasgos diferentes en $\vec{d_j}$. Aunque un rasgo puede tener una frecuencia mayor que 1, siempre se cumplirá la condición $\sum_{1...k} f_{ij} = N_j$.

5.2.2. Título

La función de ponderación respecto al criterio título para un determinado rasgo t_i en el contenido de un documento d_j se expresa como:

$$f_{tit}(\vec{t_i}, \vec{d_j}) = \frac{t_{ij}}{N_{tit}(j)} \tag{5.4}$$

siendo t_{ij} la frecuencia del rasgo t_i en el título del documento d_i y N_{tit} el número total de rasgos en el título del documento. Al igual que en el caso del criterio frecuencia, se cumple la condición de normalización $\sum_{1...k} f_{tit}(\vec{t_i}, \vec{d_j}) = 1$, donde k es el número de rasgos diferentes presentes en el

título. En este caso, $k \leqslant N_{tit}$ y $\sum_{1...k} t_{ij} = N_{tit}$.

5.2.3. Enfatizado

La función de ponderación respecto al enfatizado para un rasgo t_i en el contenido de un documento d_j se expresa como:

$$f_{enf}(\vec{t_i}, \vec{d_j}) = \frac{e_{ij}}{N_{enf}(j)} \tag{5.5}$$

siendo e_{ij} la frecuencia del rasgo t_i en el conjunto de elementos enfatizados del documento d_j, y N_{enf} el número total de rasgos enfatizados en del documento. Como en los casos anteriores, esta función está normalizada, de modo que $\sum_{1...k} f_{enf}(\vec{t_i}, \vec{d_j}) = 1$, donde k es el número de rasgos diferentes presentes en el conjunto de elementos enfatizados. En este caso, siempre se cumplirá que $k \leqslant N_{enf}$ y $\sum_{1...k} e_{ij} = N_{enf}$.

5.2.4. Posición

Calcular la relevancia de un rasgo en función de la posición requiere un análisis un poco más detallado. Mientras que un rasgo tiene una frecuencia total en el documento, el título o las partes enfatizadas, su valor respecto a la posición vendrá dado por el conjunto de posiciones en las que aparezca. Por tanto, habrá que considerar la frecuencia de aparición del rasgo en las diferentes partes en las que se divida el documento.

Para ello, el texto de una página web se divide en cuatro partes. Sin llegar a asociar ninguna de ellas con las partes específicas en las que se puede dividir los diferentes tipos de textos comentados en el punto 4.5.3, se considera, como aproximación inicial, que un documento puede seguir una ordenación de sus contenidos según este tipo de esquemas basados en 4 partes. Por otro lado, el número 4 entraría dentro de los 5 ± 2 conceptos que una persona es capaz de manejar al mismo tiempo (Spirydakis, 2000).

En este trabajo de tesis se considerará que los rasgos aparecidos en la primera y cuarta parte del texto son más relevantes que los presentes en las partes centrales. Se define, por tanto, una posición *preferente* relativa a estas partes primera y cuarta, dejando como posición *estándar* las partes segunda y tercera. Para los experimentos se asignó un peso $\frac{3}{4}$ veces mayor a los rasgos de las posiciones preferentes frente a los de las posiciones estándar. De este modo, la función de captura del criterio posición tiene la expresión:

$$f_{pos}(\vec{t_i}, \vec{d_j}) = \frac{\frac{3}{4} f_{1,4}(\vec{t_i}, \vec{d_j}) + \frac{1}{4} f_{2,3}(\vec{t_i}, \vec{d_j})}{\sum_{l=1...k} (\frac{3}{4} f_{1,4}(\vec{t_l}, \vec{d_j}) + \frac{1}{4} f_{2,3}(\vec{t_l}, \vec{d_j}))} \tag{5.6}$$

siendo $f_{1,4}(\vec{t_i}, \vec{d_j})$ la frecuencia del rasgo t_i en posiciones preferentes del documento d_j y $f_{2,3}(\vec{t_i}, \vec{d_j})$ la frecuencia del rasgo t_i en posiciones estándar en el documento d_j. Como en casos anteriores, k representa el número total de rasgos diferentes en el documento.

La función está normalizada ya que la frecuencia total del rasgo t_i en el documento d_j es $f_{ij} = f_{1,4}(\vec{t_i}, \vec{d_j}) + f_{2,3}(\vec{t_i}, \vec{d_j})$. De este modo, la expresión anterior se transforma en:

$$f_{pos}(\vec{t_i}, \vec{d_j}) = \frac{2f_{1,4}(\vec{t_i}, \vec{d_j}) + f_{ij}}{\sum_{l=1...k}(2f_{1,4}(\vec{t_l}, \vec{d_j}) + f_{lj})} = \frac{2f_{1,4}(\vec{t_i}, \vec{d_j}) + f_{ij}}{2f_{j[1,4]} + N_j} \tag{5.7}$$

donde $f_{j[1,4]}$ representa la suma de las frecuencias totales de los rasgos presentes en posiciones preferentes del documento d_j, y N_j la suma de las frecuencias del total de rasgos presentes en el documento d_j.

Por último, esta expresión puede generalizarse de modo que se pueda aplicar cualquier ponderación a las posiciones preferente y estándar. Considerando que a es el peso aplicado a las posiciones preferentes y b el peso de las posiciones estándar, e imponiendo la condición $a + b = 1$, entonces la función de asignación de relevancia para el criterio posición de un rasgo t_i en un documento d_j toma la forma:

$$f_{pos}(\vec{t_i}, \vec{d_j}) = \frac{(a-b)f_{1,4}(\vec{t_i}, \vec{d_j}) + bf_{ij}}{(a-b)f_{j[1,4]} + bN_j} \tag{5.8}$$

5.3. Establecimiento de los coeficientes de la combinación ACC

Como se vió en el capítulo 2, la función de cálculo de relevancia de un rasgo supone la función F dentro cualquier modelo de representación de documentos, definido como $\langle X, \mathbb{B}, \mu, F \rangle$.

De este modo, para generar una representación autocontenida basada en combinaciones analíticas de criterios, se propone una función:

$$F \quad : \quad ACC(\vec{t_i}, \vec{d_j}) = C_{frec}f_{frec}(\vec{t_i}, \vec{d_j}) + C_{tit}f_{tit}(\vec{t_i}, \vec{d_j}) + C_{enf}f_{enf}(\vec{t_i}, \vec{d_j}) + C_{pos}f_{pos}(\vec{t_i}, \vec{d_j}) \tag{5.9}$$

donde se impone la condición $\{C_{frec}, C_{tit}, C_{enf}, C_{pos} \in [0,1] \mid C_{frec} + C_{tit} + C_{enf} + C_{pos} = 1\}$.

El conjunto de valores para los coeficientes se estima tras el estudio estadístico basado en distribuciones de frecuencias que se explica a continuación. Se creó una colección de 200 páginas web relacionadas con 4 temas diferentes. Para la búsqueda de las páginas se emplearon términos de consulta muy diversos en distintos motores de búsqueda. Se utilizaron también directorios web temáticos preclasificados. En ninguno de los casos se descargaba un número mayor de tres documentos por servidor accedido, tratando de representar lo más posible la naturaleza heterogénea de la Web.

Se aplicaron las fases de análisis léxico, eliminación de *stop-words* y truncamiento descritas en el capítulo 7. A continuación se aplicaron por separado las funciones de captura de cada criterio –$f_{frec}(\vec{t_i}, \vec{d_j})$, $f_{tit}(\vec{t_i}, \vec{d_j})$, $f_{enf}(\vec{t_i}, \vec{d_j})$, $f_{pos}(\vec{t_i}, \vec{d_j})$–, y los rasgos de cada documento fueron ordenados en función del valor de relevancia que asignaban dichas funciones. De este modo,

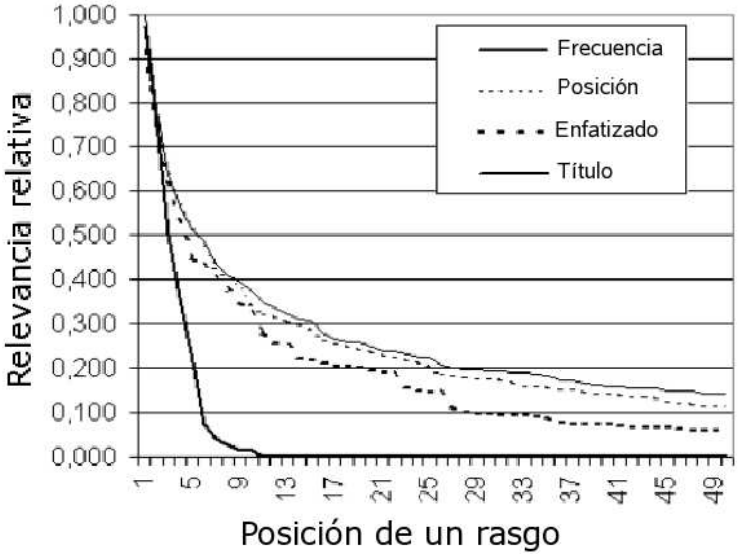

Figura 5.1: Relevancia media normalizada por criterio (relevancias relativas) para los 50 rasgos más relevantes de una página web.

se obtuvieron 4 vectores de representación para cada página web de la colección de referencia, relativos a los 4 criterios considerados. Para representar una página sólo se tomaron los 50 rasgos más relevantes de cada página, en caso de que los hubiera.

A continuación se representó gráficamente, para cada criterio, el valor medio normalizado de la relevancia de los 50 rasgos más relevantes (figura 5.1). Estas cantidades se representaron en el eje Y, mientras que la posición en ese *ranking* de los 50 rasgos más relevantes se representó en el eje X.

Analizando la figura 5.1 se pueden extraer varias conclusiones. En primer lugar, se observa que estas funciones, correspondientes a cada uno de los 4 criterios considerados, siguen aproximadamente la ley de Zipf, descrita por la expresión $P_n = \frac{1}{n^a}$, donde P_n es la frecuencia relativa del rasgo n-ésimo (frecuencia normalizada a la máxima frecuencia en el documento) en una ordenación basada en un *ranking* de frecuencias, y donde a es un coeficiente con valor entre $0,4$ y $0,5$ para los criterios *frecuencia*, *posición* y *enfatizado*, y entre $0,8$ y $0,9$ para el caso del criterio *título*. Tanto en el caso del criterio enfatizado como en la posición se observa un comportamiento muy similar al de la frecuencia, es decir, la distribución de relevancias cuando se consideran únicamente los criterios enfatizado o posición es muy parecida al caso de la frecuencia de aparición de los rasgos en un documento. Sin embargo, en estos casos es de esperar que no haya que eliminar, como hizo Luhn con la frecuencia, los rasgos más pesados por medio de una función discriminante (figura 2.1). En el caso del título, el hecho de que no se encontraran páginas con títulos mayores de 11 rasgos hace que la función decaiga mucho más rápidamente.

Por otro lado, los valores medios relativos a los criterios frecuencia y posición son mayores que en el caso del título y el enfatizado. Esto es debido a que las funciones de captura de

estos criterios siempre aportan un valor distinto de cero para todos los rasgos presentes en un documento; todo rasgo tiene una posición y una frecuencia, mientras que no siempre dicho rasgo aparecerá en el título o en algún elemento asociado con el criterio enfatizado.

En la colección de muestra el porcentaje de páginas con algún rasgo enfatizado fue del 89,30 %, mientras que en el caso del título fue de 97,05 %. Observando estos valores, y viendo la pendiente negativa que sufre su función de captura, puede pensarse que el criterio título es muy adecuado si se quieren obtener dimensiones muy pequeñas en los vectores de representación. Sin embargo, realizando un análisis de los contenidos de los títulos de las páginas que constituyeron esta colección de muestra, se observó que sólo el 51,97 % de los mismos eran realmente representativos del contenido del documento. En el resto de los casos, los títulos habían sido generados probablemente de modo automático con algún editor de documentos HTML y, en muchos casos, contenían el texto "New page 1".

Para la estimación de los valores de los coeficientes C_{frec}, C_{tit}, C_{enf} y C_{pos} se consideró, en primer lugar, que no debían ser valores específicos de una determinada colección o conjunto de documentos, por lo que los valores que se establecieran en este punto no deberían de modificarse en una posterior experimentación desarrollada con otras colecciones.

La figura 5.1 muestra que la frecuencia y la posición tienen un comportamiento muy similar, por lo que podrían llevar asociado un mismo valor para sus coeficientes, lo que supone un mismo peso en la relevancia final. Además, el hecho de que tomaran valores distintos de cero en todos los rasgos invita a considerar que la suma de sus pesos debiera ser mayor del 50 % en el cómputo final de la función ACC. Asumiendo esto, se fija el valor de los pesos para la frecuencia y posición en $C_{frec} = C_{pos} = 0{,}3$, de modo que su relevancia conjunta sume un 60 %. En el caso del enfatizado y el título se tendrán que repartir el otro 40 % y, considerando que el número de documentos con rasgos enfatizados era 1,7 veces mayor que el número de documentos con título semánticamente representativo, se elige $C_{enf} = 0{,}25$ y $C_{tit} = 0{,}15$, ya que la relación entre un 25 % y 15 % es de 1,67, es decir, de 1,7 aproximadamente.

5.4. Cálculo de la relevancia de un rasgo con ACC

Una vez establecidos los coeficientes que permiten la ponderación de cada uno de los criterios considerados, la función de proyección ACC se expresa como:

$$ACC_{0,3/0,15/0,25/0,3}(\vec{t_i}, \vec{d_j}) = 0,3 f_{frec}(\vec{t_i}, \vec{d_j}) + 0,15 f_{tit}(\vec{t_i}, \vec{d_j}) + 0,25 f_{enf}(\vec{t_i}, \vec{d_j}) + 0,3 f_{pos}(\vec{t_i}, \vec{d_j}) \tag{5.10}$$

El comportamiento de esta función $ACC_{0,3/0,15/0,25/0,3}$ se comparó con la función de ponderación TF, teniendo en cuenta que ésta se puede expresar como la función $ACC_{1/0/0/0}$.

En la figura 5.2 se observa que los valores medios de relevancia obtenidos con $ACC_{0,3/0,15/0,25/0,3}$ son menores que en el caso de TF, lo que indicaría un mayor grado

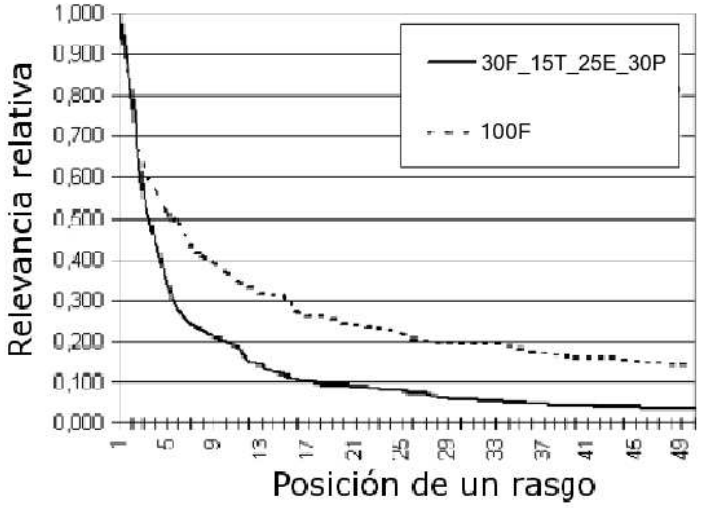

Figura 5.2: Comparación entre las relevancias medias normalizadas (relevancias relativas) de las funciones $ACC_{0,3/0,15/0,25/0,3}$ y TF para los 50 rasgos más relevantes de una página web.

de discriminación para la función de combinación analítica propuesta. Esto significa que, estableciendo un mismo umbral en las relevancias relativas en ambos casos, el conjunto de rasgos seleccionados con la función ACC sería menor que con TF. Dicho de otro modo, el conjunto de los rasgos más relevantes de una página web tendrán valores más cercanos entre sí, y más cercanos al rasgo más relevante, si se emplea la función ACC que si se emplea la función TF.

Si se toma el umbral de $0,4$ significaría que se establece un corte en un valor de relevancia igual al $40\,\%$ de la relevancia máxima. En el caso de la función ACC se seleccionarían aproximadamente 5 rasgos para llegar a este umbral, frente a los 8 de la función TF. Sin embargo, hay que tener cuidado con estas apreciaciones, ya que el grado de discriminación no siempre tiene por qué ser positivo. En la figura 5.1 el título aparecía como el criterio más discriminante y, sin embargo, como ya se ha explicado, no es conveniente emplearlo como criterio único.

En cualquier caso, cualquier función $ACC_{C_{frec}/C_{tit}/C_{enf}/C_{pos}}$, sea cual sea el conjunto de coeficientes considerados, resultará siempre más discriminante que la frecuencia (el criterio menos discriminante) y menos que el título (criterio más discriminante). En la figura 5.3 se muestra el comportamiento de la función $ACC_{C_{frec}/C_{tit}/C_{enf}/C_{pos}}$ para diferentes conjuntos de coeficientes. Éstos se escogieron variando ligeramente los valores de referencia: $C_{frec} = C_{pos} = 0,3$, $C_{tit} = 0,15$ y $C_{enf} = 0,25$.

En la figura puede verse que la función $ACC_{0,3/0,15/0,25/0,3}$ resultó ser la menos discriminante de todas las $ACC_{C_{frec}/C_{tit}/C_{enf}/C_{pos}}$ consideradas. Esto es debido a que fue la combinación que aportaba una menor ponderación al título y una mayor a la frecuencia, los criterios más y menos discriminantes respectivamente. Aún así, como ya se ha apuntado, la intención era fijar unos coeficientes que resultaran independientes de la colección de referencia y no modificarlos. A partir de este punto, y por simplicidad, se hablará simplemente de ACC en lugar de $ACC_{0,3/0,15/0,25/0,3}$.

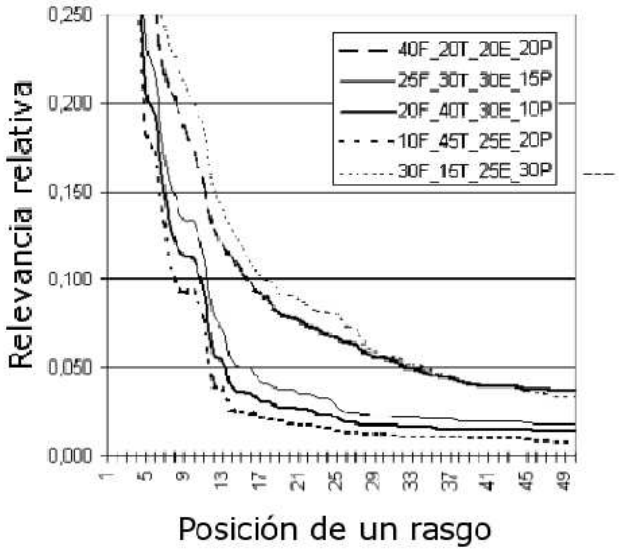

Figura 5.3: Comparación entre las relevancias medias normalizadas (relevancias relativas) para diferentes funciones $ACC_{C_{frec}/C_{tit}/C_{enf}/C_{pos}}$ para los 50 rasgos más relevantes de una página web.

5.5. Conclusiones

En este capítulo se han fijado las bases para el desarrollo general de una representación autocontenida de páginas web basada en combinaciones lineales de criterios heurísticos. Para ello, se ha definido una función de ponderación a la que se ha llamado ACC, *Analytical Combination of Criteria*.

Una vez definido un conjunto de criterios heurísticos a combinar, el siguiente paso es el establecimiento de funciones de captura para cada uno de ellos. Estas funciones deberán estar parametrizadas con información presente en la página web, asegurando así que la representación resultante sea una representación autocontenida. En esta tesis se han considerado los criterios descritos en el capítulo 4, aunque podrían establecerse otros diferentes sin que por ello se modificase el modelo.

Una vez fijados los criterios, deberán determinarse los valores de los coeficientes correspondientes a cada criterio en la combinación analítica. Por medio de estos coeficientes se asigna diferente importancia a cada criterio, tratando de recoger en su conjunto el conocimiento heurístico que se quiere aplicar al proceso de representación. Para ello, se realiza un análisis sobre una colección de páginas web de referencia parecido al que se llevó a cabo para el establecimiento de la ley de Zipf. La idea reside en encontrar unos coeficientes para la combinación que resulten generales y que funcionen adecuadamente para distintas colecciones. Los coeficientes se eligieron en función del comportamiento de cada criterio en relación a la distribución de pesos que producía cada uno de ellos, por separado, en la colección de referencia.

Capítulo 6

Representación autocontenida de páginas web a partir de un sistema de reglas borrosas

"Según aumenta la complejidad, las declaraciones precisas pierden significado y las declaraciones con significado pierden precisión"
Lofti A. Zadeh

Establecido el modelo teórico general para construir representaciones autocontenidas de páginas web y definidas las características de las representaciones basadas en combinaciones analíticas de criterios, en este capítulo se establece una representación basada en una combinación de criterios heurísticos por medio de un sistema de reglas borrosas. En primer lugar, se definen los conjuntos borrosos del sistema y se relacionan con cada uno de los criterios considerados en la representación. Seguidamente, se establecen las funciones de pertenencia para cada conjunto borroso y las funciones de captura para cada criterio. Finalmente, se define el sistema de inferencia borroso y la base de conocimiento del sistema, formada por un conjunto de reglas IF-THEN que establecen el comportamiento del mismo.

6.1. Introducción

Cuando se trata de encontrar la relevancia de un rasgo en el contenido de una página web, los criterios a combinar no siempre deberían tratarse de un modo independiente, como sucede cuando se aplica una función de combinación lineal. A menudo, un criterio toma verdadera importancia en unión con otro. Por ejemplo, podría suceder que el título de un documento tuviera una componente retórica, de modo que los rasgos presentes en él no ayudaran a describir adecuadamente el contenido del documento. Por este motivo, los rasgos presentes en el título deberían tener mayor relevancia si, además, aparecen enfatizados o en partes del documento que se consideren importantes. En ese caso se podría pensar que el título resume el tema principal de la página.

Este tipo de consideraciones no se pueden recoger con un sistema de representación basado en una combinación lineal de criterios. En ese caso, si un rasgo es muy frecuente en el título de una página, la componente relativa al criterio título tomará un valor que se sumará siempre a la

relevancia total del rasgo, independientemente del valor que tomen el resto de las componentes correspondientes a los demás criterios. Un sistema de reglas borrosas puede suponer un mecanismo más apropiado para combinar las funciones de captura asociadas a cada uno de los criterios heurísticos que se estén considerando. Con estos sistemas se combina conocimiento y experiencia en un conjunto de expresiones lingüísticas que manejan palabras en lugar de valores numéricos [(Iserman, 1998), (Hansen, 2000)].

La lógica borrosa se ha mostrado como un marco de trabajo adecuado para capturar el conocimiento experto humano, aplicando heurísticas a la hora de resolver la ambigüedad inherente a procesos de razonamiento cualitativo. Esta es una característica importante, habida cuenta de que el objetivo principal de esta tesis es encontrar una representación de páginas web a partir del conocimiento heurístico empleado en procesos de lectura y escritura de textos.

6.2. Lógica borrosa (*fuzzy logic*)

La lógica borrosa es una rama de la inteligencia artificial que permite tratar con modos de razonamiento imprecisos; en cierto modo, es una extensión de la lógica multivaluada. Sus principios fueron establecidos por Lofti Zadeh en 1965 (Zadeh, 1965) a partir de los denominados conjuntos borrosos.

En un sentido estricto, la lógica borrosa es un sistema lógico cuyo objetivo es formalizar el razonamiento aproximado. Su raíz básica es la lógica multivaluada aunque se extiende con conceptos nuevos que añaden efectividad al procesamiento borroso como, por ejemplo, las variables lingüísticas (Zadeh, 1988). De forma general, las principales características que diferencian la lógica borrosa de la bivaluada tradicional son:

- En lógica bivalente, una proposición p tiene que ser "*verdadera*" o "*falsa*". En la lógica borrosa, una proposición tendrá una condición de verdad que será un elemento del conjunto T de posibles valores de verdad.
- Los predicados no tienen necesariamente que representar conceptos concretos, sino que estos pueden ser borrosos, por ejemplo: "*grande*", "*bajo*", etc.
- Es posible emplear cuantificadores del tipo "*muchos*", "*algunos*", "*pocos*", etc.
- La lógica borrosa proporciona un método para representar el contenido de modificadores de los predicados tanto si son borrosos como si no. Para ello, será necesario procesar variables lingüísticas, es decir, variables cuyo valor son palabras o sentencias del lenguaje natural.
- La lógica borrosa tiene básicamente tres formas de cualificar una proposición p: cualificación de su verdad, su probabilidad y su posibilidad.

La lógica borrosa, como herramienta para el tratamiento de la imprecisión, fue aplicada por primera vez al control automático en 1973. Desde entonces, ha sido aplicada a muchos otros

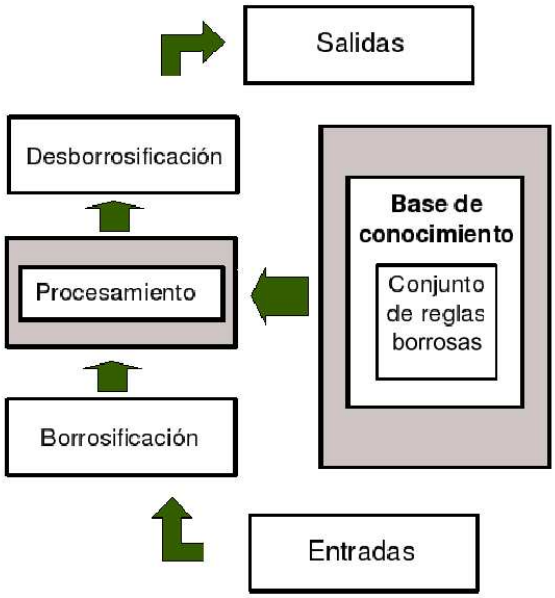

Figura 6.1: Esquema conceptual de las etapas de un controlador borroso con fase de borrosificación y desborrosificación.

problemas de muy diversas áreas, como a la recuperación de información [(Bookstein, 1981), (Cater y Kraft, 1989) (Bordogna y Pasi, 1995), (Molinari y Pasi, 1996), (Cordón et al., 1999), (Herrera-Viedma, 2001)]; la clasificación automática de textos [(Lam y Low, 1997), (Tikk et al., 2003)] o el *clustering* de documentos [(Höppner et al., 1999), (Kraaij, 2002)].

Un sistema borroso permite establecer reglas de combinación en las que se maneje cierta incertidumbre. La información que hay que aportar al sistema procede de un conocimiento experto, es decir, de un conocimiento heurístico. En general, los sistemas borrosos se pueden clasificar, según la naturaleza de sus entradas y salidas, en:

- *Sistemas borrosos puros*, donde tanto las entradas como las salidas del sistema son conjuntos borrosos. Internamente disponen de una base de reglas borrosas y de un mecanismo o motor de inferencia borroso.

- *Sistemas borrosos con fase de borrosificación y desborrosificación*. En este caso, tanto las entradas como las salidas del sistema son valores numéricos concretos. Este es el caso concreto del sistema propuesto en esta tesis. Su estructura básica se halla representada en la figura 6.1. A la entrada se sitúa una etapa de borrosificación que se encarga de traducir la entrada a conjuntos borrosos. A continuación, pasa por un sistema borroso puro, que contendrá una base de reglas obtenidas a partir de las heurísticas extraídas de los procesos de lectura y escritura de textos. Finalmente, a la salida se sitúa una fase de desborrosificación que se encarga de transformar el conjunto borroso de salida a términos numéricos[1].

[1]La fase de desborrosificación no es necesaria cuando se emplean consecuentes funcionales (sistemas borrosos Takagi-Sugeno)

Un sistema borroso se construye a partir de la Teoría de Conjuntos Borrosos.

6.2.1. Teoría de Conjuntos Borrosos

La Teoría de Conjuntos Borrosos se basa en el reconocimiento de que determinados conjuntos poseen unos límites imprecisos. Estos conjuntos están constituidos por colecciones de objetos para los cuales la transición de "pertenecer" a "no pertenecer" es gradual.

Conjuntos borrosos

Definición 28 *Sea U una colección de objetos, por ejemplo $U = R^n$, denominado universo del discurso, un* ***conjunto borroso*** *F en U está caracterizado por una función de pertenencia $\mu_F(u) \rightarrow [0,1]$, donde $\mu_F(u)$ representa el grado de pertenencia de $u \in U$ al conjunto borroso F.*

Es decir, $\mu(u)$ indica el grado de compatibilidad del valor asignado a la variable u, con el concepto representado por F, donde F es un valor lingüístico (concepto, etiqueta lingüística) asociado al conjunto borroso definido por $\mu(u)$.

Definición 29 *Sean U y V dos universos del discurso, una* ***relación borrosa*** *R es un conjunto borroso en el espacio producto $U \times V$; es decir, R posee una función de pertenencia $\mu_R(u,v) \rightarrow [0,1]$ donde $u \in U$ y $v \in V$.*

Definición 30 *Sean A y B dos conjuntos borrosos en dos universos de discurso U y V respectivamente, una* ***implicación borrosa****, denotada por $A \rightarrow B$, es un tipo especial de relación borrosa en $U \times V$. Una implicación borrosa se puede entender como una interpretación de una regla de tipo IF-THEN, expresada en términos borrosos.*

Definición 31 *Una base de conocimiento es una colección de* ***reglas borrosas*** IF-THEN *del tipo:*

$$R^{(l)} : \textit{IF } x_1 \textit{ is } F_1^l \textit{ and } \ldots x_n \textit{ is } F_n^l, \textit{ THEN } y \textit{ is } G^l$$

donde F_i^l y G_i^l son conjuntos borrosos o etiquetas lingüísticas, $x = (x_1, ..., x_n)^T \in U_1 \times \ldots \times U_n$ son variables de entrada, y donde $y \in V$ representa la salida del sistema borroso.

Definición 32 *Una* ***variable lingüística*** *es una variable que puede tomar como valor palabras del lenguaje natural (por ejemplo, "grande", "poco", etc.) o números. Estas palabras normalmente estarán relacionadas con conjuntos borrosos.*

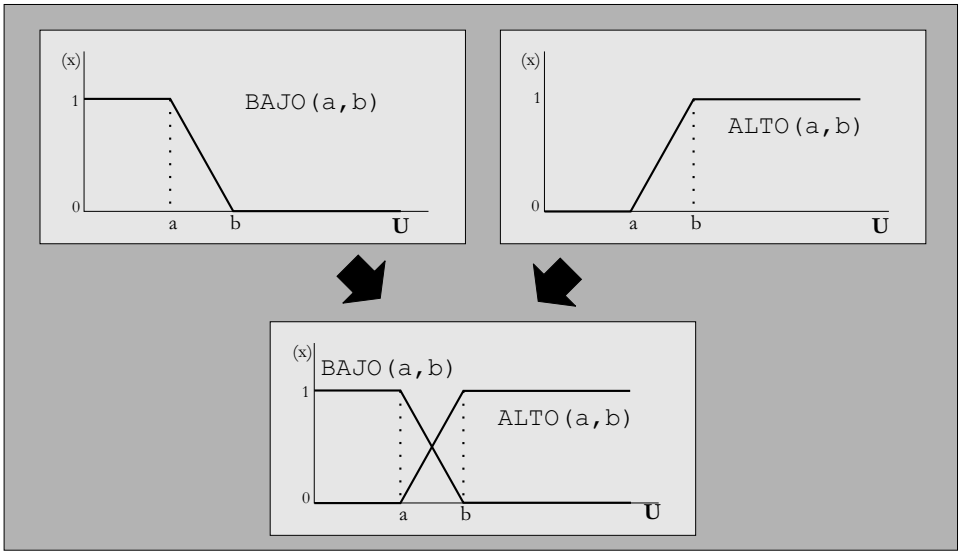

Figura 6.2: Ejemplo de funciones de pertenencia trapezoidales.

A partir de las definiciones anteriores se puede observar que los conjuntos borrosos son extensiones de los conjuntos clásicos bivaluados. En lugar de la aproximación de la lógica clásica, donde los objetos pertenecen o no a un conjunto, el concepto de conjunto borroso permite transiciones graduales de una "pertenencia" a una "no pertenencia", dando grados de parcialidad al propio concepto de pertenencia. Aquí radica el poder de la lógica borrosa para representar conceptos graduales.

Funciones de pertenencia

Un conjunto borroso permite describir el grado de pertenencia de un objeto a una determinada clase. Dicho grado de pertenencia viene descrito por una función de pertenencia $\mu_F : U \rightarrow [0, 1]$, siendo U el universo de discurso. Si el objeto $u \in U$ entonces $\mu_F(u)$ es su grado de pertenencia al conjunto borroso F.

Por simplicidad, las funciones de pertenencia más utilizadas son funciones triangulares, trapezoidales, gaussianas, o sigmoidales. En la Figura 6.2, se muestran como ejemplo las funciones de pertenencia trapezoidales de una variable lingüística para las dos etiquetas que posee: BAJO y ALTO, respectivamente. Nótese, que se obtiene un valor $\mu_{BAJO}(u) > 0$ y $\mu_{ALTO}(u) > 0$, $\forall u$ tal que $a < u < b$. En una lógica bivaluada clásica, la pertenencia a un grupo implicaría la no pertenencia al otro y viceversa.

La utilidad de un conjunto borroso para modelar un determinado concepto o etiqueta lingüística dependerá de lo apropiada que sea su función de pertenencia. Esto es de suma importancia en todas aquellas situaciones en las que se manejan términos del lenguaje natural (Garcia-Alegre, 1991).

Operaciones con conjuntos borrosos

Al igual que en la Teoría Clásica de Conjuntos, en la Teoría de Conjuntos Borrosos se definen con precisión las operaciones entre conjuntos (Garcia-Alegre, 1991):

- *Complementario* : $\mu_A(x) = 1 - \mu_A(x)$

- *Unión* : $\mu_{A \cup B}(x) = MAX(\mu_A(x), \mu_B(x))$

- *Intersección* : $\mu_{A \cap B}(x) = MIN(\mu_A(x), \mu_B(x))$

Las operaciones Complementario, Unión e Intersección suponen una generalización de sus correspondientes operadores en la Teoría Clásica de Conjuntos. Con ellas se obtienen los mismos resultados que con las operaciones clásicas cuando las funciones de pertenencia toman los valores 0 y 1.

Posteriormente, se pueden definir diferentes tipos de funciones con la condición de que verifiquen una serie de propiedades. Así, por ejemplo, para que una determinada función pueda representar el operador Unión deberá cumplir las siguientes condiciones:

- *Condiciones en los límites*:

  $u(0,0) = 0$

  $u(1,0) = u(0,1) = u(1,1) = 1$

- *Conmutatividad*: $u(a,b) = u(b,a)$

- *Asociatividad*: $u(u(a,b),c) = u(a,u(b,c))$

- *Monotonicidad*: SI $a \leqslant a^{'}$; $b \leqslant b^{'}$ ENTONCES $u(a,b) \leqslant u(a^{'},b^{'})$

Del mismo modo, para que una función i pueda considerarse Intersección de conjuntos borrosos deberá cumplir los tres últimos axiomas y, además, las siguientes condiciones:

- *Condiciones en los límites*:

  $i(1,1) = 1$

  $i(1,0) = i(1,0) = i(0,0) = 0$

Las funciones u (Unión) o i (Intersección) que verifican estos axiomas se definen en la literatura como Conorma Triangular (T-Conorma) o Norma Triangular (T-Norma), y se hayan relacionadas entre sí a través del operador complementario. Algunas de las más utilizadas son:

$$\begin{array}{cc} \underline{\text{T-Conormas}} & \underline{\text{T-Normas}} \\ \\ MAX(a,b) & MIN(a,b) \\ (a+b-ab) & ab \\ MIN(1,a+b) & MAX(0,a+b-1) \\ \begin{cases} a \ si \ b = 0 \\ b \ si \ a = 0 \\ 1 \ resto \end{cases} & \begin{cases} a \ si \ b = 1 \\ b \ si \ a = 1 \\ 0 \ resto \end{cases} \end{array} \tag{6.1}$$

La función clásica MAX es la más restrictiva de las funciones que pueden representar el operador Unión entre conjuntos borrosos. Con la función clásica MIN ocurre lo contrario:

$$\begin{array}{c} u(a,b) \geqslant MAX(a,b) \\ i(a,b) \leqslant MIN(a,b) \end{array} \tag{6.2}$$

La Teoría de Conjuntos Borrosos se denomina usualmente Teoría de la Posibilidad.

6.2.2. Sistema de inferencia borrosa

La Figura 6.1 resume la arquitectura básica de un controlador borroso como el que se empleará en esta tesis doctoral. Se compone de tres etapas de procesamiento: borrosificación de entradas, aplicación de las reglas de inferencia y una desborrosificación para producir la salida.

La definición de las variables lingüísticas supone un punto clave para tratar de acercar los conceptos propios del lenguaje natural al modelo y clasificarlos en base a unas etiquetas que, a su vez, son subconjuntos borrosos. Las funciones de pertenencia de los conjuntos borrosos se establecen también a partir del conocimiento aportado al sistema.

Las operaciones definidas entre conjuntos borrosos permiten combinar estos valores lingüísticos, expresando la base de conocimiento como afirmaciones condicionales borrosas. Así,

> *La (intersección/unión) de dos o más conjuntos se denota con el conectivo (AND/OR) y en su acepción clásica corresponde al valor (MIN/MAX) de las pertenencias.*

Durante la primera etapa del proceso de inferencia, toda variable de entrada al sistema es borrosificada de forma apropiada mediante las correspondientes funciones de pertenencia; por tanto, catalogada entre las diferentes etiquetas lingüísticas con valores $\mu_F(u) \in [0, 1]$.

El proceso de inferencia borrosa es el centro del sistema de razonamiento. Se modela mediante reglas borrosas de tipo *IF-THEN* que expresan una relación borrosa, asociando uno o más conjuntos borrosos de entrada (*antecedentes*) con un conjunto borroso de salida (*consecuente*). En un sistema de control borroso es habitual que estas reglas contengan múltiples entradas y una única salida (*Multiple-Input Single-Output* o *MISO*).

Para expresar la base de conocimiento se necesitan una serie de reglas R_i (*base de conocimiento*) que describan el sistema de la manera más precisa y donde se aporte el conocimiento heurístico o conocimiento de experto. Los antecedentes se combinan a través de operadores de Unión (conjunción) o Intersección (disyunción) que se implementan de muy diversas formas (ver expresiones 6.1).

Por tanto, mediante el mecanismo de inferencia borrosa se realiza la interpretación del conjunto de reglas *IF-THEN* disponibles en la base de conocimiento. Dependiendo de los operadores elegidos, se tendrán diferentes interpretaciones para unas mismas reglas. Las más

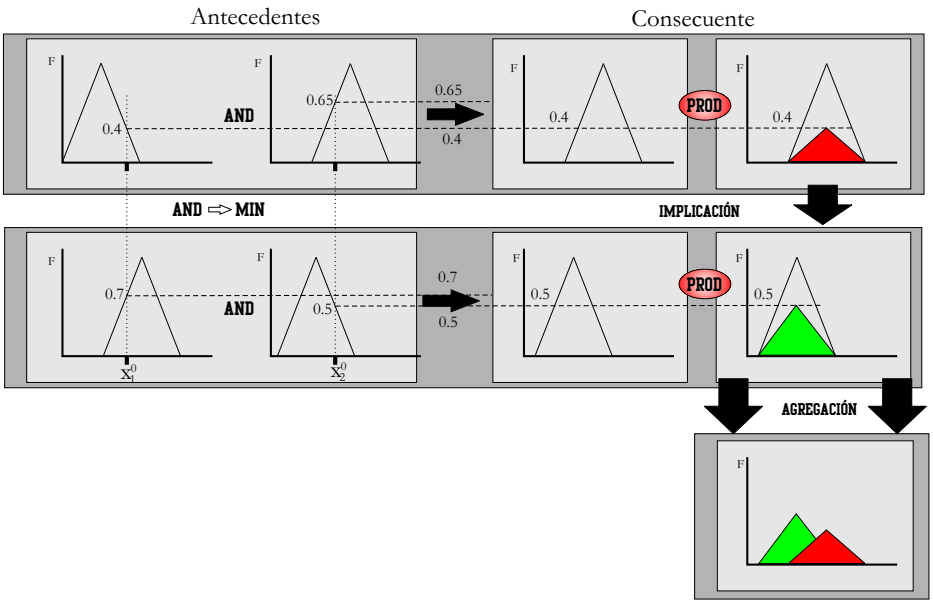

Figura 6.3: Regla de composición de Larsen (producto).

comunes, por su sencillez de análisis, son aquellas en las que tanto la implicación como el operador conjunción se reducen a la operación producto de funciones de pertenencia (regla de composición de Larsen (Larsen, 1980)), o al operador mínimo (regla de composición de Mamdani (Mamdani y Assilian, 1975)).

En la figura 6.3 se muestra una implicación a través del operador producto, que escala el conjunto de salida (en el ejemplo una función triangular) al valor resultante de la evaluación de la correspondiente regla. Por otra parte, la figura 6.4 muestra la implicación Mamdani sobre el mismo ejemplo anterior, donde se observa el comportamiento de truncado del conjunto borroso de salida debido al operador mínimo.

Tras la obtención de los consecuentes para cada regla *IF-THEN* (M conjuntos borrosos de salida), se obtiene un conjunto *agregado* final, tal y como muestran los ejemplos anteriores (tras la etapa de *agregación*). El agregado final es la entrada a la última etapa del controlador, la *desborrosificación*, necesaria para obtener un valor final no borroso. Para esta etapa también se pueden aplicar diferentes operadores, como el operador máximo o suma. En los ejemplos de las figuras se ha utilizado un operador máximo. La desborrosificación realiza una correspondencia entre un conjunto borroso en V (el de salida) con un punto concreto $y \in V$ (salida nítida o *crisp*). Para realizar esta desborrosificación se tienen, de nuevo, multitud de operadores, tales como:

- *Media Ponderada*. En aquellos casos en los que las funciones de pertenencia son monótonas.

$$MP: \ y = \frac{\sum_{i=1}^{n} w_i y_i}{\sum_{i=1}^{n} w_i} \tag{6.3}$$

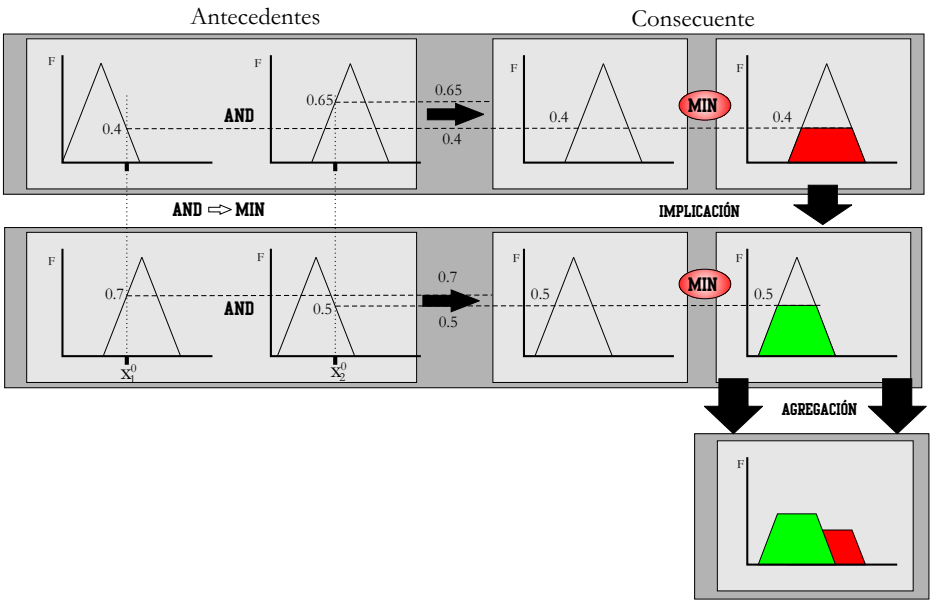

Figura 6.4: Regla de composición de Mamdani (mínimo).

donde w_i representa los grados de verdad de las condiciones de cada regla y los y_i son las salidas que se deducen de cada una de ellas.

- *Centro de masas*. Para funciones de pertenencia no monótonas, la salida se calcula como

$$CDM: \ y = \frac{\int B^*(y) y dy}{\int B^*(y) dy} \tag{6.4}$$

donde $B^* = w_1 B_1 \cup w_2 B_2 \ldots w_n B_n$, y B_i es la función de pertenencia asociada a la salida y, en la i-ésima regla.

- *Relación Funcional*. Si la salida y puede expresarse como una función lineal de las variables que aparecen en las condiciones, se obtienen funciones precisas en el consecuente.

$$y = a_0 + a_1 X_1 + a_2 X_2 + \ldots + a_n X_n \tag{6.5}$$

$$RF: \ y = \frac{\sum_{i=1}^{n} w_i y_i}{\sum_{i=1}^{n} w_i} = \frac{\sum_{i=1}^{n} w_i f_i(X_i)}{\sum_{i=1}^{n} w_i} \tag{6.6}$$

La Figura 6.5 muestra el ejemplo construido en anteriores figuras para una implicación de tipo Larsen y una desborrosificación por centro de masas. El valor y representa la salida final del controlador borroso.

En el caso de los controladores de Takagi-Sugeno, la fase de agregación se realiza simplemente agregando de forma ponderada los consecuentes funcionales correspondientes. Un ejemplo de este proceso se muestra en la Figura 6.6. La composición de las reglas borrosas puede asumirse en cualquiera de las formas descritas anteriormente.

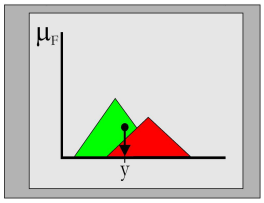

Figura 6.5: Ejemplo de desborrosificación para el controlador Mamdani de la Figura 6.4.

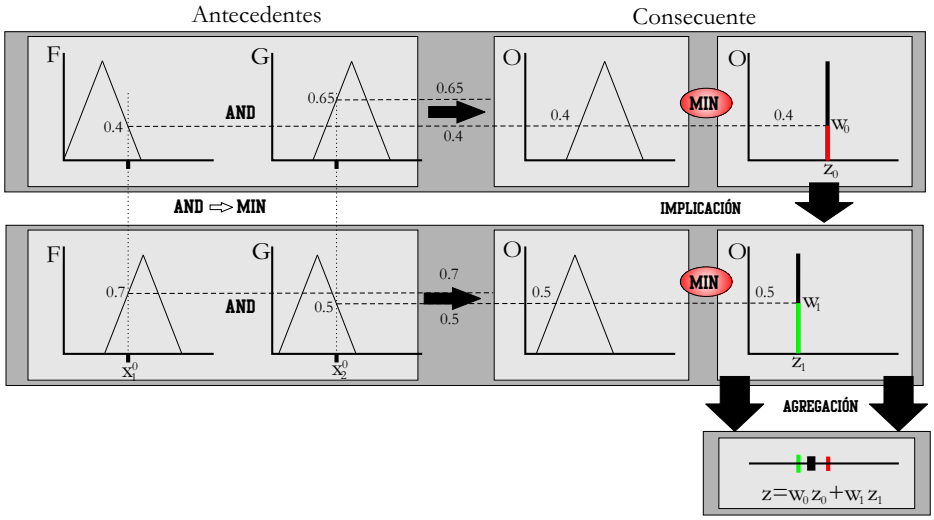

Figura 6.6: Controlador con consecuentes no borrosos (modelo de Takagi-Sugeno de orden cero).

6.3. Diseño e implementación del sistema borroso para la combinación de criterios

Como se ha visto, el diseño de un sistema borroso tiene muchos de grados de libertad. El objetivo final es la obtención de soluciones "suficientemente buenas" basadas en la combinación de conceptos que contienen aspectos relacionados con la subjetividad y la interpretación (Berkan y Trubatch, 1997). En primer lugar, es importante asegurarse de que el uso de conjuntos borrosos producirá una representación más realista que si se emplearan las mismas variables pero definidas de una forma nítida, o *crisp*. Como ya se ha indicado, la combinación de criterios heurísticos para la representación de páginas web hace pensar que esta situación se puede dar, y que una combinación borrosa podría capturar mejor la información asociada a los criterios considerados que una combinación lineal.

A partir de los criterios presentados en el capítulo 4, se definen una serie de conjuntos borrosos de entrada para cada uno de los cuatro criterios considerados: *frecuencia*, *título*, *enfatizado* y *posición*.

De este modo, la frecuencia total de un rasgo en el documento queda representada por

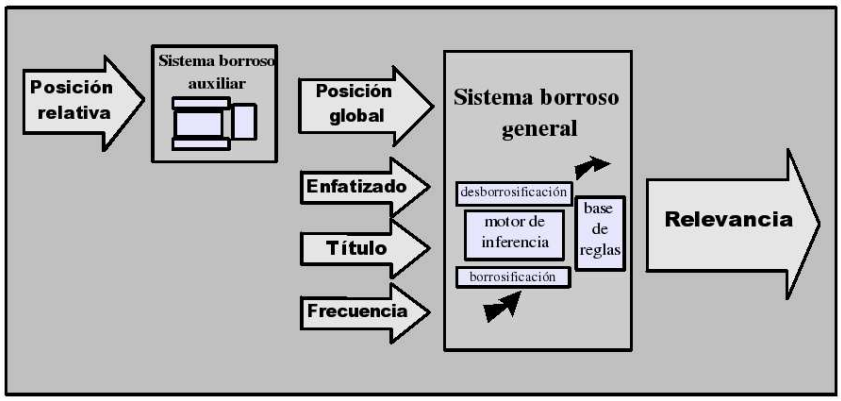

Figura 6.7: Arquitectura del sistema de representación basado en combinación borrosa de criterios.

medio de la variable lingüística **frecuencia**; la frecuencia de aparición en el título por medio de la variable **título** y la frecuencia en las diferentes partes enfatizadas del documento por medio de la variable **enfatizado**. Como a diferentes partes del documento se les asocia diferente importancia, y como cada rasgo puede tener más de una aparición en cada una de esas partes, es necesario buscar una forma de asociar la información relativa a la posición en un sentido general, o global, a cada rasgo dentro del documento. Este es el motivo por el que se define una posición global para cada rasgo, que queda representada por la variable lingüística ***posición global***.

En la figura 6.7 se presenta la arquitectura del sistema que representa la función de ponderación FCC (*Fuzzy Combination of Criteria*), donde puede verse el acoplamiento de los dos sistemas de inferencia borrosos que se van a considerar.

Para encontrar la posición global de un rasgo en un documento es necesario definir un sistema borroso auxiliar (el sistema borroso auxiliar que se muestra en la figura 6.7). Este sistema es el encargado de transformar la información correspondiente a las diferentes apariciones en cada parte del documento en la posición global de un rasgo dentro del documento. Para este sistema auxiliar se define una única variable de entrada *posición relativa* y otra de salida *posición global* que, a su vez, es una de las variables de entrada al sistema borroso general. Por último, el sistema general tiene definida una variable lingüística de salida llamada *relevancia* que representa el peso de un rasgo en el contenido de un documento.

En los siguientes puntos se definen en detalle ambos sistemas borrosos: sus variables lingüísticas y conjuntos borrosos asociados, los conjuntos de reglas que constituyen sus bases de conocimiento y las características de sus motores de inferencia. Dado que la implementación del sistema borroso auxiliar debe ser previa a la del sistema general, ya que su variable de salida supone una de las variables de entrada del sistema general, se sigue este orden para la explicación de la arquitectura del sistema de representación.

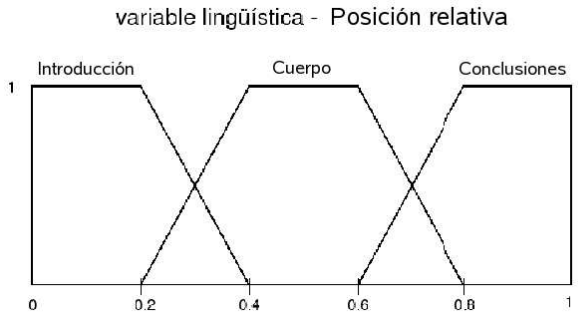

Figura 6.8: Variable lingüística *posición relativa*, entrada del sistema borroso auxiliar

6.3.1. Sistema borroso auxiliar (captura del criterio posición)

Gracias a este sistema auxiliar será posible combinar información relativa a la posición global de un rasgo en un documento con información de frecuencia en el título, enfatizado o en el documento en conjunto.

Como en todo sistema borroso, lo primero es definir las variables lingüísticas y sus conjuntos borrosos asociados, tanto para la entrada como la salida del sistema. Posteriormente, se presenta el conjunto de reglas IF-THEN que constituyen la base de conocimiento y que guardan las heurísticas que se quieren aplicar en el sistema. Por último, se presentan las características de su motor de inferencia borrosa.

Variables lingüísticas

La variable de entrada al sistema borroso auxiliar es la variable lingüística **posición relativa** (figura 6.8), para la que se definen tres conjuntos borrosos, correspondientes a las cuatro partes en las que se pretende dividir el documento.

Como el lenguaje HTML es un lenguaje interpretado, el número de líneas de una página web es diferente dependiendo de la configuración del navegador web instalado en el cliente. Por otro lado, no es muy común el uso de elementos
, correspondientes a los saltos de línea, en los documentos HTML. Por ello, para crear 4 partes en el documento, se divide el número total de rasgos en la página, N_j, por 4. De este modo, la primera de las cuatro partes se hace corresponder con el conjunto borroso "*introducción*", la segunda y tercera parte se asocian con el conjunto "*cuerpo*", y la última parte se corresponde con el conjunto borroso "*conclusión*". Las funciones de pertenencia de cada uno de estos conjuntos se muestran en la figura 6.8.

Aún sabiendo que las páginas web no tienen una estructura del tipo introducción-cuerpo-conclusión, los conjuntos borrosos toman estos nombres para facilitar la aplicación de heurísticas en la definición de la base de conocimiento. Resulta más fácil pensar en términos de introducción, cuerpo y conclusión a la hora de suponer, por ejemplo, que al inicio (el "planteamiento" o las "premisas" de un texto expositivo) o al final de un documento ("solución" o "conclusión") se

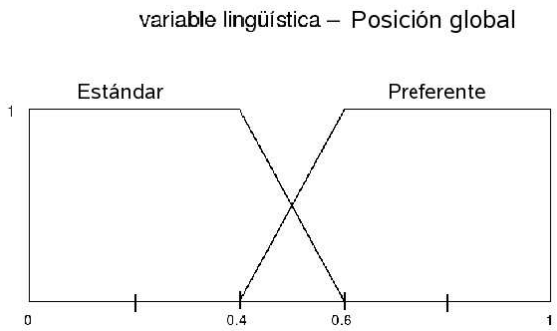

Figura 6.9: Variable lingüística *posición global*, salida del sistema auxiliar y variable de entrada en el sistema general

puede encontrar una información más relevante que en la parte central del documento.

Como la contribución de un rasgo al criterio *posición global* se computa considerando todas las apariciones del rasgo en el documento, se define la siguiente función de captura de la posición de cada ocurrencia de un rasgo en el documento en el sistema borroso auxiliar:

$$p_i = \frac{p(\vec{t_i}, \vec{d_j}, o)}{N_j} \tag{6.7}$$

donde $p(\vec{t_i}, \vec{d_j}, o)$ es la posición de la o-ésima ocurrencia del rasgo t_i en el documento d_j, y N_j es el número total de rasgos presentes en d_j. De este modo, $p(\vec{t_i}, \vec{d_j}, o)$ representa la entrada a la variable lingüística *posición relativa* en este sistema borroso auxiliar.

La salida de este sistema auxiliar se define como la variable lingüística **posición global**, que sirve a su vez de entrada al sistema borroso general. Para esta variable se definen dos conjuntos borrosos: "*preferente*" y "*estándar*" (figura 6.9). El conjunto borroso *preferente* representa las partes destacadas del documento (primera y última), aquellas en las que se espera encontrar la información más relevante, mientras que el *estándar* representará las partes restantes (segunda y tercera). En la figura 6.9 pueden verse los detalles de las funciones de pertenencia asociadas a cada conjunto borroso.

Se han tomado funciones de tipo trapezoidal para la definición de los conjuntos borrosos por la simplicidad que supone la aplicación de estas funciones de tipo lineal en el proceso de borrosificación.

Base de conocimiento

La base de conocimiento de este sistema auxiliar está compuesta por el conjunto de reglas que se muestra en la tabla 6.1. La idea que subyace bajo este conjunto de reglas es que las partes inicial y final de un documento contienen, previsiblemente, información más relevante que la parte central del documento, siempre que se quiera extraer una idea general sobre el contenido del mismo.

	posición relativa		**posición global**
IF	introducción	THEN	preferente
IF	cuerpo	THEN	estándar
IF	conclusión	THEN	preferente

Tabla 6.1: Conjunto de reglas del sistema borroso auxiliar

Sistema de inferencia

En este sistema borroso auxiliar, los operadores AND y OR se implementan como el operador mínimo y máximo respectivamente.

En el mecanismo de inferencia borrosa se emplea, por su sencillez de análisis, la regla de composición de Larsen (figura 6.3), donde tanto la implicación como el operador conjunción se reducen a la operación producto. De este modo, todos los antecedentes influyen en la obtención del conjunto agregado final.

Respecto a la fase de desborrosificación, ésta se realiza por medio del centro de masas (ecuación 6.4). La razón de esta elección es que con este método, un cambio pequeño en la entrada no supone un cambio grande en la salida, además de que ésta no resulta ambigua aunque dos o más consecuentes se cumplan en igual medida. Por otro lado, permite salidas "plausibles", es decir, que las salidas se encuentran cerca del consecuente que más se haya dado.

6.3.2. Sistema borroso general (cálculo de la relevancia)

El sistema general es el encargado de obtener la relevancia de un rasgo en el contenido de un documento mediante la combinación de los criterios frecuencia, enfatizado, título y posición global, obtenida tras la aplicación del sistema borroso auxiliar.

A continuación, como en el caso anterior, se presentan las variables lingüísticas de entrada y salida al sistema, sus correspondientes conjuntos borrosos, las reglas que constituyen la base de conocimiento del sistema y las características del motor de inferencia borroso.

Variables lingüísticas

Para la variable lingüística correspondiente al criterio **frecuencia** se definen tres conjuntos borrosos correspondientes a los conceptos de frecuencia "*alta*", "*baja*" y "*media*". En la figura 6.10 se muestran los detalles de estos conjuntos borrosos.

El hecho de que la frecuencia de un rasgo dentro de un documento pueda considerarse como *alta*, *baja* o *media* dependerá del resto de rasgos presentes en el documento. Por este motivo, se define una función de captura que normaliza la frecuencia absoluta a la mayor frecuencia presente en el documento. De este modo:

$$f_{text(ij)} = \frac{f_{ij}}{N_{max-j}} \tag{6.8}$$

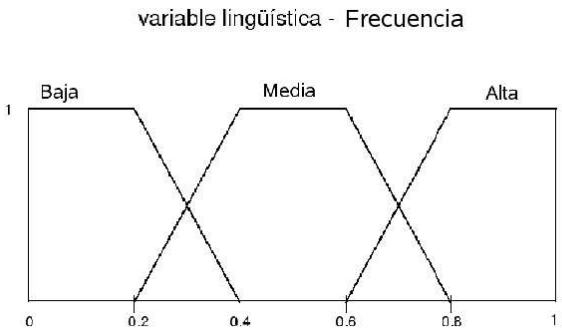

Figura 6.10: Variable lingüística *frecuencia*, entrada al sistema borroso general.

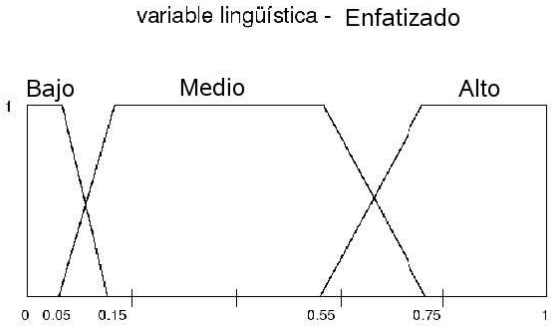

Figura 6.11: Variable lingüística *enfatizado*, entrada al sistema borroso general.

donde $f_{text(ij)}$ es la frecuencia del rasgo t_i en d_j, y N_{max-j} es la frecuencia máxima de un término en el documento d_j.

Para el criterio enfatizado se define la variable lingüística ***enfatizado***, compuesta por los conjuntos borrosos *bajo*, *medio* y *alto*. En la figura 6.11 se muestran los detalles de estos conjuntos. Al igual que con la frecuencia, la función de captura para el enfatizado debe normalizarse. De entre todos los rasgos presentes en el documento, se toma la frecuencia de enfatizado del rasgo más veces destacado y se define la función:

$$f_{emph(ij)} = \frac{e_{ij}}{N_{max-enf(j)}} \tag{6.9}$$

donde $f_{emph(ij)}$ es la frecuencia de enfatizado del rasgo t_i en d_j, y $N_{max-enf(j)}$ es la frecuencia máxima de enfatizado en el documento.

Se puede observar una falta de simetría en la definición de los conjuntos borrosos asociados a esta variable lingüística (figura 6.11). La razón es que en muchas páginas web, sobre todo en las comerciales, se pueden encontrar términos con una frecuencia de enfatizado excesivamente alta. Además, la página puede contener otros rasgos también destacados por el autor, pero con una frecuencia mucho menor. Con este diseño asimétrico se pretende evitar que un único rasgo con una frecuencia de enfatizado muy alta haga que el resto caigan en la etiqueta "bajo" y diferentes

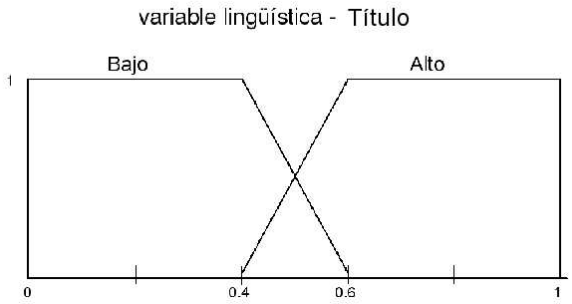

Figura 6.12: Variable lingüística *título*, entrada al sistema borroso general.

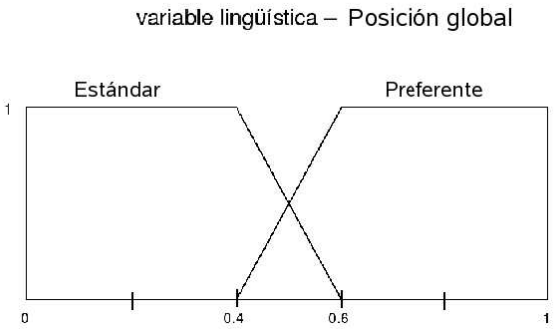

Figura 6.13: Variable lingüística *posición global*, entrada al sistema borroso general.

frecuencias de enfatizado no puedan ser matizadas.

El criterio título se modela con la variable lingüística **título**. En este caso, los conjuntos borrosos que se definen son sólo dos: "bajo" y "alto"; y se muestran en la figura 6.12. La razón es que el número de palabras del título de una página web suele ser pequeño y, además, los rasgos presentes en él no suelen aparecer repetidos. En el estudio estadístico realizado en el capítulo 5 el contenido de este elemento <title> no fue nunca mayor de 11 rasgos para la colección de referencia, valor que coincide con un estudio de J. M Pierre (Pierre, 2000). En este caso, por tanto, basta con definir dos conjuntos borrosos. La función de captura para el este criterio será:

$$f_{tit(ij)} = \frac{t_{ij}}{N_{max-tit(j)}} \tag{6.10}$$

donde t_{ij} es la frecuencia en el título de t_i en d_j, y $N_{max-tit(j)}$ es la máxima frecuencia en el título de la página web.

Además, la variable de salida del sistema borroso auxiliar sirve ahora de entrada al sistema general. En la figura 6.13 se muestran los conjuntos borrosos asociados a esta variable lingüística **posición global**, idéntica a la mostrada en la figura 6.9.

Una vez definidas las variables de entrada, se necesita una variable lingüísticas de salida, **relevancia**, (figura 6.14), que representará el peso de un rasgo en el contenido de texto de una

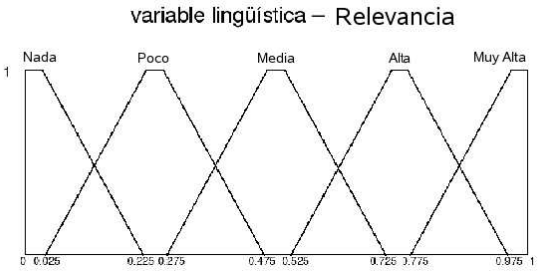

Figura 6.14: Variable lingüística *relevancia*, salida del sistema borroso general.

página web. En esta variable se definen 5 conjuntos borrosos: "nada relevante" (*Nada*), "poco relevante" (*Poco*), "relevancia media" (*Media*), "Altamente relevante" (*Alta*) y "muy relevante" (*Muy Alta*).

En este caso, los 5 conjuntos borrosos se distribuyen homogéneamente entre los valores de relevancia $[0, 1]$ y sirven para definir la base de conocimiento.

Base de conocimiento

El conjunto de reglas o base de conocimiento del sistema global guarda todo el conocimiento heurístico que se quiere aplicar en el sistema y se muestra en la tabla 6.2. El éxito de un sistema borroso está, en gran medida, condicionado por su conjunto de reglas y el conocimiento heurístico que se pueda guardar en él. Para entender esta base de conocimiento es importante considerar los siguientes aspectos:

- El contenido del título no siempre debe ser considerado como un resumen del contenido de la página.
  - En muchos casos, puede ser producto de un proceso de generación automática.
  - Por el contrario, si un rasgo, además de estar presente en el título, aparece altamente enfatizado o resulta muy frecuente, puede que sea muy relevante respecto del contenido del documento.
- La importancia de un rasgo respecto a un criterio es siempre relativa.
  - El hecho de que un rasgo aparezca enfatizado no siempre significa que haya sido destacado por el autor de la página, ya que podrían estarlo la mayoría de los rasgos presentes en el texto del documento.
  - El hecho de que un rasgo no aparezca enfatizado pudiera ser debido a que ninguno de los rasgos presentes en la página lo estuviera y no a que no sea relevante respecto del contenido del documento.

	Título		**Frecuencia**		**Enfatizado**		**Posición**		**Relevancia**
IF	Alto	AND	Alta	AND	Alto			THEN	Muy Alta
IF	Alto	AND	Media	AND	Alto			THEN	Muy Alta
IF	Alto	AND	Media	AND	Medio			THEN	Alta
IF	Alto	AND	Alta	AND	Medio			THEN	Muy Alta
IF	Alto	AND	Baja	AND	Bajo	AND	Preferente	THEN	Media
IF	Alto	AND	Baja	AND	Bajo	AND	Estándar	THEN	Poca
IF	Bajo	AND	Baja	AND	Bajo			THEN	Nada
IF	Bajo	AND	Alta	AND	Alto	AND	Preferente	THEN	Muy Alta
IF	Bajo	AND	Alta	AND	Alto	AND	Estándar	THEN	Alta
IF	Alto	AND	Baja	AND	Medio	AND	Preferente	THEN	Alta
IF	Alto	AND	Baja	AND	Medio	AND	Estándar	THEN	Media
IF	Alto	AND	Baja	AND	Alto	AND	Preferente	THEN	Muy Alta
IF	Alto	AND	Baja	AND	Alto	AND	Estándar	THEN	Alta
IF	Alto	AND	Alta	AND	Bajo	AND	Preferente	THEN	Muy Alta
IF	Alto	AND	Alta	AND	Bajo	AND	Estándar	THEN	Alta
IF	Bajo	AND	Baja	AND	Medio	AND	Preferente	THEN	Media
IF	Bajo	AND	Baja	AND	Medio	AND	Estándar	THEN	Poca
IF	Bajo	AND	Baja	AND	Alto	AND	Preferente	THEN	Alta
IF	Bajo	AND	Baja	AND	Alto	AND	Estándar	THEN	Media
IF	Bajo	AND	Media	AND	Bajo	AND	Preferente	THEN	Poca
IF	Bajo	AND	Media	AND	Bajo	AND	Estándar	THEN	Nada
IF	Bajo	AND	Media	AND	Medio	AND	Preferente	THEN	Media
IF	Bajo	AND	Media	AND	Medio	AND	Estándar	THEN	Poca
IF	Bajo	AND	Media	AND	Alto	AND	Preferente	THEN	Muy Alta
IF	Bajo	AND	Media	AND	Alto	AND	Estándar	THEN	Alta
IF	Bajo	AND	Alta	AND	Bajo	AND	Preferente	THEN	Media
IF	Bajo	AND	Alta	AND	Bajo	AND	Estándar	THEN	Poca
IF	Bajo	AND	Alta	AND	Medio	AND	Preferente	THEN	Alta
IF	Bajo	AND	Alta	AND	Medio	AND	Estándar	THEN	Media
IF	Alto	AND	Media	AND	Bajo	AND	Preferente	THEN	Media
IF	Alto	AND	Media	AND	Bajo	AND	Estándar	THEN	Poca

Tabla 6.2: Conjunto de reglas del sistema borroso global

- El criterio posición tiene mayor peso en páginas largas que en documentos cortos. Si bien una página web no tiene por qué seguir una estructura de introducción-cuerpo-conclusión, es posible que en documentos largos el autor exponga sus ideas con un esquema similar a éste.

- La frecuencia es un criterio que siempre debe tenerse en cuenta a la hora de estimar la relevancia de un rasgo:

  - Un rasgo poco frecuente puede ser relevante si aparece además en el título, en elementos de enfatizado o en posiciones destacadas dentro de la página.

  - Un rasgo poco frecuente que no aparece en el resto de los criterios puede resultar poco relevante.

  - Aún habiéndose aplicado un proceso de eliminación de *stop-words* que elimina la mayoría de rasgos con información semántica poco relevante, un rasgo muy frecuente

puede seguir siendo una palabra de uso común y es posible que su significado no ayude a revelar el contenido de la página.

6.3.3. Motor de inferencia borroso

En este sistema borroso general, el operador Unión se implementa como el valor mínimo y la Intersección como el máximo. La inferencia borrosa se realiza por medio de la regla de composición de Larsen (figura 6.3), donde tanto la implicación como el operador conjunción se realizan con el producto. La fase de desborrosificación se realiza, como en el caso del sistema auxiliar, por medio del centro de masas (ecuación 6.4). Las razones de estas elecciones son las mismas que se expusieron cuando se describió el motor de inferencia del sistema borroso auxiliar.

6.4. Conclusiones

Partiendo del modelo teórico general presentado en el capítulo 4, donde se fijaron las bases para la representación de páginas web autocontenidas, en este capítulo se ha propuesto la implementación de una posible representación basada en un sistema de reglas borrosas. Se ha presentado un sistema de reglas borrosas que supone una función de ponderación dentro de un modelo de representación de documentos.

Una vez definido el conjunto de criterios que se quiere combinar, el primer paso es el establecimiento de las variables lingüísticas y sus conjuntos borrosos asociados a cada una de ellas. Estos conjuntos deberán estar parametrizados con información presente en la página web, asegurando así que la representación final resulte autocontenida. En esta tesis doctoral se consideran los criterios seleccionados en el capítulo 4, aunque podrían establecerse otros criterios sin que por ello tuviera que modificarse el modelo.

Con las funciones de captura para cada criterio se obtienen valores numéricos que sirven de entrada a las variables de entrada del sistema borroso. Por medio de estas funciones, a cada rasgo del documento se le podrá asignar el grado de pertenencia a un conjunto de borroso. El conjunto de reglas del sistema, o base de conocimiento, almacena el conocimiento heurístico que se quiere aplicar en la combinación de criterios.

Por último, se fijan los detalles de implementación de los procesos de borrosificación y desborrosificación, así como los operadores empleados en la implementación del motor de inferencia borroso. En este trabajo de tesis se ha tomado una configuración estándar y se ha puesto el énfasis en la definición de las variables lingüísticas, las funciones de captura y el establecimiento de la base de conocimiento.

Capítulo 7

Diseño de la experimentación

"No basta saber, se debe también aplicar;
no es suficiente querer, se debe también hacer."
Johann Wolfgang von Goethe

En este capítulo se detallan aquellos aspectos relacionados con el diseño experimental que serán comunes en los experimentos en clasificación automática y en clustering de páginas web. En primer lugar, se muestran las funciones de proyección F que serán utilizadas, así como las colecciones y subcolecciones con las que se evaluará la calidad de las diferentes representaciones. Por último, se exponen los dos métodos de reducción de rasgos empleados en la reducción de la dimensión de los vocabularios; de este modo, es posible evaluar una función de ponderación dentro de un modelo de representación en función de la dimensión del vocabulario con el que se genere la representación.

7.1. Introducción

Una vez fijadas las características generales de un modelo de representación autocontenida de páginas web basado en combinaciones heurísticas de criterios (capítulo 4) y detalladas las implementaciones de dos representaciones dentro de dicho modelo, ACC (capítulo 5) y FCC (capítulo 6), la calidad de las representaciones propuestas se evaluará comparándolas con otras representaciones, por medio de un algoritmo de clasificación *Naïve Bayes* y un algoritmo de *clustering* de partición perteneciente a la librería CLUTO (Karypis, 2002). En ambos casos, se trata de ejemplos típicos de resolución de problemas de TC y DC.

Todas las representaciones consideradas en esta experimentación se pueden enmarcar en un modelo de representación definido por $\langle X, \mathbb{B}, \mu, F \rangle$, donde todas tienen un mismo álgebra $\mathbb{B}$, definido para el VSM, y una misma función de distancia μ. Por tanto, lo que diferenciará un modelo de otro será el vocabulario X y la función de proyección F.

En este capítulo se presentan los detalles de la experimentación llevada a cabo en esta tesis doctoral. En primer lugar, se establece el conjunto de funciones de proyección F que van a ser utilizadas en la evaluación de las representaciones. A continuación, se describen las colecciones sobre las que se va a realizar la experimentación, distinguiendo entre aquellas empleadas en

tareas de TC (con una parte de aprendizaje y otra de evaluación) y en problemas de DC que no requieren fase de aprendizaje. Para la creación de las diferentes subcolecciones se realiza una fase de preproceso, además de aplicar métodos de reducción de rasgos para disminuir la dimensión de los vocabularios en cada una de las representaciones evaluadas.

7.2. Representaciones evaluadas

Para evaluar la calidad de las dos representaciones propuestas, a través de contraste con otros métodos más clásicos, se consideraron modelos que empleaban funciones de proyección clásicas que no utilizan la información del etiquetado HTML, aplicadas directamente al contenido textual presente en el documento. De entre estas funciones se seleccionaron las siguientes:

- **TF**: función de ponderación que asigna como valor de relevancia la frecuencia del rasgo en el documento (véase ecuación 2.15).
- **BinIDF**: función binaria con una corrección por la frecuencia inversa de documento (véase ecuación 2.22).
- **TF-IDF**: función TF corregida con la frecuencia inversa de documento (véase 2.24).
- **WIDF**: función que asigna la frecuencia ponderada de un rasgo en un documento (véase ecuación 2.26).

  Las funciones BinIDF, TF-IDF y WIDF son de carácter global, por lo que con ellas no podrían generarse representaciones autocontenidas. Sin embargo, la información de colección que necesitan –relación entre rasgos y documentos– se puede generar sin la necesidad de un análisis de la estructura de grafo de hipertexto que forma la propia colección, algo que resultaría mucho más costoso computacionalmente que la creación de un fichero invertido.

Junto a las funciones anteriores se seleccionaron las representaciones autocontenidas más comunes que, como en el caso de ACC y FCC, emplean la información del etiquetado HTML; es decir:

- **Title**: En esta representación, la función de ponderación empleada es una función TF aplicada sobre el contenido del elemento <TITLE> de una página web, siendo, por tanto, una función de carácter local. Una característica fundamental de esta representación, como se vió en el capítulo 3, es que los contenidos de esta etiqueta, sólo visibles en la barra de título de la propia página web, suelen ser textos de tamaño muy corto, no superando en la mayor parte de los documentos las 10 palabras.
- **Molinari**: representación presentada en (Molinari et al., 2003) y descrita en detalle en la sección 3.2.1. En este caso se tomó, de entre las dos funciones de captura de

criterios presentadas en dicho trabajo, la única que tenía carácter local y permitía obtener representaciones autocontenidas (ecuación 3.3).

Por último, las dos representaciones propuestas en esta tesis:

- **ACC**: *Analytical Combination of Criteria*, desarrollada en detalle en el capítulo 5.
- **FCC**: *Fuzzy Combination of Criteria*, desarrollada en el capítulo 6.

Es importante destacar que las funciones de ponderación TF, Title, Molinari, ACC y FCC son independientes de la información de la colección; es decir, generan representaciones autocontenidas. Sin embargo, las representaciones BinIDF, TF-IDF y WIDF necesitan la información de la colección para representar cada página.

7.3. Descripción de las colecciones de referencia

Con el fin de evaluar el resultado de la clasificación y el *clustering* de páginas web con las diferentes funciones de ponderación, se seleccionaron dos colecciones: *BankSearch DataSet* (Sinka, 2002) y *WebKB* [1], a partir de las cuales se crearon los diferentes vocabularios. Ambas suponen colecciones de referencia dentro de los campos de la clasificación y el *clustering* de páginas web.

7.3.1. Colección BankSearch DataSet

Esta colección está formada por 11.000 páginas web escritas en inglés y preclasificadas manualmente en 11 categorías del mismo tamaño y organizadas en niveles jerárquicos. Fue recopilada y clasificada con el propósito de usarse como colección de referencia para evaluar el *clustering* de páginas web. Sin embargo, también es posible utilizarla en tareas de TC, ya que se trata de un conjunto de documentos HTML preclasificados manualmente. Las categorías en esta colección se organizan del siguiente modo:

- Las tres primeras categorías pertenecen al tema general "Bancos & Finanzas" (*Banking & Finance*): subcolección **ABC**.
  - Bancos Comerciales (*Commercial Banks*, **A**)
  - Sociedades de Crédito Hipotecario (*Building Societies*, **B**)
  - Aseguradoras (*Insurance Agencies*, **C**)
- Las tres siguientes se corresponden con la temática "Lenguajes de Programación" (*Programming Language*): subcolección **DEF**.

[1] http://www.cs.cmu.edu/ webkb/

  - Java (**D**)
  - C/C++ (**E**)
  - Visual Basic (**F**)

- Las dos categorías siguientes están formadas por páginas relativas al tema "Ciencia" (*Science*): subcolección **GH**.
  - Astronomía (*Astronomy*, **G**)
  - Biología (*Biology*, **H**)

- Por último, las dos últimas corresponden a "Deportes": subcolección **IJ**.
  - Fútbol (*Soccer*, **I**)
  - Deportes de Motor (*Motor Sport*, **J**)

Además de estas 10 categorías organizadas en una jerarquía de dos niveles, existe otra categoría (**X**) que constituye un superconjunto de la clase **IJ** y que supone añadir un nivel superior en la jerarquía. Esta categoría **X** está formada por páginas relativas a diferentes "Deportes" no incluidos ni en las categorías **I** ni **J**. Para los experimentos se consideraron únicamente las 10 categorías principales, identificadas con las 10 primeras letras del abecedario.

En este punto es importante destacar el carácter heterogéneo de las páginas web que forman esta colección, logrado gracias a que los documentos fueron descargados desde portales y páginas web pertenecientes a ámbitos bien distintos. Se seleccionaron inicialmente cuatro grandes temas de naturaleza bien diferente: "Bancos y Finanzas", "Lenguajes de Programación", "Ciencia" y ‘ ‘Deportes"; posteriormente, se seleccionaron 2 o 3 categorías para cada uno de ellos, hasta que se obtuvo el número total de 10 clases. Con esta forma de selección de las categorías se prentendía permitir al usuario un rango amplio a la hora de realizar el diseño experimental en el estudio de problemas de clasificación y *clustering* de páginas web (Sinka, 2002). Con esta colección se pueden realizar experimentos con diferente nivel de dificultad, pudiéndose tomar categorías "semánticamente" cercanas y lejanas, así como en dos niveles jerárquicos.

A partir de esta colección *BankSearch* se seleccionaron diferentes subcolecciones, formadas por los documentos pertenecientes a un subconjunto de estas categorías, con el fin de evaluar diferentes representaciones, en tareas de TC y DC, con diferente número de clases y grupos. También se consideraron ambos niveles jerárquicos, de modo que el hecho de considerar el nivel superior implica contar con subcolecciones de mayor tamaño.

Por último, para cada subcolección hay que distinguir dos tratamientos diferentes para el caso de clasificación automática y *clustering*, dado que en un caso se utiliza aprendizaje supervisado y en el otro no.

Subcolecciones generadas para Clasificación Automática

En el caso de la clasificación automática bayesiana, al tratarse de una tarea de aprendizaje supervisado, es necesario contar con un subconjunto de documentos para entrenar el sistema, "conjunto de estimación", y otro para probar la calidad del sistema de clasificación, "conjunto de validación".

El cálculo del error del clasificador y la verificación de los resultados se ha realizado por medio de una estimación con "conjunto de prueba" o "test". Sea C un corpus de N páginas web, se divide en dos conjuntos independientes T^l y T^t, de forma que T^l es el conjunto de estimación y T^t constituye el conjunto de validación. T^l se emplea únicamente para construir el clasificador y T^t se utiliza sólo para estimar el error del clasificador. Estos conjuntos conviene que cumplan las siguientes condiciones:

1. $T^l \cup T^t = T$ y $T^l \cap T^t = \emptyset$

2. $|T^t| = \frac{1}{3}|T|$ y $|T^l| = \frac{2}{3}|T|$

A partir de T^l se entrena y construye un clasificador d. A continuación, se clasifican todas las páginas de T^t utilizando d. La evaluación del sistema se realiza en base a estas $|T^t|$ clasificaciones; una vez realizadas, el estimador de error R^{ts} del clasificador d es:

$$R^{ts}(d) = \frac{1}{|T^t|} \sum_{(x_i,c_i)\in T^t} \Delta((x_i, c_i)) \tag{7.1}$$

donde

$$\Delta((X_i, c_i)) = \begin{cases} 1, & \text{si } d(x_i) \neq c_i \text{ (error)} \\ 0, & \text{si } d(x_i) = c_i \text{ (acierto)} \end{cases} \tag{7.2}$$

siendo x_i un elemento a clasificar y c_i una categoría posible.

Con este método se reduce el tamaño efectivo del conjunto de aprendizaje a 2/3 de su tamaño inicial, lo que hace que en conjuntos poco numerosos la consistencia de un clasificador pueda quedar comprometida; es decir, la estimación de la calidad del clasificador puede ser mala.

En esta experimentación, para la evaluación en la fase de aprendizaje se toma un 70 % del conjunto de documentos de cada una de las subcolecciones consideradas y para la de prueba el 30 % restante. Los documentos fueron seleccionados siempre de forma aleatoria. Para nuestro estudio, además, este cálculo se realizó tres veces con subconjuntos diferentes para el aprendizaje y la validación en cada caso; finalmente, la estimación final se alcanza realizando la media aritmética de las tres estimaciones.

Los vocabularios se crean a partir del subconjunto de entrenamiento que se seleccione. Posteriormente, la representación de las páginas contenidas en el subconjunto de test deberá realizarse con ese mismo vocabulario. De este modo, si un rasgo aparece únicamente en un documento perteneciente al subconjunto de test y no hubiera aparecido en ninguno de los

documentos del subconjunto de entrenamiento, no formaría parte del vocabulario, y no formaría parte tampoco de la representación.

El conjunto de subcolecciones evaluadas para la experimentación en TC se clasifica en:

1. Clasificación binaria, consistente en determinar la clase de un documento entre dos posibles.

   a) **GH_700**: colección formada por 700 páginas de cada una de las clases G y H, con las que se generan los diferentes vocabularios y se realiza la fase de aprendizaje. Se toman las otras 300 páginas por clase para llevar a cabo la fase de validación. Con esta subcolección se pretende realizar una **clasificación binaria** en el **nivel más bajo de la jerarquía** y entre **clases cercanas semánticamente**, ya que pertenecen a la misma categoría "Ciencia".

   b) **G&J_700** : colección formada por 700 páginas de cada una de las clases G y J, con las que se generan los diferentes vocabularios y se realiza la fase de aprendizaje. Las otras 300 páginas por clase se emplean en la fase de validación. Con esta subcolección se pretende realizar una **clasificación binaria** en el **nivel más bajo de la jerarquía** y entre **clases lejanas semánticamente**, ya que pertenecen a las categorías "Ciencia" y "Deportes".

   c) **GJ_700**: colección formada por 700 páginas de cada una de las clases G, H, I y J, con las que se generan los diferentes vocabularios y se realiza la fase de aprendizaje. Las otras 300 páginas por clase se emplean en la fase de validación. Con esta subcolección se pretende realizar una **clasificación binaria** en el **nivel más alto de la jerarquía** y entre **clases lejanas semánticamente**, ya que pertenecen a dos categorías diferentes.

   d) **ABC&DEF_700**: colección formada por 700 páginas de cada una de las clases A, B, C, D, E y F, con las que se generan los diferentes vocabularios y se realiza la fase de aprendizaje. Las otras 300 páginas por clase se emplean en la fase de validación. Con esta subcolección se pretende, como en el caso anterior, realizar una **clasificación binaria** en el **nivel más alto de la jerarquía** y entre **clases lejanas semánticamente**: "Bancos & Finanzas" y "Lenguajes de programación".

2. Clasificación en 3 clases:

   a) **ABC_700**: colección formada por 700 páginas de cada una de las clases A, B y C, con las que se generan los diferentes vocabularios y se realiza la fase de aprendizaje. Las otras 300 páginas por clase se emplean en la fase de validación. Con esta subcolección se pretendía realizar una **clasificación en 3 clases** en el **nivel más bajo de la jerarquía** y entre **clases cercanas semánticamente**, ya que todas pertenecen a la categoría "Bancos & Finanzas".

3. Clasificación en 6 clases:

   a) **ABC&DEF_700**: subcolección con la que se puede realizar una **clasificación en 6 clases** en el **nivel más bajo de la jerarquía** y distinguiendo entre **clases cercanas y lejanas semánticamente**, ya que pertenecen 3 a 3 a las categorías "Bancos & Finanzas" y "Lenguajes de programación".

4. Clasificación en 10 clases

   a) **AJ_700**: colección completa formada por 700 páginas de cada una de las 10 clases que constituyen el *benchmark*. Con esta colección se pretendía realizar una **clasificación en 10 clases** en el **nivel más bajo de la jerarquía** y entre **clases cercanas y lejanas semánticamente**.

Subcolecciones generadas para *Clustering* de documentos

En el caso del *clustering* de documentos, al tratarse de una tarea de aprendizaje no supervisado, los vocabularios se generan a partir del conjunto total de documentos de la colección.

Las subcolecciones evaluadas para la experimentación en DC, organizadas por el número de *clusters*, son:

1. *Clustering* binario:

   a) **GH_1000**: colección formada por 1000 páginas de cada una de las clases G y H, con las que se generan los diferentes vocabularios. Con esta subcolección se realiza un **clustering binario** en el **nivel más bajo de la jerarquía** y entre **clases cercanas semánticamente**, pertenecientes a la clase "Ciencia".

   b) **G&J_1000** : colección formada por 1000 páginas de cada una de las clases G y J, con las que se generan los diferentes vocabularios. Con esta subcolección se realiza un **clustering binario** en el **nivel más bajo de la jerarquía** y entre **clases lejanas semánticamente**, pertenecientes a "Ciencia" y "Deportes".

   c) **GJ_1000**: colección formada por 1000 páginas de cada una de las cuatro clases G, H, I y J, con las que se generan los diferentes vocabularios. Con esta subcolección se pretendía realizar un **clustering binario** en el **nivel más alto de la jerarquía** y entre **clases lejanas semánticamente**, ya que pertenecen dos a "Deportes" y dos a"Ciencia".

   d) **ABC&DEF_1000**: colección formada por 1000 páginas de cada una de las clases A, B, C, D, E y F, con las que se generan los diferentes vocabularios. Con esta subcolección, como en el caso anterior, se realiza un **clustering binario** en el **nivel**

más alto de la jerarquía y entre **clases lejanas semánticamente**: tres pertenecen a "Bancos & Finanzas" y otras tres a"Lenguajes de programación".

2. *Clustering* en 3 clases:

   a) **ABC_1000**: colección formada por 1000 páginas de cada una de las clases A, B y C, con las que se generan los diferentes vocabulario. Con esta subcolección se realiza un **clustering en 3 clases** en el **nivel más bajo de la jerarquía** y entre **clases cercanas semánticamente**, ya que todas pertenecen a la misma categoría.

3. *Clustering* en 6 clases

   a) **ABC&DEF_1000**: subcolección con la que se puede realizar un **clustering en 6 clases** en el **nivel más bajo de la jerarquía** y distinguiendo entre **clases cercanas y lejanas semánticamente**, ya que pertenecen 3 a 3 a las categorías "Bancos & Finanzas" y "Lenguajes de programación".

4. *Clustering* en 10 clases

   a) **AJ_1000**: colección completa formada por 1000 páginas de cada una de las 10 clases que constituyen la colección de referencia. Con esta colección se pretende realizar un **clustering en 10 clases** en el **nivel más bajo de la jerarquía** y entre **clases cercanas y lejanas semánticamente**.

7.3.2. Colección WebKB

Esta colección fue creada por el grupo de investigación dirigido por Tom Mitchell, de la Carnegie Mellon University, dentro del proyecto: *World Wide Knowledge Base (Web->KB)*[2].

Su fin era el estudio de la estructura de hiperenlaces que forman los documentos web y por ello se fijó un ámbito bastante reducido: el entorno universitario. En enero de 1997, se descargaron las páginas web de los departamentos de *Computer Science* de cuatro universidades estadounidenses: Cornell University[3]; The University of Texas at Austin[4]; University of Washington[5]; y University of Wisconsin-Madison[6].

Esta colección *WebKb* está formada por 8.282 páginas clasificadas manualmente en 7 clases:

- *student*, con 1641 páginas web.
- *faculty*, con 1124.
- *staff*, con 137.

[2]http://www.cs.cmu.edu/~webkb/
[3]http://www.cornell.edu/
[4]http://www.utexas.edu/
[5]http://www.washington.edu/
[6]http://www.wisc.edu/

- *department*, con 182.
- *course*, con 930.
- *project*, con 504.
- *other*, con 3764.

La última clase *other* supone un "cajón de sastre" donde se introdujeron aquellas páginas que no se sabía donde clasificar. Por ejemplo, si un miembro del profesorado tenía una página personal, una lista de publicaciones, un currículum vitae y varias páginas relativas a su investigación, sólo su página personal era clasificada como perteneciente a la clase *faculty*. El resto, se consideraban como pertenecientes a la categoría *other*. En nuestra experimentación se eliminó, ya que se trataba de una clase que no agrupaba a elementos con características comunes, quedando así una colección formada por 4.518 páginas preclasificadas en las 6 categorías restantes, cada una de ellas de un tamaño diferente.

Esta colección es menos heterogénea que la *BankSearch DataSet*, al tratarse de páginas pertenecientes a un ámbito mucho más reducido. Por este motivo, puede pensarse que no representa en gran medida la heterogeneidad existente en la Web, al contrario que la *BankSearch DataSet*, cuyo ámbito es bastante mayor. Aún así, se consideró una colección válida para realizar la experimentación.

Como en el caso anterior, las subcolecciones empleadas en clasificación y *clustering* son diferentes. Por otro lado, debido a que la colección WebKB no presenta niveles jerárquicos, se seleccionó un único corpus para el problema de TC y otro para el DC.

Subcolecciones generadas para Clasificación Automática

Los criterios para la selección del conjunto de estimación y validación son los mismos que en el caso de la colección *BankSearch*. La subcolección considerada para TC fue:

- **WebKB_0.7**: subcolección formada por el 70 % de las páginas de cada una de las seis clases principales: *student*, *faculty*, *staff*, *department*, *course* y *project*, con las que se generan los diferentes vocabularios y se realiza la fase de aprendizaje. El otro 30 % se emplea en la fase de validación. Con esta subcolección se puede realizar una **clasificación en 6 clases** y siempre entre **clases cercanas semánticamente**, ya que todas las categorías estarían relacionadas bajo la superclase "Computer Science".

Subcolección generada para *Clustering* de documentos

En este caso, los vocabularios se crean a partir de los rasgos presentes en el conjunto total de documentos de la colección, por lo que la subcolección empleada fue:

- **WebKB**: colección formada por todas las páginas de cada una de las seis clases principales: *student*, *faculty*, *staff*, *department*, *course* y *project*. Con esta subcolección se puede realizar un **clustering en 6 clases** entre **clases cercanas semánticamente**.

7.4. Selección del Vocabulario

Una vez definidas las colecciones con las que se va a realizar la experimentación, el siguiente paso es crear los vocabularios con los que representar los documentos. Para cada una de las colecciones consideradas se crean vocabularios con distintos tamaños, generados tras una fase de preproceso y empleando funciones de reducción de rasgos con diferentes parámetros.

El objetivo es observar el comportamiento de cada una de las representaciones en función de la dimensión del vocabulario, es decir, observar cómo se comportan las diferentes representaciones con vocabularios grandes (que supuestamente contienen más información) y con vocabularios muy pequeños (con menos información).

Para cada colección, se toman vocabularios con tamaños comprendidos entre un mínimo del orden de 60-100 rasgos, hasta dimensiones de un orden de magnitud menor que el tamaño del vocabulario generado sin ninguna reducción. En cada caso, para cada representación y colección considerada, se toman entre cinco y siete valores en la dimensión del vocabulario generado a partir de dicha colección.

7.4.1. Preproceso y primera selección de rasgos

Para representar cada página sólo se consideran aquellos rasgos con longitud mayor de 2 caracteres y no superior a 30 caracteres. El preproceso y la primera selección de rasgos constó de las siguientes fases:

1. Análisis léxico. Mediante este proceso se identifica el léxico presente en un texto, de forma que se pueda crear el vocabulario con el que representar el documento. En las representaciones evaluadas en esta memoria, este análisis se realiza del siguiente modo. Primero, se toman todas aquellas cadenas de caracteres separadas por espacios en blanco como potenciales rasgos iniciales. Algunos caracteres especiales (@,/,&,-,..) se mantienen ya que se considera que pueden ser útiles para reconocer direcciones de correo, URLs, palabras compuestas, etc. Otros caracteres especiales (!, #,$, %, &,_,{,},~,^,\,[,],!,",–,|,·,®,*,´,¿,¡,≪≫,º,ª,©,£,< >) se eliminan. El punto "." se trata de forma especial para intentar no perder información relativa a siglas; es decir, los puntos no se eliminaron cuando se encontraban en una sucesión letra-punto-letra-punto. . . –por ejemplo, O.N.U–. No se consideran términos multipalabra, aunque el método general que se describe en este punto es independiente de este tipo de consideraciones.

2. Eliminación de los rasgos coincidentes con los presentes en una lista de palabras vacías de contenido (*stop-words*), donde había artículos, determinantes, preposiciones... En concreto, en esta tesis se trabaja con colecciones de páginas web en inglés y se emplea una lista de *stop-words* utilizada en el CLEF [7] y en tareas de IR multilingüe.

3. Truncamiento.

   Se ha elegido el algoritmo de *stemming* de Porter para palabras en ingles (Porter, 1997) por su simplicidad y su amplia utilización en tareas de IR, TC y DC.

4. Eliminación de aquellos rasgos que sólo aparecían una vez en la colección. Estos rasgos aparecen una única vez en un sólo documento, por lo que se puede considerar que se trata de rasgos poco significativos, tanto para la página que se quiere representar, como para el conjunto de la colección.

7.4.2. Funciones de reducción de rasgos

A continuación, se presentan los dos métodos de reducción de rasgos que se aplicaron para la generación de los diferentes vocabularios de partida. Como ya se ha dicho, este proceso se hace necesario dado el enorme tamaño que toman las representaciones vectoriales cuando las dimensiones del corpus son significativas. Como ejemplo ilustrativo, a partir de una colección de 10.000 páginas web se pueden generar –después de un análisis léxico, una eliminación de stop-words y una fase de truncamiento típicas– un vocabulario del orden de 200.000 rasgos. El manejo de representaciones de estas dimensiones es una tarea costosa para muchos sistemas y por tanto es conveniente aplicar funciones de reducción de rasgos.

En estos experimentos se han realizado dos tipos de reducción de rasgos: "*Reducción term-frequency/document-frequency*" y "*Reducción con la propia función de ponderación*".

7.4.3. Reducción *term-frequency/document-frequency*

Esta función de reducción, a la que llamaremos "**MinMax**", es una variación de la reducción clásica *Document Frecuency* (Sebastiani, 2002) y está basada en la selección de rasgos por frecuencia de aparición, ya sea en el propio documento que se esté representando como en el conjunto total de páginas de la colección.

Con esta reducción, a partir de un vocabulario generado con el total de rasgos encontrados tras la fase de preproceso, se eliminan todos aquellos rasgos que aparezcan menos de $TFmin$ veces en menos de $DFmin$ documentos, y más de $TFmax$ veces en más de $DFmax$ documentos de la colección. Considerando diferentes valores para las variables $TFmin$, $DFmin$, $TFmax$, $DFmax$, es posible generar representaciones con dimensiones en un rango de valores entre cero y N, siendo N el tamaño del vocabulario sin reducir.

[7] Cross Language Evaluation Forum

Esta reducción guarda en su definición las heurística recogidas a lo largo de los años relativas a la frecuencia de un rasgo en un documento y en una colección. Se eliminan los rasgos más y menos frecuentes, así como los que presentan una mayor frecuencia de documento. Sin embargo, es posible que para aquellas funciones de ponderación que no están basadas directamente en estas frecuencias, este tipo de reducción no sea la más conveniente.

7.4.4. Reducción con la propia función de ponderación

Hay funciones de ponderación que emplean otras variables diferentes a la frecuencia de un rasgo en el documento y en la colección, por ejemplo, la frecuencia de un rasgo en un determinado elemento o conjunto de elementos HTML. En estos casos, la reducción *MinMax* no considera todos los factores que tienen en cuenta estas representaciones. Por este motivo, se propone una reducción, llamada "**PF**", donde las propias funciones de ponderación se empleen como función de reducción de rasgos.

El uso de esta reducción está pensado fundamentalmente para aquellas funciones de ponderación capaces de asignar un valor directamente proporcional a la relevancia de un rasgo. Este comportamiento no se da en el caso caso de la representación TF, por ejemplo, donde se tenderían a seleccionar los rasgos más frecuentes en una página, lo que podría llevar a una selección de rasgos de uso común. Para generar los vocabularios reducidos se ponderan todos los rasgos de una página y una vez pesados, se seleccionan los n rasgos con mayor relevancia.

Esta reducción no se puede aplicar en el caso de la función binaria, ya que si el número de rasgos en una página es mayor que este n, la selección de los N rasgos más ponderados resultaría arbitraria. En el caso de la representación BinIDF sucede algo parecido. Aunque sí es posible crear vocabularios reducidos por medio de PF; todos los rasgos toman un mismo valor en todos los documentos de la colección, ya que la componente local de la función de ponderación es la función binaria y el factor IDF sólo depende del tamaño del corpus y de la frecuencia del rasgo en la colección. Por este motivo, dos rasgos tendrán el mismo valor de relevancia sólo cuando el número de documentos en los que aparecen sea idéntico.

De igual modo que con la reducción MinMax, tomando diferentes valores para la variable n es posible generar vocabularios con dimensiones en un rango de valores entre cero y N, siendo N el tamaño del vocabulario sin reducir.

7.5. Conclusiones

En este capítulo se han presentado las características del diseño experimental. Se ha seleccionado un conjunto de funciones de proyección que serán evaluadas, junto a las propuestas FCC y ACC, para determinar la calidad de las representaciones generadas con cada una de las funciones, en tareas de TC y DC.

Se ha seleccionado un conjunto de subcolecciones extraídas de dos colecciones de referencia en clasificación y *clustering* de páginas web, para poder evaluar así la calidad de las representaciones en función de sus resultados en tareas de clasificación y *clustering* con 2, 3, 4, 6 y 10 clases.

Además, con el fin de obtener diferentes magnitudes en los vocabularios y poder evaluar la calidad de las representaciones en función del tamaño de la representación, se proponen dos funciones de reducción de rasgos: *MinMax* y *PF*, que se emplearán en la creación de los vocabularios reducidos. El carácter de cada una de estas funciones es muy diferente. La función *MinMax* se basa en una reducción por frecuencias de aparición, lo que tiene más sentido en funciones de ponderación basadas en dichas frecuencias. Por el contrario, la función *PF* considera la propia función de ponderación como método de reducción de rasgos.

Capítulo 8

Clasificación automática mediante un algoritmo Naïve Bayes

"Los conceptos están incluidos en las palabras"
Henri Bergson

En este capítulo se evalúa la calidad de las dos funciones de ponderación propuestas en esta tesis doctoral, como parte de un modelo de representación autocontenida de páginas web, mediante la aplicación de un algoritmo de clasificación automática de documentos Naïve Bayes. Se comparan ACC y FCC con funciones de ponderación clásicas aplicadas en textos (BinIDF, TF, TF-IDF y WIDF) y con otras dos funciones aplicadas en el ámbito de la representación de páginas web que utilizan información del etiquetado HTML (Title y la combinación de categorías propuesta en (Molinari et al., 2003)).

8.1. Introducción

La calidad de una representación se evalúa en función de los resultados que se obtengan tras su aplicación y depende, por tanto, de la tarea que se realice a continuación.

En este capítulo se evalúan diferentes modelos de representación de páginas web por medio de un algoritmo de clasificación automática. La idea es la siguiente: si se emplea un mismo algoritmo con diferentes representaciones, la mejor representación será aquella con la que se obtengan los mejores resultados de clasificación. Estas representaciones se diferencian unas de otras en la función de ponderación F que emplean dentro del modelo de espacio vectorial, definido en la sección 2.3.2.

El método de clasificación elegido para esta evaluación ha sido un algoritmo *Naïve Bayes* (NB), un método clásico y sencillo que ha ofrecido buenos resultados en tareas de clasificación automática de textos [(Domingos y Pazzani, 1996), (McCallum y Nigam, 1998), (Yang y Liu, 1999) y (Chakrabarti, 2003)]. En este punto es importante remarcar que el objetivo de este capítulo no es la evaluación de la calidad del algoritmo de clasificación considerado, sino comparar con un mismo algoritmo diferentes modelos de representación aplicados a documentos HTML. Se emplearán distintas funciones de probabilidad de un rasgo a una clase, así como diferentes colecciones, de modo que se puedan obtener conclusiones lo más generales posibles sobre los

resultados de la clasificación. La comparación se realizará siempre en términos relativos.

Este capítulo se estructura como sigue. En primer lugar, se trata brevemente el problema del aprendizaje automático y de la clasificación automática de documentos. A continuación, se introduce el problema de la clasificación automática de páginas web y se presentan los fundamentos teóricos sobre los que se construyen los clasificadores *Naïve Bayes*, junto con las diferentes funciones de probabilidad empleadas en esta experimentación. Por último, se muestran los resultados experimentales y las principales conclusiones extraídas a partir de ellos.

8.2. Aprendizaje Automático

Hasta finales de los años 80 del siglo pasado, el enfoque más popular dentro de la clasificación automática de textos era el de la "Ingeniería del Conocimiento", consistente en la definición manual de un conjunto de reglas que codificaban el conocimiento experto humano sobre cómo clasificar un documento dado un conjunto prefijado de clases. En la década siguiente este enfoque fue perdiendo popularidad, especialmente en la comunidad científica, en favor del paradigma del aprendizaje automático (*Machine Learning*) (Sebastiani, 2002).

El aprendizaje automático puede definirse como la disciplina que permite desarrollar sistemas capaces de realizar una tarea de un modo automático y de forma que el desempeño de dicha tarea resulte mejor con experiencia que sin ella (Mitchell, 1997). Según E. Alpaydin, el aprendizaje automático consiste en la programación de sistemas informáticos de forma que se optimice un determinado criterio de rendimiento, empleando datos de entrenamiento o con ayuda de una experiencia pasada. Se define un modelo a partir de un conjunto de parámetros y, mediante un aprendizaje, se ejecuta un programa que optimiza dichos parámetros, usando un conjunto de datos que representan experiencia pasada (Alpaydin, 2004).

Las ventajas que aporta el aprendizaje automático a la clasificación de documentos son unas tasas de precisión comparables a las ofrecidas por un experto humano, así como un ahorro en términos de trabajo potencial humano, ya que no se requiere del conocimiento de un experto en un determinado tema para la construcción de un clasificador, ni en el caso en que se desee cambiar el conjunto de categorías (Sebastiani, 2002).

En general existen dos enfoques principales dentro del aprendizaje automático: a) aprender para poder generar un nuevo conocimiento o comportamientos para un determinado sistema y, b) aprender para tratar de mejorar el comportamiento de un sistema (Alpaydin, 2004). En el primer caso se suelen utilizar técnicas basadas en razonamiento inductivo, mientras que la segunda suele estar relacionada con la utilización de técnicas analíticas. Ambos enfoques pueden emplearse también conjuntamente.

Para construir cualquier clasificador de documentos sería necesario seguir los siguientes pasos:

- Construir una base matemática que derive del sistema de clasificación y que nos permita representar un documento.

- Desarrollar procedimientos por los cuales los documentos puedan ser clasificados automáticamente dentro de categorías.

- Determinar la precisión o exactitud de la clasificación en relación a algún criterio.

Dentro del aprendizaje automático, y dependiendo de si se dispone o no de datos etiquetados, se puede distinguir entre: aprendizaje *supervisado*, *no supervisado* y *semi-supervisado*.

El aprendizaje *supervisado* se construye sobre un conocimiento a priori. En TC se debe disponer de un conjunto de documentos de ejemplo para cada una de las categorías en las que se quiere clasificar. Después de una etapa de entrenamiento, el sistema queda ajustado de tal modo que ante nuevos ejemplos, el algoritmo es capaz de clasificarlos en alguna de las clases existentes. Cuanto mayor sea el conjunto de datos etiquetados mayor será la información potencial disponible y, previsiblemente, mejor resultará el aprendizaje.

Los sistemas de aprendizaje *no supervisado* son aquellos que no disponen de conocimiento a priori, de modo que no se dispone de datos previamente etiquetados.

En problemas reales, y más en el caso de la clasificación de textos, la situación más frecuente es tener un conjunto pequeño de documentos preclasificados y un gran conjunto de documentos sin etiquetar. De este modo, compartir rasgos y similitud entre documentos etiquetados y no etiquetados es una fuente de información que puede permitir aumentar la precisión de los sistemas de TC (Nigam et al., 2000). Este procedimiento se conoce como aprendizaje *semi-supervisado*, en el cual se utiliza un conjunto de datos no etiquetados para refinar una estimación inicial obtenida a partir de los datos etiquetados de los que se disponga.

8.3. Clasificación automática de documentos

La clasificación automática de documentos se puede entender como aquella tarea en la que un documento, o una parte del mismo, es etiquetado como perteneciente a un determinado conjunto, grupo o categoría predeterminada (Yang y Pedersen, 1997).

El problema de la clasificación automática es parte de un problema mayor de análisis automático de contenidos y ya en la década de los sesenta se daban los primeros pasos en este campo. La cuestión que se planteaba entonces era saber si una máquina podría ser programada para determinar el tema del contenido de un documento y la categoría, o categorías, en las que debería ser clasificado.

En 1961 M. E. Maron aplicó un formalismo estadístico al problema de la indexación de documentos que involucraba, en primer lugar, la determinación de ciertas probabilidades relacionadas entre términos y categorías y, en segundo, el uso de estas probabilidades para determinar la clase a la que un documento con ciertas palabras pertenecía (Maron, 1961). Usando un conjunto de palabras seleccionadas, una máquina era capaz de predecir la clase a la que, con mayor probabilidad, pertenecería un documento. Éste fue el punto de partida de los clasificadores

basados en palabras clave. La evaluación de estos sistemas se debía hacer manualmente por un experto humano que determinaba si el sistema funcionaba bien y si las decisiones tomadas eran correctas o no.

Un enfoque diferente demostró que las categorías en las que es posible clasificar un documento podían ser derivadas de un proceso de análisis de factores. En (Borko, 1962), H. Borko argumentaba que si se fuera capaz de encontrar un conjunto de categorías que nos dieran la mejor descripción posible de un dominio de documentos, la tarea de clasificación sería más simple y podría afrontarse de modo automático; además, se obtendrían mejores resultados.

A partir de estas ideas se han desarrollado gran parte de los sistemas de TC utilizados en la actualidad: desde clasificadores basados en palabras clave con probabilidades asociadas a priori a cada clase, hasta sistemas que encuentran por sí mismos los rasgos característicos de cada categoría.

Dentro del campo del aprendizaje supervisado, en todo sistema de TC pueden distinguirse las siguientes etapas:

Representación-Aprendizaje-Clasificación

Además, posteriormente deberá establecerse un método de evaluación de la calidad del clasificador. El clasificador NB con el que se van a evaluar las representaciones propuestas pertenece a esta categoría.

La etapa de representación se ha tratado en el capítulo 2 y, como se ha dicho, resulta de suma importancia en cualquier sistema de clasificación. Será la fase encargada de transformar la información textual, de carácter cualitativo, ya que envuelve el tratamiento de significados, en una serie de objetos que puedan resultar tratables desde un punto de vista computacional.

El aprendizaje es la etapa en la que se obtiene la información de clase para cada una de las categorías en las que será posible clasificar un documento dado. La obtención de estos "descriptores de clase" es el resultado de la etapa de entrenamiento. Este entrenamiento se realiza sobre una jerarquía temática o árbol que puede estar estructurado en uno o varios niveles de generalidad. A partir de un número de documentos que resulte representativo de cada categoría es posible encontrar las características, o descriptores, de cada nodo del árbol de clase (Yang, 1999). Una vez se han encontrado los descriptores de clase es factible abordar la etapa de clasificación. En pocas palabras, el problema de la clasificación de un documento del que se desconoce la categoría a la que pertenece estriba en encontrar la clase a la que pertenecen, a partir de los descriptores de clase que mejor cubran la representación que se ha hecho previamente del documento.

Por último, para evaluar la calidad de la clasificación, el conjunto de datos de entrenamiento se divide en dos subconjuntos: uno servirá para entrenar el algoritmo y el otro para comprobar la calidad del mismo.

En la literatura pueden encontrarse gran cantidad de algoritmos para clasificación automática de textos. Las tareas de TC se aplican en muchos contextos, desde la indexación de documentos

basada en vocabularios controlados (Field, 1975), hasta el filtrado de documentos [(Drucker et al., 1999), (Androutsopoulos et al., 2000)], la generación automática de metadatos, en tareas de desambiguación semántica (Escudero et al., 2000), en la creación de jerarquías de páginas web (Oh et al., 2000a), etc. Otros ejemplos de trabajos que aplican técnicas de clasificación automática en textos son (Aha, 1990), (Tan y Lai, 2000), (Nigam, 2001) y (Kruengkrai y Jaruskulchai, 2002). En casi todos ellos se emplean representaciones basadas en el VSM y LSI con funciones de ponderación *binaria*, TF y TF-IDF. Además, suelen aplicar funciones de reducción de rasgos como la IG y la MI como métodos de reducción de la dimensión del vocabulario con el que se generan las representaciones.

La mayor parte de los algoritmos de clasificación que se han propuesto no son específicos para la clasificación de documentos, sino que se sirven para clasificar cualquier tipo de objeto. Entre algunos de los más usados destacan:

- **Clasificadores basados en algoritmos de aprendizaje "clásicos"**. El problema de la generación de clasificadores automáticos se puede particularizar a clasificadores automáticos de textos. Un ejemplo de sistema basado en un algoritmo clásico es el del sistema RIPPER (Cohen, 1995) que utiliza la variante del algoritmo de inducción de árboles de decisión ID3 llamada C4.5. El sistema induce reglas de decisión para obtener un clasificador de una única clase, utilizando en la etapa de entrenamiento ejemplos positivos y negativos, y la ganancia de información como función de calidad. La representación de cada documento se enmarca dentro del VSM y con una función de ponderación TF.

  Una extensión de RIPPER es FLIPPER, una versión que utiliza, para definir el clasificador, una forma de representación más rica: la lógica de predicados de primer orden, tomando como base el sistema de aprendizaje FOIL (Quilan, 1990).

- **Algoritmos probabilísticos**. Se basan en la teoría de probabilidad de Bayes, que se verá con detalle en la sección 8.5. El teorema de Bayes permite estimar la probabilidad de un suceso a partir de la probabilidad de que ocurra otro suceso, del cual depende el primero. El algoritmo más conocido dentro de este enfoque es el denominado *Naïve Bayes*, con el que se realizará la evaluación de las representaciones propuestas (el algoritmo de describe en detalle en la sección 8.6).

  Pueden encontrarse numerosos ejemplos de aplicación de algoritmos probabilísticos a la clasificación de documentos, entre otros (Lewis y Ringuette, 1994), (Heckerman, 1995), (McCallum y Nigam, 1998), (Nigam et al., 2000), (Rennie, 2001) y (Zhang y Yang, 2004).

- **Algoritmo de Rocchio**. Este algoritmo (Rocchio, 1971) ha sido aplicado en la realimentación de consultas. Una vez formulada y ejecutada una primera consulta, el usuario examina los documentos devueltos y determina cuáles le resultan relevantes y cuáles no. Con estos datos, el sistema genera automáticamente una nueva consulta

basándose en la información proporcionada por el usuario. Es capaz de construir el vector de la nueva consulta recalculando los pesos de los términos de ésta y aplicando un coeficiente a los pesos de la consulta inicial, otro a los de los documentos relevantes, y otro distinto a los de los no relevantes.

El algoritmo de Rocchio proporciona un sistema para construir patrones de cada una de las clases o categorías de documentos que se consideren. De este modo, en la fase de entrenamiento se parte de una colección de entrenamiento preclasificada manualmente y se construyen descriptores para cada una de las clases, considerando como ejemplos positivos los documentos de entrenamiento de esa categoría y como ejemplos negativos los del resto. Para categorizar un nuevo documento, bastará con encontrar la distancia entre la representación de los documentos y cada uno de los descriptores. Aquel descriptor que presente mayor similitud indicará la categoría a la que se debe asignar el documento. Algunos trabajos donde se aplica este algoritmo son (Lewis et al., 1996), (Joachims, 1997) y (Figuerola et al., 2001).

- **Algoritmo del vecino más cercano** (*Nearest Neighbour*, NN) y variantes. Este algoritmo se basa en la aplicación de una métrica que establezca la similitud entre un documento que se quiere clasificar y cada uno de los documentos de entrenamiento. La clase o categoría que se asigna al documento será la categoría del documento más cercano según la métrica establecida.

  Una de las variantes más conocidas de este algoritmo es la de los k-vecinos más cercanos (k-*nearest neighbour*, KNN), que consiste en tomar los k documentos más parecidos en lugar de sólo el primero. Como los documentos más cercanos pueden pertenecer a categorías diferentes, se asignará aquella que más veces haya aparecido. El KNN une a su sencillez una eficacia notable (Figuerola et al., 2002). En ninguno de los dos casos hay entrenamiento, ya que el clasificador se construye a partir de la propia representación automática de los documentos y la posterior aplicación del algoritmo descrito anteriormente. Algunos trabajos que emplean este algoritmo como método de clasificación son [(Hersh, 1994), (Norbert Gövert y Fuhr, 1999) y (Yang, 1999)]. KNN suele ser bastante eficaz cuando el número de categorías es alto y cuando los documentos son heterogéneos (Norbert Gövert y Fuhr, 1999).

- **Algoritmos basados en redes neuronales**. Las redes neuronales han sido aplicadas a problemas de categorización de documentos en numerosas ocasiones [(Schutze et al., 1995), (Yin, 1996), (Ruiz y Srinivasan, 1998), (Ruiz y Srinivasan, 1999) y (Lam y Lee, 1999)]. Es posible entrenar una red neuronal para que dada una entrada determinada (un vector de representación) produzca una salida deseada (la categoría a la que corresponde ese documento). Existen muchos tipos de redes neuronales, con topologías y características bien diferenciadas; por este motivo no se va a entrar a describirlas en este punto, baste

como muestra las anteriores referencias.

- ***Support Vector Machines***. Se trata de un método de aprendizaje automático presentado en (Vapnik, 1995) y (Cortes y Vapnik, 1995). Estos algoritmos pretenden encontrar una hipersuperficie de separación entre clases dentro del espacio de representación, de modo que una banda, o margen, lo más gruesa posible esté vacía alrededor de la región de separación entre clases (Cristianini y Shawe-Taylor, 2000).

  Los SVM resultan ser sistemas de aprendizaje universal que, en su forma más simple, son capaces de encontrar una función lineal discriminante que separe el espacio de representación en regiones correspondientes a cada una de las clases consideradas. Como en todo aprendizaje supervisado, la entrada al sistema es un conjunto de ejemplos de entrenamiento. Si el conjunto de entrenamiento es linealmente separable, el SVM encuentra un hiperplano que maximiza la distancia euclídea a los ejemplos de entrenamiento más cercanos. En el caso de que el conjunto no pueda ser linealmente separable, se mide el error en el entrenamiento. Así, el cálculo de este hiperplano se reduce a un problema de optimización de errores en el que se aplican restricciones (Vapnik, 1995). Las restricciones fuerzan a que todos los ejemplos de entrenamiento sean clasificados correctamente por encima de un error mínimo.

  Una característica fundamental de estos clasificadores es que son independientes de la dimensión del espacio de características. Así, la medida de la complejidad no depende del número de rasgos, sino de cómo puedan separarse los datos. Esto significa que siempre que los datos sean separables, podría generalizarse la presencia de muchos rasgos en una determinada categoría.

8.4. Clasificación automática de páginas web

Algunas de estas técnicas de TC han sido aplicadas a la construcción de taxonomías o directorios *web* temáticos [(Garofalakis, 1999), (Sebastiani, 2002)], así como en sistemas de detección de "correo basura" [(Sahami et al., 1998), (Cunningham et al., 2003)], de clasificación automática de e-mails [(Tong y Koller, 2001), (Crawford et al., 2002)] y en sistemas extractores de noticias (Mase et al., 2000). Estas técnicas permiten que un sistema de información web pueda tratar con espacios reducidos de documentos que incluyan únicamente aquellos que pertenecen a un determinado tema (Fathi et al., 2004).

Como ya se vió en el capítulo 3, los primeros clasificadores aplicados a la categorización de páginas web fueron los *Full-Text classifiers*, clasificadores que se basaban en un análisis del contenido textual de los documentos HTML y no consideraban ninguna información extraída de la estructura del grafo de hipertexto. En estos casos, las funciones de ponderación empleadas no pasaban de ser funciones clásicas: binaria, TF y TF-IDF. Dentro de este tipo se encuentran los

clasificadores bayesianos, así como los algoritmo de los k vecinos más cercanos (Yang, 1999).

Como ya se vió, además de estos clasificadores basados en contenido se han propuesto muchos otros que exploran la estructura de hiperenlaces que forma la Web. Por ejemplo, en (Chakrabarti et al., 1998c) se proponía utilizar el texto contenido en los documentos enlazados a uno dado y clasificar los documentos combinados. Este enfoque tenía el problema de que, en muchas ocasiones, documentos enlazados pueden referirse a diferentes temáticas (Fathi et al., 2004). En (Ghani et al., 2001) se realizó un análisis similar y se mostró que este tipo de información añadida a un documento debía de tratarse con sumo cuidado, ya que en ocasiones descendían las tasas de clasificación al introducir esta información. En (Oh et al., 2000b) se propuso ponderar de un modo diferente los rasgos aparecidos en un documento y en su vecindario a la hora de clasificarlo.

Otros métodos proponen el uso de los *anchortexts* para predecir la clase de los documentos apuntados por esos enlaces. En (Furnkranz, 2001) se expone un método de clasificación basado únicamente en el contenido textual de los enlaces que apuntan a un determinado documento. En (Glover et al., 2002) se realiza una clasificación similar, añadiendo además la información contenida en los *extended anchortexts*.

Otro tipo de clasificadores emplean métricas basadas directamente en la estructura de los hiperenlaces [(Calado et al., 2003), (Sun y Lim, 2003)]. En estos casos, las medidas se basan en un análisis de las correlaciones (co-citaciones) presentes en los documentos a clasificar.

Los SVM como sistemas de clasificación están siendo cada vez más aplicados al entorno de Internet. Se basan en un principio de minimización del error, es decir, en encontrar una hipótesis para la cual se pueda garantizar que se minimiza la probabilidad de error frente a un conjunto seleccionado aleatoriamente de ejemplos de test. En muchos casos, el diseño de estos sistemas se reduce a la selección y combinación de diferentes *kernels*, o funciones generadoras, de modo que se puedan hacer corresponder con una métrica empleada como medida de distancia entre datos (Thorsten Joachims y Shawe-Taylor, 2001). Empleando diferentes *kernels* se pueden conseguir no sólo funciones discriminantes polinómicas, sino redes de función básica radial (*radial basic function networks*) y redes neuronales con función de transferencia sigmoide y de tres capas (Joachims, 1998). La composición de *kernels* puede realizarse en aquellos casos en los que no hay un conocimiento real de cuál es la mejor medida de similitud entre datos. En (Thorsten Joachims y Shawe-Taylor, 2001) se clasifican páginas web considerando la combinación de *kernels* correspondientes, por un lado, a un análisis de la estructura de hiperenlaces y co-citaciones, y por otro, al análisis del contenido textual de la página. En este caso, se emplean matrices *término-documento* y *documento-documento* donde los coeficientes de dichas matrices vienen dados por una función de ponderación TF y TF-IDF. En (Joachims, 1999) se presenta una variante de las SVMs llamada "*transductive support vector machines*". Otro trabajo donde se han aplicado los SVM al contexto de los hipertextos es (Cristianini y Shawe-Taylor, 2000).

No se va a profundizar más en este punto debido a que la mayor parte de los sistemas de TC

aplicados al contexto web ya han sido detallados en el capítulo 3, cuando se realizó una revisión de los principales métodos de representación de páginas web encontradas en la literatura.

En la siguiente sección se presentan las bases de los algoritmos NB, seleccionados para la evaluación de la representaciones propuestas. Como ya se comentó al iniciar el capítulo, se ha elegido un clasificador NB por ser un algoritmo sencillo que ofrece buenos resultados [(Domingos y Pazzani, 1996), (Mitchell, 1997)]. Cabe pensar que una buena representación afectará por igual a cualquier algoritmo de clasificación, independientemente de la calidad del mismo. Por tanto, es de esperar que si las representaciones propuestas tienen mejor comportamiento que otras obtenidas con técnicas más convencionales en un algoritmo NB, también mejoren a estos métodos clásicos en otro tipo de clasificadores.

Como este estudio no se va a centrar en la calidad de los métodos de clasificación, los algoritmos NB suponen sistemas de más fácil diseño que los SVMs, y más aún cuando se pretenden realizar diferentes tipos de clasificación: binaria, no binaria, entre clases "semánticamente cercanas", "lejanas", etc. Los algoritmos NB no requieren de ninguna fase de optimización y se basan fundamentalmente en un simple conteo de frecuencias para establecer las probabilidades a priori que necesitan (Rennie, 2001). Una de las principales diferencias con los SVM es el hecho de que en un aprendizaje bayesiano se emplean únicamente ejemplos positivos durante el entrenamiento, mientras que en el caso del SVM, para cada clase se toma un conjunto de ejemplos positivos y negativos.

8.5. La teoría de Bayes aplicada a la clasificación automática de textos

En (Mitchell, 1997) se describe un clasificador probabilístico Naïve Bayes ("Bayes ingenuo") similar al que se va a utilizar en esta experimentación. El adjetivo ingenuo está relacionado con la asunción de independencia entre rasgos de una misma página, el mismo principio de independencia sobre el que se construye el VSM. Esta suposición es incorrecta pero, como se ha visto, supone una primera aproximación a la resolución del problema de la representación y clasificación de documentos. Además, en (Domingos y Pazzani, 1996) se verifica que los algoritmos NB tienen un comportamiento bueno aunque existan fuertes dependencias entre las componentes. Se demuestra que un clasificador NB no depende de la independencia total entre componentes para resultar un clasificador óptimo.

Existen diferentes técnicas que intentan capturar la dependencia real entre rasgos. El modelo más general aplicable en este contexto es la red bayesiana (Heckerman, 1995). En este caso hay un único nodo que codifica la clase de un documento. De este nodo salen conexiones hacia un conjunto de rasgos considerados del conjunto total, pudiendo estar conectados entre sí los nodos asociados a dichos rasgos. Como es de suponer, obtener este tipo de estructura a partir

del conjunto de entrenamiento es una tarea muy compleja, debido fundamentalmente a las dimensiones del espacio del problema, que resultan del orden de cientos de miles de rasgos.

En (Koller y Sahami, 1996) se propone una aproximación "voraz" (*greedy*) para disminuir la complejidad en la obtención de la red bayesiana. Con esta aproximación, el conjunto total de rasgos disminuye alrededor de 2/3 sin perder precisión en la clasificación; incluso en algunos casos, debido a la eliminación de ruido, se incrementa la precisión hasta un 5 %. Sin embargo, según los propios autores, es necesario realizar análisis experimentales en conjuntos más grandes para evaluar realmente el impacto en la clasificación con esta forma de modelar la dependencia entre rasgos.

Otra técnica que tiene en cuenta también la dependencia entre rasgos es el método de la entropía máxima. En este caso se estima la ocurrencia de un rasgo individual con probabilidades marginales que restringen las probabilidades de las celdas de una tabla gigante de riesgos que potencialmente involucra cualquier subconjunto de rasgos. Operacionalmente hay similitudes con las redes bayesiana, en el sentido de que se tiene que elegir qué regiones hay que estimar de la tabla de riesgos para posteriormente aplicar la clasificación. En los experimentos llevados a cabo con este tipo de modelos, los resultados obtenidos en comparación con un clasificador NB han sido muy variables y muy dependientes del conjunto específico de datos considerados (Nigam et al., 1999).

8.6. Clasificador Naïve Bayes

El clasificador NB está basado en la teoría de la decisión de Bayes: la teoría de las probabilidades condicionadas. Por tanto, el problema de la clasificación se reduce al cálculo de las probabilidades a posteriori de una clase dado un documento.

$$P(c_k \mid \vec{d_j}) = \frac{P(\vec{d_j} \mid c_k)P(c_k)}{P(\vec{d_j})} \tag{8.1}$$

Una vez estimados estos valores de probabilidad $P(c_k \mid \vec{d_j})$, $\forall c_j$, la tarea de un clasificador NB es simplemente elegir la clase que haga mayor esta probabilidad. Para ello, deben estimarse primero otras cantidades como son: las probabilidades a priori de cada clase, $P(c_k)$, y del documento, $P(\vec{d_j})$; además de las probabilidades condicionadas de cada documento a una clase dada, $P(\vec{d_j} \mid c_k)$.

En la mayoría de los casos se considerará que las probabilidades a priori de cada documento son iguales, con lo que la cantidad $P(\vec{d_j}) = Cte$, donde d_j representa un documento dado. Esta asunción no se podría admitir si en el corpus de entrenamiento C se tuvieran documentos repetidos. De este modo, la condición que debe imponerse es la siguiente:

$$\{P(\vec{d_j}) = Cte \ , \ \forall j \mid \vec{d_j} \neq \vec{d_{j'}} \ \forall j \neq j' \ , \text{ donde } d_j, d_{j'} \in C\} \tag{8.2}$$

En nuestras colecciones de referencia, BankSearch y WebKB, la condición 8.2 se cumple. Por otro lado, la probabilidad a priori de cada clase se expresa por medio de la cantidad:

$$P(c_k) = \frac{dim(c_k)}{dim(C)} \tag{8.3}$$

donde $dim(c_k)$ es el número de documentos etiquetados como pertenecientes a la clase k, y $dim(C)$ es el número total de documentos en el corpus C. $P(c_k)$ podría considerarse también constante si el número de documentos en cada clase dentro del corpus de referencia es igual. Así,

$$\{P(c_k) = Cte\ ,\ \forall k \mid dim(c_k) = dim(c_{k'})\} \tag{8.4}$$

La asunción del principio de independencia entre rasgos implica que la probabilidad de un documento dada una clase, $P(\vec{d_j} \mid c_k)$, sea el producto de las probabilidades condicionada de cada rasgo presente en el documento.

$$P(\vec{d_j} \mid c_k) = \prod_{i=1}^{N_j} P(\vec{t_i} \mid c_k) \tag{8.5}$$

donde N_j es el número de rasgos presentes en d_j.

Llegados a este punto, puede verse que la diferencia entre dos algoritmos NB vendrá dada fundamentalmente por la diferente función que se emplee para la estimación de la probabilidad de un rasgo a una clase: $P(\vec{t_i} \mid c_j)$.

En (McCallum, 1999) se presenta un clasificador NB que permite realizar una clasificación de los documentos multiclase y multietiqueta. Por ejemplo, dadas las clases: Norteamérica, Sudamérica, Europa, Asia y Australia, un documento acerca de las tropas estadounidenses en Bosnia podía ser etiquetado como perteneciente a las clases Norteamérica y Europa. En (Yang, 1996) y (John et al., 1997) se estudian varias estrategias de muestreo del conjunto de entrenamiento para tratar de mejorar el aprendizaje basado en modelos estadísticos sobre textos. Dependiendo del modelo utilizado para el cálculo de las probabilidades condicionadas tendremos distintos tipos de clasificadores (McCallum y Nigam, 1998): Bernuilli, Multinomiales (que emplea coeficientes multinomial), etc.

Por otra parte, los experimentos sugieren que la aplicación de métodos de reducción de rasgos produce una mejora significativa para los clasificadores binarios y una mejora moderada para los clasificadores multinomiales [(Koller y Sahami, 1996), (Yang y Pedersen, 1997), (Chakrabarti et al., 1997), (Chakrabarti et al., 1998a) y (McCallum y Nigam, 1998)]. En (Yang, 1999) se presenta una evaluación de varias aproximaciones estadísticas para la clasificación de textos.

Entonces, asumiendo la condición expresada en la ecuación 8.2, un clasificador NB asignará a

un documento d_j aquella clase c_k que cumpla:

$$c_k = argmax_{\forall k}\{P(c_k \mid \vec{d_j})\} = argmax_{\forall k}\{P(c_k)\prod_{i=1}^{N_j} P(\vec{t_i} \mid c_k)\} \tag{8.6}$$

Por motivos computacionales, se aplican logaritmos para evitar productos sobre valores muy cercanos a cero. Como toda función logarítmica es una función monótona creciente, la clase que maximice el producto de probabilidades condicionadas $P(\vec{t_i} \mid c_j)$ será la misma que maximice la suma de sus logaritmos. Así,

$$c_k = argmax_{\forall j}\{P(c_k \mid \vec{d_j})\} = argmax_{\forall k}\{log(P(c_k)) + \sum_{i=1}^{N_j} log(P(\vec{t_i} \mid c_k)\} \tag{8.7}$$

En general, sin considerar ni $P(c_k) = Cte$, ni $P(\vec{d_j}) = Cte$, la asignación que hace todo algoritmo NB de la probabilidad de una clase dado un documento sigue la expresión:

$$c_k = argmax_{\forall k}\{log(P(c_k)) + \sum_{i=1}^{N_j} log(P(\vec{t_i} \mid c_k) - log(P(\vec{d_j}))\} \tag{8.8}$$

En esta tesis doctoral se evalúa la calidad de las representaciones de documentos HTML propuestas por medio de un clasificador automático NB con aprendizaje supervisado y se experimenta con diferentes funciones para el cálculo de la probabilidad $P(\vec{t_i} \mid c_j)$.

Dependiendo de cómo sean estas funciones de probabilidad y cuáles sean los parámetros de cada una de ellas, deben definirse diferentes aprendizajes. El conjunto de funciones $P(\vec{t_i} \mid c_j)$, y sus correspondientes aprendizajes, con las que se realiza la experimentación en TC son:

- **Funciones gaussianas**:
  - Función de probabilidad "Normal" con aprendizaje por estimación de máxima verosimilitud.
  - Función de probabilidad "Normal ponderada" con aprendizaje por estimación de máxima verosimilitud.
  - Función de probabilidad "LogNormal" con aprendizaje por estimación de máxima verosimilitud.
- **Funciones basadas en eventos**:
  - Función de probabilidad "*Multinomial*" con aprendizaje *multinomial*.

A continuación, se describe cada una de estas funciones.

8.6.1. Funciones Gaussianas

Como funciones gaussianas se consideran todas aquellas funciones de probabilidad basadas en distribuciones Normales $N(\mu, \sigma)$. En nuestro caso, tanto la función de probabilidad Normal, como la Normal ponderada y la LogNormal entrarían dentro de esta categoría.

Toda función Normal está parametrizada con las variables μ (media) y σ (desviación típica), de forma que el aprendizaje que se realice deberá ser un proceso en el que, a partir de un conjunto de documentos preclasificados como pertenecientes a cada una de las clases que se vayan a considerar, se estime el valor de estos parámetros. El aprendizaje que se va a realizar para la función Normal será el mismo que el que se realice para las funciones Normal ponderada y LogNormal. En los tres casos, y en toda la experimentación de esta tesis, se realizará un aprendizaje gaussiano por estimación de máxima verosimilitud que se detalla a continuación.

Aprendizaje por estimación de máxima verosimilitud

Antes de definir un descriptor de clase para las funciones gaussianas es necesario hacer una asunción basada en el *Teorema Central del Límite*.

Teorema 1 *Si X_1, X_2, ... , X_n son variables aleatorias (discretas o continuas) independientes, con idéntico modelo de probabilidad, de valor medio μ y varianza σ, entonces la distribución de la variable:*

$$Z = \frac{(X_1 + X_2 + \ldots + X_n) - n\mu}{\sigma\sqrt{n}} = \frac{(\sum_{i=1}^{n} X_i) - n\mu}{\sigma\sqrt{n}} \tag{8.9}$$

se aproxima a la de una variable normal $N(0,1)$, mejorándose la calidad de la aproximación a medida que aumenta n. Este resultado prueba que el estadístico, o estimador media muestral,

$$\bar{X} = \frac{\sum_{i=1}^{n} X_i}{n} \tag{8.10}$$

se distribuye aproximadamente como una variable $N(\mu, \frac{\sigma}{\sqrt{n}})$ o, de manera equivalente, que $\frac{\bar{X}-\mu}{\frac{\sigma}{\sqrt{n}}}$ se distribuye aproximadamente como una variable $N(0,1)$.

De este modo, si se selecciona una muestra aleatoria suficientemente grande de una distribución, entonces, aunque esta distribución no sea ni siquiera aproximadamente Normal, una consecuencia del teorema anterior es que muchas funciones importantes de observaciones muestrales tendrán distribuciones aproximadamente Normales. En particular, para una muestra aleatoria grande de cualquier distribución con varianza finita, la distribución de la media muestral será aproximadamente Normal.

Por el Teorema Central del Límite se podría asumir que teniendo un número suficientemente elevado de páginas web de ejemplo, representadas con cualquier función de ponderación, y

correspondientes a una determinada clase, entonces la relevancia relativa de un determinado rasgo en el conjunto de ejemplos de una determinada clase seguirá una distribución Normal.

Visto el enorme tamaño de la Web, este comportamiento puede aceptarse fácilmente; es decir, se podría considerar que las relevancias con las que se presenta un determinado rasgo en el conjunto de documentos HTML contenidos en la Web, y pertenecientes a una determinada categoría, seguirán una distribución $N(\mu, \sigma)$, con media μ y desviación típica σ. Por tanto, se puede considerar que el peso relativo de un rasgo dentro de un descriptor de clase seguirá una distribución Normal.

De esta forma, los descriptores de clase en el caso de un aprendizaje gaussiano por máxima verosimilitud estarán formados por un vector:

$$\vec{DC}_k : (\vec{t_i}, \mu_{ik}, \sigma_{ik}^2, N_{ik}) \tag{8.11}$$

donde la primera componente estará constituida por cada uno de los rasgos presentes en el vocabulario t_i y la segunda será el valor medio μ_{ik} de los pesos con los que se ha ponderado el rasgo t_i en los documentos etiquetados como de la clase c_k. La varianza σ_{ik}^2 representa la tercera componente, siendo la última un valor numérico correspondiente al número de ocurrencias utilizadas para el cálculo de estas variables estadísticas. Guardar el número de palabras con el que se han calculado los valores μ_{ik} y σ_{ik}^2 servirá para poder ampliar el aprendizaje en cualquier momento, es decir, poder tener un sistema de aprendizaje incremental, de forma que en cualquier momento se puedan introducir nuevas páginas de entrenamiento, sin más que ajustar los parámetros μ_{ik} y σ_{ik}^2 a unos nuevos valores que cubran todos los ejemplos.

Hay que hacer notar que con esta definición de descriptor de clase el número de ejemplos de entrenamiento deberá ser suficientemente grande. Cada rasgo dentro de una determinada clase deberá aparecer en un número muy elevado de páginas, de forma que las estimaciones de la media y la varianza resulten significativas. Dado que no siempre aparecen los mismos rasgos en distintas páginas de una misma clase, el número de páginas de entrenamiento necesario para cada clase crece enormemente. Una posible solución a este problema sería no considerar aquellos rasgos dentro del descriptor cuyas componentes μ y σ hayan sido estimadas con un número bajo de ejemplos, limitándose de esta forma los descriptores de clase y limitando posiblemente la tarea posterior de clasificación.

Una vez descrita la forma que debe tener un descriptor de clase para funciones de clasificación gaussianas, el siguiente paso es calcular los parámetros asociados a cada término μ_{ik} y σ_{ik}^2. Para ello se empleará el método de estimación paramétrica de máxima verosimilitud. Este método considera los parámetros a estimar como valores fijos, pero desconocidos, y maximiza la probabilidad de cubrir exactamente el conjunto de ejemplos dados. Para ello se asume la independencia entre clases, de forma que los datos contenidos en los ejemplos de una clase serán independientes de los datos contenidos en otra clase diferente. Esto implica la división

del problema en tantas partes como clases queramos determinar. En nuestro caso particular, el problema de estimar los parámetros de los rasgos pertenecientes a cada clase se divide en tantos problemas como rasgos diferentes aparezcan en el conjunto de ejemplos de entrenamiento correspondientes a dicha clase, debido al principio de independencia aceptado en fase de representación.

Una vez asumida la independencia entre clases y entre rasgos de una misma página, para cada rasgo diferente dentro de cada clase tendríamos las siguientes expresiones:

$$\mu_{ik} = \frac{1}{N_{ik}} \sum_{l=1}^{N_{ik}} r_{il} \tag{8.12}$$

donde N_{ik} representa el número de veces que ha aparecido el rasgo t_i en la clase c_k y

$$\sigma_{ik}^2 = \frac{1}{N_{ik}} \sum_{l=1}^{N_{ik}} (r_{ilk} - \mu_{ik})^2 \tag{8.13}$$

donde r_{ilk} representa la l-ocurrencia del rasgo t_i en un documento de la clase c_k.

De este modo, si un mismo rasgo está presente en documentos de varias clases, sus valores μ y σ^2 deberán ser recalculados con los ejemplos pertenecientes a cada clase. Si un rasgo t_i está presente en las clases c_1 y c_2, tendrá valores medios μ_{i1} para la clase c_1 y μ_{i2} para la c_2; del mismo modo, tendrá varianzas σ_{i1}^2 y σ_{i2}^2.

Una vez calculados los descriptores de clase, el siguiente paso es buscar un algoritmo de clasificación adecuado a la representación y al entrenamiento llevados a cabo. Asociado con este aprendizaje por estimación de máxima verosimilitud se han elegido tres funciones de probabilidad. Cualquier algoritmo de clasificación es muy dependiente de la fase de aprendizaje usada, tanto en su planteamiento teórico como en su realización práctica. En el caso de estos métodos basados en distribuciones Normales, el hecho de tener un número elevado de ejemplos resulta imprescindible para que los resultados puedan considerarse realmente significativos.

Clasificación con función de probabilidad Normal

Como clasificación con función Normal se entiende aquella que utiliza como función para la estimación de la probabilidad de un rasgo a una clase una función $\mathrm{N}(\mu_{ik},\sigma_{ik}^2)$. Así, la probabilidad a posteriori de un rasgo dada una clase, seguiría la expresión:

$$P(\vec{t_i} \mid c_k) = \frac{1}{\sqrt{2\pi\sigma_{ik}^2}} e^{\frac{(r_{ij}-\mu_{ik})^2}{\sigma_{ik}^2}} \tag{8.14}$$

donde r_{ij} representa el peso, o relevancia, del rasgo t_i en el documento d_j. Esta relevancia vendrá asignada por la función de ponderación o de proyección F, utilizada dentro de la definición de un modelo de representación.

Los valores μ_{ik} y σ_{ik} suponen el valor medio y la desviación típica de la relevancia asignada al rasgo t_i en todos los documentos de la clase c_k. El cálculo de estas cantidades sigue las expresiones 8.12 y 8.13.

Clasificación con función de probabilidad Normal ponderada

Una primera corrección a la clasificación con función Normal sería relativa a la inclusión en la función de una ponderación correspondiente a la relevancia del rasgo en la propia página. De este modo, cuando se está buscando la clasificación de una nueva página, un rasgo que obtuviera un peso mayor dentro de la representación de dicha página aportaría más a la probabilidad condicionada del rasgo a la clase que un rasgo con menor peso.

De este modo, la función de probabilidad tiene la expresión:

$$P(\vec{t_i} \mid c_k) = \frac{r_{ij}}{\sqrt{2\pi\sigma_{ik}^2}} e^{\frac{(r_{ij}-\mu_{ik})^2}{\sigma_{ik}^2}} \tag{8.15}$$

Clasificación con función de probabilidad LogNormal

Si en lugar de emplear una función Normal, se usa una función LogNormal, la función de probabilidad $P(\vec{t_i} \mid c_k)$ sería:

$$P(\vec{t_i} \mid c_k) = \frac{1}{r_{ij}\sigma_{ik}\sqrt{2\pi}} e^{\frac{(ln(r_{ij})-\mu_{ik})^2}{\sigma_{ik}}} \tag{8.16}$$

Una variable tiene distribución LogNormal si el logaritmo natural de la variable se distribuye normalmente, es decir, la distribución LogNormal se obtiene cuando los logaritmos de una variable se describen mediante una distribución Normal. Esta función está caracterizada por los parámetros μ y σ, aunque estos no representan ni su valor medio, ni su desviación estándar.

Esta distribución suele utilizarse cuando las desviaciones a partir del valor de un modelo están formadas por factores, proporciones o porcentajes más que por valores absolutos (como es el caso de la distribución Normal). Se ha incluido esta función en la experimentación con el propósito de ver si la relevancia de un rasgo en un documento y en una clase tiene este comportamiento.

8.6.2. Funciones basadas en eventos

En (Mitchell, 1997) se presenta la función *m-estimate* como una función clásica para el cálculo de probabilidad a posteriori de un rasgo a una clase y que está basada en las frecuencias de aparición de un rasgo en un documento y corregida con la frecuencia en la colección.

Esta función supone una estimación que trata de eliminar la infravaloración que aportaría a un rasgo una función de probabilidad del tipo casos favorables entre casos totales. Además,

si esta función tomara un valor tendente a cero, haría que la estimación de probabilidad del clasificador lo hiciera también (Mitchell, 1997). La función *m-estimate* ha sido empleada en muchos de trabajos en el campo de la TC [(Merkl, 1997), (McCallum y Nigam, 1998), (Lewis, 1990b), (Lewis y Ringuette, 1994), (Craven y Slattery, 2001), (Bennett et al., 2002) y (Bennett et al., 2005)], y considera únicamente la frecuencia de un rasgo en un documento y en el conjunto de la colección. Así:

$$P(\vec{t_i} \mid c_k) = \frac{1 + \sum_{d \in c_j} f_{ij}}{|V| + \sum_{t \in V} \sum_{d \in c_j} f_{ij}} \tag{8.17}$$

donde f_{ij} representa la frecuencia de t_i en d_j.

Aprendizaje Multinomial

El aprendizaje, en este caso, se reduce a registrar la frecuencia de cada rasgo en cada documento, en cada clase y en el total de la colección. Esta información suele guardarse en ficheros invertidos.

Clasificación Multinomial

A partir de la función *m-estimate* se puede definir otra función de probabilidad que sigue una distribución multinomial y supone también una función clásica que ha sido empleada en multitud de trabajos de TC como (McCallum y Nigam, 1998), (Yang, 1999) y (Yang, 2001). En una clasificación multinomial, la probabilidad condicionada de un rasgo a una clase viene dada por:

$$P(\vec{t_i} \mid c_k) = \prod_{i \in d_j} P(\vec{t_i} \mid c_j)^{f_{ij}} = \prod_{i \in d_j} \left(\frac{1 + \sum_{d \in c_j} f_{ij}}{|V| + \sum_{t \in V} \sum_{d \in c_j} f_{ij}}\right)^{f_{ij}} \tag{8.18}$$

donde $|V|$ es la dimensión del vocabulario.

El aprendizaje, como en el caso de la función *m-estimate*, se reduce a registrar la frecuencia de cada rasgo en cada documento, cada clase y en el total de la colección.

8.7. Funciones de evaluación

En la fase de evaluación de cualquier proceso de clasificación, el sistema se prueba con un conjunto de páginas de las que se conoce su clase. Dependiendo de la predicción del propio sistema, se puede construir la tabla de contingencia (figura 8.1), formada por:

- **Verdaderos Positivos**: cuando el sistema clasifica un documento como perteneciente a una clase a la que sí pertenece. Cantidad '**a**' en la tabla 8.1.

	Datos Positivos	**Datos Negativos**
Asignaciones Positivas	a	b
Asignaciones Negativas	c	d

Tabla 8.1: Tabla de contingencia

- **Verdaderos Negativos**: cuando el sistema clasifica un documento como perteneciente a una clase a la que no pertenece. Cantidad '**b**' en la tabla 8.1.

- **Falsos Positivos**: cuando el sistema no clasifica un documento como perteneciente a una clase a la que sí pertenece. Cantidad '**c**' en la tabla 8.1.

- **Falsos Negativos**: cuando el sistema no clasifica un documento como perteneciente a una clase a la que no pertenece. Cantidad '**d**' en la tabla 8.1.

Así, a partir de una tabla de contingencia se pueden definir las siguientes funciones de evaluación para todo sistema de clasificación:

$$\text{Precisión, } P_k = a_k/(a_k + b_k) \tag{8.19}$$

Dada una clase k, la **precisión** (*precision*) representa la fracción de asignaciones correctas frente al total de asignaciones positivas realizadas para esa clase.

$$\text{Cobertura, } R_k = a_k/(a_k + c_k) \tag{8.20}$$

La **cobertura** (*recall*) representa la fracción de asignaciones positivas respecto al conjunto real de elementos pertenecientes a la clase k que se esté considerando.

A partir de la *precisión* y la *cobertura* se puede definir otra cantidad que las combina, la medida-F, y que viene dada por la expresión:

$$\text{medida-F } = \frac{2 \times R_k \times P_k}{(P_k + R_k)} \tag{8.21}$$

En ocasiones, en esta medida-F se introduce un coeficiente β que permite dar más peso a la precisión o a la cobertura. En esta expresión 8.21, con la que se evalúan las representaciones, se está tomando un valor $\beta = 1$, de forma que se consideran igualmente importantes.

Si se quiere evaluar el sistema globalmente, la medida-F deberá combinarse y esta combinación puede hacerse de dos formas:

1. Considerando aisladamente cada clase y haciendo después una media aritmética, o

2. Considerando las cantidades a, b, c y d del sistema globalmente y calculando posteriormente las funciones de evaluación.

En el primer caso, se tendrían evaluaciones por "micromedias", mientras que en el segundo caso, se estaría hablando de "macromedias".

De este modo, para la medida-F se definiría una medida-F micromedia (*Microaverage F-Measure*) como la media aritmética de los valores obtenidos en la medida-F para cada una de las clases consideradas,

$$\text{medida-F micromedia} = \frac{\sum_{c_k \in C} medida - F_k}{|C|} \tag{8.22}$$

donde $|C|$ representa el número de clases consideradas.

De un modo análogo, se define la medida-F macromedia (*Macroaverage F-Measure*) como el valor de la medida-F considerando los valores de a, b, c y d en el total de clases consideradas y que viene dada por la expresión:

$$\text{medida-F macromedia} = \frac{2 \times \sum_{c_k \in C} R_k \times \sum_{c_k \in C} P_k}{(\sum_{c_k \in C} P_k + \sum_{c_k \in C} R_k)} \tag{8.23}$$

8.8. Resultados experimentales

En esta tesis, la evaluación de la clasificación estadística se ha llevado a cabo por medio de la medida-F micromedia y macromedia, considerando para ambos cálculos una función de medida-F con $\beta = 1$ (expresión 8.21). Dado que los resultados obtenidos ofrecieron valores muy parecidos para los casos de micromedia y macromedia, se ha decidido mostrar únicamente los valores de la medida-F macromedia para facilitar el análisis de resultados.

Por otro lado, dada la gran cantidad de experimentos realizados –con un total de 9 subcolecciones, 4 funciones probabilidad aplicadas a cada subcolección y dos métodos diferentes de reducción de rasgos–, y con el fin de evitar que un número excesivo de tablas pudiera dificultar el análisis de los resultados, en este apartado se mostrarán comentados únicamente los resultados correspondientes a dos experimentos por subcolección; en la elección se tuvieron en cuenta dos factores.

En primer lugar, siempre se trata de seleccionar los resultados correspondientes a la mejor función de probabilidad $P(\vec{t_i}|c_k)$ encontrada en cada subcolección, entendiendo como mejor función aquella que logre los mayores valores absolutos en la medida-F; es decir, que si para una subcolección se muestra una gráfica correspondiente a la función de probabilidad Normal ponderada, por poner un ejemplo, es porque los mejores valores de medida-F se obtuvieron con dicha función.

Además, en esta selección se trata también de que queden representadas todas las funciones de probabilidad y de reducción a lo largo del análisis de las diferentes subcolecciones, de forma que puedan extraerse conclusiones generales sobre el comportamiento de cada función de probabilidad y reducción de rasgos con respecto a cada una de las funciones de proyección

evaluadas.

Los resultados experimentales se van a clasificar en función de la colección considerada, *BankSearch* y *WebKB*, así como de cada una de las subcolecciones seleccionadas en cada caso. Primero se comentan los resultados obtenidos para cada función de ponderación y probabilidad considerada, y al final del capítulo se extraen las conclusiones generales.

En las figuras en las que se presentan los resultados de clasificación se representan dos magnitudes: la "dimensión de las representaciones" y la "medida-F".

El eje X se corresponde con la dimensión de las representaciones, es decir, el tamaño del vocabulario con el que se genera cada representación. De este modo es posible analizar el comportamiento de las diferentes funciones de ponderación en relación con el tamaño de los vocabularios. Nótese que en los casos en los que se utiliza la reducción con la *propia función de ponderación* (reducción PF), las dimensiones son diferentes para cada función de proyección, mientras que cuando se emplea la reducción *term-frequency/document-frequency* (reducción MinMax), todas las funciones presentan valores de medida-F en las mismas dimensiones de vocabulario. Esto se debe a que al usar la reducción PF no tienen por qué generarse vocabularios iguales cuando se toman, por ejemplo, los dos rasgos más relevantes de cada página con la función TF-IDF o con FCC.

Por otro lado, con las dos funciones de reducción (PF y MinMax) se ha tratado de cubrir un rango que fuera desde una dimensión mínima hasta un tamaño un orden de magnitud menor que la dimensión del vocabulario sin reducir.

En el eje Y se representa la medida-F, una función con recorrido en el intervalo $[0, 1]$, de forma que valores elevados en esta cantidad suponen buenos resultados de clasificación.

8.8.1. Colección BankSearch

- Clasificación binaria:

  - **GH_700**: Clasificación binaria en el **nivel más bajo de la jerarquía** y entre **clases cercanas semánticamente**, pertenecientes a la clase "Ciencia". Para mostrar el comportamiento de las diferentes funciones de ponderación con esta colección GH_700 se han elegido los dos experimentos con los que se obtuvieron los mejores valores de medida-F, correspondientes con la función Normal ponderada con reducción PF (figura 8.1) y con la función LogNormal con reducción PF (figura 8.2).

    En la figura 8.1 se muestran los resultados de clasificación obtenidos con una función de probabilidad $P(\vec{t_i}|c_j)$ Normal ponderada y una función de reducción PF. El comportamiento más destacado lo presenta la función Title, que obtiene un valor de medida-F muy alto en todas las dimensiones para las que es posible generar vocabularios aplicándole una reducción PF. Con esta reducción, el hecho de que el

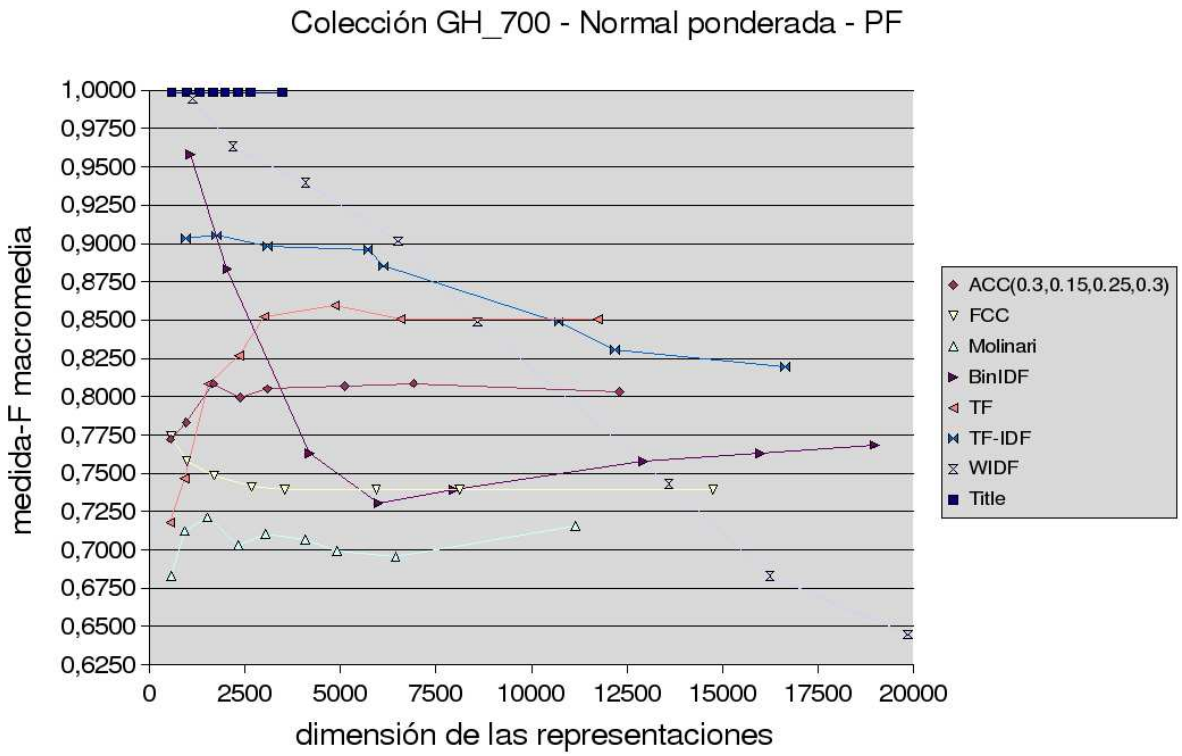

Figura 8.1: Clasificación binaria (superclases) con función Normal ponderada y reducción PF.

contenido de los elementos <title> en cada página no suela ser muy extenso, añadido a que muchas de las páginas no tienen contenido dentro de este elemento, hace que no puedan generarse vocabularios de gran tamaño.

También son destacable los resultados obtenidos por las funciones WIDF, TF-IDF y BinIDF en dimensiones reducidas del vocabulario. Este hecho parece indicar que la reducción PF se comporta bien a dimensiones de representación bajas. La función TF es la siguiente considerando la calidad de los resultados obtenidos, siendo las otras tres funciones de ponderación que emplean información del etiquetado HTML –ACC, FCC y Molinari– las que ofrecen los resultados más discretos en términos generales. De estos datos cabe destacar el hecho de que, en este caso, las funciones de ponderación globales han presentado todas ellas un comportamiento mucho mejor cuando las dimensiones de representación eran muy bajas. Por el contrario, todas las funciones de proyección locales, a excepción de FCC y Title, presentan una clara disminución en su valor de medida-F cuando las dimensiones de representación se reducen mucho. Esto hace pensar que a esas dimensiones reducidas, la pérdida de información en estas representaciones es significativa. Los resultados que se obtuvieron con una función Normal sin ponderación y con la misma reducción PF fueron muy parecidos a los presentados en la figura 8.1.

En la figura 8.2 se muestran los resultados de clasificación obtenidos con una función

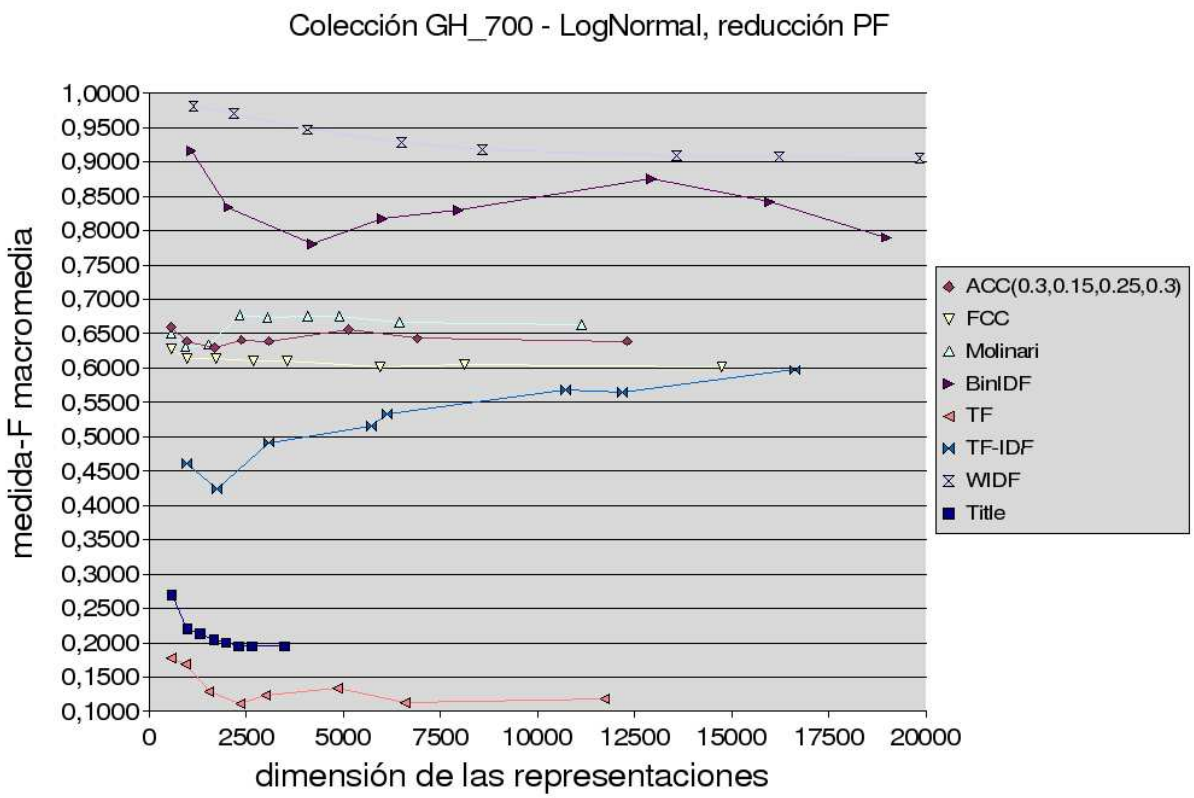

Figura 8.2: Clasificación binaria (superclases) con función LogNormal y reducción PF.

de probabilidad $P(\vec{t_i}|c_j)$ LogNormal y una función de reducción PF. En este caso, se observa que en todas las funciones de proyección la medida-F aumenta para las dimensiones de representación pequeñas. Sin embargo, en este caso, las funciones TF-IDF, Title y TF obtienen resultados bajos, mientras que WIDF y BinIDF obtienen los mejores resultados en términos de medida-F. La representación Molinari presenta un comportamiento muy similar al obtenido con la función Normal ponderada, mientras que con ACC y FCC disminuye la calidad de la clasificación.

- **G&J_700**. Clasificación binaria en el **nivel más bajo de la jerarquía** y entre **clases lejanas semánticamente**. Para evaluar el comportamiento de las diferentes representaciones con esta colección G&J_700 se vuelven a mostrar los experimentos correspondientes a la clasificación con función Normal ponderada y reducción PF (figura 8.3), y a la clasificación con función LogNormal y reducción PF (figura 8.4), por resultar estos los experimentos con mejores valores de medida-F dentro de esta subcolección. Además, de este modo es posible comprobar si el hecho de que se tomen clases cercanas o lejanas semánticamente puede influir sustancialmente en el resultado de una clasificación cuando se emplean las funciones de probabilidad Normal ponderada y LogNormal. De este modo, los resultados mostrados en las figuras 8.1 y

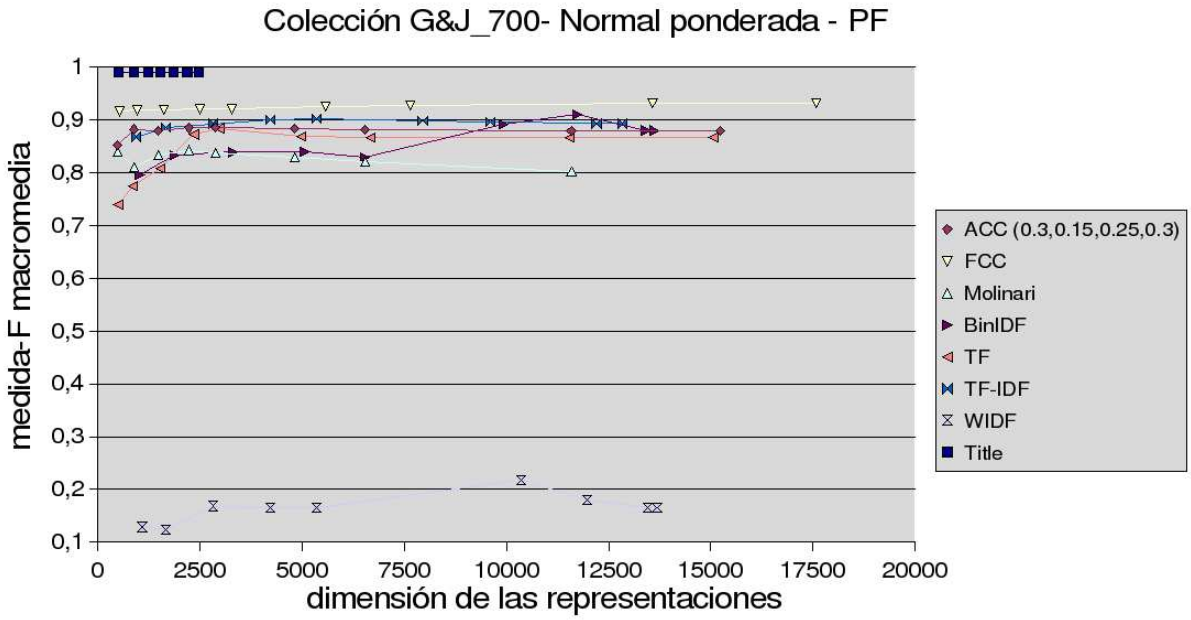

Figura 8.3: Clasificación binaria (superclases) con función Normal ponderada y reducción PF.

8.3 y 8.2 y 8.4 pueden compararse en términos de cercanía y lejanía semántica entre clases en un problema de clasificación binaria.

En la figura 8.3 se muestran los resultados obtenidos con una función Normal ponderada y una función de reducción PF. En este caso, el comportamiento de todas las funciones es muy diferente a la clasificación con clases cercanas semánticamente, salvo en el caso de Title que tiene un comportamiento idéntico. Llama la atención que la función WIDF, que antes ofrecía el mejor comportamiento entre clases semánticamente cercanas, obtenga ahora unos resultados tan bajos en el caso de realizar la clasificación con clase lejanas. Por el contrario, el resto de funciones de proyección presentan valores de medida-F bastante mejores que en el caso anterior, destacando especialmente la representación FCC, que obtiene una buena calidad de clasificación para todas las dimensiones consideradas y manteniendo sus valores de medida-F tanto en dimensiones altas como bajas. Por detrás de la FCC y con comportamientos muy parecidos entre sí, se encuentran las funciones ACC, TF-IDF y TF, quedando un poco más distanciadas BinIDF y Molinari.

Para esta misma colección G&J_700, cuando se empleó una función Normal sin ponderación con reducción PF, los resultados de clasificación obtenidos fueron muy parecidos a los de la figura 8.3, volviendo a ser Title la función que obtuvo los mejores resultados. En ambos experimentos se pone de manifiesto un comportamiento destacable de FCC, donde los valores de medida-F se mantienen estables en todo el rango de dimensiones del vector de representación, obteniendo valores elevados, en

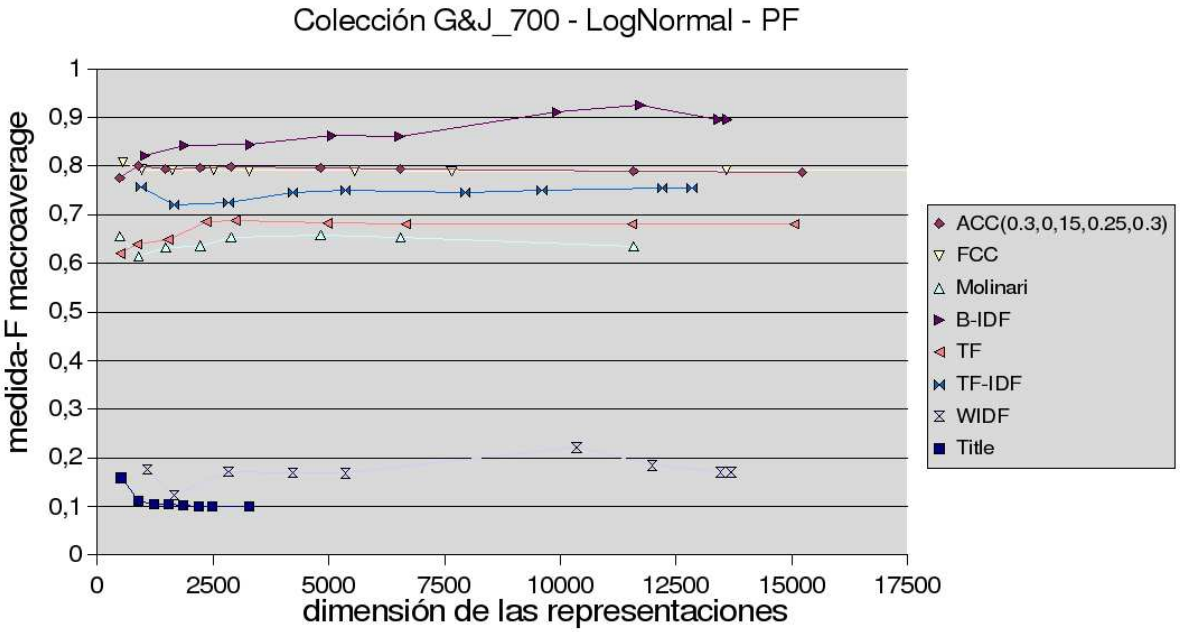

Figura 8.4: Clasificación binaria (superclases) con función LogNormal y reducción PF.

ambos casos, para el caso de la dimensión del vocabulario más reducida.

En la figura 8.4 se muestra un buen comportamiento de la función BinIDF, con valores parecidos a los que se mostraban con clases cercanas semánticamente. En este caso destaca la subida experimentada por las funciones FCC y ACC, que presentaban valores cercanos a 0,65 con la colección GH_700 y que ahora, con dos clases más separadas, obtienen valores de medida-F en torno a 0,8. De este modo, resultan las dos mejores funciones tras BinIDF, y seguidas después por TF-IDF, TF y Molinari. Como en el otro caso considerado en esta colección, la función WIDF obtiene resultados muy bajos, al igual que Title, que repite el mal comportamiento obtenido con la función LogNormal en el caso de la colección GH_700.

- **GJ_700**: **Clasificación binaria** en el **nivel más alto de la jerarquía** y entre **clases lejanas semánticamente**. En este caso, se muestran los experimentos correspondientes a una clasificación Multinomial con reducción PF (figura 8.5) y una clasificación con función Normal sin ponderación y reducción PF (figura 8.6).

  En la figura 8.5 se muestra que los resultados obtenidos en clasificación con las funciones WIDF, FCC y ACC resultan ser los mejores. Además, obtienen valores de medida-F bastante altos. Se mantienen, por tanto, unos buenos resultados de clasificación para FCC y ACC con clases separadas. Esta clasificación se realiza entre clases lejanas, pero con una colección de mayor tamaño que en el caso anterior

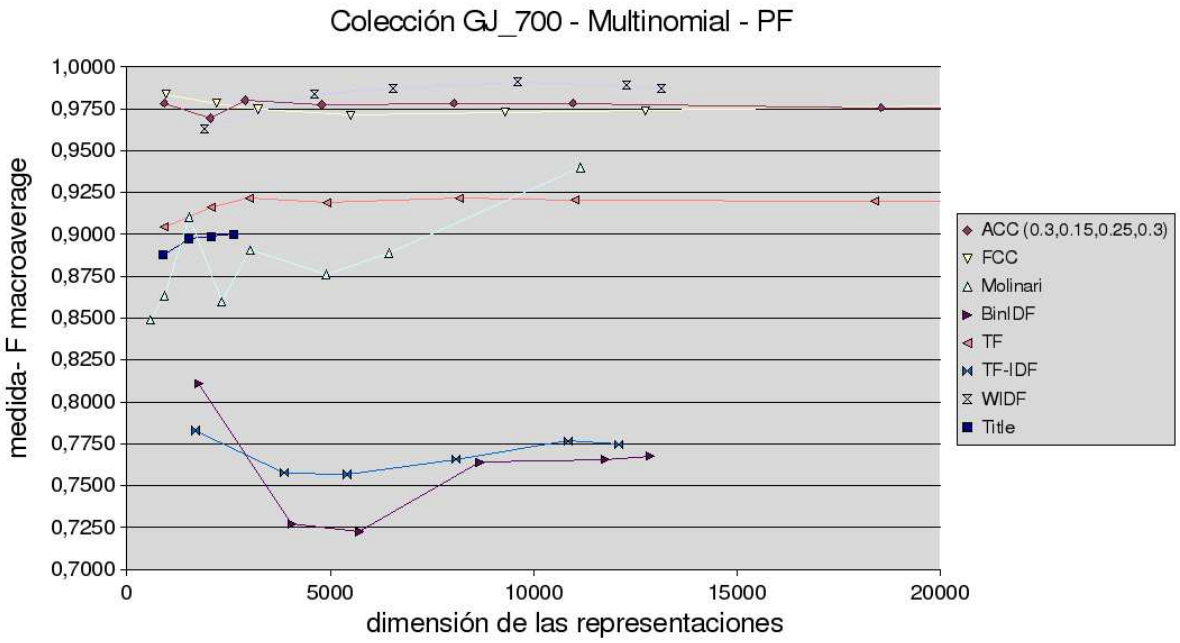

Figura 8.5: Clasificación binaria (superclases) con función Multinomial y reducción PF.

G&J_700, dado que GJ_700 se enmarca en un nivel superior en la jerarquía de la colección y estaría formada por cuatro clases del nivel inferior. Hay que destacar que en este caso la función WIDF vuelve a tomar valores competitivos, tras los malos resultados que ofreció para la otra colección de clases lejanas. Las funciones TF y Molinari obtienen peores resultados que WIDF, FCC y ACC, pero son las funciones TF-IDF y BinIDF las peores situadas, aunque en niveles de medida-F aceptables.

En el caso de la función Normal sin ponderación (figura 8.6), el resultado es muy similar al obtenido con la otra colección de clases separadas (figura 8.3). La función con mejor comportamiento vuelve a ser la FCC seguida por la ACC. A continuación se encuentran la BinIDF, TF-IDF, TF, Title y Molinari. Para el caso de la función WIDF, vuelve a ser la función con peor comportamiento, aunque a dimensiones reducidas mejore a Molinari y TF.

- **ABC&DEF_700**: **Clasificación binaria** en el **nivel más alto de la jerarquía** y entre **clases lejanas semánticamente**: "Bancos y Finanzas" y "Lenguajes de programación". Esta clasificación se realiza de nuevo al nivel más alto de la jerarquía que constituye la colección *BankSearch*, pero en este caso con 2 superclases que contienen 6 clases del nivel de jerarquía bajo. Para evaluar el comportamiento de las funciones de proyección se muestra una clasificación Multinomial con reducción MinMax (figura 8.7) y una función LogNormal también con reducción MinMax (figura 8.8).

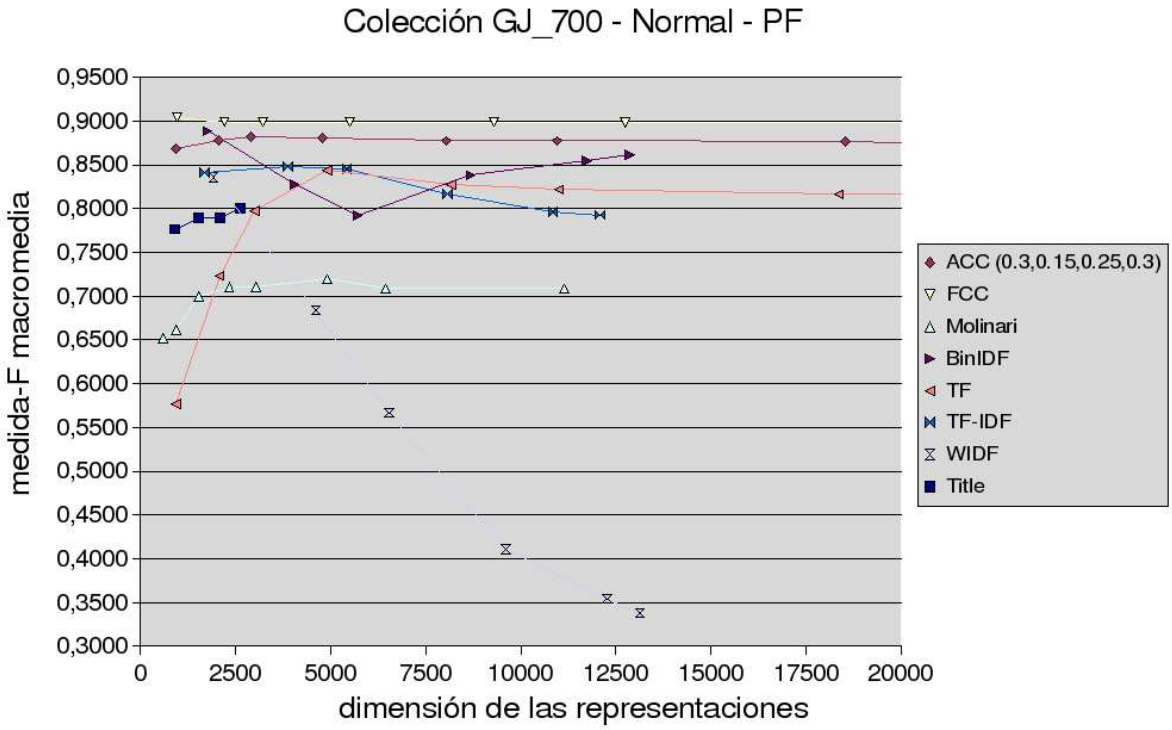

Figura 8.6: Clasificación binaria (superclases) con función Normal y reducción PF.

La primera conclusión que se puede extraer tras la aplicación de la reducción MinMax con estas funciones de probabilidad es que se trata de una reducción que se comporta mejor con una la función Multinomial que con la LogNormal, donde los resultados, a excepción de WIDF y BinIDF, son muy bajos para el resto de funciones de ponderación (figura 8.8). Puede observarse que la función LogNormal con reducción MinMax se comporta de forma muy diferente para unas funciones y otras. En el caso de la clasificación multinomial se observan unos valores bajos para el caso de TF-IDF y BinIDF, mientras que WIDF, ACC, Molinari, FCC y TF obtienen buenos valores de medida-F en torno a 0,95 (figura 8.7). El comportamiento de la función TF-IDF no es nada bueno en este caso.

- Clasificación a 3 clases.

  - **ABC_700**: **Clasificación a 3 clases** en el **nivel más bajo de la jerarquía** y entre **clases cercanas semánticamente**. Para esta subcolección, las funciones de probabilidad con las que se obtenían los valores de medida-F más altos fueron la Multinomial con reducción PF (figura 8.9) y la clasificación con función Normal ponderada y reducción PF (figura 8.10).

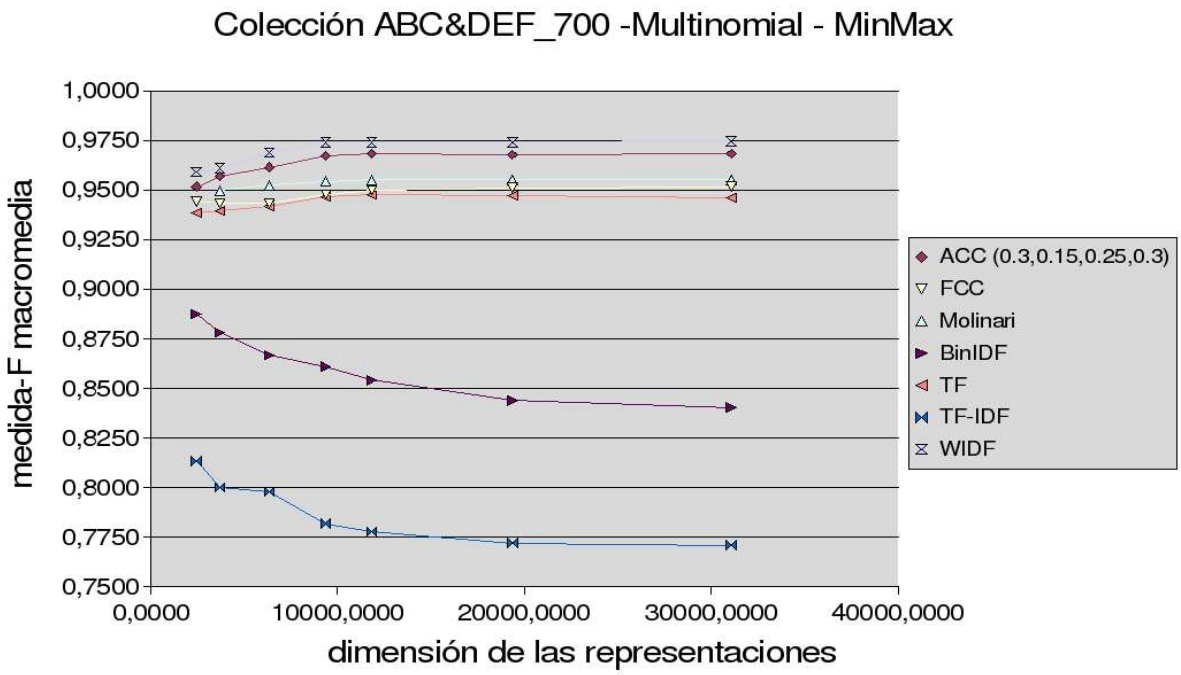

Figura 8.7: Clasificación binaria (superclases) con función Multinomial y reducción MinMax.

Lo primero que se observa al salir de la clasificación binaria y aumentar el número de clases es que los valores de medida-F obtenidos con el clasificador *Naïve Bayes*, independientemente de la función probabilidad considerada, disminuyen mucho. En el caso de la clasificación Multinomial con reducción PF (figura 8.9) los mejores resultados se obtiene con las funciones ACC, WIDF, Molinari, FCC y Title, mientras que las funciones TF-IDF, TF y BinIDF obtienen unos valores muy bajos.

En el caso de emplear la función Normal ponderada (figura 8.10) la mejor representación se obtiene de nuevo con la función Title, seguida de la función de ponderación WIDF, que vuelve destacar cuando se trata de una clasificación entre clases cercanas. El resto de funciones presentan un comportamiento más discreto, pudiéndose destacar el aumento de la medida-F para ACC y FCC en dimensiones de representación más pequeñas, donde el resto empeora.

- Clasificación a 6 clases

  - **ABC&DEF_700**: **Clasificación a 6 clases** en el **nivel más bajo de la jerarquía** y distinguiendo entre **clases cercanas y lejanas semánticamente**, ya que pertenecen 3 a 3 a las categorías "Bancos & Finanzas" y "Lenguajes de programación". En este caso, los mejores resultados se obtuvieron con las dos clasificaciones Multinomiales. Este hecho permite observar cómo se comporta esta

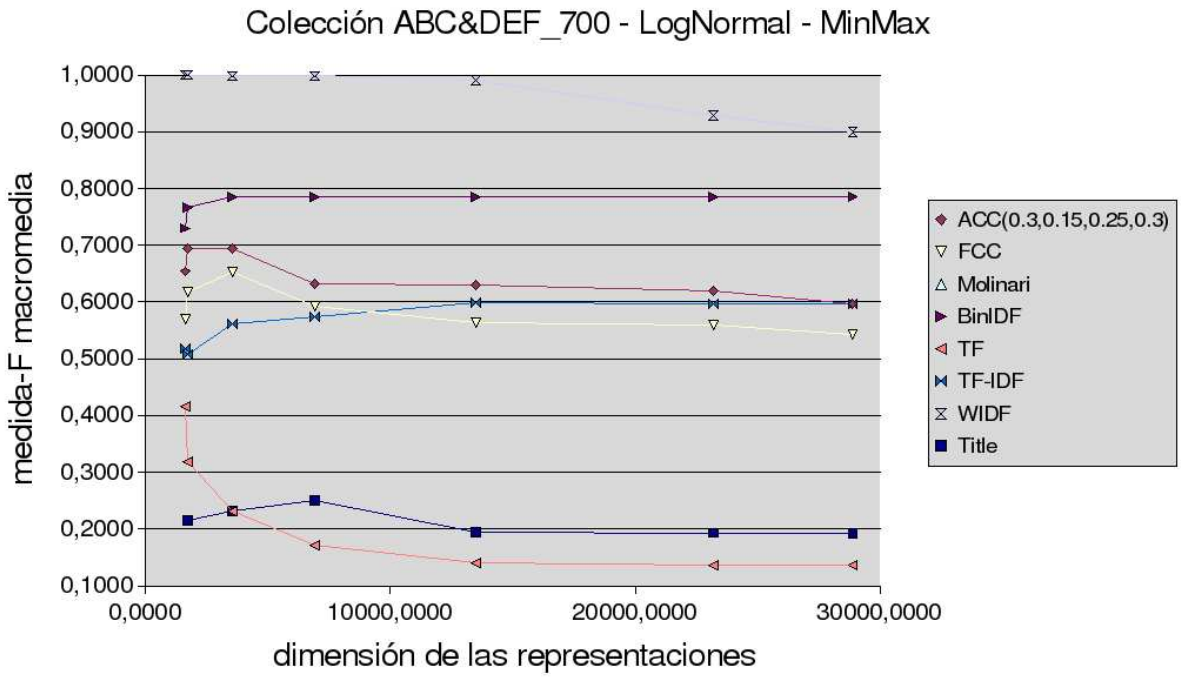

Figura 8.8: Clasificación binaria (superclases) con función LogNormal y reducción MinMax.

función de probabilidad en relación al tipo de reducción de rasgos empleado.

Las figuras 8.11 y 8.12 muestran los resultados de medida-F para una clasificación a 6 clases y con los dos tipos de reducciones consideradas. El comportamiento de las funciones de proyección es muy similar en ambos casos, con valores de medida-F muy parecidos, salvo en el caso de las funciones TF y BinIDF. Es lógico que la función TF se comporte peor con una reducción PF que con la MinMax, ya que la reducción PF selecciona los rasgos más ponderados de cada página y en este caso serían los rasgos más frecuentes, algo que introduce ruido dado que podrían seleccionarse rasgos de uso muy común que no sean verdaderamente discriminantes del contenido de una página o de la clase a la que pertenecen. Por el contrario, para funciones como ACC, FCC y Molinari es de suponer que esta reducción PF sea más adecuada, ya que en estos casos los rasgos más ponderados por cada función representan los rasgos más relevantes. En el caso de la función BinIDF, cuando se reduce mucho la dimensión de las representaciones con la reducción PF el comportamiento mejora sustancialmente. Este mismo comportamiento se ha puesto de manifiesto en casi todas las clasificaciones donde se ha empleado este tipo de reducción.

- Clasificación a 10 clases

  - **AJ_700**: **Clasificación a 10 clases** en el **nivel más bajo de la jerarquía** y entre

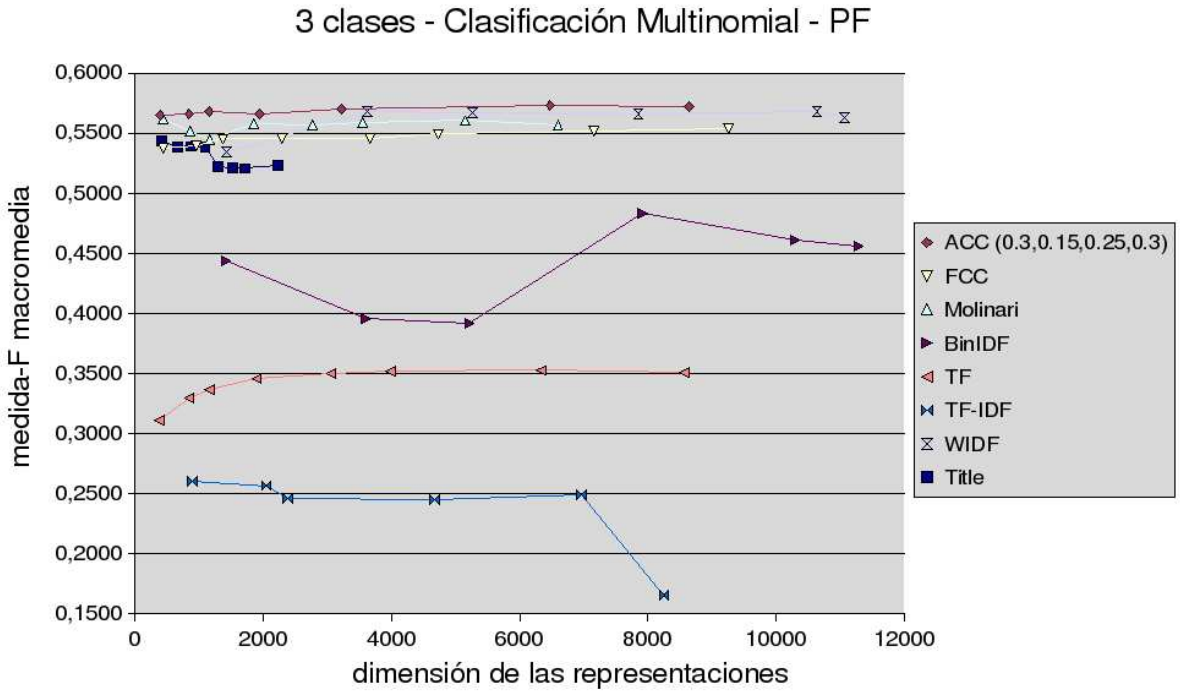

Figura 8.9: Clasificación Multinomial con reducción PF sobre la colección ABC_1000.

clases cercanas y lejanas semánticamente. Este es el ejemplo de clasificación más difícil de los que se han considerado y así los valores de medida-F, en términos generales, vuelven a ser muy bajos. Los mejores resultados se obtuvieron con la clasificación Multinomial y con una función LogNormal, aunque en este último caso sólo la función WIDF obtenía buenos resultados. Por este motivo, se han seleccionado los resultados obtenidos con una función Normal ponderada (figura 8.13) y la clasificación Multinomial (figura 8.14), en ambos casos empleándose una reducción MinMax.

Cuando se evalúa con la función Normal ponderada (figura 8.13), la función de ponderación más destacada es Title, seguida por BinIDF, FCC y ACC. En el caso de Title, la situación no cambia respecto a casos anteriores donde se empleaba la función Normal ponderada, y donde siempre obtiene los mejores resultados. Sin embargo, con el resto de funciones de probabilidad sus valores bajan aunque, en este caso, para la dimensión mínima del vocabulario obtiene el mejor valor de medida-F de todas las funciones evaluadas. En el caso de la clasificación Multinomial con reducción MinMax (figura 8.14), los valores de BinIDF son muy bajos en relación al resto, mientras que FCC y ACC aumentan su valor de la medida-F. Para dimensiones elevadas y con reducción PF, la función TF-IDF obtiene resultados comparables a los de BinIDF, FCC y ACC, mientras que con la reducción MinMax vuelven a ser muy bajos. En el

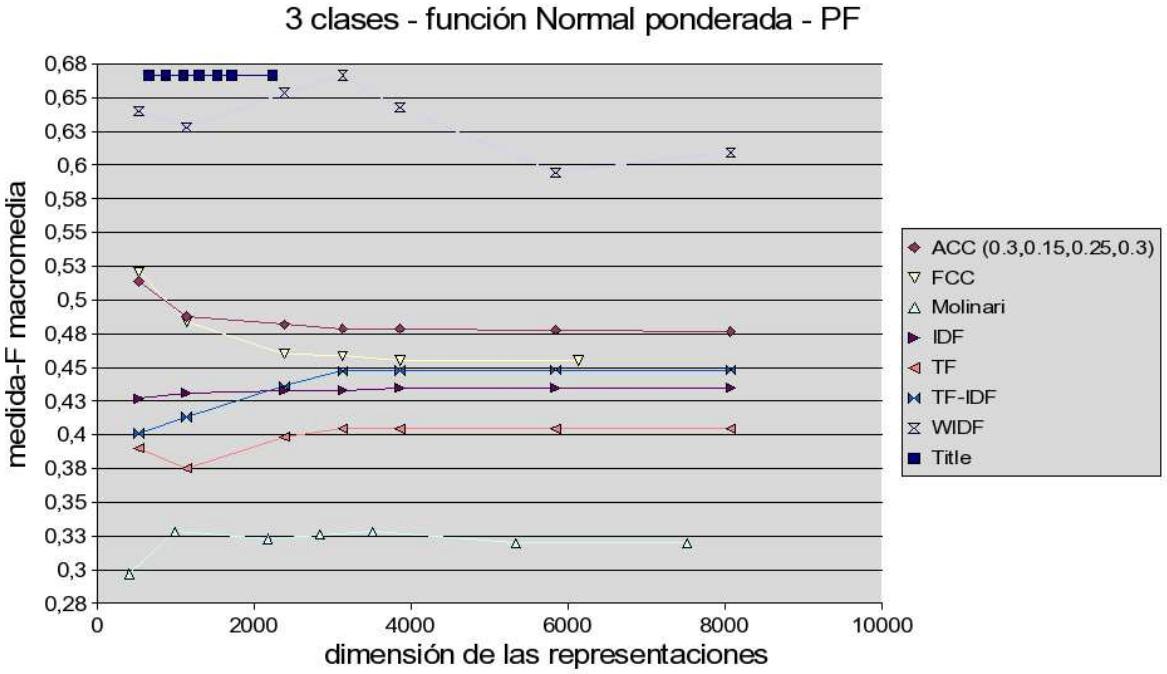

Figura 8.10: Clasificación con función Normal ponderada y reducción PF sobre la colección ABC_1000.

caso de la función WIDF, obtiene muy buenos resultados cuando se usa una reducción PF, mientras que son mucho peores en el caso de reducción MinMax.

8.8.2. Colección WebKB

- Clasificación a 6 clases

  - **WebKB_700**: **Clasificación a 6 clases cercanas semánticamente**. Con esta colección, los mejores resultados se obtuvieron con las clasificaciones Multinomial (figura 8.15) y Normal (figura 8.16), ambas con reducción PF. Como en el caso de la colección *BankSearch*, los resultados de la clasificación Normal son muy similares a los obtenidos con la función Normal ponderada, por lo que en este caso se ha seleccionado para su análisis la primera. En el caso de la clasificación con función LogNormal, volvieron a encontrarse los mejores resultados para la función WIDF, mientras que los valores de medida-F obtenidos por el resto de funciones fueron muy bajos.

    En primer lugar, hay que destacar que los valores de medida-F obtenidos con esta colección son similares a los encontrados cuando se realizó la clasificación a 6 clases con la subcolección ABC&DEF_700. En el caso de la clasificación Multinomial (figura 8.15), las representaciones propuestas en esta tesis, ACC y FCC, obtienen los mejores resultados, aunque con esta colección acusan más que con las otras la reducción

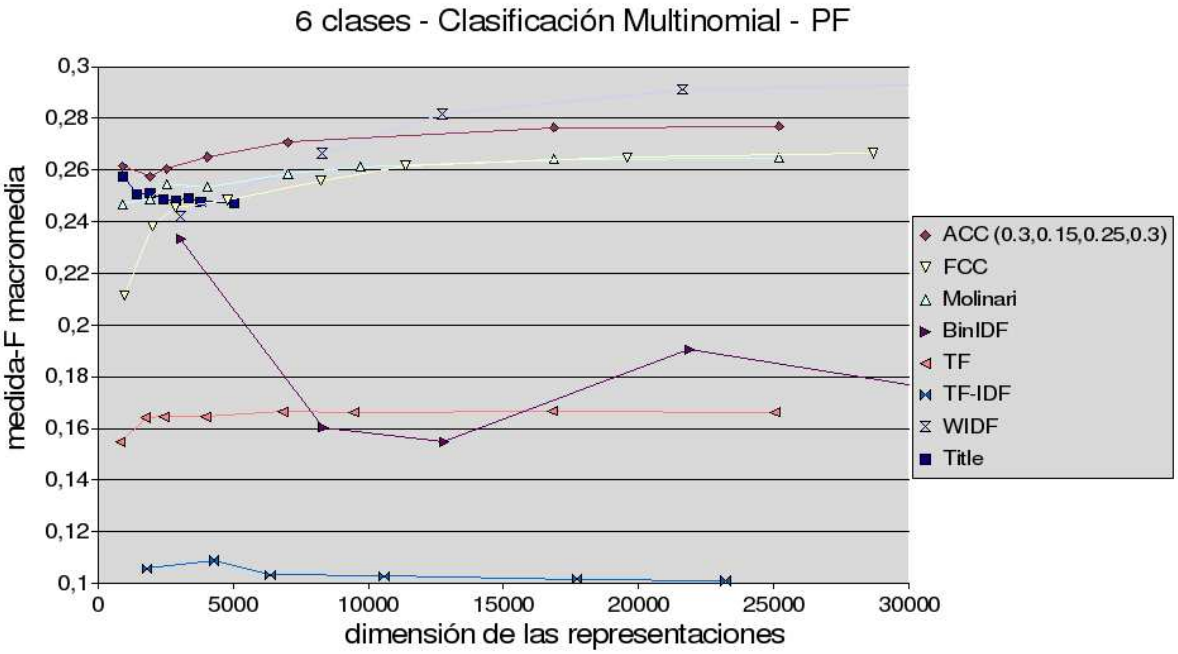

Figura 8.11: Clasificación Multinomial y reducción PF.

de las dimensiones de representación. Ya no mantienen el valor de la medida-F independientemente del tamaño de la representación. Además, otro detalle importante es el hecho de que las representaciones más pequeñas ahora no son las generadas por ACC y FCC, como sucedía con la colección *BankSearch*, donde sus vocabularios mínimos siempre solían ser menores que los del resto de las representaciones. Tras estas funciones, Molinari presenta el siguiente mejor comportamiento.

Otro detalle curioso es que observando la figura 8.16 se puede pensar que la reducción MinMax se comporta mejor que la reducción PF para esta colección *WebKB*, ya que todas las funciones mejoran para vectores de representación pequeños. Este comportamiento es contrario al encontrado en la colección *BankSearch*, donde el valor de la medida-F solía disminuir en lugar de aumentar para dimensiones pequeñas. Por último, el hecho más destacable en esta clasificación es el aumento sustancial de medida-F logrado por la función WIDF respecto al caso anterior. El resto de funciones obtuvieron con esta clasificación Normal resultados similares a los obtenidos con la Multinomial, excepto WIDF, que resultó la más destacada.

8.9. Conclusiones

En este capítulo se ha realizado una breve revisión de los métodos de clasificación automática de textos, poniendo especial énfasis en su aplicación a páginas web. A continuación, se ha descrito el algoritmo *Naïve Bayes* que ha sido usado en la experimentación. Se han considerado, además,

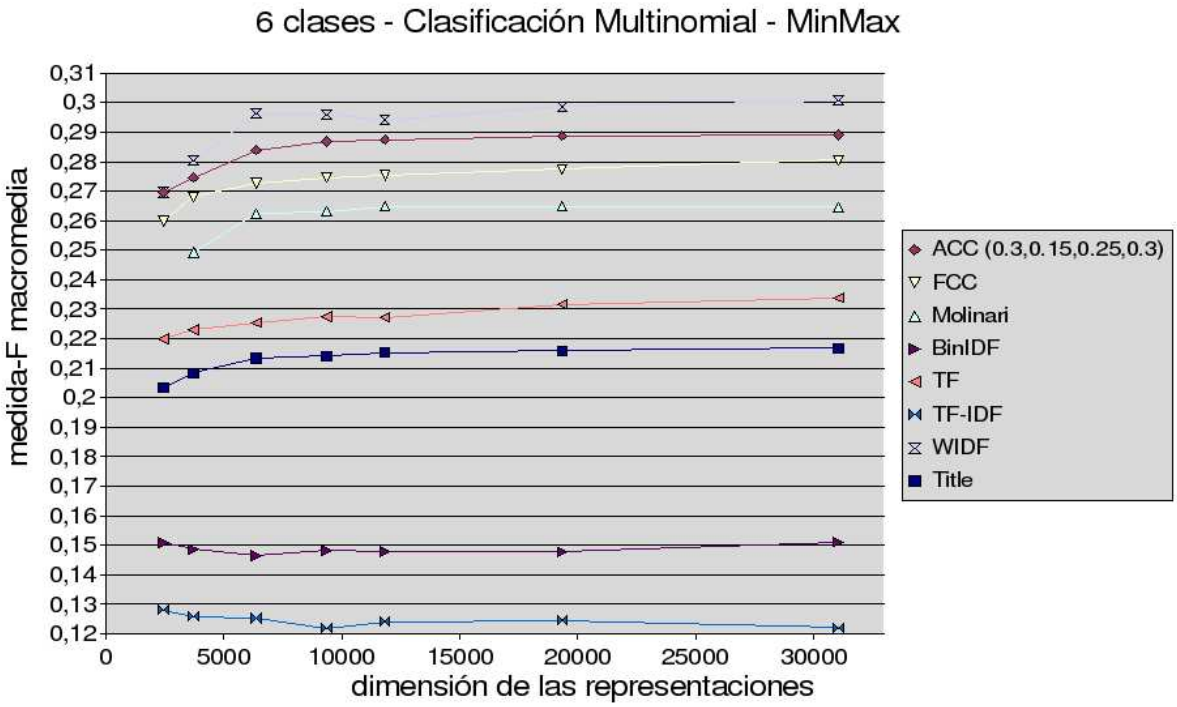

Figura 8.12: Clasificación Multinomial y reducción MinMax.

diferentes funciones de proyección, varias subcolecciones y los dos métodos de reducción de rasgos descritos en el capítulo 7.

Una vez realizada la clasificación sobre las diferentes subcolecciones de la colección *BankSearch* y sobre la colección *WebKB*, con ejemplos clasificación binaria y a 3, 6 y 10 clases, se pueden extraer las siguientes conclusiones generales. En cuanto al comportamiento del clasificador se debe concluir que sólo en el caso de las clasificaciones binarias los valores de medida-F obtenidos fueron lo suficientemente aceptables como para poder considerar este clasificador *Naïve Bayes* como un buen clasificador de páginas web. En este punto, muchos autores afirman que este comportamiento no supone ningún problema, puesto que toda clasificación a varias clases se puede descomponer como una sucesión de problemas de clasificación binarios (Sebastiani, 2002).

Tanto con la colección *BankSearch* como *WebKB*, los resultados obtenidos por las funciones ACC y FCC han sido destacables, aunque han presentado un comportamiento variable en función del número de clases, del tipo de clasificador y de la reducción empleada. En ocasiones, y tanto en problemas binarios como en problemas a varias clases, las funciones propuestas en esta tesis se han posicionado entre las que obtenían los mejores valores de medida-F. Su comportamiento ha sido, en términos generales, el más regular a lo largo de toda la experimentación y, especialmente, en relación con el resto de representaciones autocontenidas. En la mayoría de los casos no han notado la reducción en la dimensión de las representaciones, es decir, que han obtenido similares valores de la medida-F para vectores de representación grandes y pequeños.

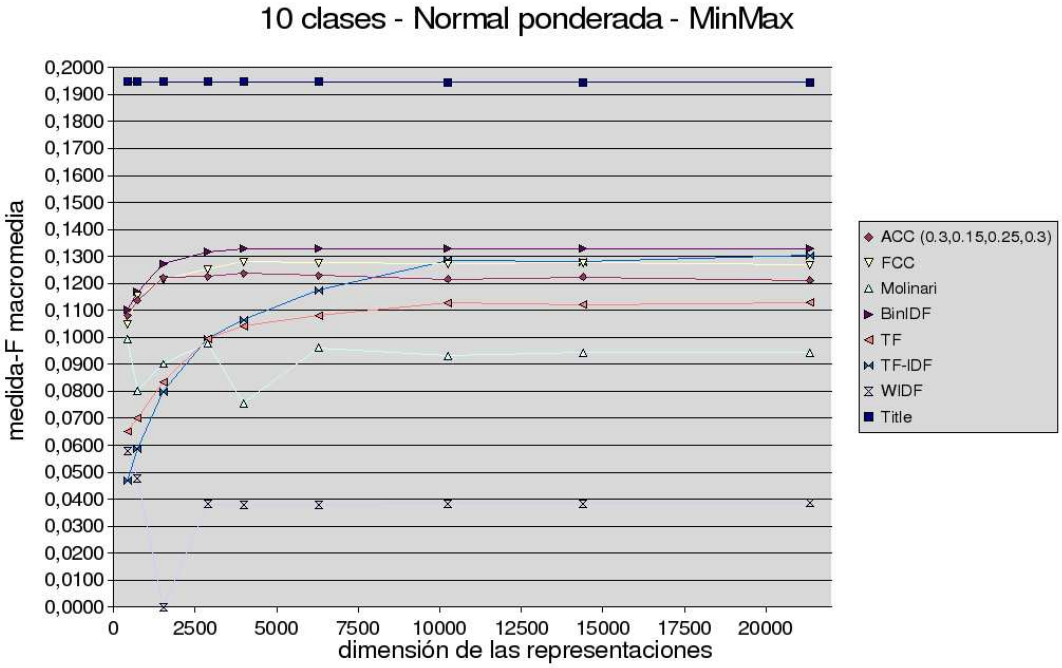

Figura 8.13: Clasificación con función Normal ponderada y reducción MinMax.

Los resultados obtenidos por las funciones WIDF y BinIDF han sido muy buenos en la mayoría de los casos en los que se aplicó una reducción PF y con dimensiones muy pequeñas en el vector de representación. En muchos casos han logrado las mejores tasas de medida-F, pero sólo con dimensiones de representación reducidas, decayendo mucho cuando aumentaban estas dimensiones. Por otro lado, para una determinada subcolección podían ser las funciones con mayor valor de medida-F, mientras que para otras ofrecían el peor comportamiento de entre las funciones evaluadas. Este comportamiento irregular también se ha puesto de manifiesto dentro de una misma colección, cuando se empleaban diferentes clasificaciones.

El comportamiento de las funciones TF y TF-IDF no ha sido nada destacable, encontrándose en la mayoría de los casos entre las funciones con peores valores de medida-F para cada subcolección. Sólo en un par de ocasiones obtuvieron valores de medida-F comparables al de las mejores funciones en ese caso. La función Molinari, en términos generales, se ha comportado mejor que TF y TF-IDF, aunque algo alejada del comportamiento general presentado por ACC, FCC y, en determinados casos, WIDF.

El comportamiento de Title ha sido muy curioso. La función Normal ponderada le ofrece unos valores de medida-F altísimos, que no se repiten ni en el caso de la función LogNormal ni con una clasificación multinomial. También es posible que los resultados empeoraran mucho cuando se emplearon subcolecciones que no presentaran texto dentro del título.

En cuanto a las funciones empleadas para el cálculo de la $P(\vec{t_i}|c_j)$, en general, la función Multinomial ha sido la que ha obtenido los mejores resultados. Con las dos variantes de las función Normal se obtuvieron siempre resultados muy parecidos, mientras que la función

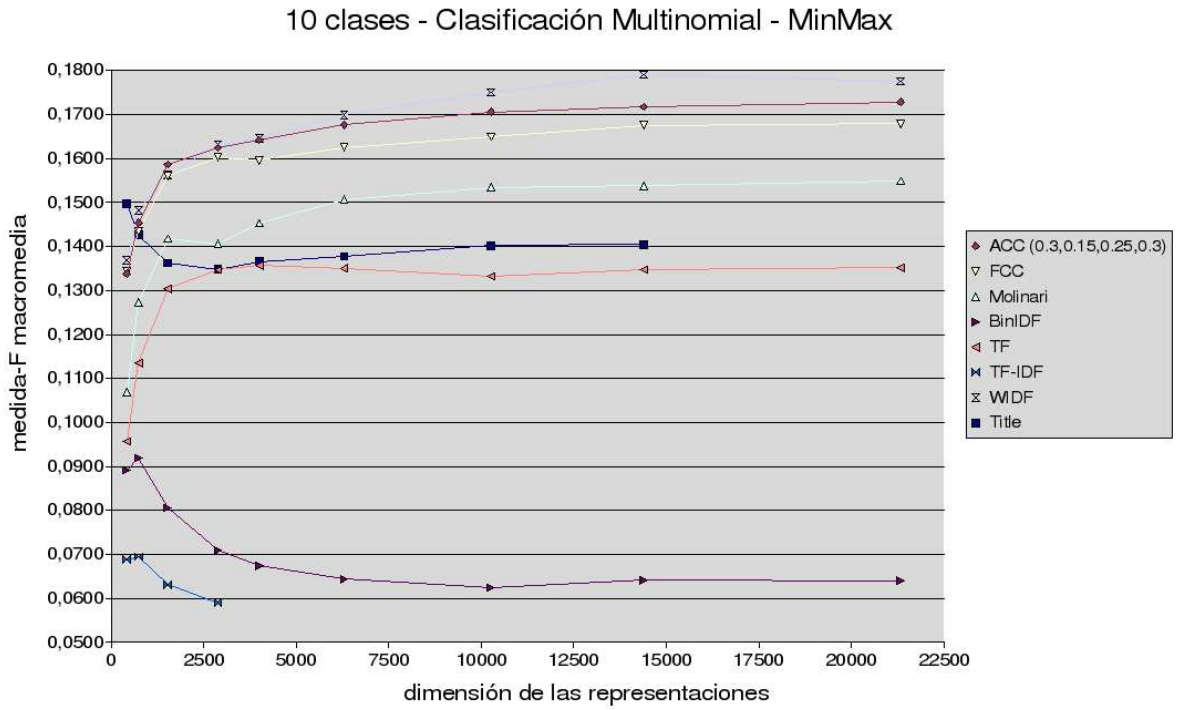

Figura 8.14: Clasificación Multinomial y reducción MinMax.

LogNormal ha mostrado un comportamiento general malo, salvo para el caso de las funciones WIDF y BinIDF cuando se usaba la reducción PF y con vectores de representación pequeños. En estos casos ha registrado los mayores valores de medida-F de todas las colecciones. Es posible que la hipótesis asumida en el aprendizaje gaussiano, y apoyada en el "teorema central del límite" no sea muy acertada cuando el conjunto de categorías aumenta sin hacerlo el conjunto de páginas de entrenamiento. La razón es evidente: según aumenta el número de categorías en las que se pretende realizar la clasificación, el número de documentos con los que se calculan los estadísticos definidos en los descriptores de cada clase disminuye. Por ejemplo, en el caso de la clasificación binaria y para la colección *BanSearch DataSet*, μ y σ se calculan con unos 3.500 documentos por clase, considerando que sobre el total de 10.000 documentos se toma un 70 % para la etapa de entrenamiento. En el caso de tratar con 10 clases, este número de páginas se reduciría a 700, por lo que para la mayoría de los rasgos de un vocabulario las relevancias medias por clase se tendrían que calcular con un número de documentos pequeño, lo que implica que el cálculo de μ y σ no resulte significativo desde el punto de vista estadístico. Este mismo comportamiento se puede dar en el caso de clasificaciones binarias cuando el número de páginas es reducido. Por supuesto, el descenso en la calidad de los resultados en TC cuando aumentaba el número elevado de clases puede haberse debido también a otros factores.

Respecto al método de reducción de rasgos empleado, ninguno de ellos ha obtenido siempre mejores resultados que el otro, de modo que la clasificación final ha resultado más dependiente del tipo de representación que se estuviera considerando en cada momento.

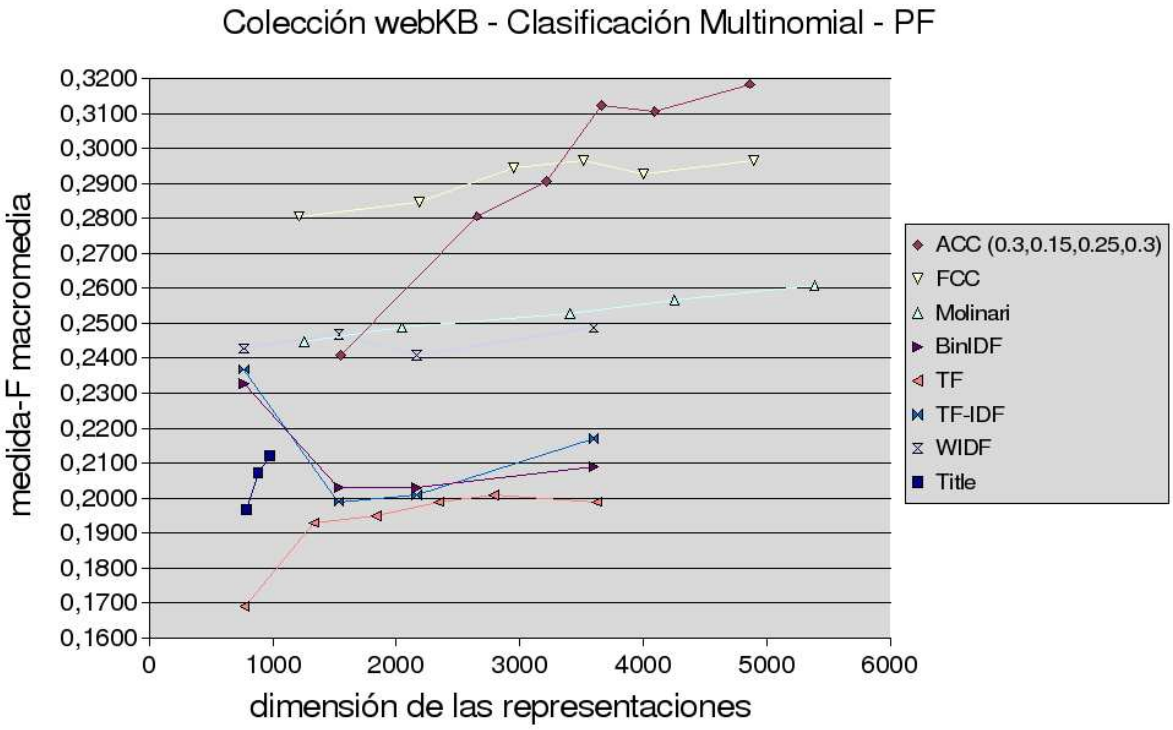

Figura 8.15: Clasificación Multinomial y reducción PF.

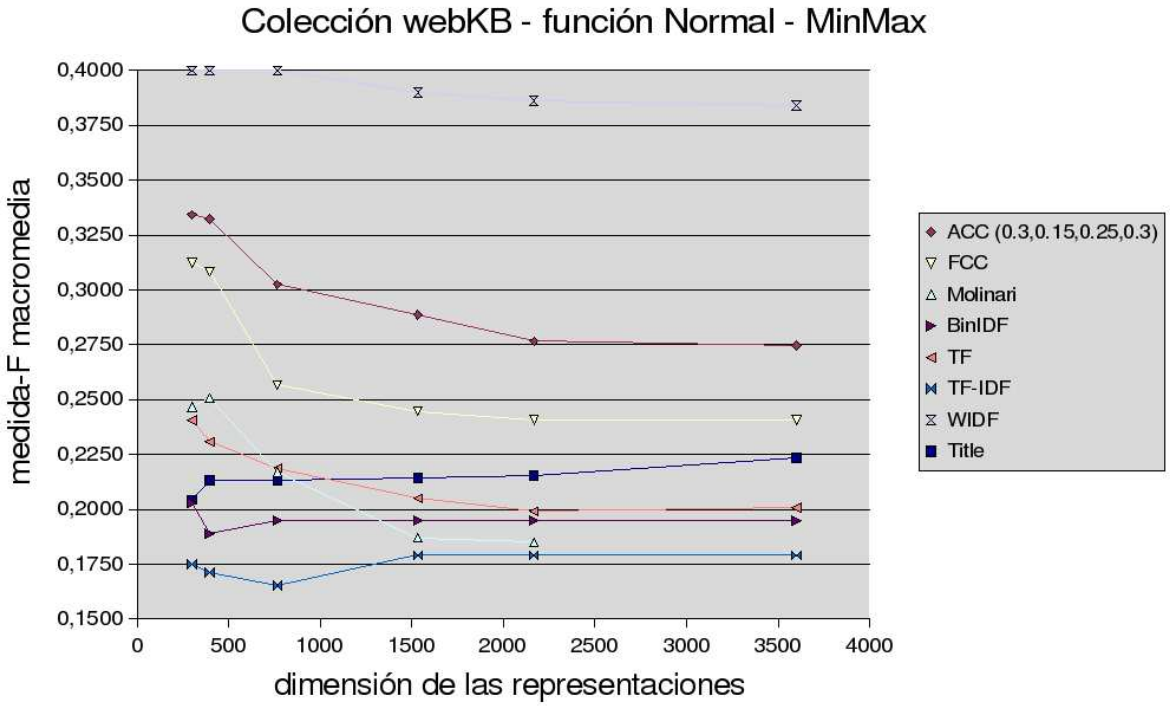

Figura 8.16: Clasificación con función Normal sin ponderación y reducción PF.

Capítulo 9

Clustering de páginas web

"Pensar es más interesante que saber,
pero menos interesante que mirar"
Johann Wolfgang von Goethe

En este capítulo se evalúa la calidad de las funciones de ponderación propuestas en esta tesis como parte de un modelo de representación autocontenida de páginas web. Mediante un algoritmo de clustering de partición, ACC y FCC se comparan con funciones de ponderación clásicas usadas en textos (BinIDF, TF, TF-IDF y WIDF) y otras funciones aplicadas en el ámbito de la representación de páginas web que emplean como información adicional el etiquetado HTML (Title y la combinación de categorías propuesta en (Molinari et al., 2003)).

9.1. Introducción

La elección del modelo de representación que se vaya a emplear en un problema de DC resulta un aspecto fundamental, ya que la elección de una representación u otra podría dar lugar a diferentes soluciones de *clustering*. De este modo, es posible evaluar la bondad de un modelo de representación en función de la calidad de la solución que se obtiene a partir de la aplicación de un determinado algoritmo de *clustering* sobre un corpus de referencia.

En este capítulo se tratará de evaluar la calidad de las funciones de ponderación propuestas en esta tesis doctoral por medio de un algoritmo de partición extraído de la librería CLUTO (Karypis, 2002).

Tomando la cuaterna $\langle X, \mathbb{B}, \mu, F \rangle$ como definición general de todo modelo de representación de documentos, en todas las representaciones que van a ser evaluadas en este capítulo se fijará como álgebra $\mathbb{B}$ el definido dentro del VSM. Además, la función de distancia μ será para todos los casos el coeficiente de correlación (que se describirá en el punto 9.4.1). Por último, los vocabularios X con los que se realiza la experimentación fueron generados según se expuso en el capítulo 7.

Dado que en esta tesis el *clustering* de documentos se empleará únicamente como método de evaluación de las diferentes representaciones consideradas, no se realizará un estudio detallado de los diferentes tipos de algoritmos de *clustering* existentes en la literatura. Aún así, se presentan brevemente las características de los algoritmos más utilizados en la literatura, y en especial de

los de *clustering* de páginas web. Por último, se muestran los resultados experimentales y las principales conclusiones extraídas a partir de ellos.

9.2. Métodos de *clustering* de documentos

El *clustering*, también llamado agrupamiento, forma parte de un proceso de aprendizaje automático no supervisado [(Banfield y Raftery, 1993), (Meila, 1999), (Hofmann, 1999) y (Szymkowiak et al., 2001)]. El objetivo fundamental es encontrar la estructura intrínseca presente en una colección de datos no etiquetados. Por tanto, en toda tarea de *clustering* se trata de agrupar un conjunto de objetos en subconjuntos llamados *clusters*, de modo que los objetos pertenecientes a un mismo *cluster* tengan un alto grado de similitud entre sí, manteniendo a su vez el menor grado de similitud posible en relación a los objetos pertenecientes a otros *clusters*.

Desde un punto de vista práctico, el *clustering* juega un papel muy importante en aplicaciones de Data Mining, tales como exploración de datos científicos, recuperación de la información, minería de textos, aplicaciones Web, aplicaciones sobre bases de datos espaciales tales como GIS (*Geographic Information Systems*) o datos procedentes de astronomía, marketing, diagnóstico médico, análisis de ADN en biología computacional, y muchas otras (Karypis, 2002).

En particular, el *clustering* de documentos ha sido ampliamente utilizado para la organización de grandes volúmenes de textos [(Willet, 1988), (Lewis, 1992), (Pereira et al., 1993), (Boley et al., 1999), (Kaban y Girolami, 2000) y (Gaussier et al., 2002)]. Inicialmente se utilizó para mejorar la precisión y cobertura en los sistemas de IR [(van Rijsbergen, 1979), (Kowalski, 1997)], y como una forma eficiente de encontrar los vecinos más cercanos a un documento (Buckley y Lewit, 1985). En este contexto, Van Rijsbergen formuló en 1979 la denominada "Hipótesis del Agrupamiento", donde sostenía que "los documentos fuertemente asociados tienden a ser relevantes para una misma consulta" (van Rijsbergen, 1979). Basándose en esta hipótesis, el *clustering* de documentos considera el contenido de los mismos para agruparlos, de forma que se asume el hecho de que documentos similares contendrán rasgos similares.

Mientras que una tarea de TC requiere información que relacione documentos con clases, en el caso de los problemas de DC no se cuenta con esta información a priori; los *clusters* que se generan quedan definidos por los documentos que les hayan sido asignados en cada caso.

En un problema típico de *clustering* pueden considerarse diferentes aspectos relativos a:

- **Selección de los atributos en los que se basarán las representaciones y posteriores agrupaciones**. En la mayoría de los casos, los documentos se representarán como vectores de rasgos con diferentes pesos que representan la relevancia de cada rasgo en el contenido del documento.
- **Selección de un método apropiado de agrupación**. Con respecto a la estructura de los grupos resultantes, los algoritmos de *clustering* se pueden clasificar en dos grandes familias:

- **Métodos Jerárquicos**, donde los documentos son agrupados en una estructura de *clusters* jerárquica, que se suele obtener tras un proceso iterativo en el que se van definiendo los *clusters* dentro de cada uno de los niveles. Este tipo de algoritmos pueden subdividirse en:
  - *Aglomerativos*; se parte inicialmente de un conjunto de *clusters* igual al número de documentos que se quiere agrupar y, posteriormente, estos se van agrupando (Jain y Dubes, 1988) hasta concluir en un único *cluster*.
  - *Divisivos*. En este caso, inicialmente se agrupan todos los documentos en un único *cluster* y sucesivamente se van separando en un proceso iterativo (Kaufman y Rousseeuw, 1990).
- **Métodos de Partición**, donde el objetivo es obtener una partición del conjunto de documentos en *clusters*, de forma que todo documento pertenezca a alguno o algunos de los k *clusters* posibles, pudiendo tener estos los límites precisos o borrosos. El número de *clusters* k podrá fijarse de antemano o ser calculado en el proceso.

Algunos métodos conocidos para realizar el *clustering* son los siguientes:

- **Clustering Probabilístico**. En este enfoque, se considera que los documentos siguen una combinación de distribuciones de probabilidad independientes (McLachlan y Basford, 1988). Algunos ejemplos de trabajos que emplean este tipo de *clustering* pueden encontrarse en (Mitchell, 1997), (Cadez y Smyth, 1999) y (Shatkay y Wilbur, 2000). En ocasiones, las fronteras entre los *clusters* pueden considerarse borrosas (*fuzzy*), en el sentido de que la asignación de un objeto a un *cluster* se regirá por las leyes de la lógica borrosa; esto es: la pertenencia de un determinado objeto a un *cluster* se expresará con un grado de posibilidad. Algunos ejemplos donde se pueda encontrar este tipo de asignación son (Bezdek, 1981), (McLachlan y Krishnan, 1997) y (Hall y Bezdek, 1999).
- **Métodos de las k-medias** (*k-means*). Este algoritmo (Hartigan, 1975) (Hartigan y Wong, 1979) es el más popular de los algoritmos de *clustering* aplicados en el ámbito científico e industrial. En primer lugar se seleccionan arbitrariamente k puntos representantes que constituyen los "centroides". Después se asigna cada documento al grupo del centroide más cercano o similar, optimizando un determinado criterio, y se actualizan los k centroides de acuerdo a la nueva composición de cada grupo. Estas fases de asignación y actualización se repiten hasta que no sea posible mejorar el criterio de optimización; normalmente, hasta que los k centroides no cambien después de una iteración. Estos métodos asumen que el valor de k es conocido.
- **Métodos de los k-vecinos** (*kmedoids*). En este caso, cada *cluster* queda representado por uno de los documentos que lo constituyen, al que se llama *medoid*

o "centroide real". De este modo, los *clusters* serán subconjuntos de documentos que rodean al documento *medoid*. Posteriormente, se define una función de distancia para medir de la similitud entre un documento y un *medoid*. Algunos trabajos donde se aplica este método son (Kaufman y Rousseeuw, 1990) y (Ng y Han, 1994).

Estos algoritmos funcionan muy bien cuando las representaciones de los objetos son de tipo numérico y no categórico (Berkhin, 2002). En (MacQueen, 1967) y (Jain y Dubes, 1988) pueden encontrarse ejemplos de aplicación de este método.

- **Algoritmos Basados en Densidad** (*Density-Based Partitioning*). En este tipo de algoritmos, se consideran conceptos relativos a la "densidad", "conectividad" y "frontera" entre *clusters*. Definen *cluster* como una componente de densidad que puede crecer en cualquier dirección y así, con estos algoritmos se pueden encontrar *clusters* con cualquier forma. Requieren un espacio medible y su aplicación natural es el *clustering* de datos de carácter espacial (Berkhin, 2002). Dentro de los algoritmos basados en densidad pueden distinguirse:
  - *Clustering de Conectividad Basada en Densidad* (*Density-Based Connectivity Clustering*); cuando la densidad y la conectividad se miden en función de la distribución local de los vecinos más cercanos (Ester et al., 1996).
  - *Clustering basado en Funciones de Densidad*; cuando se emplean funciones de densidad definidas directamente sobre el espacio de medida (Han y Kamber, 2001).
- **Métodos Basados en Rejillas** (*Grid Based Methods*). En estos métodos se considera la topología del espacio medible en el que se representan los objetos que se desea agrupar, y que se particiona. Esta partición se realiza en base a la pertenencia de los objetos a las diferentes regiones en las que puede dividirse el espacio de medida. Este tipo de *clustering* no depende del orden en el que se presenten los datos y funcionan bien con datos de tipo no numérico, al contrario que los métodos de recolocación (k-medias, k-vecinos, ...), muy dependientes de la ordenación y más eficaces si se trata con atributos de tipo numérico (Berkhin, 2002). Un ejemplo de algoritmo basado en rejilla puede encontrarse en (Sheikholeslami y Zhang, 1998).
- **Métodos Basados en la Co-Ocurrencia de Datos Categóricos**; relacionados con el concepto de "transacción", que se define como un conjunto finito de elementos llamados "items" que pertenecen a un universo común de items (Berkhin, 2002). La idea es ver si determinados elementos pertenecen o no a un determinado conjunto. Pueden encontrase ejemplos de este tipo de *clustering* en (Guha y Shim, 1999) y (Ganti et al., 1999).
- **Clustering Basado en Restricciones** (*Constraint-Based Clustering*). Este tipo de algoritmos fueron presentados en (Tung y Han, 2001) y se basan en el establecimiento

de determinadas restricciones previas al agrupamiento. Estas restricciones pueden ser: restricciones a objetos particulares, a parámetros como el número de clusters, etc. Puede encontrase una revisión de este tipo de métodos en (Han et al., 2001).

- **Algoritmos para Datos de Grandes Dimensiones**. Cuando las dimensiones de los objetos sobre los que quiere aplicarse una tarea de clustering son muy grandes, la dificultad del proceso aumenta por dos motivos principales (Berkhin, 2002). En primer lugar, bajo cualquier definición de similitud, la presencia de atributos irrelevantes (cuyo número siempre aumentará con la dimensión) dificulta el hecho de encontrar tendencias en el agrupamiento. Por otro lado, la separación espacial de los datos es mucho más difícil cuando la dimensión es muy grande. Para este tipo de datos, se han planteado diferentes tipos de algoritmos como:
  - *Clustering Subespacial* (*Subspace Clustering*), que tratan de reducir el espacio de representación de los objetos y realizar después el agrupamiento en el espacio reducido (Agrawal et al., 1998).
  - *Técnicas de Co-Clustering*, donde se trata de agrupar los atributos en subconjuntos de los que se deriva un representante. De este modo, se reduce igualmente la dimensión del espacio de representación (Dhillon, 2001).

- **Selección de las medidas de similitud o distancia**. Además de la elección del método de *clustering* podrán considerarse diferentes medidas de similitud entre documentos que representarán las métricas μ dentro de un modelo de representación. Entre las más empleadas destacan: la distancia euclídea, distancia coseno, coeficiente de correlación, distancia de Manhattan, distancia de Mahalanobis, etc. Puede encontrarse un estudio detallado del concepto de similitud semántica en (Rodríguez, 2002).

- **Selección de función criterio**. Otro aspecto a tener en cuenta en un problema de *clustering* es la elección de una función criterio. Es posible encontrar diferentes particiones de un conjunto de M objetos en k grupos, por lo que se puede definir una función criterio de modo que nos permita escoger aquella partición que proporcione un valor óptimo para dicha función. La idea es tratar de maximizar la similitud *intracluster*, a la vez que se minimiza la similitud *intercluster*. En problemas reales se pueden encontrar funciones criterio que minimicen la distancia *intracluster*, que maximicen la similitud *intercluster* o que combinen ambas medidas. La función criterio más simple y comúnmente empleada es la suma de errores al cuadrado; con esta función se trata de minimizar las distancias al cuadrado de los objetos que se quieren asociar a un determinado *cluster* con el centroide de dicho *cluster*, a la vez que se maximizan las distancias al cuadrado respecto de los centroides correspondientes al resto de *clusters* (Duda et al., 2001). En (Zhao y Karypis, 2001) se describen otras conocidas funciones criterio.

- **Validación de los resultados obtenidos**. La calidad del *clustering* podrá evaluarse en función de diferentes medidas de evaluación como son la cohesión interna, la medida-F, la entropía, índice de pureza, etc.

- **El número de *clusters* a crear**. En algunos casos se establecerá como un parámetro de entrada al algoritmo de *clustering*; en otros, será el propio algoritmo el encargado de determinar el número óptimo de *clusters* en los que se va dividir el conjunto de documentos de entrada.

Estas consideraciones muestran que cualquier proceso de *clustering* supone una tarea compleja que depende de numerosos factores. La entrada será siempre el conjunto de objetos que se quiere agrupar; en nuestro caso, páginas web. La salida dependerá del tipo de algoritmo utilizado. En unos casos, será un conjunto de *clusters* con sus respectivos documentos y en otros casos, como ya se ha visto, podría ser una organización jerárquica de los *mismos*.

9.3. *Clustering* de páginas web

Las aplicaciones más importantes del *clustering* en el contexto web son:

- Agrupación previa de los documentos almacenados en un servidor web, de modo que se pueda realizar una organización previa de todos los documentos disponibles. A este tipo de *clustering* se le conoce como *clustering* a priori.

- Agrupación de los documentos recuperados por un motor de búsqueda, facilitando la revisión de los resultados de una búsqueda por parte del usuario final, lo que se conoce como *clustering* a posteriori.

El *clustering* se ha estudiado y aplicado en diferentes campos y con diferentes fines. A menudo ha sido empleado para tratar grandes volúmenes de documentos, facilitando así al usuario la posibilidad de distinguir aquellos que puedan resultar de su interés. En (Crouch et al., 1989) se realiza un *clustering* jerárquico sobre grandes conjuntos de documentos HTML. En este trabajo se describe el funcionamiento de un navegador interactivo basado en dichas jerarquías, donde los documentos son representados dentro del VSM con una función de ponderación local TF.

Posteriormente, se propusieron algoritmos de DC para la navegación por colecciones de documentos (Cutting et al., 1992) o para la organización de los resultados devueltos por un motor de búsqueda en respuesta a una petición de usuario (Zamir et al., 1997). También se han utilizado en la generación automática de jerarquías de documentos (Koller y Sahami, 1997).

En otros sistemas, el *clustering* se realiza por medio de mapas auto-organizativos de Kohonen, un tipo especial de red neuronal no supervisada (Kohonen, 1995). Éste es el caso de (Miikkulainen, 1990), donde se explotan diferentes mapas de características jerarquizadas o

del proyecto WebSOM[1], donde se emplean este tipo de redes neuronales para el agrupamiento de páginas web. En (Merkl, 1997) y (Merkl, 1998) el *clustering* se realiza dentro de un modelo de redes neuronales con arquitectura multicapa, donde cada capa es en sí misma un mapa auto-organizativo independiente, y donde en cada capa se va reduciendo el nivel de generalidad a partir de los *clusters* encontrados. En estos casos, el vector de entrada al mapa es siempre de igual dimensión que el vocabulario y suele emplearse una función de ponderación TF o TF-IDF.

En (Ruocco y Frieder, 1997) se realiza una tarea de clustering paralelo sobre una colección de noticias digitalizadas del *Wall Street Journal* de los años 1987, 1988 y 1989. El procesamiento que se realizaba sobre la colección era, en primer lugar, un proceso típico de *clustering* para, posteriormente, complementarse con una tarea de clasificación automática dentro del ámbito de la IR. En este trabajo la representación se realizaba dentro del VSM y la función de ponderación utilizada era la función TF. La similitud entre un documento y un *cluster* se calculaba con la función *coseno*:

$$coseno(c_j, r_i) = \frac{\sum_{ij} c_{ij} r_i}{\sqrt{\sum_{ij} c_{ij}^2 \sum_i r_i^2}} \tag{9.1}$$

donde c_{ij} representa la relevancia de un rasgo t_i en un *cluster* c_j y r_i es la relevancia del rasgo t_i en el documento.

En (Zamir y Etzioni, 1998) y (Zamir y Etzioni, 1999) se realizan tareas de clustering basadas en la coaparición de sintagmas en documentos, estudiando el agrupamiento de grandes volúmenes de páginas web, y lo hacen en función de la coaparición de los rasgos en los documentos. La representación empleada en este caso es una representación con una función de ponderación TF dentro de un VSM.

En (Muresan y Harper, 1998) y (Muresan et al., 1999) se presenta el proyecto WebCluster, donde se agrupan los documentos de una colección web de ámbito restringido, de forma que el usuario navega por una estructura de *clusters*. Un enfoque similar puede encontrarse en (Leuski y Allan, 2000). En este trabajo se agrupan los documentos en un *cluster*, se selecciona un documento de un *cluster* y si resulta de interés para el usuario se examina el resto del *cluster*. La representación utilizada en este caso se enmarca dentro del VSM y la función de ponderación utilizada para encontrar la relevancia de un rasgo t_i en un documento d_j fue una función "TF de Okapi":

$$r_{ij} = \frac{f_{ij}}{f_{ij} + 0{,}5 + 1{,}5 \frac{\sum_i f_{ij}}{\frac{1}{N}\sum_i f_{ij}}} \cdot \frac{log(\frac{N+0{,}5}{df(\vec{t_i})})}{log(N+1)} \tag{9.2}$$

donde f_{ij} es la frecuencia de t_i en d_j, $df(\vec{t_i})$ es el número de documentos en los que está presente t_i y N es el número de documentos de la colección. Posteriormente, la similitud entre documentos se calculaba con la función coseno y se empleaba en un algoritmo de *clustering* de partición.

[1] http://websom.hut.fi/websom/

En (Muresan y Harper, 2001) se propone el uso de técnicas de *clustering* para encontrar la estructura de la colección sobre la que se quiere realizar una consulta y ayudar así a lo que llaman "mediación". La mediación se puede definir como la tarea, desarrollada típicamente por un humano, en la cual se asiste al usuario en la formulación de la consulta o en el refinamiento de la misma. Se aplica, por tanto, un *clustering* a priori en portales temáticos especializados; construyen jerarquías a partir de ontologías para dominios específicos. La representación de los documentos se realiza dentro del VSM y con función de ponderación TF. El cálculo de la similitud entre *clusters* se calcula con la divergencia Kullbach Liebler (KL), también llamada entropía relativa. Esta medida establece la especificidad relativa de un rasgo t_i en un *cluster* P respecto a otro Q y se expresa como:

$$KL_i = p_{i,P} log \frac{p_{i,P}}{p_{i,Q}} \tag{9.3}$$

donde $p_{i,P}$ representa la fracción entre el número de ocurrencias de t_i de P y el número total de rasgos en el *cluster*.

En (Liu et al., 2002) se realiza un *clustering* jerárquico sobre la representación del contenido textual de los documentos dentro de un sitio web. Se considera la estructura jerárquica del sitio web y su fin era el desarrollo de una técnica de visualización que permitiera a los usuarios encontrar información útil sobre sus competidores comerciales. Los documentos se representaron dentro del VSM y en el preproceso se eliminaron, además de una lista de palabras vacías típica en IR, otra relativa al dominio específico en el que se estaba trabajando. Se empleaba una función de ponderación TF y se aplicaba un proceso típico de truncamiento; con ello se trataba de encontrar una serie de palabras clave con las que representar cada sitio web. Otros trabajos centrados en representaciones de páginas web por contenido son (Honkela et al., 1997) y (Sinka y Corne, 2004).

Otra aplicación habitual para la que se ha empleado el *clustering* de páginas web ha sido la visualización de los resultados tras una consulta a un motor de búsqueda; de hecho, ha sido el principal enfoque en el que se han empleado estas técnicas dentro del contexto Web. En (Allan et al., 1997) se realizaba una visualización de *clusters* para asistir a los usuarios en la identificación de documentos relevantes tras una consulta a un motor de búsqueda. Partiendo de la "Hipótesis de Agrupamiento" de van Rijsbergen, en este trabajo los documentos HTML son representados como textos dentro del VSM y con una función de ponderación TF-IDF. En este caso, la distancia entre documentos se mide con una función *seno*, de forma que toma un valor 0 si los documentos son iguales (si tienen una representación idéntica) y 1 si sus vectores de representación son ortogonales. En (Sebrechts, 1999) se estudia también la visualización de los resultados de la respuesta de un motor de búsqueda. En este caso se emplean palabras clave que se buscan en los títulos de los documentos.

En (Carey et al., 2000) se realiza una representación empleando como función de ponderación una modificación de la función TF-IDF para detectar e identificar palabras clave en los documentos recuperados tras una consulta a un motor de búsqueda. A partir de estas palabras clave se realiza un *clustering* sobre los documentos recuperados, de forma que puedan mostrarse agrupados visualmente. Otros trabajos donde se han aplicado tareas de DC para la visualización de resultados son (Good, 1958), (Fairthorne, 1961) y (Needham, 1961).

En (Cigarrán et al., 2005) se propone el análisis de conceptos formales (*Formal Concept Analysis*) como alternativa al *clustering* clásico. En este trabajo, las páginas web devueltas por un motor de búsqueda tras una consulta son representadas por medio de sintagmas, de forma que es posible la navegación a través de los resultados obtenidos tras una búsqueda. La función de ponderación que se emplea, presentada en un trabajo previo (Cigarrán et al., 2004), trata de dar mayor peso a aquellos sintagmas que aparecen más frecuentemente en el conjunto de documentos recuperados que en la colección de referencia. Por tanto, esta función tiene un carácter global y se expresa como:

$$r_i = 1 - \frac{1}{log_2(2 + \frac{tf_{i,ret} \cdot f_{i,ret}}{tf_{i,col}+1})} \tag{9.4}$$

donde r_i es el peso que se asigna a un sintagma t_i, $tf_{i,ret}$ representa la frecuencia del sintagma en el conjunto de documentos recuperados, $f_{i,ret}$ es el número de documentos recuperados en los que está presente el sintagma y $tf_{i,col}$ es la frecuencia relativa del sintagma t_i en la colección, sin contar el conjunto de documentos recuperados.

En (Karypis y Han, 2000) se presenta un algoritmo llamado *concept indexing*, que se emplea como un método de reducción de la dimensión de las representaciones. La idea es sencilla: tras la aplicación del algoritmo de *clustering* se pueden seleccionar los centroides de cada uno de k los grupos creados y entonces usar esos k centroides como vectores base generadores de una representación vectorial en un espacio k-dimensional. Para la aplicación del algoritmo se emplearon representaciones vectoriales con función de ponderación TF-IDF y como métrica de distancia se empleó la función coseno.

Pero las tareas de *clustering* no sólo se han basado en las representaciones por contenido de los documentos. En otros trabajos, estas técnicas han sido aplicadas a documentos HTML analizando la estructura de hipertexto que la Web, como en (Moore et al., 1997) y (Boley et al., 1999), donde se comparan diferentes algoritmos de *clustering* aplicados sobre páginas web. Se utiliza, en primer lugar, un algoritmo basado en descubrimiento de reglas de asociación (*Association Rule Hypergraph Partitioning*, ARHP), y empleado en tareas de DM como, por ejemplo, la búsqueda de relaciones entre transacciones de venta y los artículos vendidos en un supermercado. La representación del contenido de cada documento fue realizada dentro del VSM y con una función de ponderación TF.

Otro algoritmo utilizado en ese trabajo fue un algoritmo de partición (*Principal Direction Divisive Partitioning*, PDDP) en el que se empleaba una representación por LSI y con función de

ponderación TF-IDF. Asimismo, fue comparado con un algoritmo de *clustering* jerárquico, donde se empleaba la función coseno como métrica de distancia dentro del modelo de representación y función de ponderación TF-IDF, y otro algoritmo de tipo probabilístico (AutoClass), presentado en (Cheeseman y Stutz, 1996). Los autores llegaron a la conclusión de que los algoritmos ARHP y PDDP ofrecían mejor comportamiento que el resto de algoritmos evaluados, considerados como clásicos en el ámbito de la selección de rasgos. Evidencian también el hecho de que utilizando la información almacenada en las etiquetas HTML, como son los enfatizados, se podrían mejorar aún más los resultados (Boley et al., 1999).

Otros trabajos que realizan un análisis similar de la estructura de hiperenlaces son (Han et al., 1997), (Craven et al., 1998a) y (Strehl et al., 2000). En (Han et al., 1998b), además de analizar la estructura de hiperenlaces, se representa el contenido de un documento analizando el texto presente en diferentes etiquetas HTML: $< title >$, los encabezados $< h1 >, \ldots, < h6 >$ y los enfatizados $< emphasized >$ y $< strong >$. Además, se presenta una arquitectura de agentes software, llamada WebACE, para la creación de categorías a partir de una colección de páginas web no etiquetadas. En este arquitectura se prueban los algoritmos de *clustering* ARPH y el PDDP.

En (Fu et al., 1999) se realiza un *clustering* sobre los patrones de acceso de usuarios web. Estos patrones se extraen de los *logs* de los servidores web y se organizan en "sesiones" que representan "episodios" de interacción entre los usuarios y el propio servidor web. El conjunto de sesiones es posteriormente agrupado mediante un algoritmo de *clustering* jerárquico. En (Cadez et al., 2000), al igual que en (Wen et al., 2001), se estudia la experiencia en la navegación web por parte de usuarios a través de un algoritmo de *clustering* que ayuda a agrupar los diferentes patrones de navegación encontrados. Trabajos más recientes donde se analizan los patrones de uso y se aplican técnicas de DC son (Baglioni et al., 2003) y (Smith y Ng, 2003).

En la actualidad, se pueden encontrar cada vez más buscadores que emplean el *clustering* sobre los resultados de búsqueda. Algunos como *Vivísimo*[2], *Clusty*[3] o *iBoogie*[4] realizan agrupaciones jerárquicas de los resultados. Otros como *Grokker*[5] y *WebClust*[6] presentan una previsualización de los sitios web, mientras que *Mooter*[7] muestra además una visualización de los *clusters*; en el caso de *KartOO*[8] se combinan ambas cosas. En el caso del buscador *Dumbfind*[9], se realiza un agrupamiento por temas y rasgos. Otro buscador que realiza clustering es *JBraindead*[10], que forma retículos con los documentos devueltos por un motor de búsqueda por medio de análisis de conceptos formales (Cigarrán et al., 2004).

[2] http://vivisimo.com/
[3] http://clusty.com/
[4] http://www.iboogie.com/
[5] http://www.grokker.com/
[6] http://www.webclust.com/
[7] http://www.mooter.com/
[8] http://www.kartoo.com/
[9] http://www.dumbfind.com/
[10] http://bender.lsi.uned.es:8080/ModuloWeb/jbraindead.html

9.4. CLUTO: un paquete *software* para el clustering de documentos

El algoritmo de partición con el que se va a evaluar la calidad de las funciones de proyección propuestas en esta tesis doctoral ha sido un algoritmo extraído de la librería CLUTO[11]. Se trata de un paquete *software* desarrollado por George Karypis en el Departamento de *Computer Science* de la Universidad de Minnesota, que permite realizar el *clustering* de documentos. Todos los detalles técnicos del paquete pueden encontrarse en (Karypis, 2002).

CLUTO está especialmente desarrollado para tratar con colecciones de datos de alta dimensionalidad y tiene capacidad tanto para generar, como para analizar las características de los *clusters* obtenidos. La principal característica de CLUTO es que trata el problema del *clustering* como un proceso de optimización en el que se trata de maximizar o minimizar una determinada "función criterio", que puede estar definida local o globalmente y cuyo dominio es siempre el espacio total de soluciones posibles.

Los algoritmos de los que dispone esta librería están desarrollados especialmente para su aplicación sobre colecciones de documentos de texto y se pueden agrupar en tres tipos: algoritmos de *partición*, *aglomerativos* y *de partición basados en análisis de grafos*. Dispone, además, de siete funciones criterio con las que es posible realizar el *clustering* de partición y aglomerativo.

CLUTO ofrece funciones para analizar el conjunto de *clusters* generados y encontrar las relaciones entre ellos, disponiendo de herramientas que ayudan a la visualización del conjunto de soluciones encontradas. Permite también encontrar las características que mejor describen cada uno de los *clusters* hallados.

En esta tesis se ha seleccionado uno de los algoritmos de partición que ofrece esta librería, el algoritmo de clustering ***k-way via Repeated Bisections***, que será descrito en la siguiente sección.

9.4.1. *Clustering* k-way via Repeated Bisections

El algoritmo de *clustering* empleado en la evaluación de las representaciones propuestas en esta tesis es el *k-way via Repeated Bisections* de la librería CLUTO. Se trata de un algoritmo de *clustering* de partición al que se le debe especificar previamente el número de *clusters* k en los que quiere realizarse la partición. Ésta se logrará mediante un proceso iterativo de sucesivas biparticiones. Se ha elegido este algoritmo por su sencillez y buen comportamiento con diferentes colecciones de documentos.

El motivo de la elección de este algoritmo fue que se trata de un algoritmo que obtiene buenos resultados y que ha sido empleado en numerosos trabajos de DC [(Casillas et al., 2003a), (Casillas et al., 2003b), (Agirre y Lopez de Lacalle, 2003), (Carrasco et al., 2003), (Jain et al.,

[11] http://www-users.cs.umn.edu/ karypis/cluto

2004), (Li et al., 2004) y (Almuhareb y Poesio, 2004)].

En primer lugar, el conjunto inicial de documentos es dividido en dos *clusters*. A continuación, se selecciona uno de ellos y se divide de nuevo en dos *clusters*. Este proceso de selección y bipartición es repetido hasta que se tiene el número k de *clusters* deseado. Cada una de estas bisecciones se realiza de forma que el conjunto resultante optimice la función criterio seleccionada. Obtener de este modo la división en k *clusters* permitiría que la solución final resultante fuera jerárquica y fácilmente visualizable, aunque en esta experimentación no se ha considerado esta posibilidad, ya que lo que nos interesaba era el nivel de partición en el que se obtenía el k deseado y no la estructura jerárquica que se generaba.

Un aspecto importante de este algoritmo es saber cómo determinar cuál será el siguiente *cluster* a biseccionar. En los experimentos realizados en el contexto de esta tesis doctoral, esta elección se realiza seleccionando el *cluster* con el que se obtenga un mayor valor de la función criterio.

En este algoritmo, cada *cluster* es representado por su centroide teórico. Siendo n el número total de documentos que se desea agrupar, N el número total de rasgos –la dimensión del vocabulario con el que se esté representando–, y k el número de *clusters* que se desea obtener, entonces S denotará el conjunto total de n documentos que se quieren agrupar. De este modo, con $S_1, S_2, \ldots, S_k$ se denotan cada uno de los k *clusters* que se quieren encontrar y $n_1, n_2, \ldots, n_k$ serán las respectivas dimensiones de dichos *clusters*. Con todo esto, si A es un conjunto de documentos que forman un *cluster*, se define su centroide teórico como:

$$\vec{C_A} = \frac{\sum_{d_j \in A} \vec{d_j}}{|A|} \tag{9.5}$$

es decir, $\vec{C_A}$ representa el vector obtenido tras el cálculo del valor medio de los pesos de las componentes del total de los documentos que forman un *cluster*.

A lo largo de los años, se han propuesto numerosas funciones para el cálculo de la similitud entre dos documentos dentro del VSM y para problemas de *clustering*. En CLUTO se puede elegir entre la *distancia euclídea*, la *función coseno* y la *función de correlación*.

La distancia euclídea viene dada por la siguiente expresión:

$$dis(\vec{d_i}, \vec{d_j}) = \sqrt{(\vec{d_i} - \vec{d_j})^t (\vec{d_i} - \vec{d_j})} = \|\vec{d_i} - \vec{d_j}\| \tag{9.6}$$

donde $\|\vec{d_i}\|$ representa la norma de $\vec{d_i}$. En este caso, $\{dis(\vec{d_i}, \vec{d_j}) = 0 \mid \vec{d_i} = \vec{d_i}\}$; la distancia es igual a 0 siempre que las representaciones de dos documentos sean idénticas y será igual a $\sqrt{2}$ si los documentos no tienen nada en común.

La función coseno, que mide la distancia entre dos documentos d_i y d_j, toma la expresión:

$$coseno(\vec{d_i}, \vec{d_j}) = \frac{\vec{d_i^t}.\vec{d_j}}{\|\vec{d_i}\|.\|\vec{d_j}\|} \tag{9.7}$$

donde $\vec{d_i^t}$ representa el vector traspuesto de $\vec{d_i}$ y $\|\vec{d_i}\|$ representa la norma de $\vec{d_i}$. De este modo, esta medida coseno tomará un valor igual a 1 si $\vec{d_i} = \vec{d_j}$ y será 0 si ambas representaciones resultan ortogonales, lo que podría interpretarse como que no tienen características comunes.

La diferencia principal entre ambas funciones de similitud es que la función coseno no toma en cuenta el módulo de las representaciones, sino el ángulo que forman los vectores de representación.

En nuestra experimentación, la función que se emplea como medida de la similitud entre documentos es el coeficiente *de correlación* entre dos documentos $\vec{d_i}$ y $\vec{d_j}$.

$$s_{ij} = \frac{\sum_{k=1}^{N}(r_{ik} - \overline{r_i}).(r_{jk} - \overline{r_j})}{(\sum_{k=1}^{N}(r_{ik} - \overline{r_i})^2 \sum_{l=1}^{N}(\overline{r_j})^2)^{1/2}} \tag{9.8}$$

con

$$\overline{r_i} = \sum_{k=1}^{N} \frac{r_{ik}}{N} \tag{9.9}$$

y donde r_{ij} representa la relevancia de un rasgo t_i en un documento d_j y N es la dimensión del vector de representación.

Este coeficiente de correlación toma valores en el intervalo $[-1, 1]$, pudiéndose transformar en valores de correlación en el intervalo $[0, 1]$ mediante la transformación:

$$s_{ij}^{*} = \frac{(1 + s_{ij})}{2} \tag{9.10}$$

Se ha seleccionado esta medida al haberse obtenido buenos resultados con ella en experimentos previos (Casillas et al., 2003b).

Por último, como función criterio se ha seleccionado la función I_2 definida en (Karypis, 2002). Esta función es una variante del popular algoritmo de partición *k-means* [(Cheng y Wei, 1991), (Dhillon, 2001), (van Rijsbergen, 1979)] y se ha seleccionado porque mostraba un buen comportamiento medio con diferentes colecciones (Zhao y Karypis, 2002).

La función criterio I_2 que debe maximizarse en el algoritmo tiene la siguiente expresión:

$$I_2 = \sum_{r=1}^{k} \sum_{d_i \in S_r} s_{ij} \tag{9.11}$$

Si se considera un conjunto de A documentos y se define $\vec{D_A} = \sum_{\vec{d_j} \in A} \vec{d_j}$; e I_2 como el vector

compuesto de A, entonces la función a maximizar podría reformularse como:

$$I_2 = \sum_{r=1}^{k} \sum_{d_i \in S_r} \frac{\vec{d_i^t} \vec{C_r}}{\|\vec{C_r}\|} = \sum_{r=1}^{k} \frac{\vec{D_r^t} \vec{C_r}}{\|\vec{C_r}\|} = \sum_{r=1}^{k} \frac{\vec{D_r^t} \vec{D_r}}{\|\vec{C_r}\|} = \sum_{r=1}^{k} \|\vec{C_r}\| \tag{9.12}$$

9.5. Funciones de evaluación

Evaluar la calidad de los resultados de un proceso de *clustering* es ver lo "naturales" que resultan los *clusters* creados por un determinado algoritmo. Esto implica el establecimiento de métricas que traten de relacionar la estructura intrínseca existente en un conjunto de datos de entrada con los *clusters* generados tras la aplicación de un algoritmo sobre dichas colecciones. Así, una solución podría considerarse buena si respecta y no viola la estructura intrínseca existente en los datos de entrada. Esta es la idea que descansa tras la mayoría de las funciones de evaluación de *clustering* (Gokcay y Principe, 2000). En una función de evaluación interna no se utiliza ninguna colección previamente agrupada para compararla con la solución obtenida, mientras que en una evaluación externa, la solución obtenida es comparada con una solución considerada de referencia, que suele haber sido creada de forma manual.

En el *clustering* llevado a cabo en esta tesis doctoral se ha realizado una evaluación externa por medio de la medida-F (van Rijsbergen, 1979), función que combina los valores de *precisión* y *cobertura*. Un mayor valor de la medida-F indicará mejor calidad del *clustering*. Si se denomina "clase" a cada uno de los grupos proporcionados por las colecciones de referencia y "*clusters*" a los grupos obtenidos tras la aplicación del algoritmo con el que se evalúa, la medida-F del *cluster* j y la clase i viene dada por la ecuación:

$$\text{medida-F } (i,j) = \frac{2 \times Recall(i,j) \times Precision(i,j)}{(Precision(i,j) + Recall(i,j)} \tag{9.13}$$

donde la cobertura (*Recall*) y la precisión (Precision) son:

$$Recall(i,j) = \frac{n_{ij}}{n_i} \ , \ Precision(i,j) = \frac{n_{ij}}{n_j} \tag{9.14}$$

siendo n_{ij} el número de documentos de la clase i en el *cluster* j, n_j el número de documentos del *cluster* j, y n_i el número total de documentos de la clase i. El valor de esta medida-F para el conjunto total de *clusters* es:

$$\text{medida-F } = \sum_i \frac{n_i}{n} max\{ \text{ medida-F } (i,j)\} \tag{9.15}$$

donde n es el número total de páginas web que se quieren agrupar, n_i es el número de documentos de un *cluster* S_i y la función "max" se aplica sobre el conjunto de *clusters*.

9.6. Resultados experimentales

En esta sección se presentan los resultados experimentales del *clustering* de páginas web en función de las diferentes colecciones y del número de *clusters* considerado, así como de cada una de las funciones de proyección seleccionadas.

Al igual que en el capítulo 8, las figuras muestran los resultados de "medida-F" en función de la "dimensión de las representaciones". En el eje X se representa el tamaño del vocabulario con el que se genera cada representación, y en el eje Y se representa la medida-F, una función con recorrido en el intervalo $[0, 1]$, de forma que los valores más altos de medida-F supondrán los mejores resultados de *clustering*.

Así, es posible analizar el comportamiento del *clustering* para las diferentes funciones de ponderación en relación con la dimensión de las representaciones. Al igual que en la experimentación en TC, en los casos en los que se utiliza la reducción con la *propia función de ponderación* (reducción PF), las dimensiones son diferentes para cada función de proyección. Esto es debido a que al usar la reducción PF no se generan vocabularios iguales al tomar, por ejemplo, los 3 rasgos más relevantes de cada página con la función WIDF o con ACC. Por el contrario, cuando se emplea la reducción *term-frequency/document-frequency* (reducción MinMax), todas las funciones presentan valores de medida-F en las mismas dimensiones de las representaciones.

Tanto en el caso de la reducción PF como en el de la MinMax se ha tratado de cubrir un rango que fuera desde una dimensión de representación mínima, hasta un tamaño un orden de magnitud menor que la dimensión del vocabulario sin reducir.

9.6.1. Colección BankSearch

- *Clustering* binario

  - **GH_1000**. **Clustering binario** en el **nivel más bajo de la jerarquía** y entre **clases cercanas semánticamente**. En las figuras 9.1 y 9.2 se muestran los resultados de este *clustering* empleando cada una de las funciones evaluadas como función de ponderación de rasgos y con las reducciones PF y MinMax respectivamente.

    Cuando se emplea la reducción PF (figura 9.1), la función que mejor resultado presenta es la FCC, seguida de la ACC, basadas ambas en combinaciones heurísticas de criterios. La función TF tiene un comportamiento general más destacado que la función TF-IDF, que sólo obtiene mejor resultado para la medida-F en un punto. Sin embargo, éste se corresponde con el menor valor para la dimensión de la representaciones. Las funciones BinIDF, WIDF y Molinari obtienen valores por debajo del 0,85, acusando mucho la reducción de rasgos PF y presentando valores

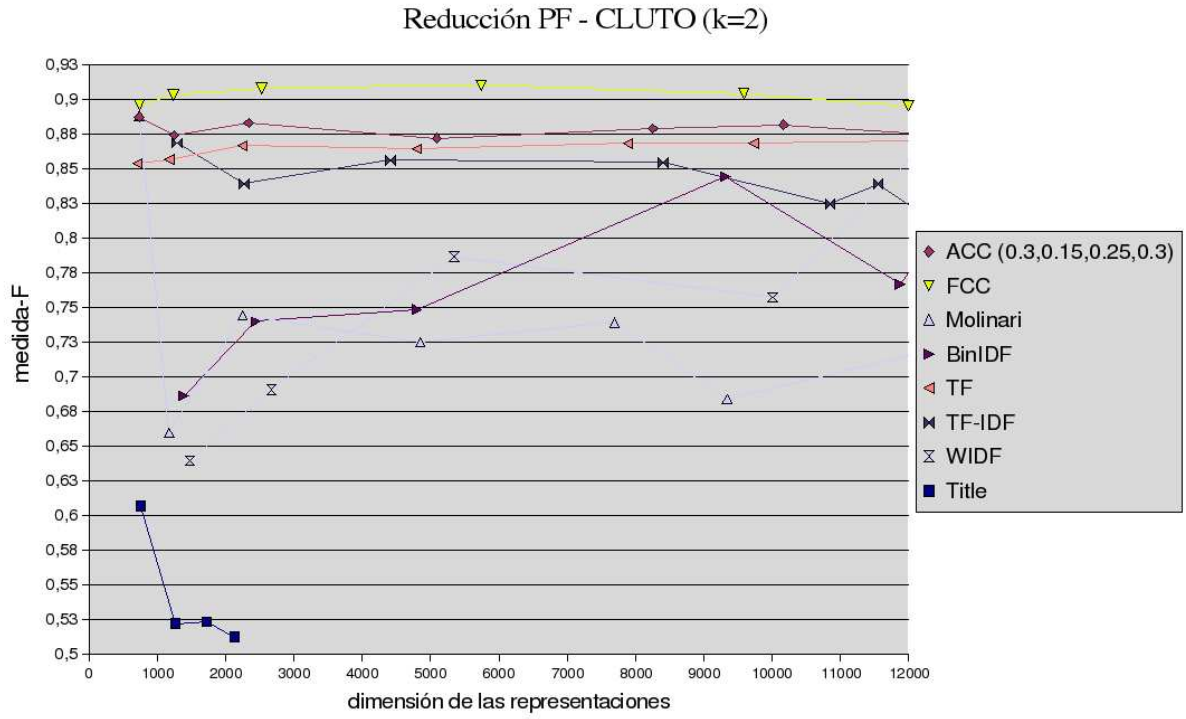

Figura 9.1: *Clustering* binario con la colección GH_1000 y reducción de rasgos realizada con la propia función de ponderación. El *clustering* se realiza entre las clases "Astronomía" y "Biología", pertenecientes a la superclase "Ciencia".

decrecientes en la medida-F según se reduce la dimensión de las representaciones. Con la función Title se obtienen los peores resultados.

Cuando se emplea la reducción MinMax (figura 9.2) se obtienen, en valor absoluto, peores resultados que con la reducción PF. Las funciones TF-IDF y WIDF resultan las funciones con un comportamiento más destacado en término general, seguidas por la función TF. Además, puede observarse un comportamiento particular en algunas de las funciones, y en especial la combinación borrosa, ya que mejoran sustancialmente sus valores de medida-F con dimensiones pequeñas de la representación. Con vocabularios grandes los resultados no son nada destacables, mientras que a dimensiones de representación bajas los resultados de FCC son comparables a los obtenidos por TF-IDF y WIDF. Las funciones BinIDF, Title y Molinari presentan en menor medida este comportamiento relativo mejor a dimensiones bajas. ACC no obtiene buenos resultados destacables, pero en el caso de la dimensión más pequeña obtiene un valor de medida-F equiparable al de TF-IDF, la función más destacada en ese rango.

- **G&J_1000**. **Clustering binario** en el **nivel más bajo de la jerarquía** y entre **clases lejanas semánticamente**. En las figuras 9.3 y 9.4 se muestran los resultados

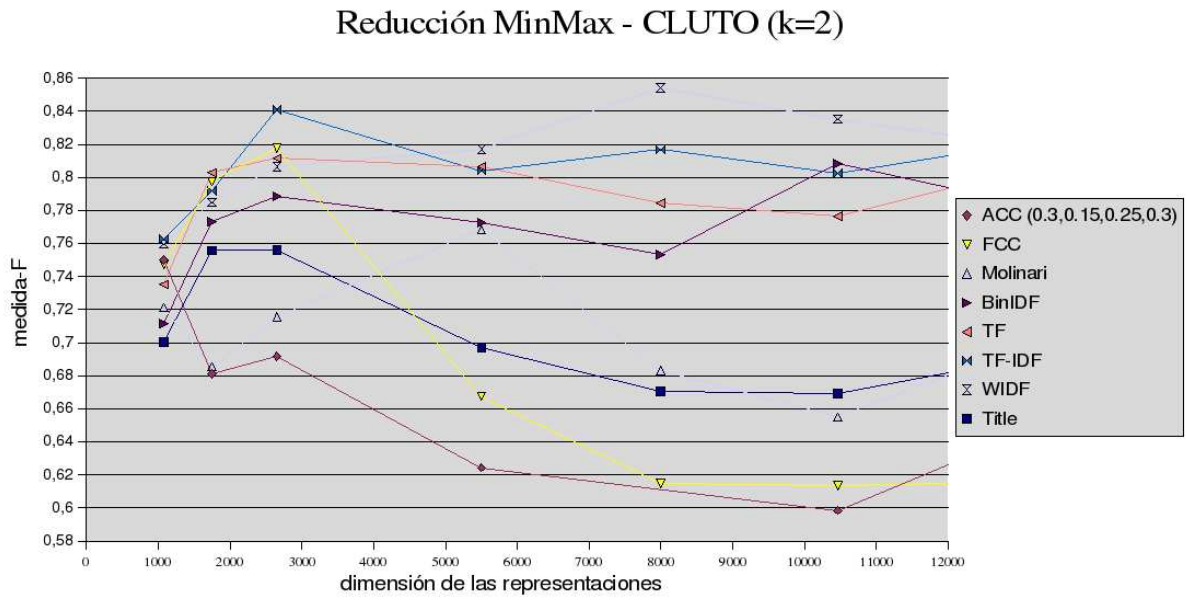

Figura 9.2: *Clustering* binario con la colección GH_1000 y reducción de rasgos MinMax.El *clustering* se realiza entre las clases cercanas semánticamente.

de este *clustering* empleando cada una de las funciones evaluadas como función de ponderación y con las reducciones PF y MinMax respectivamente.

Cuando se emplea la reducción PF (figura 9.3), las funciones que utilizan el etiquetado HTML –Molinari, ACC y FCC– obtienen similares resultados, siendo los mejores valores en medida-F a dimensiones bajas. TF y TF-IDF obtienen valores ligeramente inferiores, aunque TF-IDF presenta un comportamiento más oscilante, obteniendo el mejor valor absoluto de medida-F en dos puntos. Hay que destacar el buen resultado general de todas las funciones, salvo BinIDF, Title y WIDF que vuelven a presentar malos resultados en la evaluación del *clustering* cuando disminuye la dimensión de los vocabularios con los que se generan las representaciones.

Con la reducción MinMax (figura 9.4) las funciones ACC y FCC empeoran respecto a los valores obtenidos con la reducción PF. Las funciones TF y TF-IDF, por el contrario, obtienen valores de medida-F muy similares a los contenidos en ese caso. Cabe destacar el comportamiento de la función BinIDF, mientras que Molinari obtiene resultados malos con esta subcolección y reducción MinMax.

- **GJ_1000**. **Clustering binario** en el **nivel más alto de la jerarquía** y entre **clases lejanas semánticamente**. En las figuras 9.5 y 9.6 se muestran los resultados de este *clustering* empleando cada una de las funciones evaluadas como función de ponderación y con las reducciones PF y MinMax respectivamente.

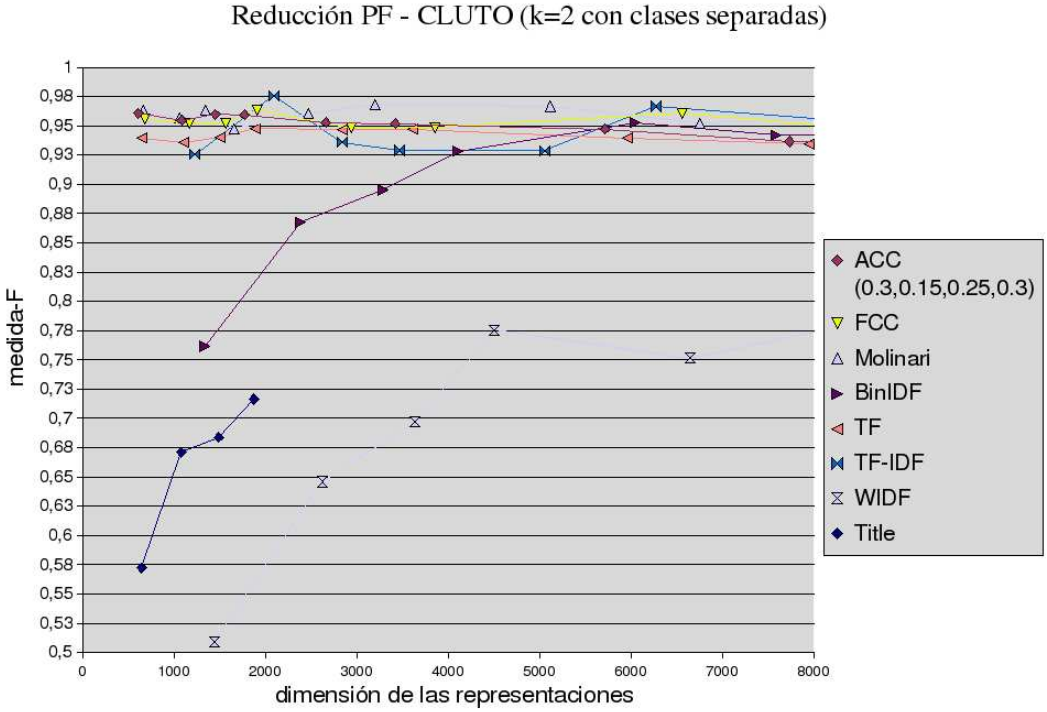

Figura 9.3: *Clustering* binario entre colecciones semánticamente lejanas, con la colección G&J_1000, k=2 y reducción con la propia función de ponderación. El *clustering* se realiza entre las clases "Astronomía" y "Deportes de Motor".

En el caso en el que se emplea la reducción PF (figura 9.5), los mejores resultados se obtienen con las funciones ACC y FCC, con valores de medida-F por encima de $0,95$, seguidas por la función TF, con valores en la medida-F similares y destacados frente al resto de representaciones.

Cuando se emplea la reducción MinMax (figura 9.6) se observa un comportamiento muy irregular en todas las funciones de proyección. En todos los casos, la diferencia entre los mejores valores de medida-F obtenidos y los peores es mayor que $0,1$. Para diferentes tamaños de vocabulario se obtienen diferentes comportamientos relativos en todas las funciones evaluadas. Esto puede ser debido a que al aplicar la reducción MinMax sobre esta colección se están eliminando determinados rasgos que pudieran resultar de alto contenido informativo.

- **ABC&DEF_1000**. **Clustering binario** en el **nivel más alto de la jerarquía** y entre **clases lejanas semánticamente**. En las figuras 9.7 y 9.8 se muestran los resultados de este *clustering* empleando cada una de las funciones de ponderación evaluadas y con las reducciones PF y MinMax respectivamente.

En el caso en el que se emplean las propias funciones de ponderación como método

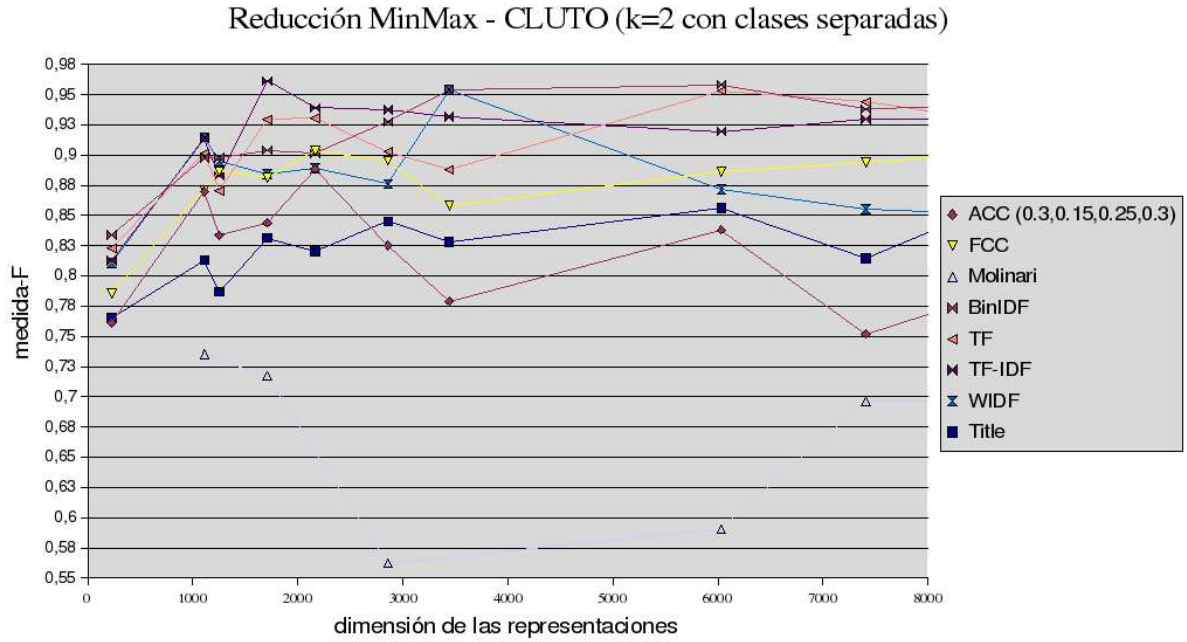

Figura 9.4: *Clustering* binario entre colecciones semánticamente lejanas, con la colección G&J_1000, k=2 y reducción MinMax.

de reducción de rasgos (figura 9.7) el mejor comportamiento se obtiene con las funciones ACC, FCC y TF, que obtienen valores de medida-F muy similares entre sí. Ligeramente inferiores resultaron los valores obtenidos por TF-IDF y BinIDF, mientras que con WIDF, Title y Molinari, por este orden, se obtuvieron los peores resultados con esta subcolección que representa un *clustering* binario sobre las dos superclases "Bancos y Finanzas" y "Lenguajes de programación". En el caso de Molinari los valores obtenidos fueron muy inferiores al resto y entorno al $0,7$.

Cuando se emplea la reducción MinMax se observa un comportamiento muy parecido con todas las funciones, así como muy similar al obtenido con esas mismas funciones y una reducción PF.

- *Clustering* a 3 clases

  - **ABC_1000**. **Clustering** en el **nivel más bajo de la jerarquía** y entre **clases cercanas semánticamente**, ya que todas pertenecen a la categoría "Bancos y Finanzas". En las figuras 9.9 y 9.10 se muestran los resultados de este *clustering* empleando cada una de las funciones de ponderación evaluadas como función de reducción de rasgos y con MinMax respectivamente.

    En ambos casos, los valores absolutos obtenidos son bastante peores que en los casos anteriores. Este *clustering* se realiza a 3 clases y salvo la función ACC, que en el

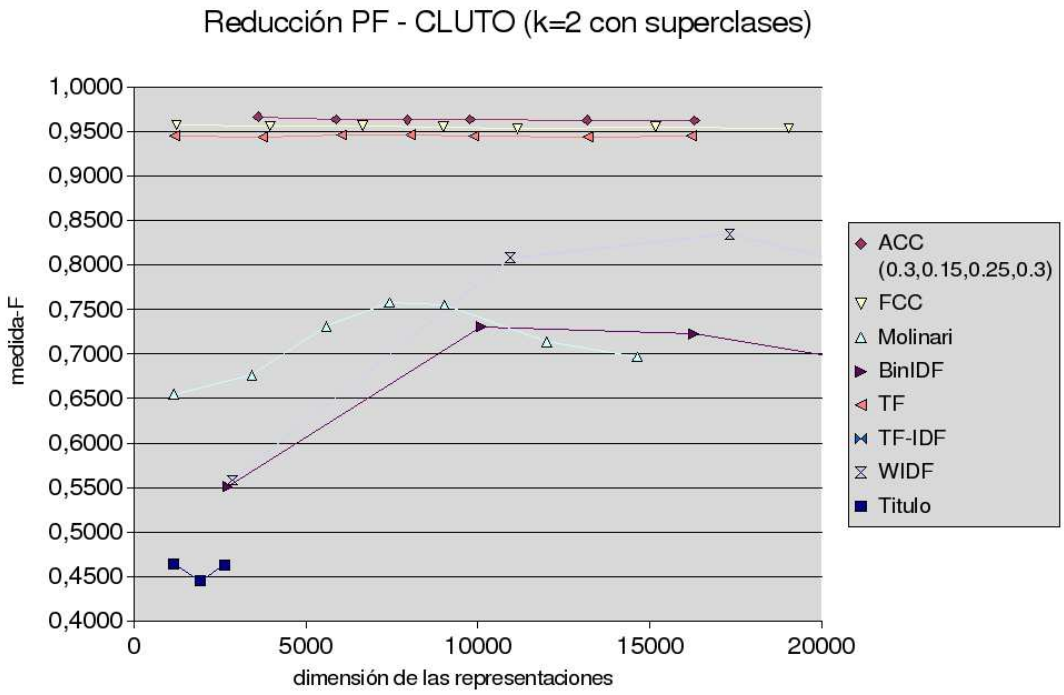

Figura 9.5: *Clustering* binario entre colecciones semánticamente lejanas, con la colección GJ_1000, k=2 y reducción con la propia función de ponderación. El *clustering* se realiza entre las superclases "Ciencia" y "Deportes".

caso de la reducción PF obtiene valores en torno al $0,8$, el resto de funciones obtiene valores en la medida-F muy bajos, entre $0,4$ y $0,68$ en todos los casos. Cuando se emplea la reducción de rasgos MinMax (figura 9.10) los resultados de la medida-F vuelven a ser muy cambiantes en todas las funciones de ponderación.

- *Clustering* a 6 clases

  - **ABC&DEF_1000**. **Clustering a 6 clases** en el **nivel más bajo de la jerarquía** y distinguiendo entre **clases cercanas y lejanas semánticamente**, ya que pertenecen 3 y 3 a la categoría "Bancos & Finanzas" y 3 a "Lenguajes de programación". En las figuras 9.11 y 9.12 se muestran los resultados de este *clustering* empleando cada una de las funciones de ponderación evaluadas como función de reducción de rasgos y con MinMax respectivamente.

    En el caso en el que se emplea la reducción PF (figura 9.11), las funciones ACC y FCC destacan frente al resto y son las únicas que obtienen valores de la medida-F en torno al $0,75$. El resto de funciones presentan valores más bajos, destacando negativamente las funciones Molinari y Title. La función Molinari, desarrollada para tareas de IR, presenta peores valores de medida-F según se aumenta el número de *clusters*.

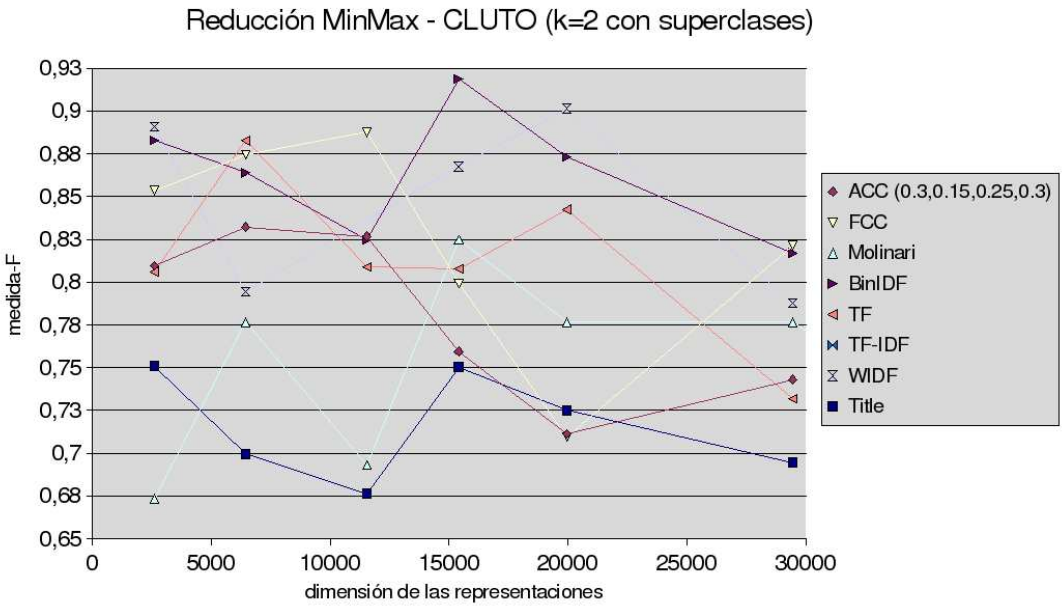

Figura 9.6: *Clustering* binario entre colecciones semánticamente lejanas, con la colección GJ_1000, k=2 y reducción MinMax. El *clustering* se realiza entre las superclases "Ciencia" y "Deportes".

Cuando se emplea la reducción MinMax (figura 9.12) los resultados obtenidos con todas las funciones de proyección son muy pobres, aunque las funciones ACC y FCC siguen siendo las más destacadas.

- *Clustering* a 10 clases

  - **AJ_1000**. **Clustering a 10 clases** en el **nivel más bajo de la jerarquía** y entre **clases cercanas y lejanas semánticamente**. En las figuras 9.13 y 9.14 se muestran los resultados de este *clustering* empleando cada una de las funciones de ponderación evaluadas como función de reducción de rasgos y con MinMax respectivamente. Este es el problema de *clustering* con más clases; se toma toda la colección BankSearch y se trata de agrupar en 10 clases, entre las que hay clases más cercanas y lejanas semánticamente. Los valores de medida-F no llegan a los obtenidos en algunos problemas de *clustering* binario, pero un valor $0,75$ puede considerarse aceptable en un problema de esta dificultad.

    Los resultados muestran que la función que mejor comportamiento presenta en este *clustering* con $k = 10$ es la función FCC, siendo la que obtiene los mejores resultados en todos los casos y con los dos tipos de reducción de rasgos empleados. La medida-F ha variado poco para las dimensiones de representación empleadas, con valores muy similares desde dimensiones del orden de 12000 rasgos hasta dimensiones en torno a

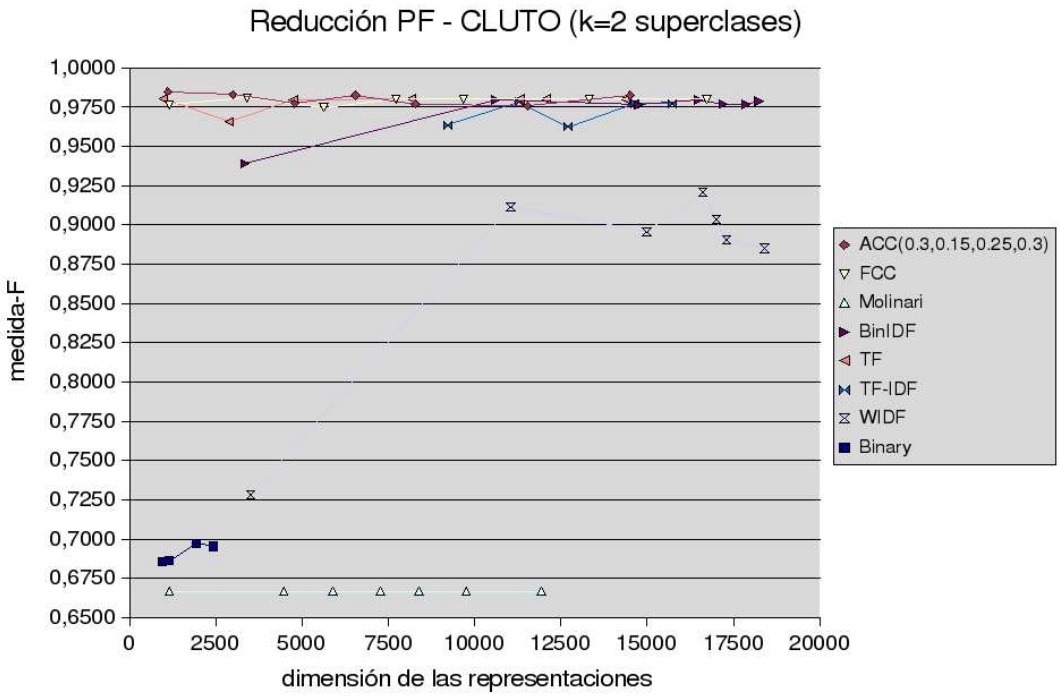

Figura 9.7: *Clustering* binario entre colecciones semánticamente lejanas, con la colección ABC&DEF_1000, k=2 y reducción con la propia función de ponderación. El *clustering* se realiza entre las superclases "Bancos y Finanzas" y "Lenguajes de Programación".

los 1000 rasgos; a partir de este tamaño, disminuye la calidad de los agrupamientos para todas las funciones. Tras la FCC, las funciones con mejor comportamiento para ambas reducciones son TF-IDF, ACC y TF.

La función BinIDF con reducción PF se comporta mejor para dimensiones altas de los vectores de representación, al igual que la función WIDF, acusando mucho la reducción de rasgos PF. Ambas funciones generan los vocabularios mayores y con ellas no fue posible crear representaciones de dimensión menor de 5000 rasgos. Esto es debido a que asignaron la mayor relevancia en un documento a más rasgos diferentes que el resto de funciones de proyección. En el caso de la función BinIDF, el descenso en el valor de la medida-F puede ser debido a que la relevancia que asigna a un rasgo dentro de un documento no se corresponde con información presente en el mismo, ya que la parte local de esta función es binaria. La relevancia que se asigna a cada rasgo con esta función viene dada por el factor IDF, que toma información únicamente de la colección. Este hecho se manifiesta en menor medida en el caso de vocabularios de gran tamaño, ya que estos se crean con un número elevado de rasgos de cada documento. Por otro lado, se puede observar que ambas funciones, BinIDF y WIDF, tienen un comportamiento más constante con la reducción MinMax (figura 9.14), que selecciona el vocabulario a partir de información de colección.

Las funciones ACC y TF tienen comportamientos similares, resultando mejores los

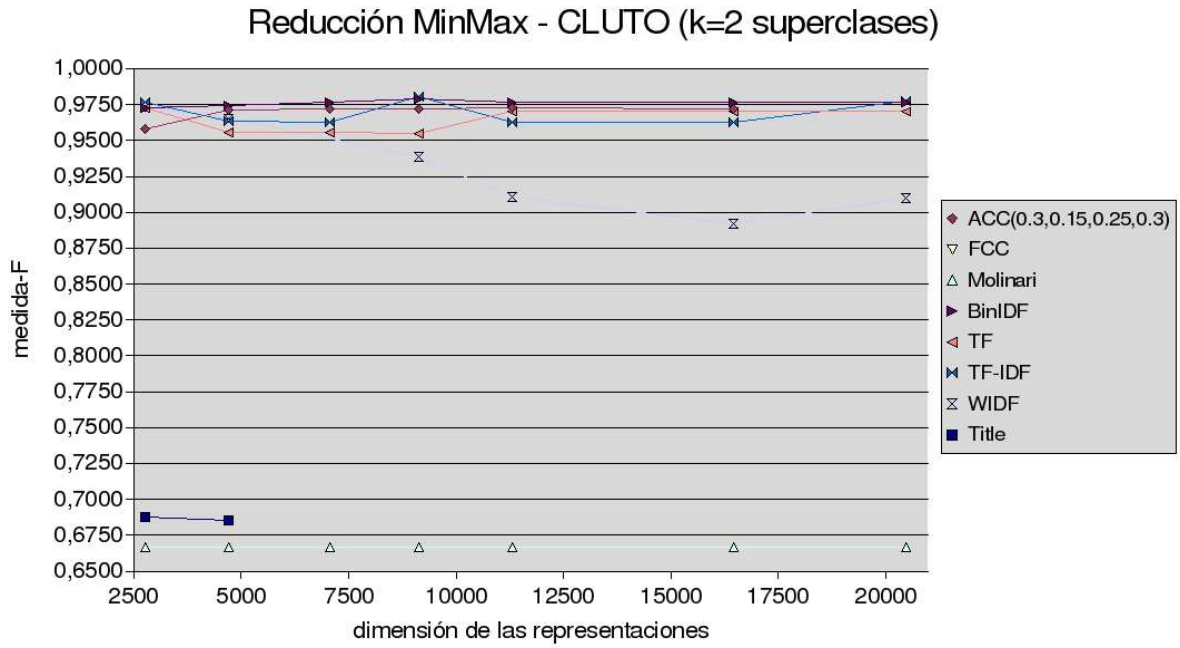

Figura 9.8: *Clustering* binario entre colecciones semánticamente lejanas, con la colección ABC&DEF_1000, k=2 y reducción MinMax. El *clustering* se realiza entre las superclases "Bancos y Finanzas" y "Lenguajes de Programación".

de ACC en el caso en el que se emplea la reducción PF y resultando mejor la función de proyección TF con la reducción MinMax. Los resultados que se obtienen con las funciones WIDF y Molinari son los peores utilizando ambas reducciones, siendo especialmente sorprendentemente bajos los valores de la medida-F para el caso de la función Molinari, lo que certifica el descenso en la calidad de los agrupamientos creados con esta función según se aumenta el número de *clusters*, es decir, con valores cada vez mayores de k.

9.6.2. Colección WebKB

- *Clustering* a 6 clases.

  - **WebKB_700**: **Clustering a 6 clases cercanas semánticamente**. En las figuras 9.15 y 9.16 se muestran los resultados de este *clustering* empleando cada una de las funciones de ponderación evaluadas como función de reducción de rasgos y con MinMax respectivamente. En este caso los resultados de *clustering* resultan muy pobres. Ninguna función obtiene un valor de medida-F por encima de $0,6$ con ninguna de las reducciones. Esta colección *webKB* es una colección que no presenta la misma heterogeneidad que la colección *BankSearch*, ya que las 6 clases pertenecen al contexto universitario. Así, podría considerarse como un problema de *clustering* con $k = 6$ y

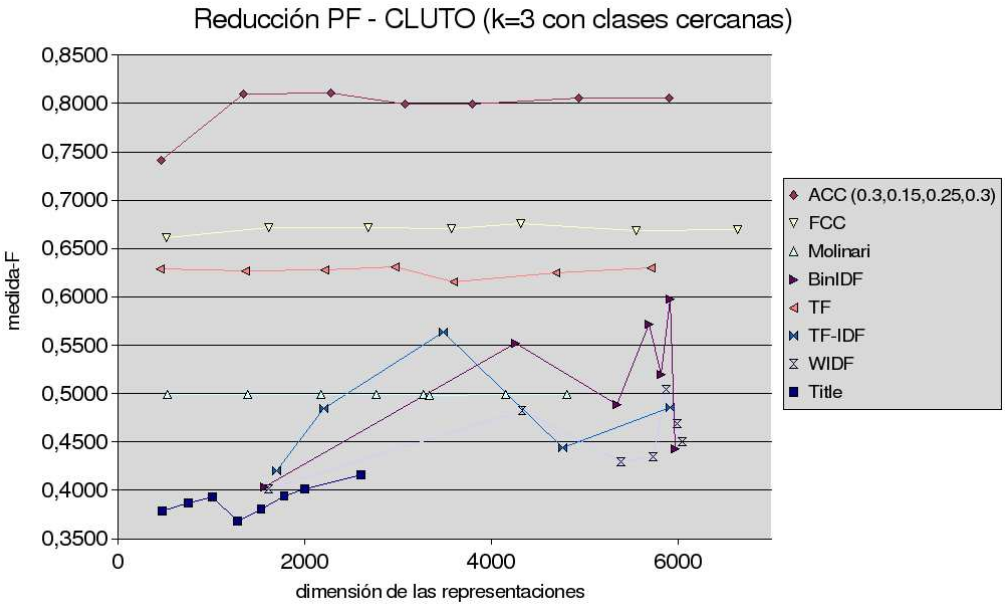

Figura 9.9: *Clustering* binario entre colecciones semánticamente cercanas, con la colección ABC_1000, k=3 y reducción con la propia función de ponderación. El *clustering* se realiza entre las clases "Bancos Comerciales", "Sociedades de crédito Hipotecario" y "Aseguradoras", clases pertenecientes a la superclase "Bancos y Finanzas".

clases cercanas semánticamente.

En el caso en el que se emplean las propias funciones de ponderación como método de reducción de rasgos (figura 9.15), la función que presenta mejor resultado es TF-IDF. Cabe destacar que la función Molinari tiene un comportamiento similar, e incluso mejor, que el resto de funciones y muy superior al valor que se obtenía con esta misma función en el *clustering* con $k = 6$ en la colección *BankSearch*.

Cuando se emplea la reducción MinMax (figura 9.16) los resultados generales obtenidos son aún peores. En este caso la función BinIDF es la que obtiene un mejor comportamiento general, seguida por la TF-IDF. Destaca el hecho de que la función Molinari no presenta decaimiento en los valores de la medida-F al reducirse las dimensiones de representación. La función Title tiene un comportamiento similar, aunque en menor medida. En este caso, FCC presenta un mal comportamiento con dimensiones bajas de las representaciones.

9.7. Conclusiones

En este capítulo se ha realizado una breve revisión de los métodos de *clustering* aplicados a documentos electrónicos, poniendo especial énfasis en su aplicación a páginas web. A continuación, se ha descrito el algoritmo empleado en la experimentación en *clustering* con las funciones de proyección, las colecciones y los métodos de reducción de rasgos seleccionados

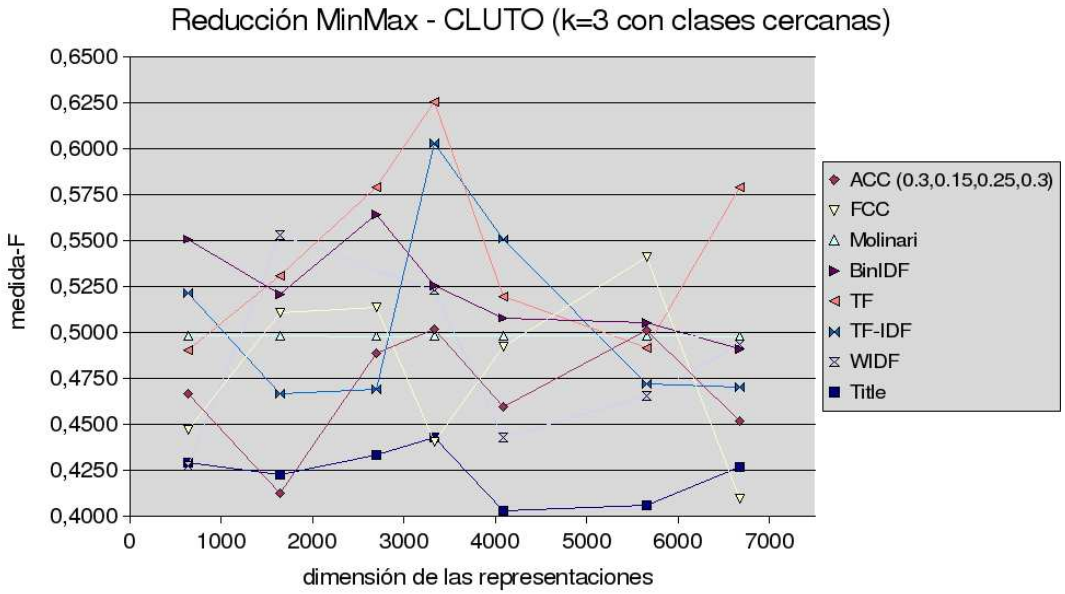

Figura 9.10: *Clustering* binario entre colecciones semánticamente cercanas, con la colección ABC_1000, k=3 y reducción MinMax. El *clustering* se realiza entre las clases "Bancos Comerciales", "Sociedades de crédito Hipotecario" y "Aseguradoras".

en el capítulo 7.

Una vez realizado el *clustering* sobre las diferentes subcolecciones de la colección *BankSearch* y sobre la colección *WebKB*, con ejemplos de *clustering* con $k = 2$, $k = 3$, $k = 6$ y $k = 10$ se pueden sacar las siguientes conclusiones generales.

Respecto de las funciones propuestas, el comportamiento general de FCC y ACC ha sido muy destacable en su aplicación a la colección *BankSearch*, obteniendo buenos resultados tanto en ejemplos de *clustering* binario como multiclase, obteniéndose para $k = 10$ el mejor agrupamiento con FCC. Destaca más aún este comportamiento si se compara únicamente con el resto de representaciones autocontenidas evaluadas (TF, Title y Molinari).

Por el contrario, con la colección *webKB* los resultados no fueron igualmente buenos. Esto se puede deber al hecho de que se trata de una colección con contenidos más homogéneos que *BankSearch*. Las heurísticas aplicadas en ACC y FCC pretenden ser generales, lo que puede perjudicar la representación cuando se tenga una colección como la *webKB*, de contenidos relativos a un entorno acotado, en este caso el universitario, ya que es posible que la presentación de estos contenidos siga determinados patrones debido al tipo de documentos que son.

Respecto a la reducción de rasgos, el método PF se ha mostrado más cercano a la naturaleza de las representaciones propuestas, ya que emplean más información que las frecuencias en el documento y en la colección. Siempre se han obtenidos mejores valores de medida-F para las funciones ACC y FCC cuando se empleaba la reducción PF. Por el contrario, las funciones clásicas TF, TF-IDF –y en menor medida, BinIDF– han obtenido valores similares con la aplicación de las reducciones PF y MinMax.

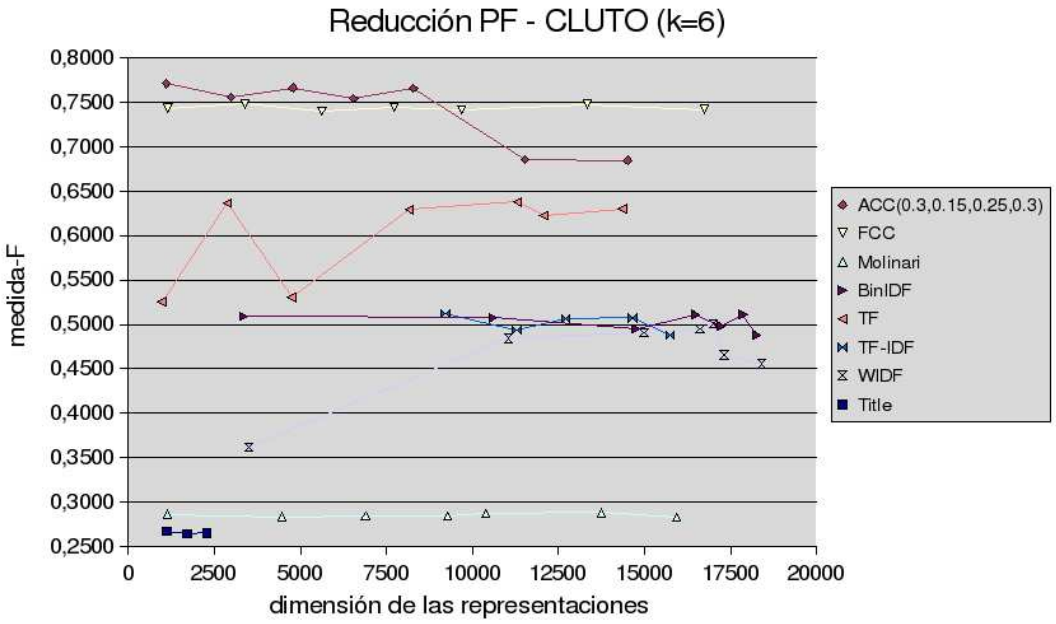

Figura 9.11: *Clustering* con k=6 y reducción con la propia función de ponderación. Colección formada por las clases "Bancos Comerciales", "Sociedades de Crédito Hipotecario", Aseguradoras", incluidas en la clase "Bancos y Finanzas",) y las clases "Java", "C/C++" y "Visual Basic", pertenecientes a "Lenguajes de Programación".

Con la reducción PF se han podido generar vocabularios más reducidos manteniendo la calidad del *clustering*. Por otro lado, cuando se reducían mucho los vocabularios con una función MinMax la calidad de los resultados descendía notablemente.

Analizando las funciones de ponderación, cabe destacar especialmente el comportamiento de las funciones FCC y ACC con reducción PF. En muchas de las colecciones han resultado ser las que ofrecían un *clustering* de mayor calidad. En todos los ejemplos de *clustering* binarios se han situado entre las funciones que han obtenido mejor evaluación, siendo en algunos casos las mejores. Además, este comportamiento se ha mantenido cuando se ha aumentado el valor de k, aumentando también la dificultad del problema. Dado que el *clustering* forma parte de un proceso de aprendizaje no supervisado, el hecho de que el comportamiento sea similar independientemente del número de *clusters* representa, por sí mismo, un aspecto muy importante en el ámbito de la web.

La función TF-IDF también ha obtenido buenos resultados tanto con reducción PF como con MinMax. Si bien esta función ha mejorado en términos generales a ACC y FCC con reducción MinMax, en el caso de aplicar una reducción PF las funciones basadas en combinaciones heurísticas de criterios han obtenido los mejores resultados, con valores absolutos de medida-F mayores que los obtenidos con TF-IDF y reducción MinMax. Por contra, TF-IDF se ha comportado bien con ambas colecciones, *BankSearch* y *webKB*. El resto de funciones han tenido un comportamiento general más discreto.

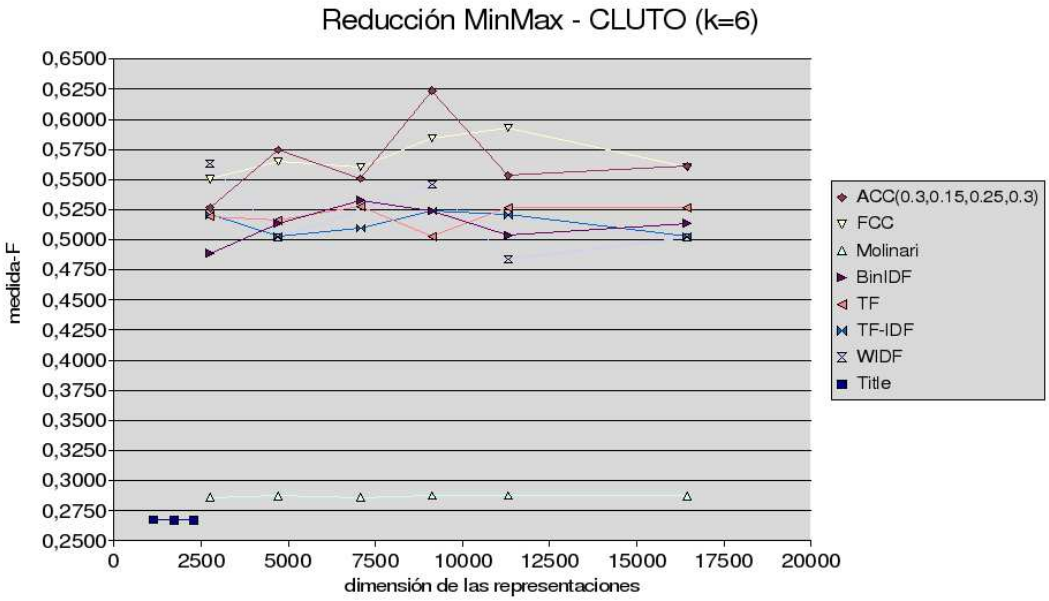

Figura 9.12: *Clustering* con k=6 y reducción MinMax. Colección formada por las clases "Bancos Comerciales", "Sociedades de Crédito Hipotecario", Aseguradoras", incluidas en la clase "Bancos y Finanzas", y las clases "Java", "C/C++" y "Visual Basic", pertenecientes a "Lenguajes de Programación".

La función Molinari, más orientada a su aplicación en tareas de IR, ha obtenido unos resultado bastante pobres en términos generales con la colección *BankSearch*, destacando negativamente cuando aumentaba el número de *clusters*. Sin embargo, con la colección *webKB* ha presentado un comportamiento mejor. En el caso de la reducción PF la calidad del *clustering* ha resultado bastante destacable, aunque no al nivel de la función TF-IDF. En el caso de la reducción MinMax, su comportamiento ha sido bastante estable, siendo la función que mejor realizaba el *clustering* en el caso de las dimensiones más reducidas. Sin embargo, sus valores de medida-F para dimensiones elevadas no fueron tan buenos. Este comportamiento tan diferenciado entre las colecciones puede deberse a las características intrínsecas de la colección *webKB*, con páginas pertenecientes a clases muy cercanas (todas del entorno universitario) y, posiblemente, con patrones de diseño bastante similares. Estos patrones pueden estar siendo capturados por alguna, o varias, de las 12 categorías que considera la representación Molinari. Por otro lado, y como era de esperar, los vocabularios reducidos usando la función Title son menores que los del resto, por lo que no es posible generar representaciones a dimensiones elevadas. Además, su comportamiento en problemas de *clustering* ha resultado bastante pobre.

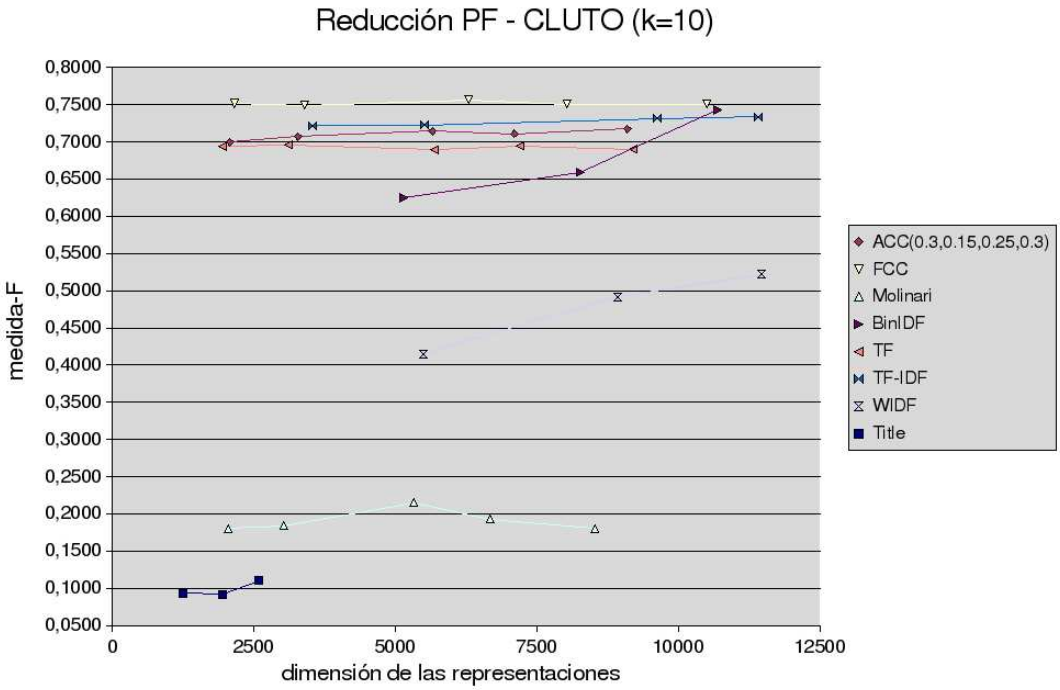

Figura 9.13: *Clustering* con k=10 y reducción con la propia función de ponderación. Colección *BankSearch* completa.

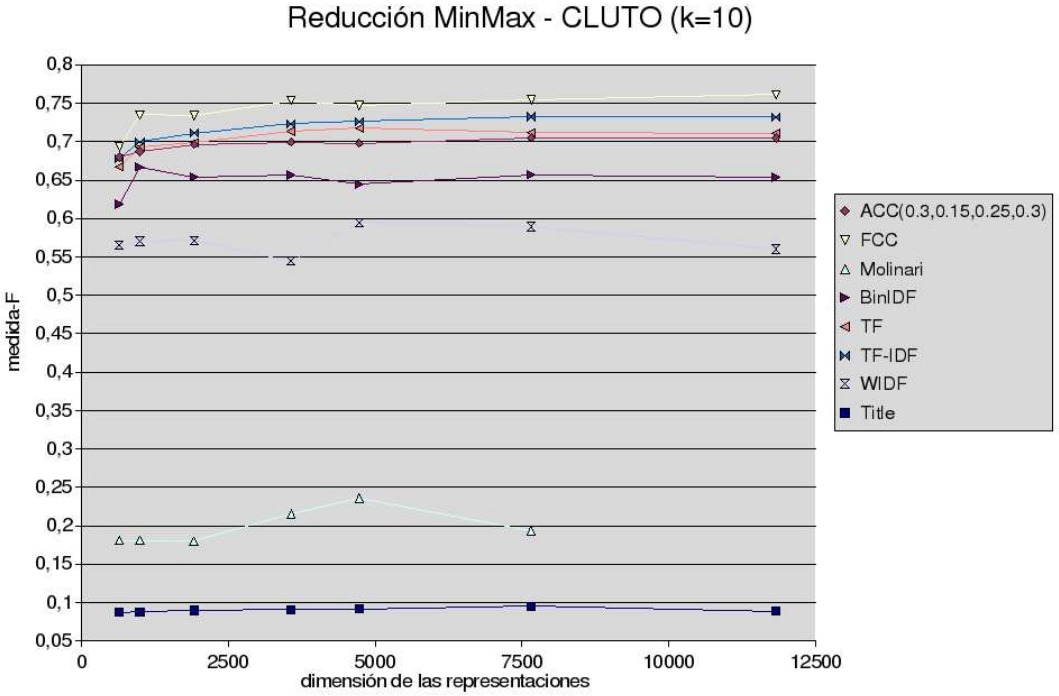

Figura 9.14: *Clustering* con k=10 y reducción MinMax. Colección *BankSearch* completa.

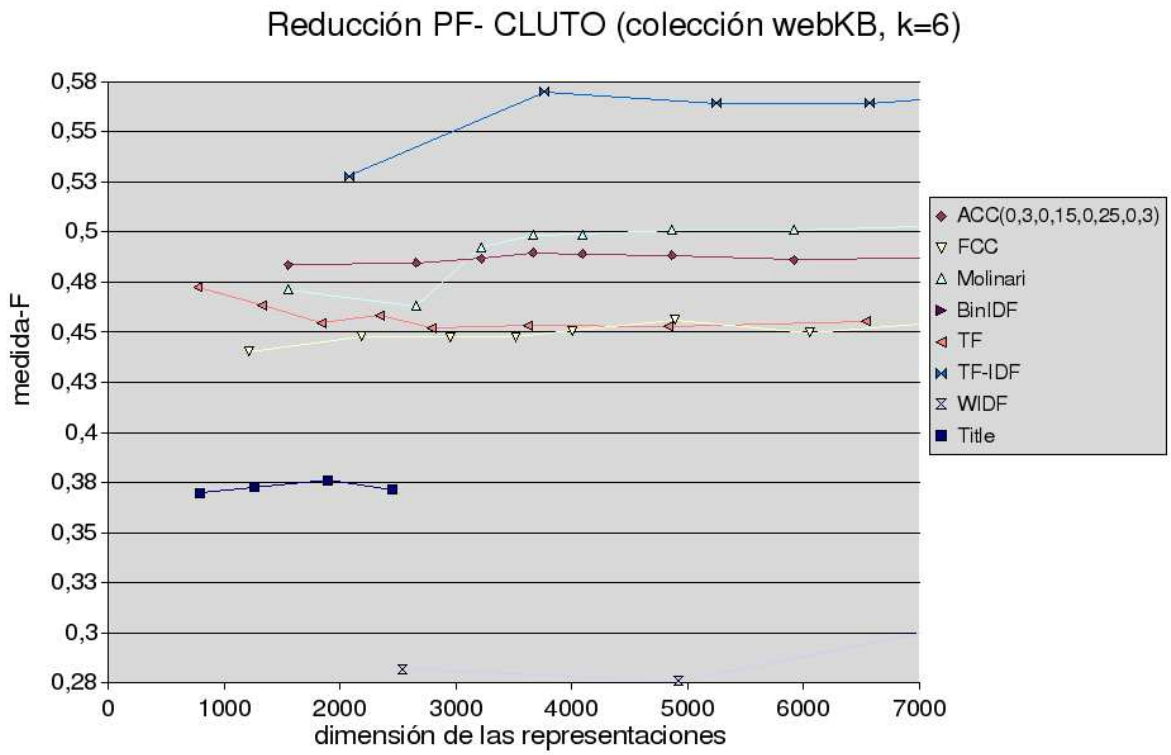

Figura 9.15: *Clustering* con k=6 entre clases cercanas semánticamente y con reducción con la propia función de ponderación. Colección *webKB*, considerada como colección homogénea.

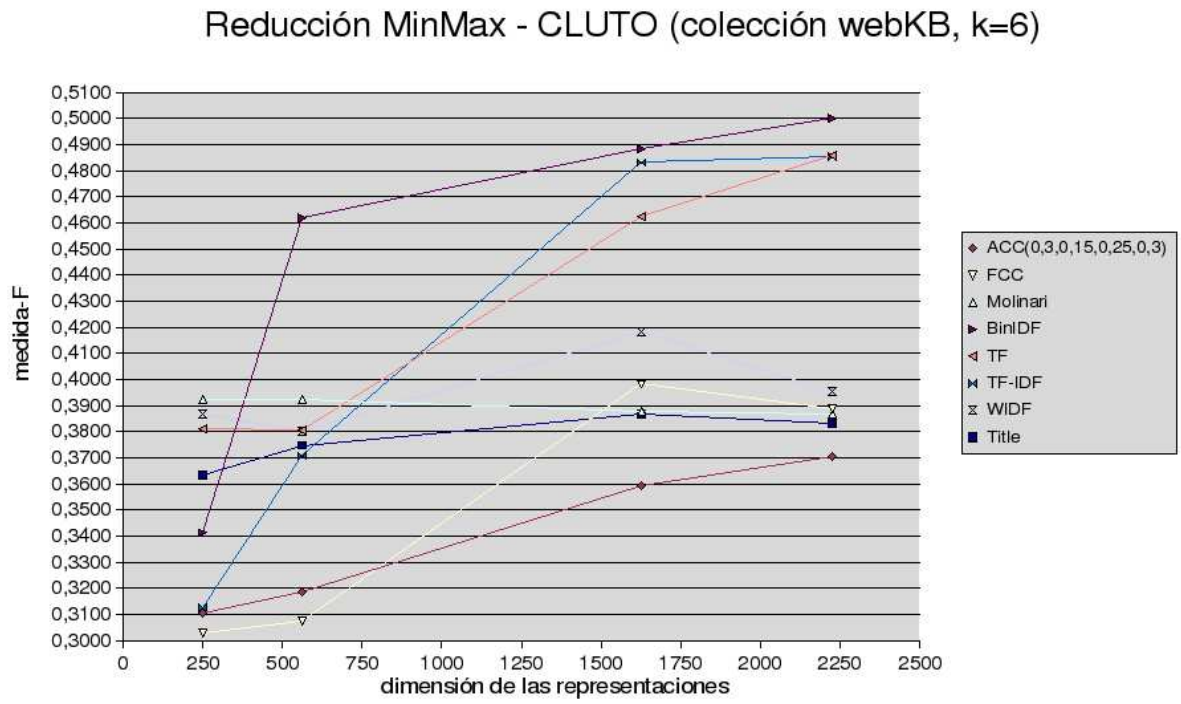

Figura 9.16: *Clustering* con k=6 entre clases cercanas semánticamente y con reducción MinMax. Colección *webKB*, considerada como colección homogénea.

Capítulo 10

Conclusiones, aportaciones y trabajo futuro

"Una vida sin reflexión no merece la pena ser vivida"
Sócrates

En este último capítulo se presentan las conclusiones y aportaciones generales derivadas de la investigación desarrollada en esta tesis doctoral. Además, se enumeran algunas de las líneas de investigación que quedan abiertas y que se pretenden cubrir en trabajos futuros.

10.1. Introducción

En esta tesis doctoral se ha abordado como objetivo general el desarrollo de un modelo de representación autocontenida de páginas web basado en una combinación de criterios heurísticos extraídos de los procesos de lectura y escritura de textos e hipertextos. Se han presentado dos funciones de ponderación, F, dentro de la definición de un modelo general de representación de documentos $\langle X, \mathbb{B}, \mu, F \rangle$. La diferencia entre ambas propuestas radica en el modo diferente en el que se realiza la combinación heurística de los criterios.

La representación automática de documentos HTML resulta un problema de gran relevancia en nuestros días, ya que supone la primera etapa a realizar en cualquier proceso de clasificación automática o *clustering* de páginas web. Estas tareas suponen problemas relevantes dentro del aprendizaje automático –supervisado y no supervisado– y tienen especial importancia hoy en día en el ámbito de Internet. La clasificación automática se puede aplicar en la construcción y mantenimiento de los directorios temáticos que ofrecen muchos de los portales web y motores de búsqueda empleados habitualmente en la Web. Por otro lado, el *clustering* de páginas web puede aplicarse tanto sobre el contenido de un servidor web, como sobre el conjunto de documentos recuperados por un motor de búsqueda tras una consulta.

10.2. Conclusiones y aportaciones principales

El objetivo propuesto inicialmente en esta tesis doctoral era encontrar un modelo de representación autocontenida, es decir, que no requiriera de información externa a la propia página web para representarla. Para ello, en el capítulo 2 se formalizó la representación automática de textos, identificando el conjunto de elementos que constituyen cualquier modelo de representación automática de documentos.

Se describieron el modelo de espacio vectorial y el índice de latencia semántica por tratarse de los modelos vectoriales más utilizados en representación automática de documentos, concluyendo que el primero resulta el más adecuado para obtener representaciones autocontenidas. La independencia que se asume entre componentes favorece la experimentación con combinaciones heurísticas de criterios, además de permitir el uso de funciones de proyección parametrizadas únicamente con información presente en la propia página que se desea representar. El LSI, por el contrario, requiere información externa a la página –y relativa a coapariciones entre rasgos– para poder establecer la representación de un documento.

Se mostraron diferentes funciones de ponderación –que dentro del modelo formal fueron denotadas como funciones de proyección F– tanto de carácter local como global, así como funciones de reducción de rasgos susceptibles de ser empleadas como funciones de ponderación. Se mostró que las funciones locales resultan las más adecuadas para una aplicación en representaciones autocontenidas; las globales requieren información externa al documento. Esta información suele estar relacionada con frecuencias inversas de documento en una colección y frecuencias absolutas de un rasgo en dicha colección. Si bien se ha argumentado que esta información podría hacer su cálculo muy costoso, visto el enorme tamaño y crecimiento que experimenta hoy en día la Web, dicha información podría tomarse de colecciones de referencia auxiliares de menor tamaño. En ese caso, se emplearían las frecuencias relativas de los rasgos en los vocabularios de dichas colecciones auxiliares. En ese caso, se corre el riesgo de que la colección auxiliar no sea suficientemente representativa de los contenidos de Internet.

Se vió que algunas funciones de reducción de rasgos no deberían de aplicarse en representaciones autocontenidas, ya que suelen estar condicionadas por el procesamiento posterior que se le vaya a dar a la representación. Estas funciones, usadas sobre todo en el contexto de la IR y TC, suelen estar parametrizadas con funciones de probabilidad dependientes de los algoritmos que se van a aplicar a continuación.

En el capítulo 3 se realizó una revisión de los procesos de análisis y representación de documentos HTML. Se presentaron en detalle las representaciones utilizadas por los principales motores de búsqueda, basadas principalmente en el análisis de la estructura de los *hyperlinks* que forma la Web. Se realizó también una revisión de los diferentes modelos de representación de documentos HTML: por contexto, uso y contenido. La conclusión que se extrajo fue que las representaciones empleadas en tareas de IR –las más estudiadas en la literatura– no tienen por qué ser las más adecuadas cuando se tiene un problema relativo a clasificación o *clustering* de

documentos, donde un análisis del contenido puede resultar mucho más adecuado.

Además, se observó que muchas de las representaciones combinaban información de contexto con información basada en contenido. En estos casos, las funciones empleadas en el análisis del contenido solían venir heredadas de métodos de representación aplicados a textos planos, donde se consideraban principalmente las frecuencias de aparición en el documento y en la colección. A partir de esta realidad se puede pensar que sería bueno desarrollar representaciones autocontenidas que utilizaran información de la estructura y el contenido HTML, y que pudieran aplicarse posteriormente en modelos de representación mixtos. Explorando por separado la estructura de hiperenlaces y el contenido de un documento HTML, es posible que puedan mejorar posteriormente las representaciones de tipo mixto.

En el capítulo 4 se presentó un marco teórico general en el que es posible desarrollar representaciones autocontenidas de documentos HTML basadas en combinaciones heurísticas de criterios. La hipótesis que subyace en este modelo es que las páginas web son documentos creados para ser leídos. A partir de esta idea se considera la lectura de un documento HTML como un proceso activo en el cual, tanto el autor de la página web como el lector aportan su experiencia y conocimiento al proceso de lectura/escritura. El vocabulario HTML tiene asociado un significado relacionado con aspectos de organización y presentación tipográfica de contenidos textuales. El modelo propuesto se basa en la suposición de que es posible extraer información de este vocabulario HTML, relativa a qué partes del documento han sido voluntariamente destacadas por el autor, y qué partes son las que más llaman la atención del lector cuando éste revisa una página en un proceso de lectura rápido.

Las representaciones propuestas en esta tesis se generan tras una secuencia de fases que termina en una evaluación de un sistema de reglas donde se trata de combinar un conocimiento heurístico. La representación ACC combina linealmente una serie de criterios heurísticos definidos a priori (capítulo 5). En este caso, el conocimiento heurístico deberá estar implícito en la elección de los coeficientes de la combinación analítica. En el caso de la representación FCC, al tratarse de un sistema de reglas borrosas, se puede definir una base de conocimiento donde se almacenen las heurísticas mediante conjuntos de reglas IF-THEN (capítulo 6). A la simplicidad conceptual que supone la representación FCC cabe añadir la facilidad en el diseño del conjunto de reglas, siempre que los criterios a combinar resulten familiares a cualquier lector.

La evaluación de las representaciones propuestas (ACC y FCC) se realizó mediante procesos de clasificación y *clustering* de páginas web. En el caso de la clasificación automática (capítulo 8) se empleó un algoritmo *Naïve Bayes* que destaca por su simplicidad, a la que hay que unir unos resultados de clasificación bastante aceptables. En el caso del *clustering* (capítulo 9) se empleó un algoritmo de partición perteneciente a la librería CLUTO.

Tras un análisis de los resultados de clasificación, se muestra un comportamiento bastante aceptable para las representaciones propuestas. De entre las representaciones autocontenidas obtienen los mejores resultados en términos generales (mejoran en casi todos los casos a las

representaciones Title, Molinari y TF), aunque no suponen las mejores representaciones en el conjunto total de funciones evaluadas.

Si bien es cierto que en algunos casos se ven superadas por otras funciones de proyección aplicadas en el ámbito de la representación de textos planos (WIDF y BinIDF), estas funciones son de carácter global y, por tanto, requieren de una información de colección previamente capturada. Además, estas funciones han mostrado comportamientos muy irregulares en el conjunto total de colecciones evaluadas. Por otro lado, habría que analizar su comportamiento en problemas reales, cuando las frecuencias necesarias para el cálculo de los factores de frecuencia inversa de documento fueran calculados con colecciones diferentes.

Respecto al tipo de clasificación, binaria o no binaria, el comportamiento de todas las representaciones empeoró según aumentaba el número de clases. Si bien los valores de la medida-F para todas las funciones y representaciones evaluadas son aceptables en el caso de la clasificación binaria, estos bajan sustancialmente al aumentar el número de clases consideradas.

La función de probabilidad que ha ofrecido mejores resultados ha sido la función Multinomial. Esto hace pensar que la hipótesis asumida en el aprendizaje gaussiano no sea muy acertada cuando el conjunto de categorías aumenta, ya que el número de documentos con los que se calculan los estadísticos μ y σ disminuye. Por otro lado, en el caso del aprendizaje Multinomial hay que destacar que sólo considera la frecuencias de aparición en el documento y en la colección, por lo que en principio no beneficia a las representaciones basadas en el análisis del etiquetado HTML, ya que no tiene en cuenta esta información adicional en la fase de entrenamiento del clasificador.

En el caso del *clustering*, el comportamiento de ACC y FCC resultó mucho más destacable. Con la colección *BankSearch*, y en términos generales, los resultados de las representaciones propuestas fueron mejores que los del resto de representaciones evaluadas, ya fuera en un *clustering* a dos, tres, seis o diez *clusters*. Dado que se trata de un proceso de aprendizaje no supervisado, el hecho de que el comportamiento de FCC y ACC sea similar independientemente del número de *clusters* ratificaría la conclusión acerca de la calidad del aprendizaje en TC y resulta, por sí mismo, un aspecto muy importante en el ámbito de la web.

Por otro lado, si bien en ciertos casos las representaciones FCC y ACC no han obtenido los mejores resultados absolutos, en muchas ocasiones lo han hecho para las dimensiones de representación más reducidas, lo que supone un hecho altamente destacable. A esto hay que añadirle que, en la mayoría de los casos, han mantenido los valores de la medida-F, tanto en dimensiones de representación grandes como en reducidas, por lo que se puede encontrar una misma calidad de *clustering* empleando vectores de representación más reducidos, lo que disminuye siempre el coste computacional. Por tanto, se han obtenido representaciones autocontenidas de dimensión reducida y con un comportamiento aceptable en tareas de TC y muy destacable en DC.

Respecto al análisis comparativo entre ACC y FCC, en general, la representación creada

con la función ACC obtuvo mejores resultados en los problemas de clasificación automática. Por el contrario, la combinación borrosa de criterios, FCC, ha resultado mejor que ACC para problemas de *clustering* de páginas web.

Un aspecto a destacar de la representación ACC es que los coeficientes de su combinación resultan independientes de las colecciones sobre las que se pueda aplicar. Se han encontrado en la literatura ejemplos de representaciones que podrían ajustarse al modelo teórico presentado en la formalización de la representación ACC pero, en esos caso, la selección de categoría que emplean no se ajusta a criterios heurísticos claros y diferenciables, lo que dificulta una asignación de ponderaciones independiente a la colección.

La función Molinari, presentada en (Molinari et al., 2003) y pensada para problemas de IR, podría expresarse teóricamente como una función ACC donde se hubieran definido 12 criterios correspondientes a las 12 clases que aparecen en la tabla 3.1. En este caso, los coeficientes de la combinación se calculan por medio de una función lineal que reparte el peso entre los 12 criterios considerados, en lugar de tratar de reflejar alguna heurística. El comportamiento de esta función Molinari no ha sido muy destacado en la colección *BankSearch*, lo que pone de manifiesto que cualquier combinación de criterios no tiene por qué resultar buena, sino que habrá que definir con sumo cuidado los criterios a combinar y sus coeficientes dentro de la combinación.

En el caso de la representación FCC, la combinación de criterios resulta también completamente independiente de la colección de páginas web sobre la que se esté trabajando. Además, puede realizarse de una manera sencilla y natural. En este caso es fácil pensar en trasladar las mismas heurísticas utilizadas en esta tesis, u otras que se nos pudieran ocurrir, a otros documentos electrónicos escritos con otros lenguajes de marcado.

Tanto la experimentación en TC como en DC se ha realizado empleando como parámetro de evaluación la dimensión de las representaciones. En este punto cabe destacar que las representaciones propuestas, además de resultar completamente autocontenidas, presentan un comportamiento muy estable, ya sea con un tamaño pequeño de vocabulario o con una dimensión elevada. Esta característica las hace especialmente adecuadas para su aplicación en el contexto web. Por otro lado, el hecho de ser representaciones autocontenidas hace que sean completamente aplicables a representaciones mixtas.

Otro aspecto a tener en cuenta es el hecho de que la fase de representación en una tarea de *clustering* puede tener más peso que en un proceso de clasificación automática basado en aprendizaje automático. La razón principal es que la fase de aprendizaje puede estar influyendo de manera sustancial en el resultado final de la clasificación. En el caso del *clustering*, la función de similitud representa una métrica aplicada directamente sobre los vectores de representación, por lo que la medida de la calidad del *clustering* está más cercana a la calidad de la representación de los documentos.

Por último, un aspecto muy importante de estas propuestas es que podrían generalizarse a otros lenguajes de marcado procedimentales, si se quieren emplear las mismas heurísticas,

o a lenguajes estructurales si se definen criterios nuevos. Bastaría con encontrar criterios de asignación de relevancia a rasgos, asociarlos con subconjuntos de elementos del vocabulario del lenguaje que se esté considerando y, a continuación, combinarlos en base a un conocimiento heurístico adquirido previamente. De este modo, la aplicación sobre lenguajes documentales como LaTex o el formato RTF son directas, pero también podría serlo sobre vocabularios XML como DocBook.

En resumen, las aportaciones principales de esta tesis doctoral son:

1. Revisión del estado del arte en representación automática de textos.
   - *a*) Formalización teórica de los modelos de representación automática de textos.
   - *b*) Revisión de las funciones de ponderación y reducción de rasgos aplicadas dentro del VSM y LSI
2. Revisión del estado del arte en representación automática de páginas web.
3. Establecimiento de un modelo teórico general para la representación autocontenida de documentos HTML basada en combinaciones heurísticas de criterios.
   - *a*) Formalización de una representación basada en una combinación lineal de criterios (representación con función de proyección ACC).
   - *b*) Formalización de una representación basada en combinación borrosa de criterios (representación con función de proyección FCC).
4. Aplicación de un algoritmo *Naïve Bayes* para la clasificación automática de páginas web basadas en contenido en las colecciones de referencia *BankSearch DataSet* y *WebKb*.
   - *a*) Establecimiento de dos funciones de probabilidad gaussianas para un algoritmo de clasificación Naïve Bayes.
   - *b*) Establecimiento de una etapa de aprendizaje basada en el Teorema Central del Límite.
5. Aplicación del algoritmo *k-way via Repeated Bisectons* al *clustering* de páginas web basadas en contenido en las colecciones de referencia *BankSearch DataSet* y *WebKb*.

10.3. Resultados parciales obtenidos

A continuación, se relacionan los artículos publicados, tanto en revistas, como en capítulos de libros y congresos nacionales e internacionales generados como parte de la difusión de los resultados parciales obtenidos durante el desarrollo de esta tesis doctoral. Se presentan en orden cronológico:

- V. Fresno, A. Ribeiro. **"Features selection and dimensionality reduction in web page representation"**. *International ICSC Congress on Computational Intelligence: Methods and Applications*. Bangor, Wales (U.K.). June 2001. pp 416-421. ISBN 3-906454-26-6.

  En este artículo (Fresno y Ribeiro, 2001a) se presenta la representación de páginas web propuesta en esta tesis basada en una combinación analítica de criterios (ACC), como función de reducción de la dimensión de la representación de páginas web dentro del VSM. Se presenta un estudio estadístico para establecer los valores de los coeficientes para los factores de la combinación analítica, a la vez que se presentan funciones para evaluar cada uno de los criterios heurísticos considerados. Dicha evaluación se realiza en comparación con la representación clásica de la bolsa de palabras (TF). El análisis de los resultados indica que la combinación analítica propuesta reduce la dimensión de la representación, a la vez que captura un porcentaje mayor de términos altamente informativos, evaluados con un clasificador *Naïve Bayes*.

- A. Ribeiro, V. Fresno. **"A Multi Criteria Function to Concept Extraction in HTML Environment"**. *International Conference on Internet Computing 2001 (IC'2001)*. June 2001. Vol I, pp 1-6. ISBN 1-892512-82-3

  En este artículo (Ribeiro y Fresno, 2001) se presenta la representación ACC, propuesta en esta tesis, como sistema de extracción de términos en páginas web, y basada en una combinación lineal de criterios heurísticos. Se realiza una comparación con una herramienta comercial, *Copernic Summarizer*[1], que permite la extracción de los términos más relevantes de una página web, así como la generación automática de resúmenes. La comparativa se realizó en términos de *tamaño* y *homogeneidad/heterogeneidad* en los contenidos de las páginas. Los resultados experimentales de este trabajo muestran un mejor comportamiento de la representación propuesta frente a la que ofrecía la herramienta comercial. La evaluación se realizó en base a una función multicriterio propuesta.

- A. Ribeiro, V. Fresno, M. García-Alegre and D. Guinea. **"A Fuzzy System for the Web page Representation"**, en *"Intelligent Exploration of the Web"*. P. S. Szczepaniak, J. Segovia, J. Kacprzyk, L. A. Zadeh - editors. Springer-Verlag Group (Physica-Verlag-Heidelberg) 2002. pp 19-38. ISSN 1434-9922. ISBN 3-7908-1529-2

  En este trabajo (Ribeiro et al., 2002) se presenta el método de representación propuesto en esta tesis en forma de combinación de criterios mediante un sistema de reglas borrosas (FCC). Se realiza una comparativa entre una combinación analítica y una combinación de criterios borrosa. Se detallan las decisiones tomadas durante el diseño del sistema, tanto en la definición del conjunto de variables lingüísticas a considerar, como en el conjunto de reglas a evaluar; y en las fases de borrosificación y desborrosificación. La evaluación de la

[1] http://www.copernic.com

calidad de la representación se realiza mediante un algoritmo de clasificación automática bayesiana en una colección de documentos pre-clasificados en dos categorías: *Medicina-Farmacología* y *Tecnología-Aeroespacial*. El algoritmo de clasificación propuesto supone una variante a las funciones típicamente empleadas con clasificadores *Naïve Bayes*. Los resultados indican que una combinación borrosa puede capturar mejor la intención del autor. Los resultados obtenidos en clasificación son mejores que en el caso de emplear una combinación analítica.

- V. Fresno, A. Ribeiro.**"Una representación Fuzzy para páginas web"**. *XI Congreso Español sobre Tecnologías y Lógica Fuzzy (ESTYLF'2002)*. León, España. Septiembre 2002. Actas del XI Congreso Español sobre Tecnologías y Lógica Fuzzy (*ESTYLF'2002*). ISBN 84-7719-933-7

  En este artículo (Fresno y Ribeiro, 2001b), a partir de varios conjuntos de etiquetas HTML, se definen criterios heurísticos que permiten establecer la relevancia de cada palabra dentro de una página. Se proponen dos formas de combinación de estos criterios: una analítica (ACC) y otra borrosa (FCC). Tanto el método como las dos propuestas de combinación se evalúan, comparándolas con la representación TF, utilizando la representación en una tarea de aprendizaje, y posterior clasificación de páginas web. Los resultados experimentales muestran que la representación FCC reporta el mejor comportamiento en clasificación bayesiana binaria, y con una colección de prueba pre-clasificada manualmente. Una de las principales ventajas que se destacan en este trabajo es que las representaciones evaluadas no requieren conocimiento previo del dominio, lo que las hace muy apropiadas para el contexto de la Web.

- A. Ribeiro, V. Fresno, M. C. García-Alegre and D. Guinea. **"Web Page Classification: A Soft Computing Approach"** en *"Advances in Web Intelligence"*. Springer-Verlag in the series of Lecture Notes in Artificial Intelligence (LNAI 2663) ISSN 0302-9743. ISBN 3-540-40124-5

  En este artículo (Ribeiro et al., 2003) se presenta un sistema de TC basado en un sistema de reglas borrosas. Este clasificador se fundamenta en la representación FCC presentada en esta tesis. Las etiquetas lingüísticas utilizadas en el proceso de representación pasan ahora a formar parte del sistema de clasificación, y mediante una búsqueda con algoritmos genéticos se establece el conjunto de reglas que forman la base de conocimiento del sistema de clasificación borroso. Este trabajo supuso una primera aproximación a la aplicación de este tipo de clasificadores a nuestra representación. Los resultados de clasificación fueron comparados con los obtenidos con el algoritmo bayesiano empleado en trabajos anteriores, resultando ser peores con la misma colección de referencia.

- V. Fresno and A. Ribeiro. **"An Analytical Approach to Concept Extraction in HTML Environments"**. *Journal of Intelligent Information Systems - JIIS*. Vol 22(3).

215-235. Kluwer Academic Publishers. ISSN:0925-9902.

En este artículo (Fresno y Ribeiro, 2004) se exponen conjuntamente, y de forma detallada, todos los trabajos realizados hasta ese momento para desarrollar el modelo de representación de páginas web basado en la combinación analítica de criterios (ACC). Se presentan en detalle las funciones de caracterización correspondientes a cada uno de los criterios considerados, se muestra y analiza el estudio estadístico realizado sobre reducción de la dimensión de las representaciones, la comparativa del método frente a la herramienta comercial *Copernic Summarizer* y, por último, el método de representación se evalúa en función de un algoritmo de clasificación *Naïve Bayes*. Se analizan también los resultados en clasificación para cada uno de los criterios por separado, para así poder extraer conclusiones que apoyen las heurísticas tomadas sobre la elección de los coeficientes de la combinación analítica.

- Arantza Casillas, Víctor Fresno, M. Teresa González de Lena and Raquel Martínez. **" Evaluation of Web Page Representations by Content through Clustering"**. *Eleventh Symposium on String Processing and Information Retrieval (SPIRE'2004)*. Springer-Verlag in the series of Lecture Notes in Computer Science (LNCS 3246) ISSN 0302-9743, ISBN 3-540-23210-9.

  En este trabajo (Casillas et al., 2004a) se presentan los resultados de evaluar siete diferentes métodos de representación de páginas web por medio de un proceso de *clustering*. Se comparan cinco funciones de pesado bien conocidas en el campo de la representación de textos (Binaria, Binaria-IDF, TF, TF-IDF y WIDF)(Salton, 1988) -aplicadas sobre el contenido de las páginas web-, con las dos representaciones propuestas en esta tesis (ACC y FCC), que toman en cuenta información extraídas del marcado HTML. Los experimentos fueron evaluados por medio de la medida F y la entropía.

- Víctor Fresno Fernández and Luis Magdalena Layos. **Text Content Approaches in Web Content Mining"**. "Encyclopedia of Data Warehousing and Mining". 1103-1108. Editor: John Wang, Montclair State University. Idea Group Inc. ISBN 1-59140-557-2.

  En este artículo (Fresno y Magdalena, 2005) se da un repaso a la representación de páginas web basadas en contenido. A su vez, se presenta una comparativa entre diferentes representaciones de páginas web por medio de un proceso de clasificación *Naïve Bayes* binario. Se comparan las funciones Binaria, Binaria-IDF, TF, TF-IDF y WIDF, aplicadas sobre el contenido de las páginas web, con las representaciones ACC y FCC propuestas en esta tesis. La colección sobre la que se evalúa es un subconjunto de 2000 páginas de la colección de Sinka y Corne. Los resultados fueron evaluados en función de la tasa de acierto en clasificación.

- Arantza Casillas, Víctor Fresno, Raquel Martínez and Soto Montalvo. **"Evaluación del**

***clustering* de páginas web mediante funciones de peso y combinación heurística de criterios"**. *XXI Congreso de la Sociedad Española para el Procesamiento del Lenguaje Natural (SEPLN)*. Granada, España, 14-16 de Septiembre. Revista de la Sociedad Española para el Procesamiento del Lenguaje, 35, 417-424. ISSN 1135-5948.

Este trabajo (Casillas et al., 2005) supone una ampliación y mayor formalización sobre el trabajo publicado en el *Eleventh Symposium on String Processing and Information Retrieval (SPIRE'2004)*.

10.4. Trabajos futuros

En esta sección se esbozan los trabajos futuros que se pretenden desarrollar para dar continuidad a las líneas de investigación abordadas durante este periodo de tesis doctoral:

- Aplicación del modelo teórico general de representación autocontenida de páginas web basada en combinaciones heurísticas de criterios empleando como rasgos términos multipalabra y entidades nombradas. Para ello, habría que definir unas nuevas fases de preproceso (selección de vocabulario), procesamiento (captura de información de criterio) y combinación de conocimiento heurístico.

- Aplicación del modelo teórico general a otros lenguajes de marcado, por ejemplo, diferentes vocabularios XML como DocBook. Si se quiere aplicar sobre lenguajes documentales como LaTex, las heurísticas empleadas en este trabajo se podrían trasladar directamente. En el caso de vocabularios más complejos, como DocBook, cabría ampliar la definición de criterios heurísticos y los detalles de su posterior combinación.

- Aplicación de las representaciones ACC y FCC a otros algoritmos de clasificación y *clustering* de documentos; sería interesante probar la calidad del modelo de representación con un clasificador*Support Vector Machines* o por medio de mapas auto-organizativos.

- Inclusión de las representaciones ACC y FCC en representaciones de páginas web mixtas, combinando la información por contenido que ofrecen con otra información de contexto, empleando análisis de correferencias y los conceptos de *authorities* y *hubs*. En este caso, se trataría de emplear las funciones ACC y FCC como parte local dentro de funciones de ponderación global.

- Aplicación de las funciones de ponderación propuestas, ACC y FCC, dentro del modelo LSI para tratar de enriquecer la información de coaparición entre rasgos en una colección. El objetivo sería transformar la información de coaparición entre rasgos por otra información de co-relevancia, comprobando si se produce con ello un enriquecimiento en la representación.

Otro aspecto importante es la posible generalización del modelo a documentos XML, y su posible aplicación en el contexto de la Web Semántica. A partir de un vocabulario XML dado se pueden establecer correspondencias, al igual que se ha hecho con HTML, entre determinados elementos del lenguaje y una serie de criterios que quieran ser considerados. A continuación, la información recogida en base a estos criterios podrá ser combinada, tanto con una combinación lineal, como por medio de un sistema de reglas borrosas.

Bibliografía

Abaitua, J. (2005). Curso sobre SGML (publicación electrónica, http://www.serv-inf.deusto.es/abaitua/konzeptu/sgml.htm).

Abu-Salem, H., Al-Omari, M., y Evens, M. (1999). Stemming Methodologies over Individual Queries Words for an Arabian Information Retrieval System. *Journal of the American Society for Information Science and Technology*, 50:524–529.

Aebli, H. (1988). *Doce formas básicas de enseñar una didáctica basada en la psicología*. Narcea, Madrid.

Agirre, E. y Lopez de Lacalle, O. (2003). Clustering Wordnet Word Senses. In *In Proceedings of the Conference on Recent Advances on Natural Language Processing (RANLP '03)*.

Agrawal, R., Gehrke, J., Gunopoulos, D., y Raghavan, P. (1998). Automatic Subspace Clustering of High Dimensional Data for Data Mining Applications. In *In Proceedings of the ACM SIGMOD Conference*, pages 94–105.

Aha, D. W. (1990). *A Study of Instance-based Algorithms for Supervised Learning Tasks: Mathematical, Empirical and Psychological Evaluations*. PhD thesis, Department of Information & Computer Science, University of California, Irvine.

Allan, J., Leouski, A., y Swan, R. (1997). Interactive Cluster Visualization for Information Retrieval. Technical Report IR-116, University of Massachusetts.

Almuhareb, A. y Poesio, M. (2004). Attribute-Based and Value-Based Clustering: An Evaluation. In Lin, D. y Wu, D., editors, *Proceedings of EMNLP 2004*, pages 158–165, Barcelona, Spain. Association for Computational Linguistics.

Alpaydin, E. (2004). *Introduction to Machime Learning*. The MIT Press, Cambridge, MA.

Anderson, J. R. (1997). *Production Systems, Learning and Tutoring*, volume Self-Modifying Production Systems: Models of Learning and Development. Bradford Books/MIT, Cambridge, MA.

Androutsopoulos, I., Koutsias, J., Chandrinos, K. V., y Spyropoulos, C. D. (2000). An Experimental Comparison of Naive Bayesian and Keywordbased Anti-spam Filtering with Personal e-mail Messages. In *Proceedings of SIGIR-00, 23rd ACM International Conference on Research and Development in Information Retrieval*, pages 160–167.

Arimura, H., Abe, J., Sakamoto, H., Arikawa, S., Fujino, R., y Shimozono, S. (2000). Text Data Mining: Discovery of Important Keywords in the Cyberspace. In *Kyoto International Conference on Digital Libraries*, pages 121–126.

Armstrong, T., Freitag, D., Joachims, T., y Mitchell, T. (1995). Webwatcher: A Learning Apprentice for the World Wide Web. In *Proceedings of the AAAI Spring Symposium on Information Gathering from Heterogeneous, Distributed Environments*.

Asirvatham, A. y Ravi, K. (2001). Web Page Classification based on Document Structure.

Attardi, G., Gulli, A., y Sebastiani, F. (1999). Automatic Web Page Categorization by Link and Content Analysis. In *Proceedings of THAI'99, European Symposium on Telematics, Hypermedia and Artificial Intelligence*, pages 105–119.

Aznar, E., Cros, A., y Quintane, L. (1991). *Coherencia Textual y Lectura*. Horsori.

Baeza-Yates, R. (2004). Excavando la Web. *El Profesional de la Información*, 13(1).

Baeza-Yates, R. y Ribeiro-Neto, B. (1999). *Modern Information Retrieval*. Addison-Wesley.

Baglioni, M., Ferrara, U., Romei, A., Ruggieri, S., y Turini, F. (2003). Preprocessing and Mining Web Log Data for Web Personalization.

Banfield, J. y Raftery, A. (1993). Model-based Gaussian and non-Gaussian Clustering. *Biomentrics*, 49(803–821).

Barfourosh, A., Motahary-Nezhad, H., Anderson, M. L., y Perlis, D. (2002). Information Retrieval on the World Wide Web and Active Logic: A Survey and Problem Definition.

Bennett, P., Dumais, S., y Horvitz, E. (2002). Probabilistic Combination of Text Classifiers using Reliability Indicators: Models and Results. In Beaulieu, M., Baeza-Yates, R., Myaeng, S. H., y Järvelin, K., editors, *Proceedings of SIGIR-02, 25th ACM International Conference on Research and Development in Information Retrieval*, pages 207–214, Tampere, FI. ACM Press, New York, US.

Bennett, P., Dumais, S., y Horvitz, E. (2005). The Combination of Text Classifiers Using Reliability Indicators. *Information Retrieval*, 8(1):67–100.

Berkan, R. y Trubatch, S. (1997). Fuzzy Systems Design Principles: Building Fuzzy IF-THEN Rules Bases. *IEEE Press*.

Berkhin, P. (2002). Survey of Clustering Data Mining Techniques. Technical report, Accrue Software, San Jose, CA.

Berners-Lee, T., Cailliau, R., Groff, J., y Pollermann, B. (1992). World-Wide Web: The Information Universe. *Electronic Networking: Research, Applications and Policy*, 1(2):74–82.

Best, J. (2001). *Psicología Cognitiva*. Madrid: Paraninfo.

Bezdek, J., editor (1981). *Pattern Recognition with Fuzzy Objective Function Algorithms*. Plenum Press, New York, NY.

Bezdek, J. y Pal, S., editors (1992). *Fuzzy Models for Pattern Recognition: Methods That Search for Structures in Data*. IEEE Press, Piscataway, NJ, USA.

Bharat, K. y Broder, A. (1998). A Technique for Measuring the Relative Size and Overlap of Public Web Search Engines. In *7th World-Wide Web Conference (WWW7)*.

Bharat, K. y Henzinger, M. (1998). Improved Algorithms for Topic Destillation in a Hyperlinked Environment. In *Proceedings of the ACM SIGIR98*, pages 104–111.

Billhardt, H. (2002). A Context Vector Model for Information Retrieval. *Journal of the American Society and Technology*, 53(3).

Boley, D., Gini, M., Gross, R., Han, S., Hastings, K., Karypis, G., Kumar, V., y Mobasher, J. (1999). Partitioning-Based Clustering for Web Document Categorization. *Decision Support Systems*.

Bookstein, A. (1981). A Comparison of two systems of Weighted Boolean queries. *Journal of the American Society for lnfornmtion Retrieval*, 32(4):275–279.

Bordogna, G. y Pasi, G. (1995). Controlling Retrieval through a User-adaptive Representation of Documents. *International Journal of Approximate Reasoning*, 12(3–4):317–339.

Bordogna, G. y Pasi, G. (2004). A model for a Soft Fusion of Information Accesses on the web. *Fuzzy Sets and Systems*, 104(105–118).

Borges, J. y Levene, M. (1999). Data Mining of User Navigation Patterns. In *Proceedings of the WEBKDD99*, pages 92–111.

Borges, J. y Levene, M. (2004). An Average Linear Time Algorithm for Web Data Mining. *International Journal of Information Technology and Decision Making*, 3.

Borko, H. (1962). The Construction of an Empirically based Mathematically derived Classification System. In *Proceedings of the Spring Joint Computer Conference*, volume 21, pages 279–289.

Borko, H. y Bernick, M. (1963). Automatic Document Classification. *Journal of the Association for Computing Machinery*, 10(2):151–162.

Botafogo, R., Rivlin, E., y Shneiderman, B. (1992). Structural Analysis of Hypertexts: Identifying Hierarchies and Useful Metrics. *ACM Transactions of Information Systems*, 2(10):142–180.

Bransford, J. y Johnson, M. (1972). Contextual Prerequisites for Understanding: Some investigations of Comprehension and Recall. *Journal of Verbal Learning and Verbal Behavior*, 11:717–726.

Brin, S., Motvani, L., y Winograd, T. (1998). What can you do with the web in your packet. *Bulletin of the IEEE Computer Society Technical Comitee on Data Engineering*.

Brin, S. y Page, L. (1998). The Anatomy of a Large-Scale Hypertextual Web Search Engine. *Computer Networks and ISDN Systems*, 30(1–7):107–117.

Bryan, M. (1988). *SGML An author's guide to the standard Generalized Markup Language*. Addison Wesley, Londres (Gran Bretaña).

Buckley, C. y Lewit, A. (1985). Optimizations of Inverted Vector Searches. In *Proceedings of the SIGIR'85*.

Caceres, S. (2004). La evolucion de los contenidos de internet.

Cadez, I., Heckerman, D., Meek, C., Smyth, P., y White, S. (2000). Visualization of Navegation Patterns on a Web Site Using Model-Based Clustering. In *Proceedings of the KDD 2000*.

Cadez, I. y Smyth, P. (1999). Probabilistic Clustering using Hierarchical Models. Technical report, no 99-16, Department of Information and Computer Science. University of California, Irvine.

Calado, P., Cristo, M., Moura, E., Ziviani, N., Ribeiro-Neto, B., y Gonzalves, M. (2003). Combining Link-based and Content-based Methods for Web Document Classification. In *CIKM '03: Proceedings of the twelfth international conference on Information and knowledge management*, pages 394–401, New York, NY, USA. ACM Press.

Calsamiglia, H. y Tusón, A. (1999). *Las cosas del decir. Manual del análisis del discurso.* Ariel, Barcelona.

Carey, M., Kriwaczek, F., y Ruger, S. (2000). A Visualization Interface for Document Searching and Browsing. In *Proceedings of the NPIVM 2000.*

Caropreso, M., Matwin, S., y Sebastiani, F. (2001). A Learner-independent Evaluation of the Usefulness of Statistical Phrases for Automated Text Classification. *In Text database and and Documment Managemen: theory and practice. A. G. Chin, ed.*, pages 78–102.

Carrasco, J., Fain, D., Lang, K., y Zhukov, L. (2003). Clustering of Bipartite Advertiser-keyword Graph. In *In proceedings of International Conference on Data Mining*, pages 1–8.

Casillas, A., Fresno, V., de Lena, M. G., y Martinez, R. (2004a). Evaluation of web page representations by content through clustering. In *Proceedings of the Eleventh Symposium on String Processing and Information Retrieval (SPIRE'2004)*, Padua, Italy.

Casillas, A., Fresno, V., Martínez, R., y Montalvo, S. (2005). Evaluzción del clustering de páginas web mediante funciones de peso y combinación heurística de criterios. *Revista de la Sociedad Española para el Procesamiento del Lenguaje Natural.*

Casillas, A., González de Lena, M., y Martínez, R. (2003a). Algoritmo de Clustering On-Line Utilizando Metaheurísticas y Técnicas de Muestreo. *Procesamiento del Lenguaje Natural*, 31:57–64.

Casillas, A., González de Lena, M., y Martínez, R. (2003b). Partitional Clustering Experiments with News Documents. *Lecture Notes in Computer Science*, 2588:615–618.

Casillas, A., González de Lena, M., y Martínez, R. (2004b). Sampling and feature selection in a genetic algorithm for document clustering. *Lecture Notes in Computer Science*, 2495:601–612.

Cassany, D. (1995). *La cocina de la escritura.* Anagrama, Barcelona.

Cater, S. y Kraft, D. (1989). A Generalization and Clarification of the Waller-Kraft Wish List. *Information Processing Management*, 25(1):15–25.

Celsi, R. y Olson, J. (1989). The Role of Involvement in Attention and Comprehension Processes. *Journal of costumer research*, 2(15):210–224.

Cerezo Arriaza, M. (1994). *Texto, contexto y situación.* Octaedro, Barcelona.

Chakrabarti, S. (2003). *Mining the Web.* Morgan Kaufmann, San Francisco.

Chakrabarti, S., Dom, B., Agrawal, R., y Raghavan, P. (1997). Using Taxonomy, Discriminants and Signatures for Navigating in Text Databases. In *VLDB*, pages 446–455.

Chakrabarti, S., Dom, B., Agrawal, R., y Raghavan, P. (1998a). Scalable Feature Selection, Classification and Signature Generation for Organizing Large Text Databases into Hierarchical Topic Taxonomies. *VLDB Journal.*

Chakrabarti, S., Dom, B., Gibson, D., Kleimberg, J., Kumar, P., Raghavan, P., Rajagolapan, S., y Tomkins, A. (1999a). Mining the Link Structure of the World Wide Web. *IEEE Computer.*

Chakrabarti, S., Dom, B., Gibson, D., Kleinberg, J., Raghavan, P., y Rajagopalan, S. (1998b). Automatic Resource List Compilation by Analyzing Hyperlink Structure and Associated Text. In *Proceedings of the 7th International World Wide Web Conference.*

Chakrabarti, S., Dom, B., y Indyk, P. (1998c). Enhanced Hypertext Categorization Using Hyperlinks. In Haas, L. M. y Tiwary, A., editors, *Proceedings of SIGMOD-98, ACM International Conference on Management of Data*, pages 307–318, Seattle, US. ACM Press, New York, US.

Chakrabarti, S., Joshi, M., y Tawde, V. (2001). Enhanced Topic Distillation using Text, Markup tags and Hyperlinks. In *SIGIR '01: Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 208–216, New York, NY, USA. ACM Press.

Chakrabarti, S., Roy, S., y Soundalgekar, M. (2003). Fast and Accurate Text Classification via Multiple Linear Discriminant Projections. *The VLDB Journal*, 12(2):170–185.

Chakrabarti, S., van der Berg, M., y Dom, B. (1999b). Distributed Hypertext Resource Discovery Through Examples. In *Proceedings of the 25th VLDB Conference*, Edimburgh, Scotland.

Cheeseman, P. y Stutz, J. (1996). *Advances in Knowledge Discovery and Data Mining*, chapter Bayesian classification (AutoClass): Theory and Results, pages 153–180. AAAI/MIT Press.

Chek, C. Y. (1997). *classification of world wide web documents*. PhD thesis, School of Computer Science, Carnegie Mellon University.

Chen, H. y Dumais, S. T. (2000). Bringing order to the Web: Automatically Categorizing Search Results. In *Proceedings of CHI-00, ACM Internation Conference on Human Factor in Computing Systems*, pages 145–152.

Cheng, C.-K. y Wei, Y.-C. A. (1991). An Improved two-way Partitioning Algorithm with Stable Performance. *IEEE Transactions on Computer Aided Design*, 10(12):1502–1512.

Cheung, D., Kao, B., y Lee, J. (1998). Discovering User Access Patterns on the World Wide Web. *Knowledge Based System Journal*, 10(7).

Chirita, P., Olmedilla, D., y Nejdl, W. (2003). Finding Related Hubs and Authorities. In Computer Society, I., editor, *Proceesingd of First Latin America Web Congress (LA-WEB 2003)*.

Cigarrán, J., Peñas, A., Gonzalo, J., y Verdejo, F. (2004). Browsing Search Results via Formal Concepts Analysis: Automatic Selection of Attributes. In *Concept Lattices, Second International Conference on Formal Concept Analysis, ICFCA 2004. Ed. P.W. Eklund*, pages 74–87. Springer Verlag - LNCS 2961.

Cigarrán, J., Peñas, A., Gonzalo, J., y Verdejo, F. (2005). Automatic Selection of Noun Phrases as Document Descriptors in a FCA-Based Information Retrieval System. In *Proceedings of the ICFCA 2005. B. Ganter and R. Godin (Eds.)*, pages 49–63. Springer Verlag - LNCS 3403.

Cohen, W. (1995). Fast Effective Rule Induction. In *Proceedings of the International Conference on Machibe Learning (ICML)*, pages 115–123.

Cohen, W. y Singer, Y. (1999). Context Sensitive Learning methods for Text Categorization. *ACM Transactions in Information Systems*, 17(2):141–173.

Conklin, J. (1987). Hypertext: An introduction and survey. *IEEE Computer*, 20:17–41.

Cooley, R., Srivastava, J., y Mobasher, B. (1997). Web Mining: Information and Pattern Discovery on the World Wide Web. In *Proceedings of the 9th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'97)*.

Cordón, O., Moya, F., y Zarco, C. (1999). Learning Queries for a Fuzzy Information Retrieval System by means of GA-P Techniques. In *EUSFLAT-ESTYLF Joint Conference*, Palma de Mallorca.

Cortes, C. y Vapnik, V. (1995). Support-Vector Networks. *Machine Learning*, 20:273–297.

Coscollola, M. (2001). *Diseño de documentación en soporte papel que se utiliza para explicar el funcionamiento de programas informáticos de gestión empresarial.* PhD thesis, Departament de Pedagogía Aplicada. UAB.

Craven, M., DiPasquo, D., Freitag, D., McCallum, A., Mitchell, T., Nigam, K., y Slattery, S. (1998a). Learning to extract Symbolic Knowledge from the World Wide Web. In *Proceedings of AAAI-98, 15th Conference of the American Association for Artificial Intelligence*, pages 509–516, Madison, US. AAAI Press, Menlo Park, US.

Craven, M. y Slattery, S. (2001). Relational Learning with Statistical Predicate Invention: Better Models for Hypertext. *Machine Learning*, 43(1–2):97–119.

Craven, M., Slattery, S., y Nigam, K. (1998b). First-order learning for web mining. In *European Conference on Machine Learning*, pages 250–255.

Crawford, E., Kay, J., y McCreath, E. (2002). IEMS - The Intelligent e-mail Sorter. In *Proceedings of the Nineteenth International Conference on Machine Learning (ICML).*

Crestani, F. (2002). Spoken Query Processing for Interactive Information Retrieval. *Data Knowledge Engineering*, 41(1):105–124.

Crestani, F. y Pasi, G. (2003). Handling Vagueness, Subjectivity and Imprecision in Information Retrieval: an introduction to the special issue. *Information Processing and Management*, 39(2):161–165.

Cristianini, N. y Shawe-Taylor, J. (2000). *An introduction to Support Vectors Machines and other kernel-based learning methods.* Cambridge University Press.

Crouch, D., Crouch, C., y Andreas, G. (1989). The Use of Cluster Hierarchies in Hypertext Information Retrieval. In *Proceedings of the Hypertext'89.*

Cunningham, P., Nowlan, N., Delany, S., y Haahr, M. (2003). A Case-based Approach to Spam Filtering that can Track Concept Drift.

Cutler, M., Shih, Y., y Meng, W. (1997). Using the Structure of HTML Documents to Improve Retrieval. In *USENIX Symposium on Internet Technologies and Systems.*

Cutting, D., Pedersen, J., Karger, D., y Tukey, J. (1992). Scatter/Gather: A Cluster-based Approach to Browsing Large Document Collections. In *SIGIR*, pages 318–329.

Dawson, J. (1974). Suffix Removal and Word Conflation. *ALLC Bulletin*, 2:33–46.

Dean, J. y Henzinger, M. R. (1999). Finding Related Pages in World Wide Web. *Computer Networks*, 31 11-16(1467–1479).

Dhillon, I. (2001). Co-clustering Documents and Words using Bipartite Spectral Graph Partitioning. In *In Proceedings of the 7th ACM SIGKDD*, pages 269–274. ACM Press.

Díaz-Pérez, P., Catenazzi, N., y Cuevas, A. (1996). *De la multimedia a la hipermedia.* Editorial Rama.

Ding, C. H. Q., Zha, H., Xiaofeng, H., Husbands, P., y Simon, H. D. (2004). Link Analysis: Hubs and Authorities on the World Wide Web. *SIAM Review*, 2(46):256–268.

Domingos, P. y Pazzani, M. J. (1996). Beyond Independence: Conditions for the Optimality of the Simple Bayesian Classifier. In *International Conference on Machine Learning*, pages 105–112.

Drucker, H., Vapnik, V., y Wu, D. (1999). Automatic Text Categorization and its Applications to Text Retrieval. *IEEE Transactions on Neural Networks*, 10(5):1048–1054.

Duda, R. O., Hart, P. E., y Stork, D. G. (2001). *Pattern Classification*. John Wiley & Sons, Inc.

Dumais, S. y Chen, H. (2000). Hierarchical Classification of Web Content. In Belkin, N. J., Ingwersen, P., y Leong, M.-K., editors, *Proceedings of SIGIR-00, 23rd ACM International Conference on Research and Development in Information Retrieval*, pages 256–263, Athens, GR. ACM Press, New York, US.

Edmonson, H. y Wyllys, R. (1961). Automatic Abstacting and Indexing Survey and Recommendations. *Communications of the ACM*, 4:226–234.

Embley, D., Jiang, S., y Ng, Y. (1999). Record-boundary Discovery in Web Documents.

Ernst, G. W. y Newell, A. (1967). Some issues of representation in a general problem solver. In *Proceedings of the 1968 Spring Joint Computer Conference - SJCC*, volume 32.

Escudero, G., Marquez, L., y Rigau, G. (2000). Boosting Applied to Word Sense Disambiguation. In *In Proceedings of ECML-00, 11th European Conference on Machine Learning*, pages 129–141.

Ester, M., Kriegel, H.-P., Sander, J., y Xu, X. (1996). A Density-based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In *In Proceedings of the 2nd ACM SIGKDD*, pages 226–231.

Ester, M., Kriegel, H.-P., y Schubert, M. (2002). Web Site Mining: a new way to spot competitors, customers and suppliers in the World Wide Web. In *KDD '02: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 249–258. ACM Press.

Everitt, B., Landau, S., y Leese, M. (2001). *Cluster Analysis*. Arnold Publishers, London.

Fairthorne, R. (1961). The Mathematics of Classification. *Towards Information Retrieval. Butterworths, London*, pages 1–10.

Fathi, M., Adly, N., y Nagi, M. (2004). Web Documents Classification Using Text, Anchor, Title and Metadata Information. In *Proceedings of the The International Conference on Computer Science, Software Engineering, Information Technology, e-Business and Applications (CSITeA)*, pages 1–8, Cairo, Egypt.

Felzenszwalb, P., Huttenlocher, D., y Kleinberg, J. (2003). Fast Algorithms for Large-State-Space HMMs with Applications to Web Usage Analysis. *Advances in Neural Information Processing Systems (NIPS)*, 16.

Feng, D., Siu, W. C., y Zhang, H. J., editors (2003). *Multimedia Information Retrieval and Management*. Springer.

Field, B. (1975). Towards Automatic Indexing: Automatic Assignment of Controlled-Language Indexing and Classification from Free Indexing. *Journal of Documentation*, 31(4):246–265.

Figuerola, C., Rodríguez, A., y Berrocal, J. (2001). Automatic vs. Manual Categorization of Documents in Spanish. *Journal of Documentation*, 57(6):763–773.

Figuerola, C. G., Gómez, R., Rodriguez, A. F. Z., y Berrocal, J. L. A. (2002). Spanish Monolingual Track: The impact of stemming on retrieval. *Lecture Notes on Computer Science*, 2406(253–261).

Foltz, P. (1996a). Comprehension, Coherence, and Strategies in Hypertext and Linear Text.

Foltz, P. (1996b). Latent Semantic Analysis for Text-Based Research.

Fresno, V. y Magdalena, L. (2005). Text content approaches in web content mining. *Encyclopedia of Data Warehousing and Mining. Editor: John Wang, Montclair State University. Idea Group Inc.*

Fresno, V. y Ribeiro, A. (2001a). Features Selection and Dimensionality Reduction in Web Page Representation. In *Proceedings of the International ICSC Congress on Computational Intelligence: Methods and Applications*, Bangor, Wales.

Fresno, V. y Ribeiro, A. (2001b). Una representación fuzzy para páginas web. In *Actas del XI Congreso Español sobre Tecnologías y Lógica Fuzzy (ESTYLF2002)*, León, España.

Fresno, V. y Ribeiro, A. (2004). An analytical approach to concept extraction in html environments. *Journal of Intelligent Information Systems - JIIS, Kluwer Academic Publishers*, pages 215–235.

Friburger, N., Maurel, D., y Giacometti, A. (2002). Textual similarity based on proper names.

Fu, Y., Sandhu, K., y Shih, M. (1999). Clustering of Web Users Based on Access Patterns. In *Proceedings of the 1999 KDD Workshop on Web Mining.*

Furnkranz, J. (1999). Exploiting structural information for text classification on the WWW. In *Intelligent Data Analysis*, pages 487–498.

Furnkranz, J. (2001). Using links for classification web pages. In *Proceedings of the 3rd International Symposium (IDA)*, pages 487–497, Amsterdam, Netherlands.

Ganti, V., Gehrke, J., y Ramakrishnan, R. (1999). Cactus-clustering categorical data using summaries. In *In Proceedings of the 5th ACM SIGKDD*, pages 73–83, San Diego, CA.

Garcia-Alegre, M. (1991). Artificial intelligence in process control: Fuzzy controllers. *Mundo Electrónico*, 214:42–49.

García Sevilla, J. (1997). *Manual de Psicología de la Atención.* Editorial Síntesis, Madrid.

Garofalakis, M. (1999). Data mining and the Web: Past, Present and Future. In *Proceedings of ACM 2nd Workshop on Web Information and Data Management*, pages 43–47, Kansas City, USA.

Gaussier, E., Goutte, G., Popat, K., y Chen, F. (2002). A hierarchical model for clustering and categorising documents. In *In Proceedings of the 24th European Colloquium on IR Research (ECIR 02)*, pages 229–247, Glasgow.

Getoor, L. (2003). Link mining: a new data mining challenge. *SIGKDD Exploration of Newsletters*, 5(1):84–89.

Getoor, L., Segal, E., Taskar, B., y Koller, D. (2001). Probabilistic models of text and link structure for hypertext classification.

Ghani, R., Slattery, S., y Yang, Y. (2001). Hypertext categorization using hyperlink patterns and meta data. In Brodley, C. y Danyluk, A., editors, *Proceedings of ICML-01, 18th International Conference on Machine Learning*, pages 178–185, Williams College, US. Morgan Kaufmann Publishers, San Francisco, US.

Glover, E. J. (2001). *Using Extra-Topical User Preferences to Improve Web-Based Metasearch*. PhD thesis, University of Michigan Computer Science and Engineering.

Glover, E. J., Tsioutsiouliklis, K., Lawrence, S., Pennock, D. M., y Flake, G. W. (2002). Using web structure for classifying and describing web pages. In *WWW '02: Proceedings of the eleventh international conference on World Wide Web*, pages 562–569. ACM Press.

Gokcay, E. y Principe, J. (2000). A new clustering evaluation function using renyi's information potential. In *Proceedings of the ICASSP 2000*.

Gonzalo, J. (2004). Hay vida después de google? Technical report, In the Software and Computing System seminars. Escuela Superior de Ciencias Experimentales y Tecnología. Universidad Rey Juan Carlos. (http://sensei.lsi.uned.es/ julio/).

Good, I. (1958). Speculations Concerning Information Retrieval. *Research Report PC-78, IBM Research Center, Yorktown Heights, New York*.

Goodman, K. S. (1967). Reading: A psicolinguistic guessing game. *Journal of the reading specialist*, (4):126–135.

Gough, P. B. (1972). *Language by ear and eye*, chapter Information processing models: a second of resding, pages 331–358. MIT Press, Cambridge, MA.

Gould, J. D., Alfaro, L., Finn, R., Haupt, B., y Minuto, A. (1987). Why reading was slower from crt displays than from paper. In *CHI '87: Proceedings of the SIGCHI/GI conference on Human factors in computing systems and graphics interface*, pages 7–11, New York, NY, USA. ACM Press.

Graham, I. S. (1997). *REL and REV attributes for hypertext relationships*.

Guha, S., R. R. y Shim, K. (1999). Rock: A robust clustering algorithm for categorical attributes. In *In Proceedings of the 15th ICDE*, pages 512–521.

Gulli, A. y Signorini, A. (2005). The indexable web is more than 11.5 billions pages. In *WWW2005: Proceedings of 14th International World Wide Web Conference*. ACM Press.

Halkidi, M., Nguyen, B., Varlamis, I., y Vazirgiannis, M. (2003). Thesus: Organizing web document collections based on link semantics.

Hall, L.O., O. B. y Bezdek, J. (1999). Clustering with a genetically optimized approach. *IEEE Trans. on Evolutionary Computation*, 3(2):103–112.

Han, E., Boley, D., Gini, M., Gross, R., Hastings, K., Karypis, G., Kumar, V., Mobasher, B., y Moore, J. (1998a). Webace: A wreb agent for document categorization and exploration. In *Proceedings of the 2nd International Conference on Autonomous Agents (Agents'98)*.

Han, E., Karypis, G., Kumar, V., y Mobasher, B. (1997). Clustering based on association rule hypergraphs. In *In Workshop of Research Issues On Data Mining and Knowledge Discovery*.

Han, E., Karypis, G., Kumar, V., y Mobasher, B. (1998b). Hypergraphs based clustering in high-dimensional data sets: A summary of results. *Bulletin of the Technical Committee on Data Engineering*, 1(21):9–13.

Han, J. y Kamber, M. (2001). *Data Mining*. Morgan Kaufmann Publishers.

Han, J., Kamber, M., y Tung, A. K. H. (2001). Spatial clustering methods in data mining: A survey. In Taylor y Francis, editors, *Geographic Data Mining and Knowledge Discovery*, pages 1–29.

Hansen, B. K. (2000). Analog forecasting of ceiling and visibility using fuzzy sets. In *AMS2000*.

Hartigan, J. (1975). *Clustering Algorithms*. Wiley, New York.

Hartigan, J. y Wong, M. (1979). Algorithm as136: A k-means clustering algorithm. *Applied Statistics*, 28:100–108.

Heckerman, D. (1995). *A Tutorial on Learning With Bayesian Networks*. Microsoft Research.

Herrera-Viedma, E. (2001). Modeling the retrieval process of an information retrieval system using an ordinal fuzzy linguistic approach. *Journal of the American Society for Information Science and Technology (JASIST)*, 52:460–475.

Hersh, W. (1994). Oshumed: An interactive retrieval evaluation and large test collection for research. In *In ACM SIGIR Conference on R&D in Information Retrieval*, pages 192–201.

Hofmann, T. (1999). Probabilistic latent semantic indexing. In *Proceedings of the 22nd ACM-SIGIR International Conference on Research and Development in Information Retrieval*, pages 50–57.

Honkela, T., Kaski, S., Lagus, K., y Kohonen, T. (1997). WEBSOM—self-organizing maps of document collections. In *Proceedings of WSOM'97, Workshop on Self-Organizing Maps, Espoo, Finland, June 4-6*, pages 310–315. Helsinki University of Technology, Neural Networks Research Centre, Espoo, Finland.

Hou, J. y Zhang, Y. (2003). Utilizing hyperlink transitivity to improve web page clustering. In *CRPITS'17: Proceedings of the Fourteenth Australasian database conference on Database technologies 2003*, pages 49–57, Darlinghurst, Australia, Australia. Australian Computer Society, Inc.

Höppner, F., Klawonn, F., Kruse, R., y Runkler, T. (1999). *Fuzzy Cluster Analysis*. Chichester, England.

Iglesias, P. V. y Veiga, I. G. (2004). *Psicología de la lectura*. Pearson. Prentice Hall.

Isakson, C. S. y Spyridakis, J. H. (2003). The influence of semantics and syntax on what readers remember. *Journal of the Society for Technical Communiication*, 50(4):538–53.

Iserman, R. (1998). On fuzzy logic applications for automatic control supervision and fault diagnosis. *IEEE Trans.Syst.Man and Cybern*, 28:221–235.

Jain, A. y Dubes, R. C. (1988). *Algorithms for Clustering Data*. Prentice Hall.

Jain, A., Topchy, A., Law, M., y Buhmann, J. (2004). Landscape of clustering algorithms. In *In Proceedings of the 17th International Conference on Pattern Recognition*, pages 260–263.

Jin, R. y Dumais, S. (2001). Probabilistic combination of content and links. In *SIGIR '01: Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 402–403, New York, NY, USA. ACM Press.

Joachims, T. (1997). A probabilistic analysis of the rocchio algorithm with tfidf for text categorization. In *In Proceedings of ICML-97, 14th International Conference on Machine Learning. Douglas H. Fisher, editor.*, pages 143–151, Nashville, US. Morgan Kaufmann Publishers, San Francisco, US.

Joachims, T. (1998). Text categorization with support vector machine: Learning with many relevant features. In *European Conference on Machine Learning (ECML)*, pages 137–142, Berlin. Springer.

Joachims, T. (1999). Transductive inference for text classification using support vector machines. In *ICML '99: Proceedings of the Sixteenth International Conference on Machine Learning*, pages 200–209. Morgan Kaufmann Publishers Inc.

John, G. H., Kohavi, R., y Pfleger, K. (1994). Irrelevant features and the subset selection problem. In *International Conference on Machine Learning*, pages 121–129. Journal version in AIJ, available at http://citeseer.nj.nec.com/13663.html.

John, G. H., Kohavi, R., y Pfleger, K. (1997). Irrelevant features and the subset selection problem. In *Proc. of the 11th International Conference on Machine Learnig ICML97*, pages 121–129.

Jones, S. (1972). A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28(1):11–21.

Kaban, A. y Girolami, M. (2000). Clustering of text documents by skewness maximisation. In *In Proceedings of the 2 nd International Workshop on Independent Component Analysis and Blind Source Separation (ICA 2000)*, pages 435–440.

Kaindl, H., Kramer, S., y Afonso, L. M. (1998). Combining structure search and content search for the world-wide web. In *HYPERTEXT '98. Proceedings of the Ninth ACM Conference on Hypertext and Hypermedia: Links, Objects, Time and Space - Structure in Hypermedia Systems, June 20-24, 1998, Pittsburgh, PA, USA*, pages 217–224. ACM.

Kan, M.-Y. y Thi, H. (2005). Fast webpage classification using url features. In *Proceedings of the Conference on Information and Knowledge Management (CIKM '05). Poster Paper.*

Karypis, G. (2002). CLUTO: A Clustering Toolkit. Technical Report Technical Report 02-017, University of Minnesota, Department of Computer Science, Minneapolis, MN 55455.

Karypis, G. y Han, E.-H. (2000). Concept indexing: A fast dimensionality reduction algorithm with applications to document retrieval and categorization. Technical report tr-00-0016, University of Minnesota.

Kaufman, L. y Rousseeuw, P. (1990). *Finding Groups in Data: an Introduction to Cluster Analysis.* John Wiley & Sons.

Kieras, D. E. (1981). The role of major referents and sentence topic in the construction of passage macrostructure. *Discourse Processes*, 4:1–15.

Kintsch, W. A. (1992). *The study of cognition: Conceptual and metodological issues*, chapter A cognitive architecture for comprehension, pages 143–164. American Psycological Association.

Kintsch, W. A. y van Dijk, T. A. (1978). Toward a model of text comprehension and production. *Psicological Review*, (85):363–394.

Klahr, D. (1989). Information processing approaches. *Annals of Child Development*, JAI Press, Inc.:131–185.

Klahr, D. (1992). Information processing approaches to cognitive development. (3rd edition). *Developmental Psychology: An Advanced Textbook*, pages 273–335. In M. H. Bornstein and M. E. Lamb (Eds.).

Klahr, D., Langley, P., y Neches, R. T. (1987). *Production System Models of Learning and Development*. MIT Press.

Klahr, D. y Wallace, J. (1976). Cognitive development: An information processing view. *Lawrence Erlbaum Associates*.

Kleimberg, J. M. (1998). Authoritative sources in a hyperlinked environment. In *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms*, pages 668–677.

Kohonen, T. (1995). *Self-organizing maps*. Springer-Verlag, Berlín.

Koll, M. (1979). WEIRD: An approach to concepr-based information retrieval. In *SIGIR Forum*, volume 13, pages 32–50.

Koller, D. y Sahami, M. (1996). Toward optimal feature selection. In Saitta, L., editor, *Proceedings of the International Conference on Machine Learning*, volume 13. Morgan-Kaufmann.

Koller, D. y Sahami, M. (1997). Hierarchically classifying documents using very few words. In *Proceedings of the International Conference on Machine Learning, 14th International Conference on Machine Learning*, pages 170–178.

Kosala, R. y Blockeel, H. (2000). Web mining research: A survey. *ACM SIGKDD Explorations Newsletter*, 2(1):1–15.

Kowalski, G. (1997). *Information Retrieval Systems. Theory and Implementation*. Kluwer Academic Publishers.

Kraaij, W. (2002). TNO at CLEF-2001: Comparing translation resources. *Lecture Notes on Computer Science*, 2406:78–93.

Kruengkrai, C. y Jaruskulchai, C. (2002). A Parallel Learning Algorithm for Text Classification. In *Proceedings of the Eighth ACM SIGKDD International Conference and Data Mining (KDD-2002)*.

Lam, S. L. y Lee, D. L. (1999). Feature reduction for neural network based text categorization. In Chen, A. L. y Lochovsky, F. H., editors, *Proceedings of DASFAA-99, 6th IEEE International Conference on Database Advanced Systems for Advanced Application*, pages 195–202, Hsinchu, TW. IEEE Computer Society Press, Los Alamitos, US.

Lam, W. y Low, K.-F. (1997). Automatic document classification based on probabilistic reasoning: Model and performance analysis. In *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, volume 3, pages 2719–2723.

Landauer, T. K., Foltz, P. W., y Laham, D. (1998). An introduction to latent semantic analysis. *Discourse Processes*, 25:259–284.

Larsen, J., Hansen, L., Szymkowiak, A., Christiansen, T., y Kolenda, T. (2001). Webmining: Learning from the world wide web. *Computational Statistics and Data Analysis*.

Larsen, P. M. (1980). Industrial Applications of Fuzzy Logic Control. *International Journal of Man-Machine Studies*, 12(1):3–10.

Lawrence, S. y Giles, C. L. (1998). Context and page analysis for improved web search. *IEEE Internet Computing*, 2(4):38–46.

Lawrence, S. y Giles, C. L. (1999). Accessibility of information on the web. *Nature*, 400:107–109.

Lempel, R. y Moran, S. (2000). The stochastic approach for link-structure analysis (salsa) and the tkc effect. *Computer Networks*, 33((1-6)):387–401.

Leuski, A. y Allan, J. (2000). Improving iunteractive retrieval by combining ranked lists and clustering. In *Proceedings of RIAO2000*, pages 665–681.

Lewis, D. (1990a). Representation quality in text classification: An introduction and experiments. In Kauffmann, M., editor, *Proceedings of a Workshop Held at Hidden Valley*, pages 288–295, Pennsylvania.

Lewis, D. (1990b). Text representation for text classification. *Text-Based Intelligent Systems: Current Research in Text Analysis, Information Extraction and Retrieval*, pages 288–295.

Lewis, D., Schapire, R., Callan, J., y Papka, R. (1996). Training algorithms for linear text classifiers. In *In Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (Special Issue of the SIGIR Forum)*, pages 298–306. ACM.

Lewis, D. D. (1992). Text representation for intelligent text retrieval: a classification-oriented view. pages 179–197.

Lewis, D. D. y Ringuette, M. (1994). A comparison of two learning algorithms for text categorization. In *Proc. Symposium on Document Analysis and Information Retrieval SDAIR-94*, pages 81–93.

Li, W., Ng, W. K., y Lim, E.-P. (2004). Spectral analysis of text collection for similarity-based clustering. In *PAKDD*, pages 389–393.

Liu, B., Zhao, K., y Yi, L. (2002). Visualizing web site comparisons. In *WWW '02: Proceedings of the eleventh international conference on World Wide Web*, pages 693–703. ACM Press.

López-Ostenero, F., Gonzalo, J., y Verdejo, F. (2004). Búsqueda de información multilingue: estado del arte. *Revista Iberoamericana de Inteligencia Artificial*, VIII (22):11–35.

Lorch, R. F. y Chen, A. H. (1986). Effects of number signals on reading and recall. *Journal of Educational Psicology*, 78:263–279.

Lorch, R. F., Lorch, E. P., y Inman, W. E. (1993). Effects of signaling topic structure on text recall. *Journal of Educational Psycology*, 2(85):281–290.

Lovins, J. (1968). Development of a Stemming Algorithm. *Mechanical Translation and Computational Linguistics*, 11:22–31.

Lu, Q. y Getoor, L. (2003). Link-based text classification. In *Proceedings of the IJCAI Workshop on Text Mining and Link Analysis*.

Luhn, H. (1953). A new method of recording and searching information. *American Documentation*, 4(1):14–16.

Luhn, H. (1957). A statistical approach to mechanized encoding and searching of literaty information. *IBM Journal of Research and Development*, 1(4):307–319.

MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. In *In Proceedings of the 5th Symposium of Mathematics, Statistics and Probability.*, pages 281–297.

Mamdani, E. H. y Assilian, S. (1975). An Experiment in Linguistic Sythesis with a Fuzzy Logic Controller. *International Journal of Man-Machine Studies*, 7(1):1–13.

Mandler, J. M. y Nancy S. Johnson (1977). Remembrance of things parsed: Story structure and recall. *Cognitive Psycology*, 9:111–151.

Manning, C. y Schutze, H. (1999). *Foundations of Statistical Natural Language Processing*. MIT Press.

Maron, M. E. (1961). Automatic indexing : an experimental inquiry. *Journal of the ACM*, 8:407–417.

Mase, H. (1998). Experiments on automatic web page categorization for ir system. Technical report, Standford University.

Mase, H., Morimoto, Y., y Tsuji, H. (2000). Classification knowledge discovery from newspaper articles. *Studies in Informatics and Control*, 9(3):167–178.

Mauldin, M. (1997). Lycos: Design choices in an internet search service. (html version). *IEEE Expert*.

McCallum, A. (1999). Multi-label text classification with a mixture model trained by em.

McCallum, A. y Nigam, K. (1998). A comparison of event models for naive bayes text classification. In *Proceedings of the AAAI-98 Workshop on Learning for Text Categorization*.

McLachlan, G. y Basford, K. (1988). *Mixture Models: Inference and Applications to Clustering*. Marcel Dekker, New York, NY.

McLachlan, G. y Krishnan, T. (1997). *The EM Algorithm and Extensions*. John Wiley & Sons, New York, NY.

McLeod, I., Barnard, D., Hamilton, D., y Levinson, M. (1991). SGML Documents and Non-linear Text Retrieval. In *Actas de RIAO'91*, pages 226–244.

Mehler, A., Dehmer, M., y Gleim, R. (2004). Towards Logical Hypertext Structure. A Graph-Theoretic Perspective. In *Proceedings of I2CS'04*, page (to appear), Berlin/New York. IEEE, Springer.

Meila, M. (January 1999). *Learning with Mixtures of Trees*. PhD thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology.

Merkl, D. (1997). Lessons learned in text document classification. In *Proceedings of WSOM'97, Workshop on Self-Organizing Maps, Espoo, Finland, June 4–6*, pages 316–321. Helsinki University of Technology, Neural Networks Research Centre, Espoo, Finland.

Merkl, D. (1998). *Text data mining. A handbook of Natural Languages Processing Techniques and Applications for the Processing of Languages as Text*. Marcel Dekker.

Miikkulainen, R. (1990). Script recognition with hierarchical feature maps. *Connection Science*, 2.

Mitchell, T. (1997). *Machine Learning*. McGraw Hill.

Mladenic, D. (1998). Feature subset selection in text learning. In *Proceedings of ECML-98, 10th European Conference on Machine Learning*, pages 95–100. IEEE Press.

Mobasher, B., Dai, H., Luo, T., Sun, Y., y Zhu, J. (2000). Integrating web usage and content mining for more effective personalization. In *EC-WEB '00: Proceedings of the First International Conference on Electronic Commerce and Web Technologies*, pages 165–176. Springer-Verlag.

Molinari, A. y Pasi, G. (1996). A fuzzy representation of html documents for information retrieval systems. In *Proceedings of the IEEE International Conference on Fuzzy Systems*, pages 8–12.

Molinari, A., Pasi, G., y Pereira, R. A. M. (2003). An indexing model of html documents. In *SAC '03: Proceedings of the 2003 ACM symposium on Applied computing*, pages 834–840. ACM Press.

Moore, J., Han, E., Boley, D., Gini, M., Gross, R., Hastings, K., Karypis, G., Kumar, V., y Mobasher, B. (1997). Web page categorization and feature selection using association rule and principal component clustering.

Morkes, J. y Nielsen, J. (1997). Concise, scannable, and objective: How to write for the web.

Mukherjea, S. y Hara, Y. (1997). Focus + context views of world wide web nodes. In *Proceedings of the eight ACM conference on Hypertext*, pages 187–196.

Muresan, G. y Harper, D. (1998). The WebCluster project, using clustering for mediating access to the world wide web. In *Proceedings of the SIGIR: ACM Special Interest Group on Information Retrieval*, pages 357–358.

Muresan, G. y Harper, D. (2001). Document clustering and language models for system-mediated information access. In *Proceedings of the ECDL 2001*, pages 438–449.

Muresan, G., Harper, D., y Mechkour, M. (1999). WebCluster, a tool for mediated information access. In *Proceedings of the SIGIR: ACM Special Interest Group on Information Retrieval*, page 337.

Musciano, C. y Kennedy, B. (1997). *HTML: The Complete Guide*. McGraw-Hill.

Musciano, C. y Kennedy, B. (2000). *HTML & XHTML: The Complete Guide*. McGraw-Hill.

Muslea, I., Minton, S., y Knoblock, C. (1998). Wrapper induction for semistructured web-based information sources.

Needham, R. (1961). *Research on information retrieval, classification and grouping*. PhD thesis, University of Cambridge; Cambridge Language Research Unit, Report M.L. 149.

Nelson, T. H. (1965). A file structure for the complex, the changing and the indeterminate. In *Proceedings of the ACM National Conference*.

Newell, A. y Simon, H. A. (1963). Gps, a program that simulates human thought. *Computers and Thought MIT Press*, pages 279–93. Feigenbaum, E. and Feldman, J. editors.

Ng, R. y Han, J. (1994). Efficient and effective clustering method for spatial data mining. In *In Proceedings of the 20th VLDB Conference*, pages 144–155.

Nielsen, J. (1997a). How users read on the web.

Nielsen, J. (1997b). Report from a 1994 web usability study.

Nigam, K. (2001). *Using Unlabeled Data to Improve Text Classification.* PhD thesis, Carnegie Mellon University, Pittsburgh, US.

Nigam, K., Lafferty, J., y McCallum, A. (1999). Using maximum entropy for text classification.

Nigam, K., McCallum, A. K., Thrun, S., y Mitchell, T. M. (2000). Text classification from labeled and unlabeled documents using EM. *Machine Learning*, 39(2/3):103–134.

Norbert Gövert, M. L. y Fuhr, N. (1999). A probabilistic description-oriented approach for categorising web documents. In *In Susan Gauch and Il-Yeol Soong, editors, Proceedings of the Eighth International Conference on Information and Knowledge Management. ACM*, pages 475–482.

Off, L. (2000). *Engine Basics.* Look Off.

Oh, H.-J., Myaeng, S.-H., y Lee, M.-H. (2000a). A practical hypertext categorization method using links and incrementally available class information. In *SIGIR*, pages 264–271.

Oh, H.-J., Myaeng, S. H., y Lee, M.-H. (2000b). A practical hypertext catergorization method using links and incrementally available class information. In *SIGIR '00: Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 264–271, New York, NY, USA. ACM Press.

O'Neill, E. T., Lavoie, B. F., y Bennett, R. (2002). Trends in the evolution of the public web. 1998-2002. 2(4).

Ortega-Binderberger, M., Mehrotra, S., Chakrabarti, K., y Porkaew, K. (2000). WebMARS: A multimedia search engine. Technical Report. tr-db-00-09, University of California at Irvine.

Pereira, F. C.Ñ., Tishby, N. Z., y Lee, L. (1993). Distributional clustering of english words. In *In Proceedings of the Association for Computational Linguistics*, pages 183–190.

Perfetti, C. y Goldman, S. (1974). Thematization of sentence retrieval. *Journal of Verbal Learning and Verbal Behavior*, 13:70–79.

Pierre, J. (2000). Practical issues for automated categorization of web sites. In *Electronic Proc. ECDL 2000 Workshop on Semantic Web.*

Pierre, J. M. (2001). On the automated classification of web sites. *CoRR*, cs.IR/0102002.

Pinkerton, B. (1994). Finding what people want: Experiences with the webcrawler. In *In Proceedings of the First International World-Wide Web Conference*, Geneva, Switzerland.

Piroli, P., Pitkow, J., y Rao, R. (1996). Silk from a sow's ear: Extracting usable structure fron the web. In Press, A., editor, *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing*, pages 118–125.

Pitkow, J. y Pirolli, P. (1997). Life, death, and lawfulness on the electronic frontier. In *Proceedings of the Conference on Human Factors in Computing Systems CHI'97.*

Porter, M. (1997). An algorithm for suffix stripping. In *Reprinted in Sparck Jones, Karen, and Peter Willet, Readings in Information Retrieval, San Francisco: Morgan Kaufmann.*

Potelle, H. y Rouet, J.-F. (2003). Effects of content representation and readers' prior knowledge on the comprehension of hypertext. *Int. J. Hum.-Comput. Stud.*, 58(3):327–345.

Quilan, J. R. (1990). Learning logical definitions from relations. *Machine Learning*, 3(5):236–266.

R. Armstrong, D. Freitag, T. J. y Mitchell, T. (1995). Webwatcher: A learning apprentice for the world wide web. In *AAAI Spring Symposium on Information Gathering*.

Raghavan, V. V. y Wong, S. K. M. (1986). A critical analysis of vector space model for information retrieval. *Journal of the American Society for Information Science*, 37(5):279–287.

Rayner, K. y Pollatsek, A. (1998). *The Psychology of Reading*. Prentice Hall, Englewood Cliffs, NJ.

Rennie, J. (2001). Improving multi-class text classification with naive bayes. Master's thesis, Massachusetts Institute of Technology.

Ribeiro, A. y Fresno, V. (2001). A multi criteria function to concept extraction in html environment. In *Proceedings of the International Conference on Internet Computing 2001 (IC'2001)*, pages 1–6.

Ribeiro, A., Fresno, V., García-Alegre, M., y Guinea, D. (2002). A fuzzy system for the web page representation. In *Intelligent Exploration of the Web. Springer-Verlag Group*, pages 19–38.

Ribeiro, A., Fresno, V., García-Alegre, M., y Guinea, D. (2003). Web page classification: A soft computing approach. In *Proceedings of the Atlantic Web Intelligence Conference (AWIC'03)*, pages 103–112.

Riboni, D. (2002). Feature selection for web page classification.

Richardson, M. y Domingos, P. (2002). The Intelligent Surfer: Probabilistic Combination of Link and Content Information in PageRank. In *Advances in Neural Information Processing Systems 14*. MIT Press.

Rocchio, J. (1971). *Relevance feedback in information retrieval. The SMART Retrieval System. Experiments in Automatic Document Processing*. Prentice Hall, Englewoods Cliffs, N. J.

Rodríguez, H. (2002). Tutorial sobre similitud semántica presentado en el curso de industrias de la lengua. (http://www.lsi.upc.es/ horacio/docencia.html).

Rosenbloom, P., Laird, J., y Newell, A., editors (1993). *The SOAR papers: Research on Integrated Intelligence*. MIT Press, Cambridge, MA.

Ruiz, M. y Srinivasan, P. (1998). Automatic text categorization using neural networks. In *Advances in Classification Research vol. 8: Proceedings of the 8th ASIS SIG/CR Classification Research Workshop. Efthimis Efthimiadis Ed.*, pages 59–72.

Ruiz, M. y Srinivasan, P. (1999). Hierarchical neural networks for text categorization. In *Proceedings of SIGIR'99, 22nd ACM International Conference on Research and Development in Information Retrieval*, pages 281–282.

Ruocco, A. y Frieder, O. (1997). Clustering and Classification of Large Document Bases in a Parallel Environment. *Journal of the American Society for Information Science*, 48(10):932–943.

Sahami, M., Dumais, S., Heckerman, D., y Horvitz, E. (1998). A bayesian approach to filtering junk E-mail. In *Learning for Text Categorization: Papers from the 1998 Workshop*, Madison, Wisconsin. AAAI Technical Report WS-98-05.

Salton, G. (1988). *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer*. Addison-Wesley.

Salton, G. y Buckley, C. (1965). The SMART automatic documento retrieval system - an ilustration. *Communications of the ACM*, 8(6):391–398.

Salton, G., Wong, A., y Yang, C. (1975). A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620.

Salton, G. y Yang, C. (1973). On the specification of term values in automatic indexing. *Journal of Documentation*, 29(4):351–372.

Salton, G., M. M. (1983). *Introduction to Modern information Retrieval*. McGraw Hill.

Sanchez, R. P., Lorch, E. P., y F., L. R. (2001). Effects of heading s on text proccessing strategies. *Contemporary Educational Psycology*, (26):418–428.

Savoy, J. (1999). A Stemming Procedure and Stop-word List for General French Corpora. *Journal of the American Society for Information Science*, 50:944–952.

Schutze, H., Hull, D., y Pedersen, J. (1995). A comparasion of classifers and document representations for the routing problem. In *Proceedings of the SIGIR'95*.

Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1):1–47.

Sebrechts, M. (1999). Visualization of search results: A comparative evaluation of text, 2d and 3d interfaces. In *Proceedings of the SIGIR'99*.

Serafini, M. T. (1992). *Cómo se escribe*. Paidós.

Shatkay, H. y Wilbur, W. J. (2000). Finding themes in medline documents: Probabilistic similarity search. In *Advances in Digital Libraries*, pages 183–192.

Sheikholeslami, G., C. S. y Zhang, A. (1998). Wavecluster: A multiresolution clustering approach for very large spatial databases. In *In Proceedings of the 24th Conference on VLDB*, pages 428–439.

Shih, L. y Karger, D. (2004). Using urls and table layout for web classification tasks.

Sinka, M. P., C. D. W. (2002). A large benchmark dataset for web document clustering. *Soft Computing Systems: Design, Management and Applications, Frontiers in Artificial Intelligence and Applications*, 87:881–890.

Sinka, M. P. y Corne, D. (2004). Measuring effectiveness of text-decorated html tags in web document clustering. In *Proceedings of the ICWI*, pages 707–714.

Slattery, S. y Craven, M. (1998). Combining statistical and relational methods for learning in hypertext domains. In *ILP '98: Proceedings of the 8th International Workshop on Inductive Logic Programming*, pages 38–52. Springer-Verlag.

Smith, K. A. y Ng, A. (2003). Web page clustering using a self-organizing map of user navigation patterns. *Decision Support Systems*, 35(2):245–256.

Song, R., Liu, H., Wen, J.-R., y Ma, W.-Y. (2004). Learning important models for web page blocks based on layout and content analysis. *SIGKDD Explor. Newsl.*, 6(2):14–23.

Sparck, J. (1972). A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28(1):11–21.

Spirydakis, J. H. (2000). Guidelines for authoring comprehensible web pages and evaluating their success. *Journal of the Society for Technical Communication*, pages 359–382.

Srihari, R. K. (1995). Automatic indexing and content-based retrieval of captioned images. *Computer*, 28(9):49–56.

Srikant, R. (1998). Title. In *Research Report: Mining Sequential Patters: Generalizations and performance improvements. EDBT.*, Avignon, France.

Stanovich, K. E. (1980). Toward an interactive-compensatory model of individual differences in the development of reading fluency. *Reading research quartely*, 1(16):32–71.

Strehl, A., Ghosh, J., y Mooney, R. (2000). Impact of similarity measures on web-page clustering. In *Proceedings of the 17th National Conference on Artificial Intelligence: Workshop of Artificial Intelligence for Web Search (AAAI 2000), 30-31 July 2000, Austin, Texas, USA*, pages 58–64. AAAI.

Sugiyama, K., Hatano, K., y Yoshikawa, M. (2003). Refinement of tf-idf schemes for web pages using their hyperlinked neighboring pages. In *Proceedings of the Hypertext'03 Conference*, Nottingham.

Sun, A. y Lim, E.-P. (2003). Web unit mining: finding and classifying subgraphs of web pages. In *In Proceedings 12th Int. Conf. on Information and knowledge management*, pages 108–115.

Svatek, V. y Berka, P. (2000). Url as starting point for www document categorisation. In *In Proceedings of the RIAO'2000 – Content-Based Multimedia Information Access, CID*, pages 1693–1702, Paris.

Svatek, V., Labsky, M., y Vacura, M. (2004). Knowledge modelling for deductive web mining. In *Lecture Notes in Computer Science, Vol. 3257*, pages 337–353.

Szymkowiak, A., Larsen, J., y Hansen, L. (2001). Hierarchical clustering for datamining. In *Proceedings of the 5th International Conference on Knowledge-Based Intelligent Information Engineering Systems and Allied Technologies (KES 2001).*

Tan, A.-H. y Lai, F.-L. (2000). Text categorization, supervised learning, and domain knowledge integration. In *Proceedings of the KDD'2000. Workshop on Text Mining*, pages 113–114.

Thelwall, M. (2001). Extracting macroscopic information from web links. *J. Am. Soc. Inf. Sci. Technol.*, 52(13):1157–1168.

Thorsten Joachims, N. C. y Shawe-Taylor, J. (2001). Composite kernels for hypertext categorization. In *ICML '01: Proceedings of the 18th International Conference on Machine Learning*, pages 250–257. Williams College, US.

Tikk, D., Yang, J., y Bang, S. (2003). Hierarchical text categorization using fuzzy relational thesaurus.

Tong, S. y Koller, D. (2001). Support vector machine active learning with applications to text classification. *Journal of Machine Learning Research*, pages 45–66.

Tung, A.K.H., H. J. y Han, J. (2001). Spatial clustering in the presence of obstacles. In *In Proceedings of the 17th ICDE*, pages 359–367.

Valle, F., Cuetos, F., y Igoa, J. (1990). *Comprensión y Producción del lenguaje (lecturas de psicolinguística).* Alianza psicología.

van Dijk, T. A. (1997). *La ciencia del texto: Un enfoque interdisciplinario.* Paidós Comunicación.

van Rijsbergen, C. J. (1979). *Information Retrieval*. Butterworths, London, 2nd edition.

Vapnik, V.Ñ. (1995). *The Nature of Statistical Learning Theory*. Springer, New York.

Vlajic, N. y Card, H. C. (1998). Categorizing web pages on the subject of neural networks. *J. Netw. Comput. Appl.*, 21(2):91–105.

Wang, J., Hodges, J. E., y Tang, B. (2003). Classification of Web Documents Using a Naïve Bayes Method. In *Proceedings if the ICTAI*, pages 560–564.

Wang, K. y Su, M.-Y. T. (2002). Item selection by "hub-authority" profit ranking. pages 652–657.

Wang, Y. y Kitsuregawa, M. (2001). Use link-based clustering to improve web search results. In *WISE (1)*, pages 115–124.

Wang, Y. y Kitsuregawa, M. (2002). Evaluating contents-link coupled web page clustering for web search results. In *CIKM '02: Proceedings of the eleventh international conference on Information and knowledge management*, pages 499–506, New York, NY, USA. ACM Press.

Wasserman, S. (1994). *Social Network Analysis: Methods and Applications*. Cambridge University Press, Cambridge, England, and New York.

Weigend, A., Weiner, E., y Pedersen, J. (1999). Exploiting hierarchy in text categorization. *Information Retrieval*, 1(3):193–216.

Weiss, R., Velez, B., Sheldon, M., Namprempre, C., Szilagyi, P., Duda, A., y Gifford, D. (1996). Hypursuit: A hierarchical network search engine that exploits content-link hypertext clustering. In *In ACM Conference on Hypertext*, Washington, USA.

Wen, J.-R., Nie, J.-Y., y Zhang, H.-J. (2001). Clustering user queries of a search engine. In *Proceedings of the WWW'01 Conference*, pages 162–168.

Widyantoro, D. H. (1999). Dynamic modeling and learning user profile in personalized news agent.

Wikipedia (2006). Web semántica — wikipedia, la enciclopedia libre. [Internet; descargado 1-abril-2006].

Willet, P. (1988). Recent trends in hierarchical document clustering: a critical review. *Information Processing and Management*, 24(5):577–597.

Wong, W. y Fu, A. (2000). Incremental document clustering for web page classification.

Wright, P. (1991). Cognitive overheads and prostheses: some issues in evaluating hypertexts. In Press, A., editor, *Proceedings of the ACM conference on Hypertext: Hypertext '91*, pages 1–12, New York.

Yang, Q., Wang, H., Wen, J., Zhang, G., Lu, Y., Lee, K., y Zhang, H. (2000). Toward a next generation search engine. In *Proceedings of the Sixth Pacific Rim Artifact Artificial Intelligent Conference*, Melborne, Australia.

Yang, Y. (1996). Sampling stretegies and learning efficiency in text categorization. In *Proceedings of the AAAI Spring Symposium on Machine Learning in Information Access*, pages 88–95.

Yang, Y. (1999). An evaluation of statistical approaches to text categorization. *Journal of Information Retrieval*, 1(1/2):67–88.

Yang, Y. (2001). A study on thresholding strategies for text categorization. In Croft, W. B., Harper, D. J., Kraft, D. H., y Zobel, J., editors, *Proceedings of SIGIR-01, 24th ACM International Conference on Research and Development in Information Retrieval*, pages 137–145, New Orleans, US. ACM Press, New York, US.

Yang, Y. y Liu, X. (1999). A re-examination of text categorization methods. In *22nd Annual International SIGIR*, pages 42–49, Berkley.

Yang, Y. y Pedersen, J. O. (1997). A comparative study on feature selection in text categorization. In Fisher, D. H., editor, *Proceedings of ICML-97, 14th International Conference on Machine Learning*, pages 412–420, Nashville, US. Morgan Kaufmann Publishers, San Francisco, US.

Yang, Y., Slattery, S., y Ghani, R. (2002). A study of approaches to hypertext categorization. *J. Intell. Inf. Syst.*, 18(2-3):219–241.

Yi, L. y Liu, B. (2003). Web page cleaning for web mining through feature weighting. In *IJCAI*, pages 43–50.

Yin, L.-L. (1996). *Learned Text Categorization By Backpropagation Neural Network*. PhD thesis, Hong Kong University of Science & Technology.

Young, R. M. (2001). *Production Systems in Cognitive Psychology*, volume International Encyclopedia of the Social and Behavioral Sciences. Pergamon.

Yu, H., Hana, J., y Chang, K.-C. (2004). Pebl: Web page classification without negative examples. *IEEE Transactions on Knowledge and Data Engineering*, 16, Issue 1:70–81.

Yu, S., Cai, D., Wen, J.-R., y Ma, W.-Y. (2003). Improving pseudo-relevance feedback in web information retrieval using web page segmentation. In *WWW '03: Proceedings of the twelfth international conference on World Wide Web*, pages 11–18. ACM Press.

Zadeh, L. A. (1965). Fuzzy sets. *Information and control*, 8:338–353.

Zadeh, L. A. (1988). Fuzzy logic. *IEEE Computer*, pages 83–93.

Zamir, O. y Etzioni, O. (1998). Web document clustering: A feasibility demostration. In *Proceedings of the SIGIR: ACM Special Interest Group in Information Retrieval*.

Zamir, O. y Etzioni, O. (1999). Grouper: A dynamic clustering interface to web search results. In *WWW'8: Proceedings of the World Wide Web Conference*.

Zamir, O., Etzioni, O., Madani, O., y Karp, R. M. (1997). Fast and intuitive clustering of web documents. In *Knowledge Discovery and Data Mining*, pages 287–290.

Zhang, J. y Yang, Y. (2004). Probabilistic score estimation with piecewise logistic regression. In *ICML '04: Twenty-first international conference on Machine learning*. ACM Press.

Zhao, Y. y Karypis, G. (2001). Criterion functions for document clustering: Experiments and analysis. Technical Report 01-40, University of Minnesota, Department of Computer Science, Minneapolis, MN 55455.

Zhao, Y. y Karypis, G. (2002). Evaluation of hierarchical clustering algorithms for document datasets. In *CIKM '02: Proceedings of the eleventh international conference on Information and knowledge management*, pages 515–524, New York, NY, USA. ACM Press.

Zipf, G. K. (1949). *Human behavior and the principle of least effort*. Addison-Wesley, Reading, MA.